

**II CONGRESO ESPAÑOL
SOBRE
TECNOLOGIAS Y LOGICA FUZZY**

Boadilla del Monte

2-4 Noviembre 1992

**II CONGRESO ESPAÑOL
SOBRE TECNOLOGIAS Y
LOGICA FUZZY**

**BOADILLA DEL MONTE
2-4 NOVIEMBRE 1992**

COMITE DEL CONGRESO:

Presidente: Julio Gutierrez Ríos

Secretario: Antonio Ruiz Mayor

Vocales: Susana Cubillo Villanueva

José Angel Olivas Varela

ENTIDADES COLABORADORAS:

**FACULTAD DE INFORMATICA. UNIVERSIDAD POLITECNICA DE
MADRID**

**DEPARTAMENTO DE TECNOLOGIA FOTONICA. UNIVERSIDAD
POLITECNICA DE MADRID**

SOCIEDAD ESPAÑOLA DE TECNOLOGIAS Y LOGICA FUZZY

INSTITUTO NACIONAL DE TECNICA AEROSPAIAL

II CONGRESO ESPAÑOL SOBRE TECNOLOGIAS Y LOGICA FUZZY

INDICE

1.	<i>Condicionales exactos.....</i>	1
	E. Trillas, S. Cubillo.	
2.	<i>En torno a la condición de idempotencia.....</i>	17
	G. Mayor, J. Torrens.	
3.	<i>Contextos y Expertos (Un intento de caracterizar formalmente dos nociones importantes en Lógica Borrosa e Inteligencia Artificial).....</i>	23
	A. Sobrino.	
4.	<i>Indistinguibilidades Quasi-Unidimensionales.....</i>	41
	J. Jacas, J. Recasens.	
5.	<i>Una posible aplicación de la Teoría de Prototipos de Zadeh a la gestión de la evolución de siniestralidad de los incendios forestales en un motor de inferencia.....</i>	47
	R. Lago, J.A. Olivas, A. Sobrino.	
6.	<i>Un método inductivo para el aprendizaje de reglas difusas.....</i>	59
	A. González.	
7.	<i>Experiencias en el reconocimiento de caracteres mediante un sistema basado en una red neural y un autómatas fuzzy.....</i>	73
	R. Reina, J.R. González de Mendivil, J. Echanobe, J.R. Garitagoitia.	
8.	<i>La R-FN: un procesador elemental de información difusa.....</i>	81
	I. Requena, M. Delgado.	
9.	<i>Perceptrón Borroso.....</i>	93
	J. Rives, R. Pereira, J. A. Díaz.	

10.	<i>Culturergia: un modelo económico difuso (sabiduría, economía, energía, ingeniería).....</i>	103
	M. Luna de Toledo.	
11.	<i>Modelos de razonamiento posibilístico: Análisis comparativo.....</i>	111
	F. Esteva, P. García Calves, L. Godó.	
12.	<i>Validación de datos mediante Lógica Borrosa en Sistemas Expertos en tiempo real.....</i>	127
	J. A. Aguilar, E. de Pablo, X. Alamán.	
13.	<i>Estructuras algebraicas para conjuntos intuicionistas fuzzy.....</i>	135
	P. Burillo, H. Bustince.	
14.	<i>Algoritmos genéticos con función de evaluación difusa y sus aplicaciones.....</i>	147
	J.L. Castro, M. Delgado, F. Herrera, J.L. Verdegay.	
15.	<i>Toma de decisiones con probabilidades lingüísticas.....</i>	161
	M.T. Lamata, S. Moral.	
16.	<i>Estudio de un sistema interactivo de ayuda a la decisión en problemas de optimización difusa.....</i>	175
	J.M. Cadenas, F. Herrera.	
17.	<i>Herencia difusa.....</i>	185
	R. Conejo, J. L. Pérez de la Cruz, R. Morales, F.A. Triguero.	
18.	<i>Modificaciones sobre ID3 para el tratamiento de incertidumbre.....</i>	193
	L.I. Arranz.	
19.	<i>Gramáticas con atributos: estudio del caso fuzzy.....</i>	203
	R. Morales, B. Clares, R. Conejo, J. L. Pérez de la Cruz.	
20.	<i>Interface entre BD Relacionales Difusas y el entorno de Prolog.....</i>	215
	O. Pons, M.A. Vila.	
21.	<i>GEFRED. Un Modelo Generalizado de Bases de Datos Relacionales Difusas.....</i>	233
	J. M. Medina, M. A. Vila.	

22.	<i>Medidas de bondad para sistemas de control borroso.....</i>	253
	F. Matía, A. Jiménez, R. Sanz, R. Galán.	
23.	<i>Realización de un controlador fuzzy sobre un procesador digital de señal.....</i>	267
	I. del Campo, J.R. González de Mendívil, J.R. Garitagoitia, J.M. Tarela.	
24.	<i>Controlador Difuso basado en contextos.....</i>	283
	J.L. Castro, J.M. Zurita.	
25.	<i>Seguimiento frecuencial mediante lógica difusa.....</i>	289
	M. Reyes de los Mozos, E. Valderrama, J. Carrabina, N. Avellana, F. Lisa.	
26.	<i>Estudio comparativo de circuitos básicos de lógica difusa.....</i>	301
	S. Sánchez, I. Baturone, A. Barriga, J.L. Huertas.	
27.	<i>Realización CMOS de un circuito de función de pertenencia programable: el bloque básico de procesamiento borroso.....</i>	313
	A. Ruiz, J. Gutiérrez.	
28.	<i>Soluciones sistólicas en control borroso.....</i>	323
	A. Bugarín, S. Barro, R. Ruiz.	
29.	<i>Diseño sistemático de un procesador sistólico de inferencias difusas..</i>	339
	F. Fernández, A. Ruiz, J. Gutiérrez.	
30.	<i>Una aplicación de las cadenas inciertas de Kaufmann y Gil Aluja frente a las cadenas de Markov al control de gestión de tesorería de las empresas.....</i>	355
	E. López, C. Mendaña.	
31.	<i>Tratamiento fuzzy de los intangibles de empresa.....</i>	371
	A. Núñez, J. Sancho.	
32.	<i>Conjuntos difusos y reconocimiento de rasgos del terreno.....</i>	381
	R. Conejo, R. Morales, J.L. Pérez de la Cruz, J.A. Urbano.	

CONDICIONALES EXACTOS

Enric Trillas y Susana Cubillo

Dpto. Inteligencia Artificial y Dpto. Matemática Aplicada
Facultad de Informática. Universidad Politécnica de Madrid.

Resumen. *Para modelizar la implicación, es conveniente trabajar con relaciones que verifiquen la Regla del Modus Ponens, es decir, con las llamadas T-condicionales. Se estudian algunas de estas relaciones, y entre ellas, el Condicional Material.*

Palabras clave. *Implicación, T-Condicionales, Condicional Material, Monotonía.*

I. INTRODUCCION.

La Lógica Clásica ha sido el instrumento matemático utilizado comúnmente para modelizar el razonamiento realizado sobre una información cierta y exacta; en este caso, los valores de certeza con los que se trabaja son 0 y 1. Sin embargo, en muchas ocasiones la información de que disponemos es incompleta, bien porque los datos son inciertos, bien porque ellos mismos involucran términos o conceptos vagos... A pesar de esta incertidumbre, el hombre es capaz de saltar a conclusiones, de tomar decisiones. El intento de la Inteligencia Artificial de imitar en lo posible el razonamiento humano, le lleva a la necesidad de disponer de una herramienta capaz de modelizar estas situaciones de vaguedad, situaciones en las que los grados de certeza ya no se encuentran en el conjunto $\{0, 1\}$, sino en el intervalo $[0, 1]$.

Tras el primer artículo de Zadeh (4) introduciendo el concepto de "fuzzy set" en 1965, el continuo desarrollo de la Lógica Fuzzy como una rama de la Lógica, le ha permitido llegar a ser hoy día el instrumento más potente para trabajar con vaguedad y con razonamiento aproximado. Y sin embargo, sorprende que pocos autores hayan reconocido la incidencia o "permeabilidad" entre estos dos tipos de Lógica. En la mayoría de los textos de Lógica Clásica, no se hace ninguna mención a esta nueva rama, o si se la trata

en algún capítulo es sin apenas incisión posterior, como si se tratase de algo totalmente distinto, sin ninguna interrelación.

Nuestra sorpresa procede del hecho de que realmente la Lógica Clásica se puede considerar como un caso extremo de la Fuzzy, y los resultados que se obtienen al trabajar con vaguedad pueden ser aplicados al caso en que la información es exacta o totalmente conocida. Más aún, el estudio de la Lógica de la incertidumbre, permite en muchas ocasiones tener una visión más amplia de las herramientas del caso clásico, así como en el campo científico, observar un sistema exteriormente, nos puede aportar un mayor conocimiento del mismo, o en el mismo campo de las matemáticas, para un análisis completo de las series de potencias con coeficientes reales, es necesario pasar el campo complejo. Por ejemplo, estudiar la monotonía en condicionales fuzzy nos ha permitido ver con más claridad qué ocurre con los condicionales clásicos, y comprobar entre otros resultados, que el Condicional Material no es siempre monótono, lo que un estudio superficial podría hacer creer.

En este trabajo revisamos algunos conceptos y resultados de la Lógica Clásica relativos a condicionales, a los que hemos llegado extrapolando los obtenidos anteriormente al trabajar en Lógica Fuzzy. En realidad, se trata de un artículo didáctico con el que pretendemos hacer ver que el estudio de la Lógica Borrosa permite ver, al alejar el punto de vista, un campo mayor y que, con ello, puede llegarse a resultados elementales clarificadores. El análisis desde el enfoque borroso puede dar más nitidez al planteo de los elementos de la Lógica Clásica elemental.

II. CONDICIONALES.

Definición 2.1.- Dado un conjunto E y un subconjunto no vacío $T \subseteq E$, una relación $\Rightarrow$ se dice que es un T -condicional sobre E si verifica la Regla del Modus Ponens: si $a \in T$ y $a \Rightarrow b$, entonces $b \in T$.

Utilizando las funciones características φ_T y $\varphi_{\Rightarrow}$, esta regla es equivalente a: si $\varphi_T(a) = 1$ y $\varphi_{\Rightarrow}(a, b) = 1$, entonces $\varphi_T(b) = 1$.
o bien a: $\text{Min}(\varphi_T(a), \varphi_{\Rightarrow}(a, b)) \leq \varphi_T(b)$, para todo par $(a, b) \in E \times E$.

En este caso, diremos de T que es un estado lógico para $(E, \Rightarrow)$. Al conjunto de todos los estados lógicos para $(E, \Rightarrow)$ lo denotaremos por $\mathcal{F}(\Rightarrow)$. Como el conjunto $T = E$ es un estado lógico para cualquier $(E, \Rightarrow)$, siempre $\mathcal{F}(\Rightarrow) \neq \emptyset$.

Es trivial comprobar que si $\Rightarrow \subset \Rightarrow_1$, entonces será $\mathcal{F}(\Rightarrow_1) \subset \mathcal{F}(\Rightarrow)$.

Observemos que el conjunto $V_0 = \{b \in E ; \exists a \in E \text{ con } a \Rightarrow b\} \in \mathcal{F}(\Rightarrow)$.
En efecto, si $c \in V_0$ y $c \Rightarrow d$, trivialmente $d \in V_0$.

Teorema 2.2.

Si $\Rightarrow$ es reflexiva, $V_a = \{x \in E ; a \Rightarrow x\} \in \mathcal{F}(\Rightarrow)$ para todo $a \in E$ si y sólo si $\Rightarrow$ es transitiva.

Demostración. Como $a \Rightarrow a$ para todo $a \in E$, $V_a \neq \emptyset$. Si $\Rightarrow$ es transitiva, $c \in V_a$ y $c \Rightarrow d$, como $a \Rightarrow c$ y $c \Rightarrow d$, llegamos a $a \Rightarrow d$ y $d \in V_a$.

Supongamos ahora que para todo $a \in E$, $V_a \in \mathcal{F}(\Rightarrow)$; si $b \Rightarrow c$ y $c \Rightarrow d$, $c \in V_b$ y $c \Rightarrow d$, luego $d \in V_b$ y $b \Rightarrow d$. $\square$.

Se define en E la relación $\Rightarrow^*$, dual de $\Rightarrow$, como : $a \Rightarrow^* b$ si y sólo si $b \Rightarrow a$.

Teorema 2.3.

$T \in \mathcal{F}(\Rightarrow)$ si y sólo si $E - T \in \mathcal{F}(\Rightarrow^*)$.

Demostración. Supongamos $T \in \mathcal{F}(\Rightarrow)$. Sea $a \in E - T$ y $a \Rightarrow^* b$; $a \notin T$ y $b \Rightarrow a$, por lo que ha de ser $b \notin T$, es decir $b \in E - T$.

Sea ahora $E - T \in \mathcal{F}(\Rightarrow^*)$, $a \in T$ y $a \Rightarrow b$; será $a \notin E - T$ y $b \Rightarrow^* a$, por lo que $b \notin E - T$ y $b \in T$. $\square$

Teorema 2.4.

El conjunto $\mathcal{F}(\Rightarrow)$ es cerrado bajo uniones arbitrarias.

Demostración. Sea $T = \bigcup_{i \in I} T_i$, donde $T_i \in \mathcal{F}(\Rightarrow)$ para todo $i \in I$. Si $a \in T$ y $a \Rightarrow b$, existe $i \in I$ tal que $a \in T_i$ y $a \Rightarrow b$, por lo que $b \in T_i$ y $b \in T$. $\square$.

Como $E \in \mathcal{F}(\Rightarrow)$, está claro que $\bigcup_{T \in \mathcal{F}(\Rightarrow)} T = E$.

Teorema 2.5.

El conjunto $\mathcal{F}(\Rightarrow)$ es cerrado bajo intersecciones no vacías arbitrarias.

Demostración. Sea $T = \bigcap_{i \in I} T_i$, donde $T_i \in \mathcal{F}(\Rightarrow)$ para todo $i \in I$. Si $a \in T$ y $a \Rightarrow b$,

para todo $i \in I$ $a \in T_i$ y $a \Rightarrow b$, luego para todo $i \in I$ será $b \in T_i$ y $b \in T$. $\square$

Teniendo en cuenta esto, existen dos posibilidades: o bien $\bigcap_{T \in \mathcal{F}(\Rightarrow)} T \neq \emptyset$ es el menor estado lógico para $(E, \Rightarrow)$, o bien $\bigcap_{T \in \mathcal{F}(\Rightarrow)} T = \emptyset$, y se pueden buscar estados lógicos minimales.

Ejemplos 2.6.

1) Sea (E, p) un álgebra de Boole probabilizada, donde definimos la relación $a \Rightarrow b$ si y sólo si $p(a) = 0$ ó $p(a) > 0$ y $p(b/a) > 0$.

El subconjunto $E^+ = \{x \in E ; p(x) > 0\} \neq \emptyset$ por ser $p(1) = 1 > 0$, donde 1 es el mayor elemento del álgebra E . Veamos que $E^+ \in \mathcal{F}(\Rightarrow)$; si $a \in E^+$ y $a \Rightarrow b$, como $p(a) > 0$ ha de ser $p(b/a) = \frac{p(a \cdot b)}{p(a)} > 0$, con lo que $p(b) \geq p(a \cdot b) > 0$ y $b \in E^+$.

Sea ahora $T \in \mathcal{F}(\Rightarrow)$. Si $a \in T$, o bien $p(a) > 0$ y $a \in E^+$, o bien $p(a) = 0$ y $a \Rightarrow x$ para todo $x \in E$, con lo que sería $T = E$; así tenemos que, o bien $T = E$, o bien $T \subseteq E^+$.

Además, $\Rightarrow$ no es un preorden, ya que no verifica la propiedad transitiva. En efecto, dados a, b, c verificando $p(a \cdot b) > 0$, $p(b \cdot c) > 0$ y $p(a \cdot c) = 0$, obtenemos $a \Rightarrow b$, $b \Rightarrow c$ y $a \not\Rightarrow c$.

2) Sea E un conjunto dotado de una función no nula $\mu : E \rightarrow [0, 1]$. Definimos la relación:

$a \Rightarrow b$ si y sólo si $\mu(a) = 0$ o $\mu(a) > 0$ y $\text{Min}(1, \frac{\mu(b)}{\mu(a)}) > 0$.

El subconjunto $E^+ = \{x \in E ; \mu(x) > 0\}$ es no vacío. Comprobemos que $E^+ \in \mathcal{F}(\Rightarrow)$; si $a \in E^+$ y $a \Rightarrow b$, como $\mu(a) > 0$, $\text{Min}(1, \frac{\mu(b)}{\mu(a)}) > 0$, $\mu(b) > 0$ y $b \in E^+$.

Si $T \in \mathcal{F}(\Rightarrow)$ y $a \in T$, o bien $\mu(a) > 0$ y $a \in E^+$, o bien $\mu(a) = 0$ y para todo $x \in E$ $a \Rightarrow x$ y $x \in T$; es decir, $T = E$ o $T \subseteq E^+$.

Por otra parte, $\Rightarrow$ es un preorden (propiedades reflexiva y transitiva). En efecto, si $a \in E$ y $\mu(a) = 0$, $a \Rightarrow a$; si $\mu(a) > 0$, $\text{Min}(1, \frac{\mu(a)}{\mu(a)}) = 1 > 0$ y $a \Rightarrow a$. Por otra parte, dado $a \Rightarrow b$ y $b \Rightarrow c$, si $\mu(a) = 0$ entonces $a \Rightarrow c$; si $\mu(a) > 0$, $\text{Min}(1, \frac{\mu(b)}{\mu(a)}) > 0$, $\mu(b) > 0$ y $\text{Min}(1, \frac{\mu(c)}{\mu(b)}) > 0$; por tanto, $\text{Min}(1, \frac{\mu(c)}{\mu(a)}) \geq \text{Min}(1, \frac{\mu(b)}{\mu(a)}) \cdot \text{Min}(1, \frac{\mu(c)}{\mu(b)}) > 0$ y $a \Rightarrow c$.

Teorema 2.7.

Si $\Rightarrow$ es un preorden, los únicos estados lógicos para $(E, \Rightarrow)$ son las uniones de conjuntos V_a .

Demostración. Como los V_a son estados lógicos, su unión arbitraria también lo es. Sea $T \in \mathcal{F}(\Rightarrow)$; veamos que $T = \bigcup_{a \in T} V_a$. Si $b \in T$, por verificar $\Rightarrow$ la propiedad reflexiva, $b \in V_b$ y entonces $b \in \bigcup_{a \in T} V_a$; por otra parte, si $b \in \bigcup_{a \in T} V_a$, $b \in V_a$ para algún $a \in T$, $a \Rightarrow b$ y por ser T un estado lógico, $b \in T$. $\square$

Observemos que si $V \neq \emptyset$ no es un estado lógico para $(E, \Rightarrow)$, se puede construir $V^* = \bigcup_{a \in V} V_a$, que es el menor estado lógico para $(E, \Rightarrow)$ conteniendo a V .

Lema 2.8.

Sea E un conjunto, $T \subseteq E$ y $\Rightarrow$ un T -condicional. Si $\Rightarrow_1 \subseteq E \times E$, con $\Rightarrow_1 \subseteq \Rightarrow$, entonces $\Rightarrow_1$ es un T -condicional.

Demostración. Si $a \in T$ y $a \Rightarrow_1 b$, entonces $a \in T$ y $a \Rightarrow b$, luego $b \in T$. $\square$

Nada nos dice que un T -condicional fijo sea un preorden, es decir que sea reflexivo y transitivo, propiedades deseables si queremos modelizar la propagación de la verdad.

Si la relación $\Rightarrow$ no es reflexiva, puede extenderse a su clausura reflexiva

$$\Rightarrow_r = \Rightarrow \cup \{(a, a) ; a \in E\}$$

Obsérvese que $\Rightarrow \subseteq \Rightarrow_r$.

Teorema 2.9.

Una relación $\Rightarrow$ es un T -condicional si y sólo si $\Rightarrow_r$ es un T -condicional.

Demostración. Si $\Rightarrow_r$ es un T -condicional, por el lema anterior, también lo es $\Rightarrow$. Si $\Rightarrow$ es un T -condicional, $a \in T$ y $a \Rightarrow_r b$, entonces o bien $a = b$ y $b \in T$, o bien $a \Rightarrow b$ y $b \in T$, por ser $\Rightarrow$ un T -condicional. $\square$

Si la relación $\Rightarrow$ no es transitiva, también puede extenderse a su clausura transitiva

$$\Rightarrow_t = \Rightarrow \cup \{(a, b) ; \exists a_1, \dots, a_n \in E \text{ verificando } a \Rightarrow a_1 \Rightarrow \dots \Rightarrow a_n \Rightarrow b\}$$

También en este caso $\Rightarrow \subseteq \Rightarrow_t$.

Teorema 2.10.

Una relación $\Rightarrow$ es un T -condicional si y sólo si $\Rightarrow_t$ es un T -condicional.

Demostración. Por el lema 2.8., si $\Rightarrow_t$ es un T -condicional, también lo será $\Rightarrow$.

Si $\Rightarrow$ es un T -condicional, $a \in T$ y $a \Rightarrow_t b$, entonces existirán $a_1, \dots, a_n \in E$ tales que $a \Rightarrow a_1 \Rightarrow \dots \Rightarrow a_n \Rightarrow b$; de $a \in T$ y $a \Rightarrow a_1$, tenemos que $a_1 \in T$; de $a_1 \in T$ y $a_1 \Rightarrow a_2$, llegamos a $a_2 \in T$, y sucesivamente hasta $b \in T$. $\square$

Corolario 2.11.

Si $\Rightarrow$ no es reflexiva ni transitiva, podemos extender $\Rightarrow$ a $\Rightarrow_r$, y $\Rightarrow_{rt}$, con $\Rightarrow \subset \Rightarrow_r \subset \Rightarrow_{rt}$ y se verifica que $\Rightarrow$ es un T -condicional si y sólo si $\Rightarrow_{rt}$ es un T -condicional.

Además, $\mathcal{F}(\Rightarrow_{rt}) \subset \mathcal{F}(\Rightarrow_r) \subset \mathcal{F}(\Rightarrow)$.

III. EL CONDICIONAL MATERIAL.

Definición 3.1.- Dado un conjunto E y un subconjunto no vacío $T \subseteq E$, se define el Condicional Material $\rightarrow_T$ como la relación en E :

$$\rightarrow_T = (F \times E) \cup (T \times T)$$

donde $F = E - T$.

Es inmediato el

Lema 3.2.

El Condicional Material $\rightarrow_T$ es un T -condicional.

Lema 3.3.

El Condicional Material $\rightarrow_T$ es un preorden.

Demostración. La propiedad reflexiva es inmediata.

Para ver la transitiva, sea $a \rightarrow_T b$ y $b \rightarrow_T c$. Si $a \notin T$, entonces $a \rightarrow_T c$. Si $a \in T$, ha de ser $b \in T$ y por tanto $c \in T$; luego también en este caso es $a \rightarrow_T c$. $\square$

Teorema 3.4.

$\Rightarrow$ es un T -condicional si y sólo si $\Rightarrow \subseteq \rightarrow_T$.

Demostración. Supongamos primeramente que $\Rightarrow$ es un T -condicional. Si $a \Rightarrow b$, o bien $a \in F$, o bien $a \in T$ y $b \in T$, luego $(a, b) \in (F \times E) \cup (T \times T)$, y $a \rightarrow_T b$.

Sea ahora $\Rightarrow \subseteq \rightarrow_T$. Si $a \in T$ y $a \Rightarrow b$, entonces $a \in T$ y $a \rightarrow_T b$, con lo que ha de ser $b \in T$. $\square$

Por tanto, $\rightarrow_T$ es el mayor T -condicional en E .

Corolario 3.5.

Para toda relación $\Rightarrow$, $\Rightarrow \subseteq \bigcap_{T \in \mathcal{F}(\Rightarrow)} \rightarrow_T$.

Demostración. Si $T \in \mathcal{F}(\Rightarrow)$, $\Rightarrow$ es un T -condicional y $\Rightarrow \subseteq \rightarrow_T$. $\square$

Lema 3.6.

Si $\Rightarrow$ es un preorden sobre E , entonces $\Rightarrow = \bigcap_{T \in \mathcal{F}(\Rightarrow)} \rightarrow_T$.

Demostración. Nos falta ver $\Rightarrow \supseteq \bigcap_{T \in \mathcal{F}(\Rightarrow)} \rightarrow_T$.

Sea $a \rightarrow_T b$ para todo $T \in \mathcal{F}(\Rightarrow)$; supongamos $a \not\Rightarrow b$; llamamos

$$T_a = \{c \in E; \exists a_1, \dots, a_n \in E; a \Rightarrow a_1 \Rightarrow \dots \Rightarrow a_n \Rightarrow c\}$$

T_a es un estado lógico para $\Rightarrow$, luego $T_a \in \mathcal{F}(\Rightarrow)$; $a \in T_a$ por la reflexividad; sin embargo, $b \notin T_a$ ya que entonces, por la transitividad, sería $a \Rightarrow b$. Por tanto, no puede ser $a \rightarrow_{T_a} b$, en contra de las hipótesis. Así pues, ha de ser $a \Rightarrow b$. $\square$

Del Teorema 3.6 y del hecho de que toda intersección de preórdenes es un preorden, se deduce el siguiente

Teorema 3.7.

$\Rightarrow$ es un preorden sobre E , si y sólo si $\Rightarrow = \bigcap_{T \in \mathcal{F}(\Rightarrow)} \rightarrow_T$.

Teorema 3.8

$\mathcal{F}(\rightarrow_T) = \{T, E\}$. Es decir, dado un subconjunto no vacío $S \subseteq E$, $\rightarrow_T$ es un S -condicional si y sólo si $S = T$ o $S = E$.

Demostración. Está claro que si $S = T$ o $S = E$, $\rightarrow_T$ es un S -condicional.

Supongamos ahora que $S \neq E$ y $S \neq T$; si $\rightarrow_T$ fuese un S -condicional, por Teorema 3.4. habría de ser $\rightarrow_T \subset \rightarrow_S$; pero por ser $S \neq T$, o bien i) existe $a \in E$, con $a \in S$ y $a \notin T$, o bien ii) existe $a \in E$, con $a \in T$ y $a \notin S$.

i) en el primer caso, como $S \neq E$, existe $b \notin S$; por tanto, $(a, b) \in \rightarrow_T$ y $(a, b) \notin \rightarrow_S$; contradicción.

ii) por ser $S \neq \emptyset$, existe $b \in S$; $(b, a) \in \rightarrow_T$ y $(b, a) \notin \rightarrow_S$; contradicción. $\square$

IV. CONDICIONALES MONOTONOS.

La caracterización de preorden que nos da el Teorema 3.7. nos va a permitir estudiar la monotonía de las relaciones.

Definición 4.1.- Sea $(E, \cdot)$ con " $\cdot$ " operación binaria y conmutativa en E . Una relación $\Rightarrow$ definida en E se dice que es $\cdot$ -monótona, si $a \Rightarrow b$ implica $a \cdot c \Rightarrow b$ para todo c de E .

Utilizando funciones de pertenencia, esta propiedad es equivalente a: si $\varphi_{\Rightarrow}(a, b) = 1$, entonces $\varphi_{\Rightarrow}(a \cdot c, b) = 1$; o lo que es lo mismo: $\varphi_{\Rightarrow}(a, b) \leq \varphi_{\Rightarrow}(a \cdot c, b)$, para todo c de E .

Ejemplos 4.2.

1) La relación $\leq$ en un álgebra de Boole es siempre $\cdot$ -monótona, ya que si $a \leq b$, entonces para todo c de E se verificará $a \cdot c \leq a \leq b$.

2) En el primer ejemplo de la sección anterior, si $\cdot$ es la intersección del álgebra de Boole E , la relación $\Rightarrow$ no es $\cdot$ -monótona. En efecto, tomando $a, b, c \in E$ tales que $p(a) > 0$, $p(b/a) > 0$, $p(a \cdot c) > 0$ y $p(a \cdot b \cdot c) = 0$, obtenemos $a \Rightarrow b$, pero no $a \cdot c \Rightarrow b$.

3) En el segundo ejemplo, tomemos como μ una probabilidad. Si $a \Rightarrow b$, o bien

$p(a) = 0$ y por tanto $p(a \cdot c) = 0$ con lo que $a \cdot c \Rightarrow b$, o bien $p(a) > 0$ y $\frac{p(b)}{p(a)} > 0$; en este caso, si $p(a \cdot c) = 0$ entonces $a \cdot c \Rightarrow b$, y si $p(a \cdot c) > 0$, como $p(b) > 0$, $\text{Min}(1, \frac{p(b)}{p(a \cdot c)}) > 0$.

Teorema 4.3.

Si $\Rightarrow$ es un T -condicional reflexivo sobre E y $\cdot$ -monótono, entonces $a \cdot c \in T$ implica $a \in T$ y $c \in T$.

Demostración. Si $a \cdot c \in T$, como $a \Rightarrow a$ y $\Rightarrow$ es $\cdot$ -monótona, $a \cdot c \Rightarrow a$, y por ser T -condicional, $a \in T$.

De la misma forma, se demostraría $c \in T$. $\square$

Nota 4.4.- No existe un Teorema recíproco del 3.11., como muestra el ejemplo 1 de la sección 2: en el álgebra de Boole probabilizada (E, p) la relación $\Rightarrow$ de dicho ejemplo es reflexiva, ya vimos que no es $\cdot$ -monótona, y sin embargo, si $T \in \mathcal{F}(\Rightarrow)$ y $a \cdot c \in T$, como $a \cdot c \Rightarrow a$ y $a \cdot c \Rightarrow c$, se sigue que $a \in T$ y $c \in T$.

Teorema 4.5.

Si $\Rightarrow$ es una relación en E tal que $\Rightarrow = \bigcap_{T \in \mathcal{H}} \rightarrow_T$ siendo $\mathcal{H}$ un subconjunto de $\mathcal{F}(\Rightarrow)$, y para todo $T \in \mathcal{H}$ se verifica $a \cdot c \in T$ implica $a \in T$ y $c \in T$, entonces $\Rightarrow$ es $\cdot$ -monótono.

Demostración. Sea $a \Rightarrow b$ y supongamos $a \cdot c \not\Rightarrow b$, existirá $T \in \mathcal{H}$ con $a \cdot c \not\rightarrow_T b$, con lo que $a \cdot c \in T$ y $b \notin T$; luego $a \in T$ y $b \notin T$, $a \not\rightarrow_T b$ y $a \not\Rightarrow b$, llegando a una contradicción, con lo que ha de ser $a \cdot c \Rightarrow b$. $\square$

Teorema 4.6.

Un preorden $\Rightarrow$ en E es $\cdot$ -monótono si y sólo si para todo estado lógico T de $(E, \Rightarrow)$, $a \cdot c \in T$ implica $a \in T$ y $c \in T$.

Demostración. Inmediata, teniendo en cuenta que si $\Rightarrow$ es un preorden, $\Rightarrow = \bigcap_{T \in \mathcal{F}(\Rightarrow)} \rightarrow_T$. $\square$

Nota 4.7. Como el preorden $\Rightarrow$ del segundo ejemplo es $\cdot$ -monótono, en un álgebra de

Boole probabilizada (E, p) para cualquier $T \in \mathcal{F}(\Rightarrow)$ debe verificarse la condición del Teorema 3.11. Para $T = E$ es obvio. Si $T \subseteq E^+$ y $a \cdot c \in T$, $0 < p(a \cdot c) \leq p(a), p(c)$, $\text{Min}(1, \frac{p(a)}{p(a \cdot c)}) > 0$, $\text{Min}(1, \frac{p(c)}{p(a \cdot c)}) > 0$; entonces $a \cdot c \Rightarrow a$, $a \cdot c \Rightarrow c$; y siendo $a \cdot c \in T$, llegamos a $a, c \in T$.

Nota 4.8.- Como consecuencia de los resultados anteriores, el Condicional Material $\Rightarrow_T$ no siempre es $\cdot$ -monótono; será necesario y suficiente que se verifique la propiedad:

$$a \cdot c \in T \text{ implica } a \in T, b \in T$$

V. LA RELACION $\Rightarrow_T$ EN UN ALGEBRA DE BOOLE

Dado un conjunto E donde se han fijado la operación binaria $+$, y la unaria $'$, y un subconjunto $T \subseteq E$, podemos definir la relación $\Rightarrow_T$ como: $a \Rightarrow_T b$ si y sólo si $a' + b \in T$.

Primeramente, hemos de observar que si las operaciones $+$ y $'$ son tales que:

i) $a' \in T$ si y sólo si $a \in E - T$.

ii) $a + b \in T$ si y sólo si $a \in T$ o $b \in T$.

entonces las relaciones $\Rightarrow_T$ y $\rightarrow_T$ coinciden. En efecto, $a \Rightarrow_T b$ si y sólo si $a' + b \in T$, si y sólo si $a' \in T$ o $b \in T$, si y sólo si $a \in E - T$ o $b \in T$, si y sólo si $a \rightarrow_T b$, por la definición de Condicional Material.

Nuestro estudio se va a limitar a los conjuntos dotados de una estructura de Algebra de Boole $(E, +, \cdot, ')$. Sabemos que en este caso, la relación $\leq$ viene dada por: $a \leq b$ si y sólo si $a \cdot b = a$.

Primeramente, nos interesa conocer para qué subconjuntos T de E , la relación $\Rightarrow_T$ es un T -condicional.

Teorema 5.1.

La relación $\Rightarrow_T$ es un T -condicional si y sólo si para todo $a, b \in T$, con $a + b = 1$, el conjunto $V_{ab} = \{x \in E; a \cdot b \leq x \leq a \text{ o } a \cdot b \leq x \leq b\} \subseteq T$.

Demostración. Sea $\Rightarrow_T$ un T -condicional, y $a, b \in T$, con $a + b = 1$ (y por tanto, $b' \leq a$). Si $a \cdot b \leq x \leq a$, $a = a \cdot (b' + b) = a \cdot b' + a \cdot b \leq b' + x \leq a$; por tanto, $b' + x = a$,

$b \Rightarrow_T x$ y $x \in T$. De la misma forma, si $a \cdot b \leq x \leq b$, también será $x \in T$.

Recíprocamente, sea para todo $a, b \in T$, con $a + b = 1$, $V_{ab} \subseteq T$; si $a \in T$ y $a \Rightarrow_T c$, será $a \in T$ y $a' + c \in T$; por otra parte, $a + (a' + c) = 1$ y $a \cdot (a' + c) \leq c \leq a' + c$; por tanto, $c \in V_{a, a'+c} \subseteq T$ y $c \in T$. $\square$

Ya vimos que $\rightarrow_T$ es siempre un preorden, y es fácil demostrar que también lo es la relación $\leq$. ¿Qué ocurre con la relación $\Rightarrow_T$?

Lema 5.2.

$\Rightarrow_T$ es reflexiva si y sólo si $1 \in T$.

Demostración. $\Rightarrow_T$ es reflexiva si y sólo si para todo $a \in T$ $a \Rightarrow_T a$, lo que ocurre si y sólo si $a' + a = 1 \in T$. $\square$

Lema 5.3.

Si $\Rightarrow_T$ es un T -condicional reflexivo, $a \in T$ y $a \leq b$, entonces $b \in T$.

Demostración. Como $a \leq b$, $a' + b = 1 \in T$; por tanto $a \Rightarrow_T b$ y $b \in T$. $\square$

Lema 5.4.

Si $\Rightarrow_T$ es un T -condicional reflexivo, y $a, b \in T$, entonces $a \cdot b \in T$.

Demostración. $b = (a' + a) \cdot b = a' \cdot b + a \cdot b \leq a' + a \cdot b$ y por el lema 4.3., $a' + a \cdot b \in T$, $a \Rightarrow_T a \cdot b$ y $a \cdot b \in T$. $\square$

Dado $a \in E$, sea $V_a = \{x \in E; a \leq x\}$.

Lema 5.5.

Para todo $a \in E$, $\Rightarrow_{V_a}$ es un V_a -condicional.

Demostración. Si $b \in V_a$ y $b \Rightarrow_{V_a} c$, es $b' + c \in V_a$ y $a \leq b' + c$. Por $a \leq b$, $b' \leq a'$ y $a \leq b' + c \leq a' + c$; así, $a \leq c$. Por tanto, $c \in V_a$. $\square$

Teorema 5.6.

$\Rightarrow_T$ es un T -condicional reflexivo si y sólo si $T = V_a = \{x; a \leq x\}$ para algún $a \in E$.

Demostración. Los lemas 5.3. y 5.4. nos dan la condición necesaria, mientras que del lema 5.5. obtenemos la suficiente. $\square$

Lema 5.7.

Para todo $a \in E$, $\Rightarrow_{V_a}$ es transitiva.

Demostración. Debemos probar que si $b \Rightarrow_{V_a} c$ y $c \Rightarrow_{V_a} d$, entonces $b \Rightarrow_{V_a} d$.

Si $b \Rightarrow_{V_a} c$ y $c \Rightarrow_{V_a} d$, tenemos $b' + c \in V_a$ y $c' + d \in V_a$ y así, $a \leq b' + c$ y $a \leq c' + d$.

Para cualquier átomo h de E tal que $h \leq a$, es $h \leq c$ o $h \leq c'$.

Si $h \leq c$, $h \not\leq c'$ y por ser $a \leq c' + d$, $h \leq d$; así, $h \leq b' + d$.

Si $h \leq c'$, $h \not\leq c$ y por ser $a \leq b' + c$, $h \leq b'$; así, $h \leq b' + d$.

Por tanto, en cualquier caso es $h \leq b' + d$ para cualquier átomo h con $h \leq a$. Así, $a \leq b' + d$ y $b' + d \in V_a$. $\square$

Corolario 5.8.

Los únicos subconjuntos $T \subseteq E$ para los que $\Rightarrow_T$ es un preorden y T -condicional, son los $V_a = \{x; a \leq x\}$ para cualquier $a \in E$. Por tanto, existen $\text{Card}(E)$ subconjuntos de este tipo.

Teorema 5.9.

$\Rightarrow_T$ es $\cdot$ -monótono si y sólo si la relación $\leq$ es un T -condicional.

Demostración. Sea $\Rightarrow_T$ $\cdot$ -monótono. Si $a \in T$ y $a \leq b$, $0 + a = a \in T$ y $1 \Rightarrow_T a$.

Así, por la monotonía $b' \cdot 1 \Rightarrow_T a$, $b + a \in T$ y $b = b + a \in T$.

Recíprocamente, sea $\leq$ un T -condicional; si $a \Rightarrow_T b$, $a' + b \in T$. Entonces, para todo $c \in E$, $a' + b \leq a' + b + c'$ y $a' + b + c' \in T$; así, $(a \cdot c)' + b \in T$ y $a \cdot c \Rightarrow_T b$. $\square$

Corolario 5.10

Si $\Rightarrow_T$ es $\cdot$ -monótono, entonces $\Rightarrow_T$ es reflexiva.

Demostración. Si $\Rightarrow_T$ es $\cdot$ -monótono, $\leq$ es un T -condicional y $1 \in T$; por el lema 4.2.

$\Rightarrow_T$ será reflexiva. $\square$

Observemos por tanto, que los únicos subconjuntos $T \subseteq E$ para los que $\Rightarrow_T$ es un T -condicional $\cdot$ -monótono son los $T = V_a$ para algún $a \in E$.

VI. COMPARACION DE ALGUNOS CONDICIONALES EN UN ALGEBRA DE BOOLE.

Dada un Algebra de Boole $(E, +, \cdot, ')$, y un subconjunto $T \subset E$, hemos definido tres relaciones: $\rightarrow_T$, $\Rightarrow_T$, $\leq$. En el caso en que $T = V_a = \{x \in E; a \leq x\}$ para algún $a \in E$, las tres son T -condicionales. ¿Qué relación existe entre ellas?

Comparando las dos primeras, por el Teorema 3.4. $\Rightarrow_{V_a} \subset \rightarrow_{V_a}$. ¿Cuándo se da la inclusión recíproca?

Lema 6.1.

$\rightarrow_{V_a} \subset \Rightarrow_{V_a}$ con $a \in E$, si y sólo si $a = 0$ o a es un átomo de E .

Demostración. Sea $\rightarrow_{V_a} \subset \Rightarrow_{V_a}$. Supongamos que a no es ni 0 , ni un átomo de E . Elijamos h, l átomos con $h, l < a$; será $h \notin V_a$ y $l \notin V_a$, $h' + l = h' \not\leq a$; por tanto, $h' + l \notin V_a$, $h \not\rightarrow_{V_a} l$ y sin embargo, $h \Rightarrow_{V_a} l$, llegando a una contradicción con la hipótesis.

Recíprocamente, si a es un átomo de E y $b \rightarrow_{V_a} c$, tenemos dos casos:

- i) $b \notin V_a$; $a \not\leq b$, $a \leq b'$, $a \leq b' + c$; así, $b' + c \in V_a$ y $b \Rightarrow_{V_a} c$.
- ii) $b, c \in V_a$; $a \leq c \leq b' + c$; $b' + c \in V_a$ y $b \Rightarrow_{V_a} c$.

Finalmente, si $a = 0$, $V_a = E$ y para todo $a, b \in E$, $a \rightarrow_{V_a} b$ y $a \Rightarrow_{V_a} b$. $\square$

Corolario 6.2.

Las relaciones $\rightarrow_{V_a}$ y $\Rightarrow_{V_a}$ coinciden si y sólo si a es 0 o a es un átomo de E .

Lema 6.3.

Si $1 \in T$, entonces $\leq \subset \Rightarrow_T$.

Demostración. Si $a \leq b$, $a' + b = 1 \in T$; luego, $a \Rightarrow_T b$. $\square$

Corolario 6.4.

$\leq \subset \Rightarrow_{V_a}$ para todo $a \in E$.

Por tanto, se tiene: $\leq \subset \Rightarrow_{V_a} \subset \rightarrow_{V_a}$.

Lema 6.5.

$\Rightarrow_{V_a} \subset \leq$ si y sólo si $a = 1$.

Demostración. Supongamos $a = 1$; $V_a = 1$; si $c \Rightarrow_{V_a} b$, $c' + b \in V_a$; $c' + b = 1$, y $c \leq b$. Por otra parte, si $a \neq 1$, como $a + a = a \in V_a$, y $(a')' = a$, tenemos que $a' \Rightarrow_{V_a} a$ y sin embargo, $a' \not\leq a$; por tanto, no será $\Rightarrow_{V_a} \subset \leq$. $\square$

Corolario 6.6.

Si $E \neq \{0,1\}$, el contenido $\leq \subset \rightarrow_{V_a}$ es estricto para todo $a \in E$.

Demostración. Si fuese $\leq = \rightarrow_{V_a}$, por el corolario 6.4., $\leq = \Rightarrow_{V_a} = \rightarrow_{V_a}$.

Por ser $\leq = \Rightarrow_{V_a}$, ha de ser $a = 1$, y por ser $\Rightarrow_{V_a} = \rightarrow_{V_a}$, obtenemos $a = 0$ o a es un átomo. Contradicción. $\square$

REFERENCIAS

[1] TRILLAS, E. *On Exact Conditionals*, en curso de publicación en STOCHASTICA (1992).

[2] TRILLAS, E., CUBILLO, S. *On Monotonic Fuzzy Conditionals*, presentado para su publicación en el INTERNATIONAL JOURNAL OF NON-CLASSICAL APPLIED LOGICS (1992).

[3] TRILLAS, E. *Some reflexions on Logic and Fuzzy Logic* , conferencia inaugural del JOINT JAPANESE-EUROPEAN SYMPOSIUM ON FUZZY SYSTEMS. Manuscrito. (Berlín, 1992).

[4] ZADEH, L.A. *Fuzzy Sets*, INFORMATION AND CONTROL 8, pp.338-353 (1965).

EN TORNO A LA CONDICION DE IDEMPOTENCIA

Gaspar Mayor y Joan Torrens
Depto. de Matemáticas
e Informática
Universitat de les Illes Balears
Palma de Mallorca (España)

Resumen: Este trabajo tiene como principal objetivo el estudio del « comportamiento idempotente » de una amplia clase de operaciones binarias en el intervalo unidad, dedicando especial atención a las obtenidas por agregación lineal convexa de una t-norma y una t-conorma (arquimedianas). También se consiguen resultados satisfactorios, en la búsqueda de operaciones idempotentes, si nos movemos en el ámbito de las funciones representables en el sentido de Łukasiewicz.

Palabras Clave: intervalo unidad, operación idempotente, t-norma, t-conorma, negación, función de agregación, ecuación funcional.

1. Introducción

Desde que G. Boole publicó en 1847 "The Mathematical Analysis of Logic", obra en la que por primera vez se planteaban ecuaciones (lógicas) del tipo $xx = x$, gran número de investigadores estudiosos de la lógica clásica y no clásicas han dedicado esfuerzos en el estudio de los conectivos lógicos.

A partir de la introducción por L. Zadeh (1965) de los conectivos Min y Max en la teoría de los conjuntos difusos, se ha escrito mucha literatura en torno a la utilización de conectivos del tipo t-norma y t-conorma (operaciones asociativas pero no idempotentes salvo las ya citadas Min y Max) (véase [2]). Más recientemente, a través de investigaciones en diferentes contextos, se han utilizado otro tipo de operaciones (en general no asociativas) con resultado satisfactorio (véase [10]).

Nuestro trabajo se sitúa en el estudio de este tipo de conectivos "alternativos" y en especial en aquellos que mantienen la propiedad de idempotencia ya introducida en la lógica de Boole. No obstante nos hemos servido principalmente de t-normas y t-conormas (arquimedianas) para construir, en base a determinados tipos de agregación, nuestras operaciones idempotentes.

2. Definiciones y resultados básicos.

Definición 1. Una t-norma (t-conorma) es una operación binaria en el intervalo unidad $[0,1]$ que es asociativa, conmutativa, con neutro 1 (con neutro 0) y creciente en cada variable.

Definición 2. Una t-norma T (t-conorma S) es arquimediana si es continua (en cada variable) y verifica $T(x,x) < x$ ($S(x,x) > x$) para todo $x \in (0,1)$.

Definición 3. Una t-norma (t-conorma) es estricta si es continua y estrictamente creciente en $(0,1]^2$.

Definición 4. Una negación es una función de $[0,1]$ en $[0,1]$ que es involutiva y decreciente.

Si T es una t-norma y N una negación, podemos considerar la operación N -dual de T , T^* , definida por: $T^*(x,y) = NT(N(x),N(y))$. T^* es, obviamente, una t-conorma. Se dice que (T,T^*,N) es una terna de De Morgan.

Ejemplo 1. Son especialmente importantes las siguientes ternas (con $N(x) = 1-x$):

$$\begin{array}{ll} T(x,y) = \text{Min}(x,y) & T^*(x,y) = \text{Max}(x,y) \\ T(x,y) = xy & T^*(x,y) = x + y - xy \\ T(x,y) = \text{Max}(0, x+y-1) & T^*(x,y) = \text{Min}(1, x+y) \end{array}$$

Teorema 1. Una operación T es una t-norma arquimediana estricta si, y sólo si, existe $f: [0,1] \rightarrow [0,+\infty]$ estrictamente decreciente, continua con $f(0) = +\infty$ y $f(1) = 0$ tal que se verifica $T(x,y) = f^{-1}(f(x) + f(y))$ para todo $x,y \in [0,1]$.

Teorema 2. Una operación T es una t-norma arquimediana no estricta si, y sólo si, existe $f: [0,1] \rightarrow [0,+\infty)$ estrictamente decreciente, continua con $f(1) = 0$ y tal que $T(x,y) = f^{[-1]}(f(x) + f(y))$ para todo $x,y \in [0,1]$. La función $f^{[-1]}$ (seudoinversa de f) viene definida por: $f^{[-1]}(t) = f^{-1}(t)$ si $t \in [0,f(0)]$ y $f^{[-1]}(t) = 0$ si $t > f(0)$.

La función f que aparece en estos dos teoremas se llama generador aditivo de la t-norma, y es única a menos de una constante multiplicativa positiva. Decimos que $N(x) = f^{-1}(f(0) - f(x))$ es la negación asociada a la t-norma, verificándose que $T(x,y) = 0$ si, y sólo si, $y \leq N(x)$. Estos resultados, que permiten la manipulación de las t-normas arquimedianas a partir de funciones de una variable, son fundamentales en nuestro trabajo y aparecen en [5].

Definición 5. Una función de agregación lineal (FAL), F , es una operación binaria en $[0,1]$ que es conmutativa, creciente en cada variable y verifica:

$$F(x,0) = F(1,0)x, \quad F(x,1) = (1-F(1,0))x + F(1,0) \quad \text{para todo } x \in [0,1]$$

Ejemplo 2. Las t-normas y t-conormas son funciones de agregación lineales. De hecho las t-normas y t-conormas son las únicas funciones de agregación lineales asociativas. Otros ejemplos de FAL se consiguen al formar combinaciones lineales convexas de una t-norma y una t-conorma $((1-k)T + kS, \text{ con } k \in (0,1))$. Un

ejemplo de FAL distinto de los citados puede ser $F(x,y) = \text{Min}(xy + \text{Max}(x,y), 1)$. Para más detalles sobre FAL véase [7].

A continuación se define la familia, $\mathcal{L}$, de funciones $\mathcal{L}$ -representables (Lukasiewicz-representables). $\mathcal{L}$ es una subfamilia de FAL que contiene la clase de las t-normas arquimedianas no estrictas y en particular la t-norma de Lukasiewicz $W(x,y) = \text{Max}(0, x+y-1)$.

Definición 6. Una operación binaria en $[0,1]$, F , es $\mathcal{L}$ -representable si es de la forma $F(x,y) = f^{-1}[\text{Max}(0, f(\text{Min}(x,y)) - fN(\text{Max}(x,y)))]$, siendo N una negación y $f: [0,1] \rightarrow [0,+\infty]$ estrictamente creciente, continua y $f(0) = 0$.

Indicaremos $F = \langle (f,N) \rangle$ y diremos que (f,N) es un par generador de F . Estas operaciones son arquimedianas (continuas y $F(x,x) < x$ para todo $x \in (0,1)$). Un resultado importante en el estudio de estas operaciones es el siguiente: $F = \langle (f,N) \rangle$ es una t-norma si, y sólo si, $f(1) < +\infty$ y $N(x) = f^{-1}(f(1) - f(x))$ para todo $x \in [0,1]$. Esta clase de operaciones han sido estudiadas en [8].

Teorema 3. Si $k \in (0,1)$, la única solución $R: [0,1] \rightarrow [0,1]$ del sistema de ecuaciones funcionales

$$\begin{aligned} R\left(\frac{x}{2}\right) &= k R(x) \\ R\left(\frac{x+1}{2}\right) &= (1-k) R(x) + k \end{aligned} \quad 0 \leq x \leq 1 \quad (\text{DR}[k])$$

viene dada por

$$R(x) = \sum_{n=1}^{\infty} \left(\frac{1-k}{k}\right)^{n-1} k^{\alpha_n} \quad \text{si } x = \sum_{n=1}^{\infty} \frac{1}{2^{\alpha_n}} \quad \text{con } 0 < \alpha_1 < \alpha_2 < \dots \quad \text{enteros}$$

$$R(0) = 0$$

Esta función es estrictamente creciente con $R(1) = 1$ y continua. Por otra parte, si $k \neq 1/2$ entonces su derivada es cero casi por todo en $[0,1]$. Si $k = 1/2$, entonces $R(x) = x$ para todo $x \in [0,1]$.

Este teorema es, también, básico para nuestro estudio. Para más detalles véase [3].

Definición 7. Sea $k \in (0,1)$, una negación N es una k -negación si verifica la condición $N(kx) = (1-k)N(x) + k$ para todo $x \in [0,1]$. Véase [6].

Ejemplo 3. Se demuestra fácilmente que para $k = 1/2$ la única $1/2$ -negación es $N(x) = 1 - x$ ($N = 1-j$).

Teorema 4. Para cada $k \in (0,1)$ existe una única k -negación, ésta puede ser expresada mediante $N(x) = R(1 - R^{-1}(x))$ siendo R la solución del sistema de De Rham correspondiente al valor k .

Definición 8. Si $g: [0,1] \rightarrow [0,1]$ es estrictamente creciente, continua, con $g(0) = 0$ y $g(1) = 1$, diremos que

$$Q(x,y) = g^{-1} \left(\frac{g(x) + g(y)}{2} \right)$$

es una media casi aritmética (definida por g).

3. La condición de idempotencia.

Una operación binaria en $[0,1]$, H , diremos que verifica la condición de idempotencia, o simplemente que es idempotente, si

$$H(x,x) = x \quad \text{para todo } x \in [0,1].$$

3.1. Caso de t-normas, t-conormas y medias casi aritméticas.

Es bien sabido que en este caso la situación es la siguiente:

Teorema 5.

- a) La única t-norma idempotente es la t-norma Min.
- b) La única t-conorma idempotente es la t-conorma Max.
- c) Todas las medias casi aritméticas son idempotentes.

3.2. Caso $H = (1-k)T + kT^*$

$k \in (0,1)$; T t-norma arquimediana no estricta; T^* N_T -dual de T

En este supuesto, el resultado que hemos obtenido es el siguiente:

Teorema 6. H es idempotente si, y sólo si, el generador aditivo normalizado ($f(0) = 1$) de T es $f = 1 - R^{-1}$, siendo R la solución del sistema $DR[k]$.

Observaciones / Comentarios:

- 1) De acuerdo con este teorema, para cada $k \in (0,1)$ existe una única operación H de la forma exigida.
- 2) La t-norma de que se habla en el teorema tiene como negación asociada la k -negación.
- 3) Para $k = 1/2$, entonces $H = 1/2 (T + T^*)$, y resulta que la única operación idempotente de este tipo es $H(x,y) = 1/2 (x + y)$.
- 4) Para cada $k \in (0,1)$, la operación idempotente de que habla el teorema puede expresarse en la forma

$$H(x,y) = R \left(\frac{R^{-1}(x) + R^{-1}(y)}{2} \right)$$

así que H es una media casi aritmética y por tanto bisimétrica ($H(H(x,y), H(u,v)) = H(H(x,u), H(y,v))$).

3.3. Caso $H = (1-k)T + kT^*$

$k \in (0,1)$; T t-norma arquimediana estricta ; T^* $(1-j)$ dual de T

En este caso, el resultado conseguido es:

Teorema 7.

- a) Si H es idempotente, entonces $k = 1/2$.
- b) Sea $H = 1/2 (T + T^*)$. H es idempotente si, y sólo si, $d(x) - d(1-x) = 2x - 1$ para todo $x \in [0,1]$ (condición δ), siendo $d(x) = T(x,x)$ (función diagonal de T).

Observaciones / Comentarios:

- 1) La ecuación δ admite infinitas soluciones (con $d: [0,1] \rightarrow [0,1]$ estrictamente creciente, continua, con $d(0) = 0$, $d(1) = 1$ y tal que $d(x) < x$ para todo $x \in (0,1)$). Concretamente, las soluciones vienen dadas por:

$$d(x) = d_0(x) \quad \text{si } x \in [0, 1/2]$$

$$d(x) = d_0(1-x) + 2x - 1 \quad \text{si } x \in [1/2, 1]$$

siendo $d_0: [0, 1/2] \rightarrow [0, 1]$ estrictamente creciente, continua, $d_0(0) = 0$ y $d_0(x) < x$ para todo $x \in (0, 1/2]$. Por lo demás, d_0 es arbitraria.

- 2) De acuerdo con este teorema, podemos construir operaciones idempotentes del tipo $H = 1/2 (T + T^*)$ sin más que fijar una función d que sea solución de δ y a continuación hallar funciones $f: [0,1] \rightarrow [0, +\infty]$ estrictamente decrecientes, continuas, con $f(0) = +\infty$ y $f(1) = 0$ tales que verifiquen $f(d(x)) = 2f(x)$ para todo $x \in [0,1]$. Las soluciones de esta ecuación (de Schröder) generan t-normas arquimedianas estrictas, T , con diagonal d , y por tanto, $H = 1/2 (T + T^*)$ es idempotente.

- 3) A diferencia del supuesto 3.2., la situación actual ($k = 1/2$) permite, en principio, la construcción de gran número de operaciones idempotentes ya que así se deduce de la resolución de la ecuación de Schröder (véase [4]).

3.4. Caso $H = (1-k)F + kF^*$

$k \in (0,1)$; $F = \langle (f, N) \rangle$; F^* N - dual de F

En este caso, que comprende en particular la situación 3.2., se ha conseguido demostrar:

Teorema 8.

- a) Si H es idempotente, entonces N es la k -negación.
- b) Sea N k -negación. H es idempotente si, y sólo si, $f = \phi \circ R^{-1}$ siendo R la solución del sistema $DR[k]$ y $\phi: [0,1] \rightarrow [0, +\infty]$ estrictamente creciente, continua, con $\phi(0) = 0$ y tal que $\phi(1-x) + \phi(x/2) = \phi(1 - x/2)$ para todo $x \in [0,1]$.

Observaciones / Comentarios :

- 1) La resolución (algo tediosa) de la ecuación restringida de Cauchy que aparece en este teorema nos proporciona una infinidad de funciones ϕ en las condiciones establecidas.
- 2) Al tomar $\phi = j$ se obtienen las operaciones idempotentes de que habla el teorema 6.

4. Comentario final.

De acuerdo con los resultados obtenidos, tenemos a nuestra disposición una amplia gama de operaciones idempotentes, entresacadas de la clase FAL, que pueden ser utilizadas (una vez comprobada su adecuación) como conectivos lógicos. Vale la pena observar que las operaciones conseguidas lo han sido en base a agregaciones lineales de operaciones arquimedianas. Quedan por estudiar otras situaciones que puedan generar otras familias de operaciones idempotentes.

5. Referencias.

- [1] Aczél, J., Lectures on functional equations and their applications. Academic Press, New York-London, 1966.
- [2] Alsina, C., Trillas, E. y Valverde, L., On some logical connectives for fuzzy sets theory. Jour. Math. An. and Appl. 92-1 (1983), 15-26.
- [3] De Rham, G., Sur quelques courbes définies par des equations fonctionelles. Rend. Sem. Mat. dell'Univ. e del Politecnico di Torino 16 (1956), 101-113.
- [4] Kimberling, C., On a class of associative functions. Publ. Math. Debrecen 20 (1973), 21-39.
- [5] Ling, C.H., Representation of associative functions. Publ. Math. Debrecen 12 (1965), 189-212.
- [6] Mayor, G. y Torrens, J., Duality for a class of binary operations on $[0,1]$. Fuzzy Sets and Systems 47 (1992), 77-80.
- [7] Mayor, G. y Trillas, E., On the representation of some aggregation functions. Proc. ISMVL'86, 110-114.
- [8] Torrens, J., Estudi de models matemàtics de connectius en lògica multivaluada. Tesis doctoral. U.I.B. 1990.
- [9] Zadeh, L.A., Fuzzy Sets. Inf. and Control 8 (1965), 338-353.
- [10] Zimmermann, H.J. y Zysno, P., Latent connectives in human decision making. Fuzzy Sets and Systems 4 (1980), 37-51.

CONTEXTOS Y EXPERTOS.

(Un intento de caracterizar formalmente dos nociones importantes en lógica borrosa e inteligencia artificial).

A. Sobrino.
Area de Lógica y Filosofía de la Ciencia.
Campus Universitario, s. n.
16706. Santiago de Compostela.

0. RESUMEN.

El objetivo de este trabajo se centra en el intento de caracterizar de un modo más concreto dos conceptos muy familiares para la lógica borrosa y sus aplicaciones: la noción de contexto, útil en la caracterización de dominios restringidos de los que hace uso normalmente esta lógica y el concepto de 'experto', presente en el diseño de prototipos de sistemas expertos.

Palabras clave: Vaguedad, Lógica borrosa, Contexto, Experto.

I. ANTECEDENTES.

Como es sabido, una forma habitual de caracterizar a la lógica borrosa es indicar que es una *lógica local*; esto es, una lógica en la que la evaluación de las oraciones se hace normalmente respecto a un dominio de objetos muy restringido [Bellman & Zadeh, 1976]. Esta característica tiene una señalada vinculación con sus primigenias huellas de identidad: lógica orientada computacionalmente, en ella lo que prima es reflejar directamente algunas características significativas (y de significado) del lenguaje humano, como la vaguedad o la imprecisión de sus oraciones.

Pues bien, dado que hay un cierto consenso en que la naturaleza del lenguaje es más bien vaga que precisa, y puesto que el manejo de información vaga por parte de los humanos constituye una propiedad especie-específica, una lógica orientada para la I. A. debe ser capaz de simular razonamientos o inferencias que incluyan oraciones vagas. Esta reflexión, que hoy a fuerza de ser repetida parece sencilla, fue el *leit motiv* de la lógica borrosa y, por ende, de una parte importante de la I. A.: aquella que trata de simular con artefactos mecánicos situaciones mal definidas o deficientemente caracterizadas. Esta era la idea germinal a partir de la cual desarrolló Zadeh su teoría de subconjuntos difusos [Zadeh, 1965].

El objetivo de la lógica borrosa es, por tanto, simular con un formalismo adecuado las oraciones vagas. Esta simulación se hace desde una concepción de filosofía del lenguaje determinada. Frente a la idea de Frege de que el lenguaje es preciso y de que la vaguedad oscurece la verdadera estructura de las oraciones, con Russell [Russell, 1923] se afianza la convicción de que el lenguaje humano es fundamentalmente vago. La lógica, sin embargo, no debe ocuparse en opinión de Russell del lenguaje vago: su objeto de estudio se sitúa en el reino celestial de los significados precisos, no en la vaguedad del lenguaje cotidiano. Pero si no se

quiere adoptar una actitud de 'avestruz' y se reconoce que la presentación usual del lenguaje incluye, entre otros, aspectos significativos vagos, entonces deben proporcionarse soluciones formales específicas para su correcta estructuración conforme a los cánones lógicos [Sobrino, 1989]. Este paso lo dio Black [Black, 1963], al proponer los perfiles de consistencia, antecedente histórico de las funciones de pertenencia generalizadas de Zadeh.

Dado que la lógica borrosa es una lógica orientada computacionalmente para el razonamiento borroso, la captura de la vaguedad del lenguaje debe ser el primer objetivo, constituyendo el desarrollo formal de la propia lógica borrosa (su metalógica), una cuestión periférica [Belmann & Zadeh, 1976]. El análisis lógico del lenguaje ordinario pasa así a tener un papel primordial.

Las restricciones que la lógica borrosa se autoimpone para un correcto análisis lógico del lenguaje vago son de diversa índole. Aquí hemos tratado de sistematizarlas en tres apartados,

(1.1). Restricciones sobre el dominio de objetos.

Al manifestar una vocación computacional, los problemas a los que la lógica borrosa se enfrenta y gestiona plausiblemente bien son aquellos que, como indicábamos al principio, tienen un contexto muy restringido o un dominio más bien pequeño y bien definido. Por tanto, en esta lógica, el concepto de universo de discurso juega un papel primordial y, dentro de este, la noción de núcleo (kernel) indica en que porción del universo de discurso focalizamos nuestra atención cuando expresamos un estado de hechos de una manera vaga. La importancia del concepto de núcleo queda reflejada en la siguiente característica: en la teoría acerca de lo borroso no se suele hablar de 'conjuntos borrosos', sino de 'subconjuntos borrosos'; esto es, de una elección concreta de objetos sobre el universo de discurso propuesto.

Ejemplo: Sea el Universo de los números. Deseamos evaluar en qué medida un número, por ej. 10, es grande. (esto es, deseamos evaluar o caracterizar de alguna forma el significado de la oración '10 es un número grande'). Si no introdujésemos el matiz 'grande' -predicado vago-, el papel del universo sería secundario y la evaluación de la oración '10 es un número' sería clara; esto es, podría decirse -matizaremos a continuación un poco más en qué sentido- que su significado es 'verdadero'.

Pero la valoración de una oración que contiene predicados vagos, como es el caso señalado arriba, sería imposible si antes no se indicase en qué subconjunto de los números hemos puesto especial énfasis o se nos ha obligado a prestar especial atención. Cualquier respuesta acerca de la evaluación de la cuestión ¿Es 10 un número grande? está condicionada por el núcleo que se haya considerado.

La influencia en la elección del núcleo para la representación formal del significado de una oración vaga es enorme y fácilmente mostrable. Adoptando el principio fregeano de que el significado de una oración es su referencia y ésta es un valor de verdad, (en el caso de la lógica borrosa, a diferencia de la fregeana, los valores pueden ser muchos y de naturaleza distinta- como, p. ej., es el caso de los valores de verdad lingüísticos-) puede observarse fácilmente como cambia la valoración de una oración según el núcleo seleccionado.

Ejemplo: Sea el $U = \{\text{números naturales}\}$, $N_1 = \{0, \dots, 10\}$ y el predicado vago ‘ser número grande’. Entonces 10 es número grande con grado 1. Es decir, es ‘absolutamente verdadero’ que 10 es grande -por utilizar un valor de verdad lingüístico-. En cambio, si $N_2 = \{0, \dots, 100\}$ el significado de ‘10 es un número grande’ disminuye hasta 0.10. Otra vez, usando rótulos lingüísticos, se podía decir que es ‘casi absolutamente falso’ que 10 sea un número grande. Obsérvese como cambia la valoración de la misma oración en función del N escogido.

(1.2.). Restricciones sobre el lenguaje.

La lógica borrosa no intenta reflejar formalmente el significado de cualquier término lingüístico, aunque el uso de procedimientos borrosos en otras lógicas le ha dado un buen margen de generalidad. Como es sabido, se fija sólo en una clase seleccionada de los mismos: aquellos que matizan o hacen más cauto a nuestro discurso. Y es la matización y la cautela de nuestras oraciones un criterio sin duda a tener en cuenta a la hora de ‘medir’ la racionalidad que transmiten, sobre todo si lo que se trata de mensurar es su racionalidad común. Tres son las entidades lingüísticas en las que la lógica borrosa se fija de una forma específica:

(1.2.1). Los predicados vagos.

Del predicado vago, en una definición no formal particularmente interesante, puede decirse que es aquella entidad lingüística que da nombre a un subconjunto borroso [Trillas, 1991]. (No hay en esta definición relación alguna de preferencia: puede invertirse el orden e indicar que un subconjunto borroso intenta ser una buena caracterización formal de un predicado vago). Como ya se ha indicado, el predicado vago matiza el significado del concepto al que acompaña. Así, de ‘grande’ se puede decir que hace flexible el significado de ‘número’, al que acompaña. En efecto, ‘grande’ matiza o cualifica a ‘número’ en ‘número grande’ y ello se refleja, por supuesto, en la representación de su significado.

Ejemplo: Sea $U = \{\text{números naturales}\}$, $N_1 = \{0, \dots, 10\}$ y $N_2 = \{0, \dots, 100\}$. La oración ‘10 es un número’ tiene siempre el mismo valor de verdad, esto es, 1 o ‘absolutamente verdadero’ tanto en N_1 como en N_2 . Para cualquier otro núcleo imaginable (con sentido), el valor de verdad de la oración de arriba sería bivaluado; esto es, 1 o 0. Esto indica que el predicado objeto de análisis no tiene ninguna restricción sobre sí. Una consecuencia de la ausencia de restricción es que, por ser tan genérico, puede decirse que el predicado ‘ser número’ tiene escaso contenido informativo.

Los predicados precisos, por tanto, y en contra de lo que desde la tradición verificacionista [para una visión crítica vease Waismann, 1945] se había intentado transmitir, tienen en ocasiones deficiencias para transmitir significados deseados. Estas surgen, como indica Popper [Popper, 1974], cuando se quiere precisar un término más de lo que es posible (no dudando que su precisión, si ésta es obtenida de manera natural, es beneficiosa) o cuando no se desea especificar más su significado.

Ejemplo: El concepto de ‘número’ es, de no cualificarlo con alguna restricción, - como indicábamos antes- genérico y poco informativo. Decir que ‘10 es un número’ transmite

menos información que decir '10 es un número entero', siendo 'entero' una restricción precisa del predicado 'ser número'. Decir que '10 es un número entero grande' supone, además de la informatividad que introduce la cualificación 'entero', la ampliación de información que genera, por su laxitud, el predicado vago 'grande'.

Por otra parte, y aludiendo al comentario anterior, según en qué contextos puede ser mucho más informativa la oración vaga: 'el nivel de agua ha crecido considerablemente' (relativa a una presa), y no la oración precisa 'el nivel de agua ha subido en 8075 cm³'. Nótese que hemos introducido en el ejemplo la palabra 'contexto' y no 'universo'. Algunos desarrollos ulteriores matizarán en qué sentido el primer concepto puede ayudar a caracterizar mejor al segundo concepto.

(1.2.2.) Los modificadores o cercas.

Otra restricción bien conocida, que acentúa el carácter local de la lógica borrosa, es el que se introduce a través de los modificadores lingüísticos, términos que constriñen o matizan más el significado, no de cualquier predicado, sino de los predicados vagos. Por usar una metáfora, podría decirse que son parásitos exclusivos de esa especie de términos.

Ejemplo. Sea la oración, '10 es un número entero extraordinariamente grande', siendo 'extraordinariamente' el modificador de grande. Si es bueno el principio de que a mayor compromiso, mayor informatividad, obsérvese como con la introducción de 'extraordinariamente' se logra matizar más el significado de la oración '10 es un número entero grande', ya que el compromiso del hablante con su interlocutor es más acusado cuando concreta un poco más una oración con una referencia ya de por sí muy matizada.

Convencionalmente, en lógica borrosa se ha adoptado el criterio de que si el predicado vago (P) -sobre el núcleo escogido- tiene un valor de verdad μ , 'extraordinariamente P' tiene el valor de verdad μ^3 y esta potencia parece que, conforme a la teoría fregeana de reducir el significado a un valor de verdad, representa bastante bien el contenido semántico de 'extraordinariamente P'. En efecto, si $N = \{0, 100\}$, y la oración '10 es un número grande' tiene valor de verdad 0.1, '10 es un número extraordinariamente grande' tendrá un valor de verdad menor, ya que 'extraordinariamente' acentúa las propiedades que tendría que tener 10 para juzgarlo como número grande. Siguiendo la fórmula indicada antes, su valor de verdad sería, en efecto, 0.001. Otras cercas o modificadores son bien conocidos del lector, por lo que no damos más información acerca de ellas aquí.

(1.3). Restricciones mixtas (sobre el lenguaje -que a su vez restringen el dominio-).

Por fin, una última acotación lingüística en lógica borrosa, que genera una forma peculiar de tratar las propiedades cuyas extensiones son, posiblemente con matices, elementos del dominio elegido, es la cuantificación aproximada, que matiza el alcance de los cuantificadores clásicos 'todo' o 'algunos', bien porque abarquen al total de las entidades (todos), bien porque, abarcando a una parcela de las mismas, sean demasiado inespecíficos o

poco concretos ('algunos', 'al menos uno'). La lógica borrosa provee a este respecto herramientas para gestionar formalmente cuantificadores intermedios, de los que son ejemplo habitual, 'pocos', 'la mayoría', 'una buena parte', 'casi todos', etc.

Ejemplo: Sea el dominio $N = \{0, \dots, 10\}$ y P un predicado (borroso o preciso), definido sobre N . Pues bien '*la mayoría* de los x , ($x \in N$), son P ', permite una reevaluación de P en función de la restricción introducida por el cuantificador '*la mayoría*'. Definido el significado de '*la mayoría*', normalmente mediante un subconjunto borroso, '*la mayoría* de los x son P ' alcanza una caracterización inmediata. P. ej., si el significado de '*la mayoría*' queda definido por el siguiente subconjunto borroso, $\{5/0.1; 6/0.5; 7/0.9; 8/1; 9/0.9\}$, entonces, si el predicado es preciso y son 6 los elementos que verifican la propiedad P , la evaluación (el significado) de '*la mayoría* de los x son P ' sería igual a 0.5. En el caso borroso, una ponderación de los grados de pertenencia sería necesaria.

Esta clasificación recoge algunas restricciones que la lógica borrosa incorpora en su aparato teórico y formal para el análisis de las oraciones vagas. No es exhaustiva, pero a grandes rasgos, sí puede considerarse completa. Con esas restricciones, como indicábamos antes, la pretensión de esta lógica es deshechar la idea de un afán de análisis lógico universalizador y remarcar su adecuabilidad a la captura formal de una pequeña parte del lenguaje: aquél que incluye términos vagos.

Por tanto, una de las restricciones primarias que se adoptan en lógica borrosa es aquella que se efectúa sobre el dominio de objetos. Orientada en buena parte hacia la gestión de la incertidumbre en los sistemas expertos, se tiene la convicción de que los peritos humanos saben mucho, pero no todo, y de que cuanto más concreto sea un problema, mejores oportunidades habrá de que el experto exprese sus conocimientos acerca del mismo y sea posible una caracterización formal a éstos. Por tanto, intentar desentrañar un poco más acerca del contexto concreto suscitado por el análisis de un problema, (el núcleo en el que nos fijamos) y acerca de qué es un experto en cada contexto de uso, parece una tarea útil. Lo primero ayudará a matizar lo que en lógica borrosa se denomina núcleo o dominio de análisis del problema. Lo segundo contribuirá a caracterizar una noción muy usada en el marco de la Inteligencia Artificial, pero a menudo de una manera demasiado intuitiva o primaria. A ello nos dedicaremos en las dos próximas secciones de este trabajo.

2. ACERCA DE LA NOCIÓN DE CONTEXTO.

Como indicábamos, la noción de contexto sirve para delimitar el ámbito de referencia de nuestras oraciones. Se aproxima, por tanto, a la noción de núcleo, bien conocida y ejemplificada en la sección anterior y usada de modo abundante en el marco de la lógica borrosa. Como se puso de manifiesto en ese caso, es claro que, según el contexto, la referencia de una oración puede cambiar. Este cambio, como se mostró también antes, es más acusado si la oración incluye predicados vagos.

Ejemplo: No significa lo mismo la oración precisa 'Hay una distancia de 600 km.' en el contexto de un viajante de comercio español que en el de un viajante de comercio australiano. En una oración que incluye términos vagos, tampoco significa lo mismo, 'Es una persona alta',

en el contexto de los habitantes nilóticos que en el contexto de los que viven en Galicia.

Por tanto, como ocurría con el concepto de núcleo, una caracterización de lo qué es un contexto parece de interés ya que delimita de una manera muy acusada la referencia de las oraciones que incluyen tanto predicados precisos como vagos. En el marco de la lógica borrosa, el contexto quedaba caracterizado por un subconjunto del universo de referencia, normalmente por un subconjunto borroso. Esta explicitación, si bien operativa, esconde criterios implícitos que sería bueno hacer sobresalir. A ello se dirigirá ahora nuestra investigación.

2.1. Dos filosofías del lenguaje subyacentes a la noción de contexto.

De una manera quizás simplificadora pero que clasifica bien algunas aproximaciones a lo qué es un contexto, podríamos decir que hay dos tipos de acercamiento al tema. Estas dos vías de aproximación se fundamentan en dos concepciones distintas de filosofía del lenguaje.

(I). Al igual que ocurría con los predicados vagos, puede adoptarse el criterio de que los contextos reales de uso del lenguaje sólo son cortinas que ensombrecen habitaciones del conocimiento que deberían estar perfectamente iluminadas. El reto, por tanto, no está en decir que hay salas de conocimiento claras y salas de conocimiento oscuras, sino en iluminar las salas oscuras con claridad. Según esta filosofía, hay contextos ideales de transmisión significativa y es una tarea de la investigación científica el aprehenderlos. No obstante, desde una postura más pragmática y menos idealista, es notorio también que los humanos necesitamos comunicarnos a veces de forma inmediata; por tanto, haciendo uso de aquellos contextos a mano, normalmente con fronteras mal definidas o infracaracterizadas. Pero esto es una necesidad de los humanos de a pie, no de los científicos, cuya misión es caracterizar de forma precisa esas fronteras; esto es, hallar los contextos ideales de transmisión comunicativa. Como se observa, subyace a este planteamiento una concepción fregeana del lenguaje, en donde el lenguaje científico es considerado como superior y más deseable que el ordinario.

(II) Coincidiendo con el surgimiento de la I. A., se produce en los años 50 un giro en la filosofía del lenguaje, donde filósofos que habían mantenido una visión idealizada del mismo (p. ej., el Wittgenstein del *Tractatus*) acentúan desde entonces la importancia de su utilización según para qué propósito, (p. ej., el mismo Wittgenstein de las *Investigaciones filosóficas*)-, Alcanza entonces relevancia que el uso de contextos a mano, más que un objeto de reflexión teórica -que también lo puede ser- es un hecho social. Como tal, una teoría de contextos debería reflejar este aspecto. Concepciones realistas del significado, entre otras, pueden fundamentar una aproximación dinámica a la noción de contexto.

Según uno u otro criterio, pueden agruparse las teorías de contextos en dos clases: Teorías de contextos ideales y teorías de contextos ordinarias.

(2.2.). Teorías de contextos ideales.

Se basan en el precepto fregeano de que la referencia de una expresión es un concepto y un concepto es una función de objetos en valores de verdad. Desde esta aproximación

(importante porque fue el embrión de la lógica moderna), si esa función no está definida para todos los objetos, el predicado es vago, y por tanto, tiene una referencia parcial: es decir, no tiene referencia y, como tal no tiene acogida posible dentro del alcance de la lógica. Pero los predicados vagos, de esto era consciente Frege- ocupan una buena parte de la comunicación ordinaria. ¿Es posible, entonces, que la lógica se ocupe de alguna manera de ellos?. Autores post-fregeanos han dado una respuesta positiva a este interrogante, señalando precisamente como mediador en este conflicto al concepto de 'contexto' [Boch, 1983].

En estas teorías de contextos, la referencia de las expresiones que contienen predicados vagos se hacen dependientes de contexto: son, por tanto, funciones de contextos a conceptos, que a su vez, son funciones de objetos en el espacio valorativo $\{0, 1\}$. Un supuesto añadido en este modelo, usual en los post-fregeanos que tratan el tema de la vaguedad, es asumir el principio -también ideal- de que para cada concepto hay al menos un contexto que lo hace preciso (*Completeness Principle*) [Fine, 1975]. Bajo el paraguas de este principio, todo predicado vago puede ser transformado en preciso con la ayuda del contexto. Pero un punto importante a destacar es que, según esto, en las teorías ideales de contextos el papel del contexto no sería prestar la ayuda necesaria para que, en las oraciones vagas, la matización de su vaguedad fuese más racional, sino que su tarea más destacada consistiría propiamente en hacer desaparecer las oraciones vagas.

Dos problemas se le presentan inmediatamente a este tipo de aproximación son,

Problema I:

- (I). Que un mismo concepto pueda ser convertido en preciso por varios contextos. Pero si dentro de esta filosofía del lenguaje el contexto desvela la verdadera naturaleza del concepto, en un concepto que es precisado por dos o más contextos, tiene sentido la siguiente pregunta -llena de perplejidad-: ¿cuál es el auténtico?.
- (II). Que varios conceptos puedan ser precisados por un mismo contexto. Pero entonces, la relación entre contexto y concepto no sería funcional.

Por tanto, sólo se puede admitir una relación uno a uno entre contexto y concepto; esto es, que cada contexto precise a un sólo concepto y que éste sea precisado por un solo contexto.

Problema II:

- (I). Los contextos, en ocasiones, son imprecisos.

¿Cómo convertirlos en precisos? vuelve a ser la cuestión soberana.

(a). Particularizando los contextos, bien para cada concepto, bien para cada individuo. Esta tarea, idealmente posible, no es fácticamente realizable ni operativamente rentable dado su carácter hercúleo, sobre todo si se quiere, como es el caso de Frege, que la lógica abarque sin excepción a todas las oraciones (*Principle of Completeness*).

(b). Generalizando los contextos. Pero si los contextos son imprecisos, generalizándolos no conseguimos más que introducirles más imprecisión. La generalidad, aunque no se confunde con la vaguedad, introduce como ella imprecisión. Para que sea efectiva la generalización, debe orientarse hacia la noción de contexto-modelo, esto es, hacia contextos prototípicos (en el sentido de una teoría de prototipos ingenua [Cfr., Sobrino, 1992]) que pudiesen ser captados fácilmente (a causa de su prototipicidad) por el analista de oraciones vagas. Por tanto, si en la comunicación ordinaria un contexto es precisado por un contexto-modelo, la referencia de un predicado vago será una función de contextos-modelo a conceptos.

El problema señalado entonces de que el dominio de la función pudiera ser inexacto queda solucionado al reducir la expresión 'contextos apropiados para cada concepto' a 'contextos prototípicos' o 'contextos modelo'. Con la cobertura del *Completeness Principle* -un principio teórico y a priori- se garantiza que cada concepto tenga un contexto modelo que lo convierte en preciso.

La idea, si bien desde una perspectiva teórica es interesante, es de problemática legitimación y de dudosa utilidad. Además de los supuestos filosófico-lingüísticos subyacentes problemáticos y de principios -como el que acabamos de referir- muy comprometidos, no es en absoluto fácil caracterizar un prototipo e indicar, de un modo concreto o formal, qué es un contexto prototípico. El concepto de prototipo que usa Bosch es el proveniente de una teoría ingenua de prototipos.

Esta teoría, que ha facilitado que la noción de prototipo sea operativa: como tal, podría ser utilizada también en este ámbito, pero no como un modo de reducir la imprecisión del concepto de contexto, sino para, y ésta es una acotación que nos interesa destacar, para marcar diferencias de imprecisión o, de otra forma, la gradación de un contexto no prototípico (menos ideal) a otro más prototípico (más ideal). No obstante, en el uso de predicados vagos, la propuesta de L. Zadeh [Zadeh, 1983] de considerar prototipos borrosos hace que las versiones provenientes de la psicología sean vistas como intuitivas, pero no muy racionales. Una ejemplificación del concepto de prototipo borroso (del que en este congreso se da una muestra en [Lago, Olivas & Sobrino, 1992]) debería mostrar que cabe una aproximación al concepto de 'contexto prototípico' compleja pero interesante y que, para el caso de los predicados vagos, el concepto de 'contexto-ideal' es una noción más 'ideológica' (en lo que tiene de presuposiciones) que realista. Es así porque, como indicamos antes, su objetivo no es aproximarse de una manera directa a la vaguedad del lenguaje, sino convertir el lenguaje vago en una entidad distinta del mismo, en lenguaje preciso.

Una caracterización formal del concepto de contexto de carácter no ideal, sino real (decimos así por estar basado en una semántica de corte realista; p. ej. en la semántica de [Putnam, 1975]), da pie a otra interesante aproximación.

(2.3). Teorías de contextos ordinarias.

Las teorías de contextos ordinarias están basadas en semánticas que podrían ser denominadas 'realistas', en tanto que, de alguna manera, pretenden reflejar aspectos reales o fácticos del lenguaje de una manera directa, no sublimándolos a sus posibles correlatos ideales. Una semántica de este corte es la putnamiana [Putnam, 1975], asentada en el curioso principio de división de la labor lingüística. Este principio considera a la lengua como una actividad humana más, y así como, respecto a tareas laborales, no todo el mundo hace de todo, sino que hay una división del trabajo según su especialización, así también se puede considerar la oportunidad de establecer una división en la tarea de verbalizar el medio y las actividades que en torno a él se generan.

El contar con una teoría del significado en la que valga el Principio de división de la labor lingüística permite introducir los dos temas centrales de este trabajo:

(I). Si hay distintas tareas lingüísticas, existen diferentes contextos en los que las palabras o las oraciones pueden ser pronunciadas. Hay, según qué actividad, diferentes comunidades de

hablantes, cada una de ellas con contextos de uso específicos. Bajo esta consideración, puede darse la siguiente combinatoria:

- Comunidades de habla distintas, contextos diferentes.

Ejemplo: La palabra 'enlatar' puede ser objeto de uso diferente en al menos dos contextos distintos. Uno, el más genérico y común, el de una comunidad amplia de personas, utilizaría esta palabra para indicar que un producto alimenticio se puede introducir y cerrar en un recipiente de lata. Otro, usado en un contexto más específico, el del lenguaje de los médicos, significa que la persona objeto de su atención es un cadáver y puede meterse ya en un ataúd ('ya se puede enlatar').

- Comunidades de habla iguales, contextos diferentes.

Ejemplo: La palabra 'empapelar', en la comunidad de hispanohablantes españoles, tiene al menos dos significados según el contexto: En un contexto de uso genérico, significa 'envolver o rodear un objeto u objetos con papel'. En el contexto de una carretera donde hay unos agentes de tráfico significa 'poner una multa'. ('Le están empapelando').

- Comunidades de habla distintas, contextos iguales.

Ejemplo: El uso de cualquier término de los que propone Kuhn en su memorable libro *La estructura de las revoluciones científicas*. Por ejemplo, en el contexto de la investigación en física, y en un determinado período histórico suyo, no significaba lo mismo la palabra 'flogisto' que la palabra 'oxígeno'. Personas que analizaban el mismo elemento, verbalizan los mismos hechos, en el mismo contexto, de forma distinta.

- Comunidades de habla iguales, contextos iguales.

Desde el Principio de división de la labor lingüística, parece difícil encontrar un ejemplo que revele algún tipo de diferencia en este caso. No obstante, en un marco teórico distinto (para un análisis también diferente del problema que aquí se está analizando) si es posible encontrar ejemplos.

Ejemplo: La homonimia de palabras puede ser un caso. La referencia de los pronombres personales, como 'el', puede variar según la dirección de la comunicación: Si de mi para 'él' o de 'él' para mi. El sujeto de la acción es alternativo en uno u otro caso.

Esta categorización debe dar la oportunidad de reflexionar acerca de tres características importantes para el tema objeto de esta investigación:

(a). Estaba en lo cierto Black [Black, 1963] cuando en su seminal trabajo sobre un posible tratamiento formal de los términos vagos, indicaba que éste tendría que estar siempre referido a una comunidad de hablantes. Los ejemplos puestos arriba no incluyen términos específicamente vagos; sin embargo, puede observarse la conveniencia en cualquier caso de tener en cuenta tanto a la comunidad de habla como al contexto. Si estos factores son importantes en el análisis de los términos precisos, con más motivo lo serán en el caso de los

términos vagos, ya que como intentamos ejemplificar al principio del trabajo, su evaluación es, si cabe, mucho más dependiente de contexto.

Qué es una comunidad de habla y qué un contexto se revela, por tanto, como algo que necesita someterse a consideración.

(b). Los contextos de uso de palabras, no sólo son distintos, sino que además tienen diferente rango de especialización.

Ejemplo: Refiriendo un célebre caso relatado por [Putnam, 1975], la palabra 'agua' usada en un contexto amplio es utilizada como sinónimo de líquido incoloro, inodoro e insípido. Pero en un contexto más especializado o específico, es sinónimo de sustancia que consta de dos átomos de hidrógeno y uno de oxígeno. Evidentemente, este último contexto es más específico que el anterior. La tarea lingüística de definir al agua así puede ser hecha en propiedad por menos gente que la anterior: es más especializada. Esta reflexión nos conduce al apartado siguiente.

(c). Según la labor lingüística a realizar, hay diferentes especialistas, expertos o peritos en el uso de las palabras de nuestra lengua. Para evitar cualquier etnocentrismo, avisamos que el uso de la relación 'más especializada que' no debe comportar más juicio de valor que el, en cada caso, inevitable. Somos conscientes de que, según para que propósitos o problema, el orden de la relación puede invertirse.

Clarificar por tanto qué es 'ser un experto' parece también interesante. Dada su relación con las nociones de 'comunidad de habla' y 'contexto' (un experto lo es para una comunidad de habla y para un contexto), la nociones de 'comunidad de hablantes', 'contexto' y 'experto' presentan dependencias entre sí. Su explicitación, por tanto, no podrá ser lineal, sino genética (en el sentido de que se desarrollan a partir de un origen). El origen de esta explicación es el concepto de contexto.

2.3.1. Caracterización de la noción de contexto.

Notación: Sea

U = universo de discurso,

K = contexto,

H = hablantes, y

P(H) = conjunto potencia de H (los subconjuntos posibles de hablantes).

Una posible caracterización formal del concepto 'contexto' sería la siguiente [Cfr. Ballwegg, 1983:

Notación:

La de un cálculo de primer orden con identidad, más las siguientes abreviaturas,

$a : VI \rightarrow U$, (función de asignación de una entidad de U a cada variable de individuo).

$\langle T, * \rangle$, T = conjunto de puntos temporales; * = ordenamiento sobre T irreflexivo y transitivo.

$\langle \Sigma_{\alpha, s, t}, <_{\alpha, s, t}, \mu_{\alpha, s, t}, \tau_{\alpha, s, t} \rangle \alpha \in CP_n$

α = expresión bien formada del cálculo lógico de primer orden que incluye constantes predicativas de orden n .

$s \in S$. S = conjunto de situaciones posibles. Entorno en el cual se pronuncia una palabra u oración.

$t \in T$. T = conjunto de puntos en el tiempo

$\langle \Sigma_{\alpha, s, t}, \langle \alpha, s, t, \mu_{\alpha, s, t}, \tau_{\alpha, s, t} \rangle \alpha \in CP_n = \alpha$ referida a

Σ = conjunto de variables o constantes individuales que pueden tener uno o dos o más elementos; Si $|\Sigma_{\alpha}| \geq 2$, en este caso, las variables o constantes son atribuidas a predicados comparativos. El conjunto de variables tiene su ubicación en una ebf del cálculo lógico de primer orden que incluye constantes predicativas de orden n y es referida a un hablante y a un tiempo.

$<$ relación de orden irreflexiva y transitiva en Σ_{α} .

$\mu: U^n \rightarrow \Sigma_{\alpha}$. Función que especifica el grado de α -idad de cada n -tupla de individuos.

τ = umbral. Elemento de Σ_{α} , que puede señalar la transición de una α -riedad a otra. El establecimiento de un umbral debe ser justificado racionalmente, a fin de evitar sospechas de arbitrariedad. Pueden establecerse tantos umbrales como la especificación del contexto lo requiera. P. ej., si tenemos $\langle \tau^+_{\alpha, h, t}, \tau^-_{\alpha, h, t} \rangle$ señala que hay elementos que aún refiriendolos a una situación y a un tiempo, son en ese contexto indeterminados (los que quedan entre el umbral positivo y el umbral negativo).

Por tanto, un contexto K (de uso de una palabra) podría quedar caracterizado por la siguiente estructura:

$$K = \langle U, a, \langle T, * \rangle, \langle \Sigma_{\alpha, s, t}, \langle \alpha, s, t, \mu_{\alpha, s, t}, \tau_{\alpha, s, t} \rangle \alpha \in CP_n \rangle$$

Evidentemente, elementos de semántica situacional y de lógica temporal son imprescindibles para capturar de una forma completa el concepto de contexto. Pero una exposición detallada de las herramientas de estas lógicas útiles para esta definición queda fuera de los objetivos de este trabajo. Aquí intentamos simplemente indicar algunos elementos estructurales que deberían tenerse en cuenta en la definición de contexto.

(2.3.2.). Caracterización del concepto ‘comunidad de hablantes’.

Una comunidad de hablantes puede definirse como cualquier conjunto de hablantes suficientemente uniforme. La uniformidad debe sobresalir de algunas relaciones que están subyacentes en el uso del lenguaje por parte de esa comunidad. Algunas relaciones interesantes para caracterizar a una comunidad de habla como tal son las que se establecen entre los hablantes y

-El contexto.

Son relaciones interesantes las siguientes:

c.a.p. = contexto accesible a una persona.

c.a.p.: $H \rightarrow P(K)$. Función que asocia a cada hablante H_i el conjunto de contextos accesibles para él.

c.a.g. = contexto accesible a un grupo.

c.a.g. : $G \rightarrow P(K)$. Función que asocia a cada grupo de hablantes el conjunto de todos los contextos accesibles para él (grupo).

Un contexto se dice 'accesible' respecto a un sujeto cuando este no es pasivo respecto a los objetos a los que da nombre (en ese caso, hablaríamos de contexto natural), sino que, aún implicando un esfuerzo, es capaz de ordenarlos o clasificarlos teórica o empíricamente, justificando, si fuere necesario, ese ordenamiento.

Se usan las abreviaturas, K^S para la $\bigcup_{h_i \in H} \text{c.a.p.}(h_i)$ y K^G para la $\bigcup_{h_i \in G} \text{c.a.g.}(h_i)$.

Tanto c.a.p. como c.a.g. admiten una relación de orden parcial c.m.e. ^{h_i} ('contexto más especializado que').

-Los presupuestos .

Los presupuestos (que no deben confundirse con presuposiciones valorativas) constituyen aquello que nos permite analizar los objetos de una forma adecuada al contexto. Si es así, le denominamos presupuestos adecuados al contexto. A fin de caracterizar mejor esta noción, se establecen las siguientes funciones:

p.a.p = presupuesto accesible para cada persona.

p.a.p: $K^H \times H \rightarrow P(P)$. Función que asocia a cada hablante y a cada contexto accesible para él, el conjunto de presupuestos apropiados para ese contexto.

Se establecen los siguientes supuestos añadidos: Para todo $k \in \text{c.a.p.}(h_i)$, el conjunto p.a.p. (k, h_i) no es vacío y los conjuntos (k_1, h_i) y p.a.p. (k_2, h_i) son disjuntos para contextos diferentes k_1 y k_2 .

Esta suposición no es razonable en el estudio del lenguaje natural, y es un tema de investigación estudiar qué ocurre cuando esta condición no se da.

p.a.g. = presupuesto accesible para cada grupo.

p.a.g. : $K^H \times H \rightarrow P(P)$. Función que asocia a cada grupo de hablantes y a cada contexto accesible para él (grupo), el conjunto de presupuestos apropiados para ese contexto. Se establecen los mismos presupuestos que arriba, para los que vale también las mismas objeciones.

Se usarán las abreviaturas p.a.p. (h_i) para la $\cup_{h_i, k_i \in H, K}$ c.a.p. (h_i) y p.a.g. (h_i) para la $\cup_{h_i, k_i \in H, K}$ c.a.g. (h_i).

Sobre los presupuestos se establece también una relación de orden:

p.m.e. = 'presupuesto más especializado que', que debe respetar el ordenamiento ya asentado para los contextos accesibles ('contexto más especializado que').

Sea $k_1, k_2 \in$ c.a.p. (h_i), $k_1 \neq k_2$ y k_1 c.m.e. $_{h_i}$ k_2 , entonces,

- Para todo $p_1 \in$ p.a.p. (k_1, s_i) hay al menos un $p_2 \in$ p.a.p. (k_1, s_i) tal que p_1 p.m.e. ^{s_i} p_2 .
- Para todo $p_1 \in$ p.a.p. (k_1, s_i) y $p_2 \in$ p.a.p. (k_2, s_i) no se verifica que p_2 p.m.e. ^{s_i} p_1 .

Ambas propiedades se verifican también para el conjunto p.a.g. (S_i) y su relación de orden p.m.e. ^{S_i} .

- El foco.

Denota aquella parte de la oración en la que el hablante pone más interés. Fonéticamente suele estar anunciada por el énfasis en la entonación. El foco de interés puede definirse como una función del hablante y de sus presupuestos dominante (que normalmente condiciona ese foco) en el significado de la oración.

f.i.p. $^{S_i} : H \times P_d \rightarrow$ Significado S_i .

f.i.g. $^{S_i} : G \times P_d \rightarrow$ Significado S_i .

Al igual que en los casos anteriores, se puede establecer una relación de orden entre focos de interés de menor a mayor especificidad. P. ej., alguien que en el análisis lógico de la oración 'La mayoría de los gitanos son morenos' focalice su atención sobre la representación formal del significado de 'La mayoría', está denotando que hace uso de un foco de interés no sólo acertado, sino especializado (desde el punto de vista de la vaguedad).

-Mundos significativos.

Las relaciones 'más especializado que...' referidas a contextos, presupuestos o focos de interés permiten pensar que existen mundos significativos distintos, en tanto que unos son más especializados que otros. En efecto, si un mundo significativo depende de esas relaciones, podemos establecer una función generadora (gen) de mundos posibles (de espacios significativos concebibles) definida así:

$$\text{gen}^{s_i} \text{ c.m.a. } (s_i) \times \text{p.m.a. } (s_i) \rightarrow M_{s_i}.$$

$$\text{gen}^{S_i} \text{ c.m.a. } (s_i) \times \text{p.m.a. } (S_i) \rightarrow M_{S_i}$$

según sea para hablantes o para grupos de hablantes.

Sobre los mundos se pueden definir una vez más relaciones de orden 'más especializado que...(m.m.e.)'. Un mundo M_1 es más especializado que un mundo M_2 si M_1 es generado por un contexto, un presupuesto y/o un foco de interés más especializado que el de M_2 .

Así, sean M_1 y M_2 elementos de M_{S_i} . Entonces,

$M_1 R^s_i M_2$ syss M_1 es el resultado de $gen^s_i(k, p_1)$ y M_2 es el resultado de $gen^s_i(k, p_2)$.

M_1 m.m.e. M_2 syss $M_1 = gen^s_i(k_1, p_1)$ y $M_2 = gen^s_i(k_2, p_2)$ y k_1 c.m.e^s_i k_2 y p_1 p.m.e^s_i p_2 . Lo mismo se verifica para M^S_i .

En torno a estas definiciones, pueden establecerse propiedades interesantes.

(a). Sobre el conjunto de mundos pueden definirse relaciones de equivalencia, según su tasa de especialización.

(b). Elementos de diferentes mundos no son comparables, de entrada, bajo la relación m.m.e.

(c). En principio, para dos mundos cualesquiera diferentes (atingentes al problema que se quiere resolver), uno de ellos siempre es más preciso que el otro.

(d) Todos los mundos que son menos precisos que un mundo dado, son comparables a través de la relación m.m.e..

Constituye un ejercicio de lógica transcribir formalmente las cuatro propiedades enunciadas.

Contexto, presupuesto, foco... permiten, entre otras nociones, caracterizar por fin a una comunidad de hablantes.

Una comunidad de hablantes es tal si

(i). $c.a.g.(S_i) = \cup c.a.p.(s_i)$.

El contexto accesible al grupo es la reunión de los contextos accesibles a cada una de las personas del grupo.

(ii). $p.a.g.(S_i) = \cup p.a.p.(s_i)$.

El presupuesto accesible al grupo es la reunión de presupuestos accesibles a cada una de las personas del grupo.

(iii). Hay contextos $k_1, k_2 \in c.a.p.(s_i)$ donde se verifica que k_1 c.m.e. k_2 para todo $k_1 \in c.a.g.(S_i)$, $k_1 \neq k_2$.

Hay contextos accesibles compartidos por personas que son precisados por un contexto accesible al grupo. Esta condición garantiza la comunicación.

(iv). Para todo $m \in M_{S_i}$, se verifica que $f.i.g.^{S_i}(m) = \cap f.i.p.(m)$.

El foco de interés del grupo es compartido por todos los miembros del grupo.

(v). Para todo $k_1, k_2, k_3 \in (S_i)$, si se verifica que k_1 c.m.e.^s_i k_2 y k_1 c.m.e.^s_i k_3 entonces se

verifica que k_2 c.m.e. $S_i k_3$ o que k_3 c.m.e. $S_i k_2$.

Esto es, para el grupo siempre hay un contexto más especializado que otro.

(vi). $M_{S_i} \times \text{f.i.g.} (M_{S_i}) \rightarrow P(S_i) \times (S_i)$.

Hay un mapeo que asocia con cada mundo y foco de interés asociado a él una jerarquía dentro del grupo. Este mapeo provoca un ordenamiento total del conjunto de los subconjuntos no vacíos de S_i . Esta condición es muy interesante para el siguiente paso en este trabajo, ya que permite hablar de expertos (respecto a algún tópico, a algún foco de interés) dentro del grupo.

Esto es, si hay mundos significativos a nadie se le escapa que las cuestiones de significado deben referir siempre a un mundo, o mejor, a una clase de equivalencia de mundos y si hay mundos significativos más especializados que otros, habrá habitantes (hablantes) de esos mundos más especializados también que otros. Estamos en condiciones de acercarnos ya al otro concepto que pretendíamos analizar en este trabajo: El concepto de experto.

3. ACERCA DEL CONCEPTO DE EXPERTO.

Los expertos son elementos $s_i \in S_i$ (S_i homogéneo) que conocen los criterios relevantes para darle el significado adecuado a las palabras; esto es -en términos putnamianos- conocen sus intensiones, sus propiedades objetivas y reales. Por el contrario, el significado común adquiere representatividad en los estereotipos, que recogen creencias generalizadas del colectivo de los hablantes y que por ser tal resultan a menudo inespecíficos o poco desarrollados. Por tanto, los expertos serían aquellos que aprehendiendo las intensiones significativas, señalan las propiedades reales de los objetos. Sin embargo, no puede decirse que el significado de un término sea sólo fruto del criterio del experto, sino también de aquellos que aportan propiedades más usuales o menos refinadas al lenguaje. Los motivos de esta mixtura se encuentran en que:

(a). Las intensiones no son siempre totalmente aprehendibles. Sólo los expertos pueden acercarse a conocerlas, pero no siempre logran su perfecta comprensión. La dinámica de las teorías científicas está llena de ejemplos que pueden confirmar esto.

(b). Identificar el significado con las intensiones es olvidar el Principio de división de la labor lingüística, según el cual un objeto es significado no sólo por un especialista, sino, según en qué contextos, por personas que no lo son.

(c). Vinculado con (a), se puede afirmar que nunca puede aprehender totalmente la intensión de una palabra, dado que su significado no es estático, sino dinámico; avanza a medida que progresa el conocimiento científico y cambia a medida que se altera nuestra cultura.

No obstante estas dificultades, si se acepta que hay contextos, presupuestos o focos más especializados que otros para el mundo posible elegido, también hay que aceptar que, dentro de cada comunidad de habla, puede establecerse un orden de 'hablantes más especialistas que...' según el tema y foco de interés determinado por el asunto objeto de análisis. Dado que una función primordial del lenguaje es comunicar, el experto deberá vincular

las intensiones que como especialista tiene con los estereotipos que como ciudadano-hablante común percibe en su entorno y ha adquirido. Esto permitirá que su labor que sea interactiva y, por tanto, que su especialidad revierta en beneficio del público. Este es el objetivo que se fija también para los interfaces en los sistemas expertos, que tienen que aunar rigor e interactividad.

Por tanto, las relaciones c.m.e., p.m.e. y f.m.e. puede ayudar a definir la relación m.exp.q. 'más experto que', referida a individuos. Una forma de caracterizar esta relación es estableciendo algún grado de orden en el uso de las intensiones. Una semántica categorial [Cresswell, 1976] puede ofrecer un marco apropiado en el que se observe como un término se caracteriza más y mejor según se progresa en el grado de 'intensionalidad alcanzada'. A continuación ofrecemos un breve ejemplo.

Ejemplo:

Se trata de exponer distintos significados de la palabra 'sal' en distintos contextos y en diferentes niveles de pericia. La descripción de estos significados incluye intensiones y estereotipos, de modo que se recoja esa doble dimensión del significado de un término: su comprensión más acabada y su comunicatividad, a fin de hacer su significado valioso, pero también útil.

A tal fin se dispone una progresividad en contextos, focos, mundos, etc. El contexto base, constituye aquel que no presupone conocimiento especial alguno, que alude a la experiencia cotidiana y normal. En la escala de la división del trabajo lingüístico, ocuparía el peldaño más compartido. En este contexto, sal puede ser identificada, p. ej., con 'lo que da sabor a la comida'. Esta sería una propiedad estereotípica. Otras propiedades estereotípicas serían las siguientes: 'lo que en demasía es malo para la salud' (menos específico que el anterior), 'lo que es malo para los hipertensos', 'lo que es de color blanco'. En un contexto bastante más especializado, sal sería aquella sustancia cuya estructura química es ClNa ; esto es, cloruro sódico.

Pues bien, usado de un modo laxo una notación próxima a las gramáticas categoriales, es posible caracterizar el significado dinámico de 'sal' aludiendo a su estructura, denotada por $\langle 0, 1, 1 \rangle$, donde el primer argumento refiere a la categoría 'oración', el segundo representa un grado de comparación, es decir, un par $\langle u, \rangle$, donde $\rangle$ es la relación 'más específico que' y u un término del campo de esa relación y, por fin, el tercer argumento representa a la categoría 'nombre'.

Por tanto,

'sal₀' = $\langle 0, \langle \text{'lo que da gusto a la comida'} \rangle, \text{sal} \rangle$ para $k_0(f_0, m_{0..})$. Pero

'sal₁' = $\langle 0, \langle \text{'lo que es dañino para los hipertensos'} \rangle, \text{sal} \rangle$ para $k_1(f_1, m_{1..})$ y

sal₂ = $\langle 0, \langle \text{'cloruro sódico'} \rangle, \text{sal} \rangle$ para $k_2(f_2, m_{2..})$,

donde,

k_1 c.m.e k_0 , f_1 f.m.e f_0 , m_1 m.m.e m_0 y k_2 c.m.e k_1 , f_2 f.m.e f_1 , m_2 m.m.e m_1 .

Con una notación más compleja que ésta, que por el alcance de este trabajo y espacio disponible no es posible ensayar aquí, se puede hacer una descripción completa para la progresión del significado de un término desde los arquetipos hasta las intensiones. Toda la cadena, no obstante, constituye el significado del término, y según las necesidades que demande el problema, puede ser pertinente una caracterización intermedia del mismo, que no agote el recorrido hasta la última intensión. Por otra parte, en la mayoría de los términos, no

está clara cual es la última intensidad, porque hay una textura abierta o espacio de horizonte significativo nunca totalmente oteado [Waismann, 1945].

Lo dicho hasta aquí referido a contextos o expertos es útil para predicados precisos (donde se simplificaría mucho el tema) pero alcanza realmente relevancia para predicados vagos. En el caso de los predicados precisos ayudaría simplemente a desambiguarlos. En el caso de los predicados vagos, contribuye a caracterizar mejor la vaguedad, a profundizar un poco más, en conceptos frecuentes en su modelización formal, así como en la aplicación de la misma a casos prácticos, como acontece en el área de los sistemas expertos.

4. CONCLUSIONES.

Brevemente se puede indicar que un intento de caracterizar formalmente los conceptos de contexto y experto es útil y necesario en el área de lógicas que, como la lógica borrosa, han alcanzado interesantes aplicaciones a la Inteligencia artificial y, más concretamente, a los Sistemas Expertos. Caracterizar del modo más concreto conceptos como 'contexto', del que la lógica borrosa hace un uso central en la formalización de los predicados vagos, y experto, uno de los primeros conceptos en boca de un diseñador de prototipos de sistemas expertos, parece una inversión necesaria y rentable. En este trabajo se ha explorado una posible forma de realizar esta tarea. No obstante, creyentes en el Principio de división de la labor lingüística, también comulgamos con el Principio de división del trabajo científico, de manera que no dudamos que la aproximación emprendida en este trabajo puede ser mejorada, completada o incluso falsada por otra mejor. Una posible orientación que promete resultados esperanzadores, es la que se puede emprender desde una teoría pragmática del lenguaje. A tal empresa invitamos a todos a los que esta indagación les haya parecido de interés y estén dispuestos a capturar 'mayores' intensiones.

5. REFERENCIAS.

- BALLWEG, J. (1983), 'Vagueness or context-dependence?. Supervaluation revisited in a semantics based on scales', en T.T. Ballmer & M. Pinkal (eds.), *Approaching Vagueness*. Elsevier Science Pub. North-Holland, 1983, pp.59-78
- BAUERLE, R. & EGLI, U. & STECHOV, A. (eds.), [1976], *Semantics from different points of view*. Berlin. Heidelberg.
- BELLMAN, R. & ZADEH, L. A. [1976], 'Local and fuzzy logics', en Dunn & epstein (eds.), *Modern Uses of Multiple-Valued Logic*. Reidel Pub. Co., 1977, pp. 105-165.
- BLACK, M. [1963], 'Reasoning with Loose Concepts', en *Margins of Precision*. Cornell U. Press, 1950.
- BOSH, P. [1983], 'Vagueness is context-dependence. A solution to the Sorites Paradox', en T.T. Ballmer & M. Pinkal (eds.), *Approaching Vagueness*. Elsevier Science Pub. North-Holland, 1983, pp. 189-210.
- CRESSWELL, M. J. [1976], 'The Semantics of Degree', en Partee, B. H. (ed.), *Montague Grammar*. Academic Press. New York, pp. 261-292.
- EIKMEYER, H. Y. & RIESER, H. [1981], 'Meanings, Intensions and Stereotypes: A new

- approach to Linguistic Semantics', en *Words, Worlds and Contexts. New Approaches in Word Semantics*. Walter de Gruyter. N. Y., pp.131-187.
- FINE, K. [1975], 'Vagueness, Truth and Logic', *Synthese*, 30, pp. 265-300.
- LAGO, R. & OLIVAS, J. A. & SOBRINO, A. [1992], 'Una posible aplicación de la teoría de prototipos de Zadeh a la gestión de la evolución de la siniestrabilidad de los incendios forestales en un motor de inferencias'. En estas *Actas*.
- PUTNAM, H. [1975], 'The Meaning of the Meaning', en K. Gunderson (ed.), *Language, Mind and Knowledge*. University of Minnesota Press. V. cast. de J.J. Acero en *Teorema*, vol. XIV/3-4. Madrid, 1984.
- POPPER, K. R. [1974], *Búsqueda sin término*. Tecnos. Madrid, 1974.
- RUSSELL, B. [1923], 'Vagueness', *Australasian J. of Philosophy*, 1, pp. 84-92.
- SOBRINO, A [1989], *La vaguedad: Hacia un estudio crítico de algunos aspectos formales y de gestión*. Tesis Doctoral. Universidad de Santiago de Compostela.
- SOBRINO, A [1992], Is there a prototypical logic for prototypes and classifications founded on them?, *Proceedings of the IPMU' 92*. July, 6-10 . Palma de Mallorca, pp. 697-703.
- TRILLAS, E. [1991], 'Una introducción breve a la lógica borrosa', *Curso de Conferencias sobre Inteligencia artificial y Robótica*. Real Academia de Ciencias exactas, físicas y naturales.
- WAISMANN, F. [1945], 'Verifiability', en *Proceedings of the Aristotelian Society*. Supp. Vol. 19, pp., 119-150.
- ZADEH, L. A. [1965], 'Fuzzy Sets', *Information and Control*, 8, pp. 338-353.
- ZADEH, L. A. [1982], 'A note on prototype theory', *Cognition*, 12, pp. 291-297.

6. AGRADECIMIENTOS.

Este trabajo ha sido financiado por los proyectos XUGA 20505B90 y PS89-0175 de la Xunta de Galicia y D.G.D.I.C.Y.T., respectivamente.

INDISTINGUIBILIDADES QUASI-UNIDIMENSIONALES.

J. Jacas y J. Recasens

Sec. Matemàtiques i Informàtica.

E.T.S. d'Arquitectura de Barcelona

Univ. Politècnica de Catalunya

Diagonal, 649

08028 Barcelona, Spain

Resumen: Dada una relación probabilística E , se desarrolla un algoritmo para hallar otra relación probabilística E' "próxima" a E y unidimensional en el sentido del Teorema de Representación para igualdades difusas explotando técnicas de Saaty.

El algoritmo es fácilmente implementable y práctico, ya que con un sólo conjunto difuso se puede almacenar toda la información contenida en E' .

Palabras Clave: Relación probabilística, Teorema de Representación, Dimensión, Matriz recíproca.

1. Introducción

Los operadores de indistinguibilidad [10,3] extienden el concepto de relación de equivalencia al entorno borroso. Incluyen similitudes [12], "likeness" [8] y relaciones probabilísticas [5].

En [11] L. Valverde presenta un teorema de representación para T-indistinguibilidades que generaliza los resultados de S. Ovchinnikov para relaciones probabilísticas [6].

Diferentes familias de conjuntos difusos generan la misma T-indistinguibilidad E . El menor de los cardinales de las familias de generadores de E se define como la dimensión de E .

Si la dimensión es baja, es más ventajoso almacenar un sistema minimal de gene-

radores de E que el total de la matriz asociada. En este sentido es útil trabajar con T-indistinguibilidades de “baja dimensión”.

En [2] se demuestra que para similitudes la dimensión es razonablemente baja. Sin embargo, en el caso de ser T una t-norma arquimediana, la dimensión de E puede llegar a ser la mitad del cardinal de X [7].

Por lo tanto, en lugar de almacenar una T-indistinguibilidad de alta dimensión, será preferible construir T-indistinguibilidad próximas a la dada pero de menor dimensión.

En este sentido se introduce el concepto de relaciones probabilísticas quasi-unidimensionales y siguiendo ideas de Saaty [9] se presenta un algoritmo para generar una relación probabilística unidimensional “próxima” a la dada.

2. Preliminares.

Esta sección contiene resultados conocidos sobre relaciones probabilísticas y matrices recíprocas y se incluye para hacer autocontenido el trabajo.

Definición 2.1. (Menger) [5] Una relación probabilística E en un conjunto X es una relación difusa en X que satisface las siguientes propiedades:

2.1.1. Reflexividad: $E(x, x) = 1, \forall x \in X$,

2.1.2. Simetría: $E(x, y) = E(y, x), \forall x, y \in X$,

2.1.3. Transitividad: $E(x, y) \cdot E(y, z) \leq E(x, z), \forall x, y, z \in X$.

Si $E(x, y) = 1$ implica $x = y$, entonces E se dice que es separadora.

Teorema de Representación 2.2. (Ovchinnikov, Valverde) [6,11]. Una relación difusa E en un conjunto X es una relación probabilística sí, y sólo sí, existe una familia $\{h_i\}_{i \in I}$ de subconjuntos difusos de X tal que para cada x, y de X se tiene

$$E(x, y) = \inf_{i \in I} \left(\min \left\{ \frac{h_i(x)}{h_i(y)}, \frac{h_i(y)}{h_i(x)} \right\} \right)$$

Definición 2.3. (Jacas, Ovchinnikov) [1,6]. La dimensión de una relación probabilística es el mínimo de los cardinales de familias generadoras de E .

Tras esta definición, del Teorema anterior se deduce que toda relación probabilística puede generarse como el ínfimo de relaciones probabilísticas unidimensionales. Estas relaciones juegan por tanto un papel importante y es interesante su caracterización:

Teorema de Caracterización 2.4. (Jacas, Ovchinnikov) [1,6]. Una relación probabilística E en un conjunto X tal que $E(x, y) > 0, \forall x, y \in X$, es unidimensional sí, y sólo sí, existe un orden total $\leq$ en X tal que si $x \leq y \leq z$, entonces $E(x, y) \cdot E(y, z) = E(x, z)$.

De este Teorema se deduce que una relación probabilística E en X separadora y unidimensional define una relación de “betweenness” total B_E en X del siguiente modo:

$$(x, y, z) \in B_E \quad \text{sí, y sólo si} \quad E(x, y) \cdot E(y, z) = E(x, z) \text{ y } x \neq y \neq z \neq x.$$

La igualdad del Teorema anterior es muy similar a la que aparece en la definición de matriz recíproca consistente. Esta semejanza será explotada en las siguientes secciones.

Definición 2.5. (Saaty). Una matriz real A de n filas y n columnas con entradas $a_{ij} > 0, 1 \leq i, j \leq n$ es recíproca sí, y sólo sí, $a_{ij} = \frac{1}{a_{ji}} \forall i, j = 1, 2, \dots, n$. Una matriz recíproca es consistente sí, y sólo sí,

$$a_{ik} = a_{ij} \cdot a_{jk} \quad \forall i, j, k = 1, 2, \dots, n$$

En las siguientes secciones convendrá interpretar A como una aplicación $A : X \times X \rightarrow \mathbb{R}^+$ donde $X = \{x_1, x_2, \dots, x_n\}$ y $a_{ij} = A(x_i, x_j)$.

Las matrices recíprocas quedan caracterizadas por el siguiente Teorema (cf. Teorema 2.4.).

Teorema 2.6 (Saaty) [9]. Una matriz real $A = (a_{ij})$ de n filas y n columnas es consistente sí, y sólo sí, existe un subconjunto difuso h de X tal que

$$a_{ij} = \frac{h(x_i)}{h(x_j)} \quad \forall i, j = 1, 2, \dots, n.$$

h se llamará un generador de A .

3. Relaciones probabilísticas y matrices consistentes.

Una matriz consistente A en X , con $a_{ij} \neq 1$ si $i \neq j$, define una relación de “betweenness” B_A en X de la siguiente manera:

$$(x_i, x_j, x_k) \in B_A \quad \text{sí, y sólo sí,} \quad a_{ik} = a_{ij} \cdot a_{jk} \text{ y } i \neq j \neq k \neq i.$$

Teorema 3.1. La matriz consistente A_h y la relación probabilística E_h definidas por un mismo conjunto difuso h de X definen la misma relación de “betweenness” en X (i.e., $B_{A_h} = B_{E_h}$).

Demostración.

$$\begin{aligned}
(x_i, x_j, x_k) \in B_{E_h} &\iff E(x_i, x_j) \cdot E(x_j, x_k) = E(x_i, x_k) \iff \\
&\iff \frac{h(x_i)}{h(x_j)} \cdot \frac{h(x_j)}{h(x_k)} = \frac{h(x_i)}{h(x_k)} \iff \\
&\iff a_{ij} \cdot a_{jk} = a_{ik} \iff (x_i, x_j, x_k) \in B_{A_h}.
\end{aligned}$$

Por otro lado, dada una matriz recíproca A , Saaty [9] obtiene una matriz consistente A' “próxima” a A generada a partir de un vector propio asociado al mayor valor propio de A . Es interesante hacer notar que pequeñas modificaciones en las entradas de A dan lugar a pequeñas modificaciones en las de A' y que si A ya es consistente, entonces $A' = A$.

Si E es una relación probabilística “próxima” a otra unidimensional, entonces E “casi satisfará” el Teorema 2.4. Esto sugiere la siguiente definición:

Definición 3.2. Dado $\epsilon \in [0, 1]$, una relación probabilística E en X se dirá ϵ -unidimensional si, y sólo si, existe un orden total $\leq$ en X tal que si $x \leq y \leq z$, entonces se cumple la siguiente desigualdad:

$$|E(x, y) \cdot E(y, z) - E(x, z)| < \epsilon.$$

Si E es una relación probabilística ϵ -unidimensional en un conjunto finito X de cardinal n y $x_1 \leq x_2 \leq \dots \leq x_n$ una relación de orden total determinada por E en X , la siguiente definición permite conectar relaciones probabilísticas con matrices recíprocas.

Definición 3.3. Siguiendo con la notación anterior, la matriz $A : X \times X \rightarrow R^+$ definida por

$$A(x_i, x_j) = \begin{cases} E(x_i, x_j) & \text{si } i \leq j \\ \frac{1}{E(x_i, x_j)} & \text{en caso contrario} \end{cases}$$

se dirá la matriz ϵ -consistente asociada a la relación probabilística ϵ -unidimensional E .

4. El algoritmo.

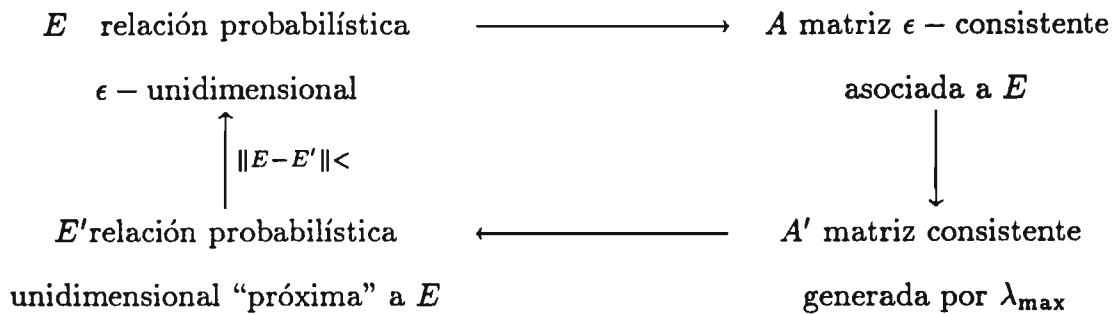
Reuniendo los resultados de la sección anterior, se obtiene un algoritmo para hallar una relación probabilística E' unidimensional “próxima” a una ϵ -unidimensional dada.

Si A es una matriz recíproca de n filas y n columnas, entonces la suma de sus valores propios es n y A es consistente si, y sólo si, n es el único valor propio distinto de cero (Saaty, [9]). Pequeñas variaciones en A dan lugar a pequeñas variaciones en sus valores propios.

Estos resultados permiten la siguiente construcción:

Sea E una relación probabilística ϵ -unidimensional. A partir de E se genera una matriz A ϵ -consistente como en la Definición 3.3. El vector propio asociado al mayor valor propio $\lambda_{\max}$ de A genera una relación probabilística E' unidimensional “próxima” a E .

El algoritmo queda condensado en el siguiente esquema:



Ejemplo. La siguiente relación probabilística E es 0,1 unidimensional

$$E = \begin{pmatrix} 1.00 & 0.74 & 0.67 & 0.50 & 0.41 \\ 0.74 & 1.00 & 0.87 & 0.65 & 0.53 \\ 0.67 & 0.87 & 1.00 & 0.74 & 0.60 \\ 0.50 & 0.65 & 0.74 & 1.00 & 0.80 \\ 0.41 & 0.53 & 0.60 & 0.80 & 1.00 \end{pmatrix}.$$

Una matriz 0,1-consistente asociada a E es

$$A = \begin{pmatrix} 1.0000 & 1.3514 & 1.4925 & 2.0000 & 2.4390 \\ 0.7400 & 1.0000 & 1.1494 & 1.5385 & 1.8868 \\ 0.6700 & 0.8700 & 1.0000 & 1.3514 & 1.6667 \\ 0.5000 & 0.6500 & 0.7400 & 1.0000 & 1.2500 \\ 0.4100 & 0.5300 & 0.6000 & 0.8000 & 1.0000 \end{pmatrix},$$

$\lambda_{\max}$, el mayor valor propio de A es 5.0003 y un vector propio de $\lambda_{\max}$ es

$$h = (1.00, 0.76, 0.67, 0.50, 0.40)$$

h genera la relación probabilística unidimensional

$$E' = \begin{pmatrix} 1.00 & 0.76 & 0.67 & 0.50 & 0.40 \\ 0.76 & 1.00 & 0.88 & 0.66 & 0.53 \\ 0.67 & 0.88 & 1.00 & 0.74 & 0.60 \\ 0.50 & 0.66 & 0.74 & 1.00 & 0.81 \\ 0.40 & 0.53 & 0.60 & 0.81 & 1.00 \end{pmatrix}$$

6. Conclusiones.

En este trabajo se da un algoritmo para hallar una relación probabilística unidimensional E' próxima a una dada E , usando técnicas de Saaty.

Con ello la información contenida en E se puede almacenar con poca distorsión usando un sólo conjunto difuso.

Si E está “lejos” de ser unidimensional, entonces será útil hallar técnicas para aproximarla mediante relaciones de dimensión “baja”, lo que nos conduce a estudiar relaciones de “betweenness” difusas cuyo estudio será objeto de futuros trabajos.

Referencias.

- [1] Jacas, J., On the generator of T-indistinguishability operator. Stochastica XII-1, (1988), 49-63.
- [2] Jacas, J., Similarity Relations-The Calculation of Minimal Generating Families. Fuzzy Sets and Systems (1990), 151-162.
- [3] Jacas, J., Relaciones de equivalencia difusas y clasificaciones. Comunicaciones del Primer Congreso Español sobre Tecnologías y Lógica Fuzzy. Granada (1991), 7-11.
- [4] Katz, M., Inexact Geometry, Notre Dame J. of Journal Logica 21 (1980).
- [5] Menger, K., Probabilistic theory of relations. Proc. Nat. Acad. Sci. USA 37. (1951). 178-180.
- [6] Ovchinnikov, S., Representations of transitive fuzzy relations in H.J. Skala et al. Eds. Aspects of Vagueness (Reidel, Dordrecht (1984), 105-118.
- [7] Recasens, J., Sobre la representació i generació de relacions d'indistingibilitat. Ph.D. Dissertation. Prog. Matemàtica Aplicada, UPC, Barcelona, 1992.
- [8] Ruspini, E., A theory of Mathematical Classification. Ph. D. Dissertation. Dept. of System Science. School of Engineering. Univ. of California. Los Angeles (1977).
- [9] Saaty, T.L., The Analytic Hierarchy Process, McGraw- Hill, (1980).
- [10] Trillas, E., Assaig sobre les relacions d'indistingibilitat. Primer Congrés Català de Lògica Matemàtica. Barcelona (1982), 51-59.
- [11] Valverde, L., On the structure of F-indistinguishability operators. Fuzzy Sets and Systems 17 (1985), 313-328.
- [12] Zadeh, L.A., Similarity relations and fuzzy orderings. Inform. Sci. 3, (1971), 177-200.

UNA POSIBLE APLICACION DE LA TEORIA DE PROTOTIPOS DE ZADEH A LA GESTION DE LA EVOLUCION DE LA SINIESTRALIDAD DE LOS INCENDIOS FORESTALES EN UN MOTOR DE INFERENCIA

R. Lago¹, J. A. Olivas^{**} y A. Sobrino^{**}.

* Dpto. de Inteligencia Artificial, Facultad de Informática, U.P.M.

** Area de Lógica y Filosofía de la Ciencia. Universidad de Santiago.

Resumen: Presentamos una aplicación de la teoría de prototipos de Zadeh a la previsión de evolución de la siniestralidad de los incendios forestales. Tratamos de interpretar una situación real a partir de un paradigma previamente establecido. Comenzamos determinando los prototipos "ideales" de progresión de la siniestralidad; los calificamos como ideales ya que son paradigmas de progresión para cualquier periodo de tiempo. A continuación se define el periodo actual a partir de la modificación del prototipo "ideal" más afín.

Palabras clave: Prototipo, Sistema Experto.

1. INTRODUCCION DEL PROBLEMA.

En la lucha diaria contra los incendios forestales en la comunidad gallega es constante el problema de asignación y optimización de recursos. Esto es debido a la frecuencia y simultaneidad de los incendios junto con la limitación de medios disponibles. Por estos factores los expertos necesitan prever el desarrollo de los siniestros a corto y medio plazo en una zona determinada. Para satisfacer esta necesidad real de previsión se presenta este estudio, como una muestra de la investigación llevada a cabo para la realización de un prototipo de sistema experto.

Tras un análisis de los casos sucedidos en los dos últimos años, los expertos¹ han observado que la evolución de la siniestralidad puede representarse como una función creciente de tipo sigmoideal que hemos dividido en tres sectores y que comenzaría en el día posterior a un periodo de lluvia en el que el número de los incendios se ha reducido a cero. Este patrón de crecimiento se repite de forma cíclica después de cada periodo de este tipo, pero puede sufrir modificaciones debido a factores específicos. Entiéndase por siniestralidad la combinación de número y peligrosidad de los incendios. La figura 1 muestra la representación del patrón de crecimiento.

1

Agradecemos la colaboración prestada por el equipo del Servicio de Defensa Contra Incendios Forestales de Lugo de la Xunta de Galicia, dirigido por Don Tomás Fernandez-Couto Juanas.

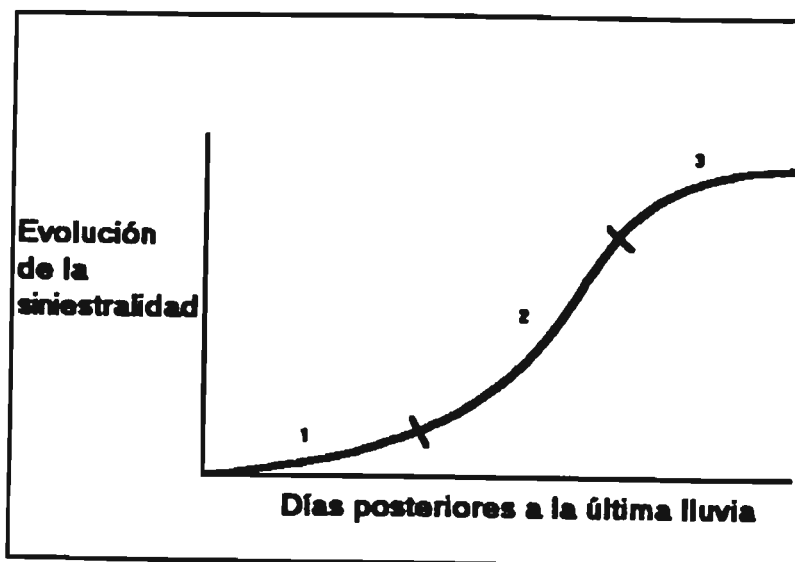


Figura 1. Patrón de crecimiento.

El primer sector representa un lento crecimiento de la siniestralidad en los días inmediatamente posteriores al periodo de lluvias.

El segundo sector expresa un alto crecimiento de la siniestralidad, especialmente en cuanto al número de incendios.

En el tercero queremos representar una estabilización en cuanto al número, pero un progresivo aumento en la peligrosidad de los fuegos.

En nuestra vida cotidiana es usual asociar un hecho o conjunto de hechos con un paradigma, de tal forma que el paradigma interpreta la situación y de él dependen las acciones que llevemos a cabo. Por ejemplo, si estamos conduciendo y empieza a granizar adecuamos nuestra conducción al esquema de conducción bajo condiciones potencialmente peligrosas. Como se ha visto en el ejemplo, muchas de las acciones que realizamos en nuestra vida dependen de una interpretación. Lo que aquí se plantea es que interpretar una situación es encontrar el paradigma o prototipo más afín a las circunstancias del problema. En nuestro caso se trata de simular la capacidad del experto para interpretar la situación, es decir, para encontrar el prototipo de evolución de la siniestralidad más adaptado a las circunstancias reales.

2. CONCEPTO Y PROTOTIPOS.

Tomando como marco de referencia la teoría de prototipos de la psicología (para una noticia crítica véase por ejemplo [Adan y Sobrino, 1991] y [Sobrino, 1992]), podría entenderse que nuestra representación es prototípica del avance de la siniestralidad. Sin embargo, en el proceso de adquisición del conocimiento pudimos observar que esta representación simplifica en exceso las pautas del comportamiento de los expertos. Cuando un técnico se enfrenta a una situación real maneja un abanico de prototipos determinados por una serie de factores, es decir, debe decidir que tipo de evolución de la siniestralidad es previsible. Dicho de otro modo, el prototipo "Evolución de la siniestralidad" no es único, sino que existen diferentes formas de evolución dentro de la misma estructura sigmoideal.

Zadeh alude a las teorías clásicas de prototipo desde el punto de vista de la psicología [Zadeh, 1982], criticando precisamente lo que aquí hemos expuesto: su falta de adecuación a la función que debe cumplir un prototipo. La aproximación de Zadeh a lo que debe entenderse por prototipo es menos intuitiva que las concepciones de las teorías psicológicas, pero más racional, más próxima a lo que en un examen detenido muestra el significado de un concepto prototípico. En nuestro caso hemos observado que la idea de Zadeh sugiere que un concepto engloba un conjunto de prototipos, los cuales representan la buena, baja o media compatibilidad de los ejemplares con el concepto.

Los conceptos pueden entenderse como predicados unarios o bien como relaciones n -arias con $n > 1$. Tras el proceso de adquisición del conocimiento pudimos observar que el concepto del que nos hablan los expertos es el atributo "Progresivo". Con este concepto realizamos algo análogo a lo que ha sido esbozado en [Olivas y Sobrino, 1991], en donde el concepto "Alto" era descompuesto en prototipos según el grado de compatibilidad con él, es decir, los buenos representantes del concepto (bastante alto, muy alto y altísimo), los representantes limítrofes (normal, ligeramente alto y ligeramente bajo) y los pobres (bajísimo, muy bajo y bastante bajo).

Desde este punto de vista podemos hablar de siniestralidad altamente progresiva, medianamente progresiva o escasamente progresiva. Esto se puede representar, tal y como refleja la figura 2, con tres sigmoides, entendidas simplemente como representaciones gráficas de los tres prototipos borrosos.

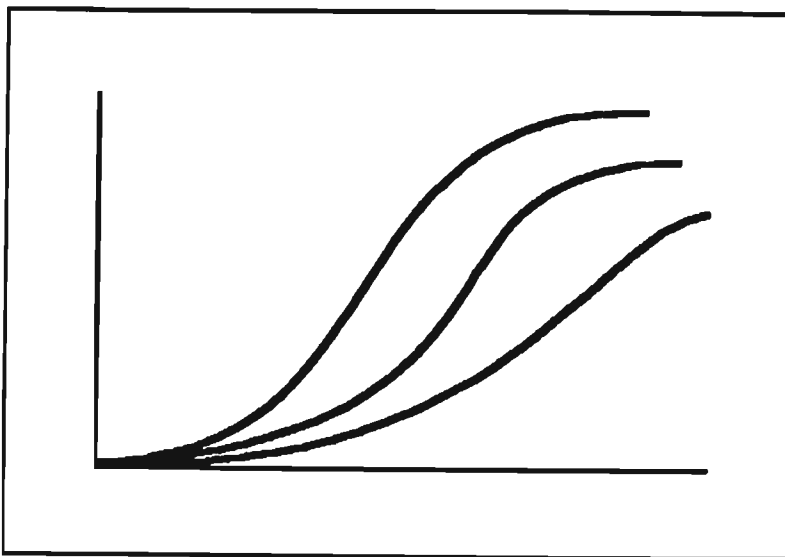


Figura 2. Representación de prototipos.

3. LOS PROTOTIPOS.

3.1. FACTORES DE CICLO.

Como vimos anteriormente la evolución de la siniestralidad viene dada por ciclos que se inician a partir de cada periodo de lluvias. Los factores que determinan a priori la progresión de la siniestralidad desde un periodo de lluvias al siguiente los reducimos a tres para facilitar el ejemplo:

A. Sequedad estacional.

La sequedad estacional es la propia de la época del año; entendiendo que puede haber un Invierno seco o un Verano lluvioso. El usuario introduce el valor al principio de cada ciclo: [Alta/0.95, Media/0.45, Baja/0.05].

B. Intensidad de la última lluvia.

En nuestro ejemplo consideramos que la influencia de los tipos de intensidad vendría dada por [Llovizna/0.9, Tormenta/0.5, Borrasca/0.1].

C. Influencia social.

Representa la influencia del factor humano en la evolución de la siniestralidad (conflictividad de la zona, rencillas entre vecinos, etc). El usuario determina el tipo de influencia por [Mucha/0.85, Media/0.45, Escasa/0.05].

3.2. DETERMINACION DE LOS PROTOTIPOS.

Los prototipos han sido representados como "números borrosos", los cuales nos van a permitir obtener un grado de pertenencia a cada prototipo a partir del grado de pertenencia al concepto. En bien de la sencillez del modelo han sido representados como triángulos isósceles e iguales. Por tanto, para construir los prototipos (números borrosos triangulares) sólo necesitamos conocer su centro ("Centro de prototipo").

Como hemos visto los factores descritos en el apartado anterior nos servían para definir los prototipos de forma genérica; estos prototipos actúan como patrones de crecimiento para cualquier ciclo posible.

Además el experto ha sido capaz de determinar los sectores de cada prototipo. Por ejemplo y de forma simplificada veamos la definición de "Siniestralidad medianamente progresiva" en la tabla 1:

Tabla 1.

Sectores	Sequedad local (probabilidad de ignición)	Virulencia de incendios	Número de incendios
Sector 1: 1º-8º días	De muy baja hasta media.	Aumento lento desde virulencia nula hasta mediana.	[0,3]
Sector 2: 9º-13º días	De media hasta alta.	Se estabiliza en mediana, pero crece el número de incendios cada día.	[4,9]
Sector 3: 14º-...	De alta en adelante	Aumento medio desde virulencia mediana hasta extrema.	[10,16]

El problema de definir los prototipos surge en el proceso de adquisición del conocimiento experto; más concretamente, del hecho de que el experto no nos podía dar reglas que asociasen unos valores de los factores con el prototipo,

pero si era capaz de mostrar ejemplos o casos que caen bajo un prototipo. Dicho de otro modo, el experto no realizaba un proceso deductivo, pero si un proceso de ejemplificación. Como primera aproximación entendimos que algún tipo de media (armónica, aritmética, geométrica, etc) de los casos dados por el experto nos podía servir para determinar el centro de cada prototipo.

El proceso para determinar los centros de los prototipos es el siguiente:

- I. Se le presenta al experto una batería de casos elegidos al azar y se le pide que señale el prototipo bajo el que cae cada caso. Véase un ejemplo en la tabla 2:

Tabla 2.

SEQUEDAD ESTACIONAL	INTENSIDAD LLUVIA	INFLUENCIA SOCIAL	SINIESTRALIDAD PROGRESIVA	MEDIA
Alta	Llovizna	Mucha	Elevada	0.900
Alta	Borrasca	Mucha	Media	0.633
Media	Tormenta	Media	Media	0.467
Media	Llovizna	Mucha	Elevada	0.733
Baja	Borrasca	Media	Escasa	0.200
Baja	Llovizna	Mucha	Media	0.600
Alta	Tormenta	Escasa	Media	0.500
Media	Borrasca	Escasa	Escasa	0.200
Baja	Tormenta	Baja	Escasa	0.200

Las tres primeras columnas señalan la información que se le da al experto (los casos); la cuarta es rellenada por el experto (los prototipos). Con el fin de respetar el lenguaje del experto sólo manejamos etiquetas lingüísticas en la entrevista, sin mostrarle las medias (quinta columna).

- II. Si algún prototipo no tuviese casos o no los suficientes realizamos el proceso inverso, es decir, preguntar al experto por ejemplares de dichos prototipos. Ejemplo:

Como observamos que había pocas ocurrencias de los prototipos "Progresión elevada" y "Progresión escasa", le pedimos al experto que enunciase casos de estos prototipos (tabla 3):

Tabla 3.

SEQUEDAD ESTACIONAL	INTENSIDAD LLUVIA	INFLUENCIA SOCIAL	SINIESTRALIDAD PROGRESIVA	MEDIA
Alta	Tormenta	Mucha	Elevada	0.767
Alta	Llovizna	Media	Elevada	0.767
Media	Borrasca	Media	Escasa	0.333

III. Por tanto, los centros de prototipos son las medias de los casos de cada prototipo. En nuestro ejemplo:

Centro de "Progresión elevada":	0.791
Centro de "Progresión media":	0.550
Centro de "Progresión escasa":	0.233

IV. El último paso es la representación de los números borrosos triangulares, a partir de dos factores:

- A. El centro de cada prototipo.
- B. La distancia del cero al centro del primer prototipo se usa como distancia estándar para construir la base de los tres triángulos (figura 3).

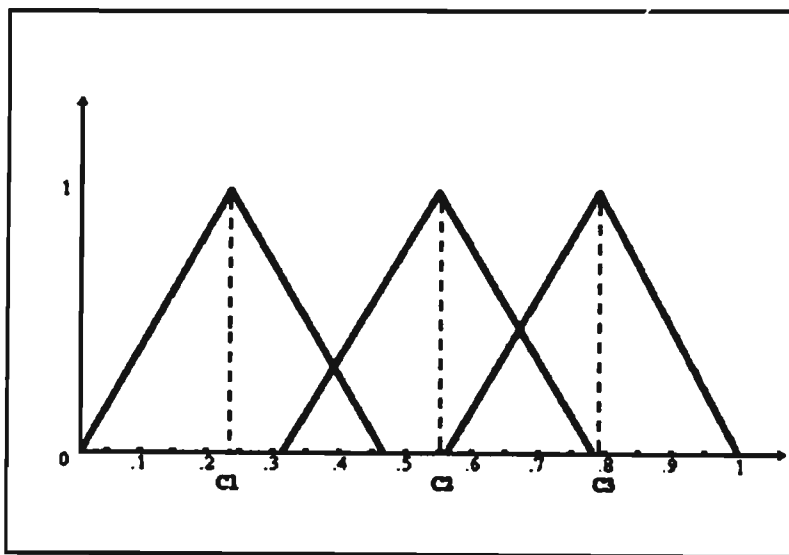


Figura 3. Formalización de prototipos.

4. APROXIMACION DEL PROTOTIPO AL CICLO REAL.

No obstante es necesario tener en cuenta que entre diferentes ciclos existen diferencias que vienen dadas por el cambio del valor de los factores de ciclo. Por tanto, es necesario que el día del inicio del ciclo (primer día sin lluvias) se genere un nuevo ciclo a partir de su afinidad con uno de los prototipos. Para ello se pregunta al usuario el valor real de los factores de ciclo. Su combinación nos dará unos grados de compatibilidad con los prototipos, en particular el que nos interesa es el grado de compatibilidad con el prototipo más afín.

Para encontrar el prototipo más afín establecimos el siguiente proceso:

- I. Combinar los valores seleccionados por el usuario para el ciclo en curso (r).

M : operador de combinación Media.

f_i : valor de un factor. $i=1$ para el factor "Sequedad estacional", etc.

p_i : prototipo. $i=1$ para el prototipo "Siniestralidad escasamente progresiva", etc.

$$M(f_1, f_2, f_3) = r$$

- II. Hallar los grados de compatibilidad con cada prototipo: $\mu_{p_1}(r)$, $\mu_{p_2}(r)$, $\mu_{p_3}(r)$.

- III. El prototipo más afín es el que mayor grado de compatibilidad tenga con r : un p_i tal que $\mu_{p_i}(r) = \max(\mu_{p_1}(r), \mu_{p_2}(r), \mu_{p_3}(r))$.

Ahora se podría adoptar una solución menos fina, consistente en asimilar el comportamiento del ciclo actual al de su prototipo más afín. Pero de lo que se trata es de modificar el prototipo más afín hacia el ciclo que ahora se inicia. Ejemplo:

El usuario el primer día del ciclo introduce los siguientes valores: [Sequedad: Media/0.45, LLuvia: Tormenta/0.5, Influencia social: Mucha/0.85]. Lo cual da $M=0.6$.

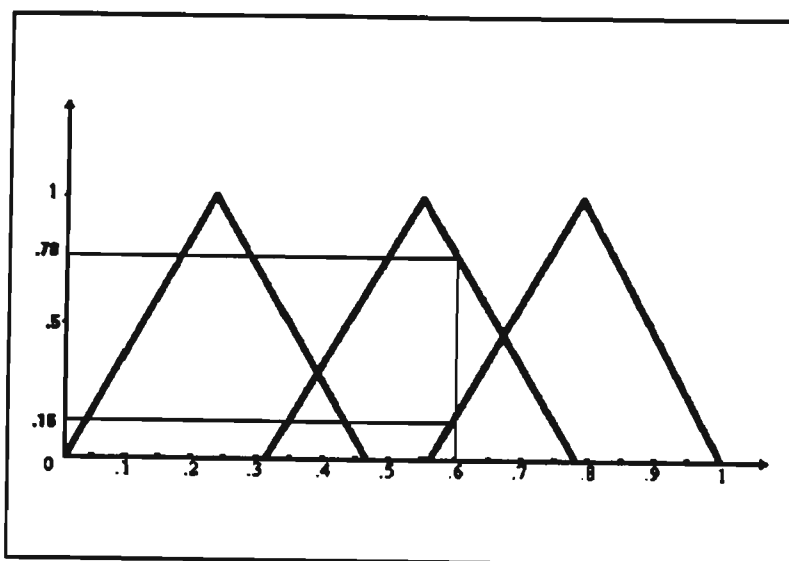


Figura 4. Grado de compatibilidad con los prototipos.

Como se puede observar en la figura 4 los grados de compatibilidad son los siguientes:

$$\mu_{p_1}(r) = 0$$

$$\mu_{p_2}(r) = 0.78$$

$$\mu_{p_3}(r) = 0.16$$

De lo que se deduce que el prototipo más afín es p_2 ("Siniestralidad medianamente progresiva").

Como antes dijimos es necesario asimilar el comportamiento del ciclo actual al de su prototipo más afín. Para lo cual es preciso modificar la definición que el experto ha dado del prototipo, evidentemente esta modificación debe realizarse en base al grado de compatibilidad con el prototipo.

Hemos definido una función que refleja el hecho de que el conocimiento experto sea capaz de actualizar los prototipos de una forma bastante eficiente:

Sean:

- c: Centro del prototipo afín.
 r: Resultado de combinar los factores de ciclo actual.
 $\mu(r)$: Grado de compatibilidad del ciclo actual con el prototipo afín.

Para cada columna ("Sectores" y "Número") se establece un factor de corrección (lo denominamos k_i).

El factor de corrección para los sectores (k_1) se define por:

$$k_1 = 2 - \mu(r), \text{ si } r \leq c.$$

$$k_1 = \mu(r), \text{ si } r > c.$$

El factor de corrección para el número de incendios (k_2) se define por:

$$k_2 = 2 - \mu(r), \text{ si } r > c.$$

$$k_2 = \mu(r), \text{ si } r \leq c.$$

La modificación de intervalos se realiza partiendo de los intervalos anteriores (del prototipo): $[a_1, b_1]$, $[a_2, b_2]$, $[a_3, b_3]$, y obtenemos los intervalos del nuevo ciclo: $[a_1, b_1 \cdot k_1]$, $[(b_1 \cdot k_1) + 1, b_2 \cdot k_1]$, $[(b_2 \cdot k_1) + 1, b_3]$.

La virulencia y la sequedad local, puesto que son etiquetas lingüísticas, las tratamos como atributos propios de los días de cada sector.

Para nuestro ejemplo: $c=0.55$, $r=0.6$, $\mu(r)=0.78$. Con lo cual los factores de corrección son los siguientes: $k_1 = 0.78$, $k_2 = 1.22$. Por tanto la definición del nuevo ciclo queda así (tabla 4):

Tabla 4.

Sectores	Sequedad local (probabilidad de ignición)	Virulencia de incendios	Número de incendios
Sector 1: 1º- 6º días	De muy baja hasta media.	Aumento lento desde virulencia nula hasta mediana.	[0,4]
Sector 2: 7º- 10º días	De media hasta alta.	Se estabiliza en mediana, pero crece el número de incendios cada día.	[5,11]
Sector 3: 11º-...	De alta en adelante	Aumento medio desde virulencia mediana hasta extrema.	[12,20]

Como es obvio, aunque desde el punto de vista de la máquina se trabaja con decimales para no acumular error, desde el punto de vista del usuario aparecen enteros.

Resumiendo brevemente, empezamos determinando los prototipos "ideales" de progresión de la siniestralidad; los calificamos como ideales ya que son paradigmas de progresión para cualquier ciclo. Lo que la tabla anterior muestra es la definición del ciclo actual a partir de la modificación del prototipo "ideal" más afín. Dicho de otra forma, interpretamos la situación actual a partir de un paradigma previamente establecido.

5. APROXIMACION DEL CICLO REAL A LA SITUACION DIARIA.

La previsión establecida sobre el ciclo actual sería válida si los factores diarios no tuviesen peso alguno, cosa que no ocurre; más bien sucede que el ciclo no debería modificarse salvo que hubiese un cambio inesperado en la situación diaria. Por ejemplo, un aumento brusco de la sequedad local debido a vientos secos y constantes. Por ello, establecemos una posible corrección del ciclo sobre la base de los factores diarios.

5.1. FACTORES DIARIOS.

Para este ejemplo los factores diarios se pueden reducir a los siguientes:

- A. Número de días después de la última lluvia (valor no borroso).
- B. La serie de incendios de los días anteriores (valor no borroso).
- C. Virulencia de los siniestros del día anterior. El usuario dispone de { Extrema/0.95, Alta/0.8, Media/0.5, Baja/0.2, Nula/0 }.
- D. Sequedad local. Vendría dada por la probabilidad de ignición de los combustibles en la comarca que estudiamos. El usuario la recibe diariamente en porcentaje, aunque el sistema asocia cada posible entrada (probabilidad de ignición) con una etiqueta y un grado, de la siguiente forma: {Extrema($\geq 60\%$)/1, Muy Alta(50%)/0.85, Alta(30%)/0.6, Media(40%)/0.4, Baja(20%)/0.25, Muy baja(10%)/0.05}.
- E. Influencia social diaria. Este factor debe recoger hechos tales como el que exista una mayor tendencia a provocar incendios durante el fin de semana, o también una mayor tendencia cuando hay una fiesta local. El usuario dispone de { Alta/0.9, Baja/0.6}. Si no hay ninguna influencia social conocida se ignora este valor.

5.2. REGLAS DE MODIFICACION DEL CICLO REAL.

No es necesario para modificar el ciclo real realizar una combinación de factores, tal y como hicimos en la modificación del prototipo. La razón de esta diferencia se encuentra en el conocimiento experto adquirido, el cual señala que en la modificación diaria no es pertinente una aproximación gradual. En la modificación diaria el motor del cambio son determinados valores de algunos de los factores, no la combinación de todos los factores. La modificación de los

prototipos era esencialmente cuantitativa, sin embargo la modificación del ciclo real es más bien cualitativa.

Todos los factores diarios, salvo la influencia social, nos ayudan a validar el buen comportamiento del ciclo real preestablecido. Una disparidad muy acusada con los factores previstos en el ciclo real nos llevaría a modificaciones, por ejemplo, un cambio prematuro de sector. La influencia social simplemente nos ayuda a avisar al usuario de que debe tomar las medidas necesarias en esa zona, por ejemplo, asignar más medios. También nos sirve de explicación de un aumento inusual del número de incendios en un momento determinado del ciclo (máximo local).

Las reglas que determinen una posible modificación del ciclo son típicas representantes de heurísticas de experto. Tienen la forma:

```
SI
    Factori ≤ Valor
    &
    Factorj ≤ Valor
    &
    ...
ENTONCES
    Modificacióni
```

Como ejemplo:

```
R. 01SECTOR1.
SI
    Virulencia ≥ Alta
    &
    Sequedad local ≥ Media
ENTONCES
    Reducir en 1 el extremo superior del intervalo de días.
```

6. RECAPITULACION Y CONCLUSIONES.

Nuestro objetivo ha sido tratar de automatizar un proceso de razonamiento experto, a pesar de que dicho razonamiento no es formalizable como un conjunto completo de reglas. Este razonamiento tiene como meta la interpretación de una situación real. Parece adecuado representar este proceso de interpretación como la modificación de un prototipo hacia la situación real, tal y como observamos que hacía el experto. Este proceso ha tenido los siguientes pasos:

I. Representación conceptual.

- A. Determinación del concepto que vamos a estudiar: ciclo o periodo de "Siniestralidad progresiva", representable gráficamente como una sigmoide dividida en tres sectores. El ciclo comienza y termina cuando hay un periodo de lluvias (siniestralidad nula).
- B. Determinación de los factores que influyen en la progresión de la siniestralidad (factores de ciclo).
- C. Determinación de los tres prototipos asociados al concepto: "Alta", "Mediana" y "Escasa", también son gráficamente representables como sigmoides divididas en sectores.

- D. Se le pide al experto una descripción de cada prototipo, descompuesta en sectores.

II. Representación formal.

- A. Hallar el centro de cada prototipo mediante petición de ejemplos al experto.
- B. Representación de los prototipos mediante números borrosos triangulares.
- C. Combinación de los factores del ciclo real o actual para obtener su afinidad con cada uno de los prototipos.
- D. Definición del ciclo actual a partir de la modificación del prototipo más afín.
- E. Aproximación del ciclo actual a la situación diaria. Esta modificación se hace en base a reglas que tienen en cuenta los factores diarios.

Consideramos que el tipo de problema para el que es adecuado este proceso es aquel en que es necesaria la interpretación de una situación real a partir de paradigmas (prototipos) previamente establecidos; teniendo en cuenta que no puede ser completamente formalizado mediante reglas. Por ejemplo, el problema de interpretar una intoxicación alimentaria por Salmonelosis como "Aguda", "Normal" o "Leve".

7. REFERENCIAS BIBLIOGRAFICAS.

- ADAN, M. & SOBRINO, A. [1991], "Nota sobre la teoría de prototipos", en *Actas del Primer Congreso Español sobre Tecnologías y Lógica Fuzzy*. Universidad de Granada: 173-177.
- BONISSONE, P. P. & TONG R. M. [1984], "Linguistic solutions to fuzzy decision problems", en *TIMS/Studies in Management Sciences*, 20. 323-334. Elsevier Science Pub.
- CUENA, J. (ed.) [1986], *Inteligencia Artificial: Sistemas Expertos*. Alianza Editorial. Madrid
- MATE, J., L. & PAZOS, J. [1988], *Ingeniería del conocimiento*. SEPA. Cordoba, Argentina.
- OLIVAS, J. A. & SOBRINO, A. [1991], "Aproximación Lingüística y Teoría de Prototipos de Zadeh", en *Actas del Primer Congreso Español sobre Tecnologías y Lógica Fuzzy*. Universidad de Granada: 179-183.
- SOBRINO, A. [1992], "Is there a prototypical logic for prototypes and classification founded on them?", en *Proceedings of the IPMU' 92*. Mallorca, July 6-10, pag. 697-704.
- TRILLAS, E. [1984], "El tratamiento cualitativo de la vaguedad. ¿Hacia una nueva lógica descriptiva?", en R. G. Jaimez (ed.), *Algunas cuestiones actuales en teoría de la probabilidad y procesos estocásticos*. Publicaciones de la Universidad de Granada: 15-32.
- ZADEH, A. [1982], "A note on prototype set theory and fuzzy sets". *Cognition* 12. 291-297, 1982.

UN METODO INDUCTIVO PARA EL APRENDIZAJE DE REGLAS DIFUSAS

Antonio GONZALEZ
Departamento de Ciencias
de la Computación e I.A.
Universidad de Granada
18071-Granada (España)

e-mail: gonzalez@robinson.ugr.es

Resumen: Los métodos de aprendizaje automático pueden ser de gran ayuda en el diseño de sistemas basados en reglas difusas. Estos métodos nos permiten simplificar el proceso de obtención del conjunto de reglas que representan al sistema. En este trabajo presentamos un modelo que asocia una medida de incertidumbre a cada regla. Esta medida se obtiene a través de la información contenida en una serie de ejemplos, y representa una medida de la creencia sobre la afirmación "esta regla representa al sistema". En el cálculo de esta medida de incertidumbre es posible incluir algún tipo de información adicional sobre la credibilidad de los ejemplos, y sobre la estabilidad de la evidencia asociada a cada regla.

Palabras clave: Aprendizaje automático, inducción, reglas difusas, medidas de incertidumbre, evidencias.

1. Introducción

Muchos sistemas pueden describirse mediante un conjunto finito de reglas difusas, por ejemplo, algunos sistemas de control [6,14] y algunos sistemas expertos [17]. La representación mediante reglas difusas ha demostrado ser muy útil en la resolución de diferentes problemas complejos. Sin embargo, las piezas básicas de esta metodología, es decir, las reglas difusas, usualmente se obtienen o bien por un procedimiento de prueba y error, o por una siempre difícil consulta a los expertos.

En muchos casos, el uso de técnicas automáticas para obtener el conjunto de reglas difusas, pueden simplificar el diseño de sistemas de control en lógica difusa o sistemas expertos, y además permite añadir una interesante capacidad de adaptación al

propio sistema.

En la literatura se han propuesto diferentes métodos para el aprendizaje de reglas difusas. Por ejemplo, Pedrycz [11] utilizó técnicas de agrupamiento y discretización difusa, para identificar sistemas relacionales difusos. Otros autores, por ejemplo Takagi y Sugeno [16] y Sugeno y Tanaka [15], investigaron la identificación de un sistema, en el que el consecuente de la regla difusa es una relación lineal entre la entrada y la salida. El uso del producto matricial para expresar el grado de certeza de una regla ha sido utilizado por Binaghi [1] y Peng and Wang [12]. Campos y Moral [3] proponen un método que nos permite extraer una asignación de pesos sobre las reglas, óptima con respecto al método de inferencia presentado en [2]. También se ha propuesto la utilización de otros métodos en la identificación de la reglas difusas, como es el caso de las redes neuronales [7,10,13] y los algoritmos genéticos [8].

En muchos casos, los algoritmos de aprendizaje de reglas difusas intentan encontrar una adecuada asignación de grados de creencia sobre las reglas. En general, estos grados pueden interpretarse como la fuerza de la implicación de la regla, y tomados como un simple peso de la misma sin ninguna interpretación teórica adicional. En [5], Delgado y González proponen un método inductivo de aprendizaje de reglas, en el que estos pesos se calculan a través de frecuencias condicionales y se interpretan como medidas de evidencia.

Dada una estructura para una regla, es decir, las variables de entrada y las variables de salida,

$$R_{ij} = \text{if } X \text{ is } A_i \text{ then } Y \text{ is } B_j \quad (1)$$

con $X=(X_1, X_2, \dots, X_n)$, $Y=(Y_1, Y_2, \dots, Y_m)$, $A_i=(A_{i1}, A_{i2}, \dots, A_{in})$, $B_j=(B_{j1}, B_{j2}, \dots, B_{jm})$, y siendo A_{ki} y B_{kj} conjuntos difusos, el algoritmo de aprendizaje propuesto en [5] produce como salida una distribución de incertidumbre sobre cada regla difusa, de la siguiente forma

$$\text{grado}(R_{ij}) = \text{medida\_de\_incertidumbre}(B_j/A_i) \quad (2)$$

Esta medida de incertidumbre puede interpretarse como un grado representando la creencia acerca de la afirmación "esta es una regla que representa al sistema". Esta medida se asignará a cada regla usando la información de los distintos ejemplos por el algoritmo de aprendizaje.

En algunos casos, es posible que tengamos información adicional sobre los distintos

ejemplos que se han obtenido a partir del sistema, y sería conveniente que cualquier algoritmo de aprendizaje, que tratara de identificar el conjunto de reglas difusas, pudiese beneficiarse de esta información.

Por ejemplo, en el método de aprendizaje propuesto en [5] ningún ejemplo se consideró más importante que otro. En general, sería deseable que un algoritmo de aprendizaje fuese capaz de incorporar cualquier información adicional disponible sobre los ejemplos, como es el caso de aquella que pueda ser proporcionada por un experto. Así, el algoritmo de aprendizaje debería ser capaz de incorporar información acerca de la alta o baja importancia que puede tener un ejemplo en la identificación de la reglas. La importancia alta significaría una alta representatividad del ejemplo acerca del comportamiento del sistema, por el contrario, un ejemplo puede considerarse de poca importancia si no confiamos mucho en el mismo o ha sido considerado como un ejemplo poco frecuente.

El método propuesto en [5] utiliza frecuencias condicionales para la estimación de los pesos de cada regla. Estas frecuencias condicionales pueden interpretarse como información sobre los ejemplos que afirman y que niegan la regla. En muchos casos, se obtienen reglas con una alta creencia, pero esta ha sido calculada a partir de uno muy pocos ejemplos, y el resultado puede no ser muy fiable. Por ejemplo, cuando existe un único ejemplo positivo a una regla (aunque no sea muy representativo de la misma) y ninguno negativo, la creencia final puede ser muy alta. Efectivamente, este resultado es coherente con el método, ya que hay una información acerca de la verdad de la regla, pero en muchos casos esta información es muy inestable. Un nuevo ejemplo puede cambiar totalmente la anterior asignación si se demuestra que la regla no era tan cierta. Así desde el punto de vista práctico, sería deseable conocer las reglas que por estar soportadas por un número adecuado de ejemplos, son más estables en su asignación de incertidumbre.

En este trabajo, proponemos una extensión del concepto de frecuencia presentado en [4] y [5] que incorpora la relativa importancia de un ejemplo, y este nuevo concepto de frecuencia genera un algoritmo de aprendizaje que es capaz de usar esta clase de información adicional. También estudiamos el problema de la estabilidad de una asignación de incertidumbre a una regla, y proponemos una definición simple que nos permita rechazar informaciones poco creíbles o poco estables.

2. Frecuencia extendida sobre dominios difusos

Con la idea de definir una frecuencia sobre dominios difusos, en [4] definimos el grado de compatibilidad entre un subconjunto A del dominio $D=\{a_1, a_2, \dots, a_N\}$, compuesto

por conjuntos difusos a_i sobre un referencial U , y el ejemplo e de una variable X , que toma valores sobre el mismo referencial U o sobre partes difusas de U , de la siguiente forma

$$\Pi_A(e) = \sup_{a \in A} \Pi_a(e) \quad (3)$$

en donde

$$\Pi_a(e) = \frac{\pi_a(e)}{\sup_{a \in D} \pi_a(e)} \quad (4)$$

con $\pi_a(e) = \sup_u \{\mu_a(u) * \mu_e(u)\}$, en donde $\mu_a(\cdot)$ y $\mu_e(\cdot)$ son las funciones de pertenencia de a y e respectivamente y $*$ es una t-norma. Cuando e es un ejemplo crisp entonces $\pi_a(e)$ coincide con $\mu_a(e)$.

Obviamente, $\Pi_A(e)$ puede ser interpretado como la medida de posibilidad de A dada la evidencia contenida en el ejemplo e , esto es,

$$\Pi_A(e) = \text{Poss}(A/e). \quad (5)$$

Esta distribución de posibilidad nos permitió en [4] definir la frecuencia de cualquier subconjunto A de D a través del conjunto de ejemplos $E = \{e_1, e_2, e_3, \dots, e_h\}$ como el promedio de todas las medidas de posibilidad de un suceso A dados los distintos ejemplos de E , esto es,

$$g(A) = \begin{cases} 0 & \text{si } A = \emptyset \\ \frac{1}{h} \sum_j \Pi_A(e_j) & \text{en otro caso} \end{cases} \quad (6)$$

Asociado con la medida g podemos definir la medida dual

$$f(A) = 1 - g(\bar{A}) = 1 - \frac{1}{h} \sum_{j=1}^h \sup_{a \notin A} \Pi_a(e_j) \quad (7)$$

El siguiente resultado, que fue demostrado en [4], nos proporciona un contexto teórico para las frecuencias sobre dominios difusos, que nos permite considerarlas como una generalización apropiada del concepto clásico de frecuencia.

Teorema

El par de medidad duales (f, g) es un par creencia-plausibilidad en el sentido de la teoría de Dempster/Shafer. $\square$

Cuando el dominio y los ejemplos están en el mismo nivel (por ejemplo, los elementos del dominio, y los ejemplos son conjuntos difusos con el mismo nivel de granularidad), entonces g es una frecuencia clásica que induce una medida de probabilidad, que como es conocido es un caso particular de función de plausibilidad en la que $f=g$, y la incertidumbre queda determinada por un único valor. En otro caso, cuando los datos están en un diferente nivel al del dominio (por ejemplo los ejemplos son número reales y el dominio está compuesto por subconjuntos difusos) obtenemos una frecuencia que tiene asociado como modelo de medida las evidencias de la teoría de Dempster/Shafer, y la incertidumbre se representa mediante un intervalo.

El concepto de frecuencia que define (6) puede interpretarse como un promedio de las diferentes medidas de posibilidad de A variando en los distintos elementos de E . En este caso, podemos interpretar que cada ejemplo se ha considerado como igualmente importante, y con un peso relativo de $1/h$. Una modificación simple de esta expresión nos permitirá incluir la importancia de los diferentes ejemplos.

Sea $E=\{e_1, e_2, e_3, \dots, e_h\}$ el conjunto de ejemplos y sea $I(.)$ una función de E a un conjunto ordenado. Supondremos que $I(.)$ representa la importancia que se le ha dado a cada uno de los ejemplos, y que toma valores en una determinada escala. La función $I(.)$ puede construirse, por ejemplo, a partir de la información dada por un experto. El valor $I(e_j)$ puede ser dado de diferentes formas, en este trabajo proponemos una forma muy simple que nos permite obtenerla fácilmente y además nos permite incluirla en la definición de frecuencia.

Tomaremos una escala lingüística representando cuatro grados de importancia para un ejemplo

$$\{\text{esencial}, \text{alta}, \text{normal}, \text{baja}\} \quad (8)$$

en donde *esencial* significa que el efecto de este ejemplo es equivalente al efecto de diez ejemplos *normales*. La etiqueta *alta* significa que este ejemplo es equivalente a cinco ejemplos *normales*, y *baja* significa que un ejemplo *normal* es equivalente a diez ejemplos de importancia *baja*. Fijando el valor de un ejemplo normal a 1 conseguimos la siguiente relación lingüística/numérica

$$\text{esencial}=10, \text{alta}=5, \text{normal}=1, \text{baja}=0.1 \quad (9)$$

Una vez que hemos obtenido la función $I(.)$ a partir del experto, necesitamos transformarla para conocer el impacto relativo de ese ejemplo dentro del conjunto completo de ejemplos. De esta forma, si notamos a $W(E)=\sum_{e \in E} I(e)$, la importancia relativa de un ejemplo es definida por

$$I_r(e_j) = I(e_j)/W(E) \quad (10)$$

y ahora podemos proponer la siguiente definición extendida de frecuencia.

Definición 2.1

Sea $D = \{a_1, a_2, \dots, a_N\}$ el dominio difuso asociado a una variable X , en donde a_i son conjuntos difusos sobre un conjunto referencial U . Sea $E = \{e_1, e_2, \dots, e_h\}$ una secuencia de ejemplos de la variable X obtenidos a través de algún proceso de muestreo y tomando valores sobre D o sobre cualquier dominio coherente con D , y sea $I_r(e_j)$ la importancia relativa de cada ejemplo. La w -frecuencia de cualquier subconjunto A de D a través del conjunto E es

$$gw(A) = \sum_j \Pi_A(e_j) I_r(e_j) \quad (11)$$

si $A \neq \emptyset$ y $gw(A) = 0$ en otro caso. $\square$

Obviamente, esta definición extiende a la definición anterior ya que cuando todos los ejemplos son tomados como normales (o de la misma importancia), entonces $I(e_j) = 1 \forall j$ y $I_r(e_j) = 1/h \forall j$, por lo tanto, $gw(A) = g(A)$.

En otro caso, $gw(A)$ es capaz de incorporar información adicional acerca de los ejemplos.

La w -frecuencia $gw(A)$ puede interpretarse como el promedio ponderado de todas las medidas de posibilidad de un suceso A dados los diferentes ejemplos de E , y además podemos probar fácilmente que es una medida de plausibilidad ya que es una combinación convexa de medidas de posibilidad. Por lo tanto, podemos definir su medida dual de la siguiente forma

$$fw(A) = 1 - gw(\bar{A}) = 1 - \sum_{j=1}^h I_r(e_j) \sup_{a \notin A} \Pi_a(e_j) \quad (12)$$

y el par de medidas duales (fw, gw) es de nuevo una creencia-plausibilidad en el sentido de la teoría de Dempster/Shافر.

En la definición anterior podemos modificar la escala lingüística que representa los grados de importancia de los ejemplos. Por ejemplo, podemos tomar una representación difusa de los términos lingüísticos {esencial, alta, normal, baja}. En este caso, la frecuencia gw es también un conjunto difuso, y la relación existente entre frecuencia sobre dominios difusos y las medidas de evidencia desaparece. En

general, si cambiamos o extendemos la escala lingüística (8) la anterior definición no necesita ser modificada.

Ejemplo

Supongamos un conjunto de ejemplos $E=\{e_1, e_2\}$, en donde conocemos que el primer ejemplo tiene una credibilidad baja y el segundo ejemplo es bastante representativo. A partir de esta información tomaremos la siguiente asignación de importancia

$$I(e_1)=\text{baja y } I(e_2)=\text{alta.}$$

Los grados de compatibilidad entre los ejemplos y el subconjunto A del dominio vamos a suponer que son

$$\Pi_A(e_1)=0.1 \quad \Pi_A(e_2)=1$$

La importancia relativa de cada ejemplo es muy fácil de calcular y toma los siguientes valores

$$I_r(e_1)=0.02 \quad I_r(e_2)=0.98$$

y en definitiva la w-frecuencia es

$$gw(A)=0.1*0.02+1*0.98=0.982$$

mientras que la anterior definición de frecuencia (que no contempla la función $I_r(\cdot)$) genera el siguiente valor

$$g(A)=0.1*0.5+1*0.5=0.55$$

Por lo tanto $gw(A)$ es más adecuada que $g(A)$ cuando tenemos algún tipo de información adicional sobre la importancia de los ejemplos. □

3. Modelo de aprendizaje de reglas

Supongamos que tenemos el siguiente conjunto de ejemplos $E=\{e_1, e_2, \dots, e_h\}$. El método de aprendizaje que proponemos consiste en calcular para cada regla posible con una estructura fijada, el grado en el que la regla puede ser considerada como una "regla verdadera para el sistema". Este grado se interpretará como un grado de incertidumbre y se calculará a partir de la evidencia que se encuentra en los ejemplos.

Supongamos la siguiente estructura para una regla

$$R_{ij} = \text{if } X \text{ is } A_i \text{ then } Y \text{ is } B_j [\alpha_{ij}, \beta_{ij}] \quad (13)$$

el algoritmo de aprendizaje tratará de estimar los dos parámetros asociados a la regla $[\alpha_{ij}, \beta_{ij}]$, que representan la incertidumbre asociada a esa regla. Por tanto, las fórmulas de aprendizaje consisten en la estimación de ambos parámetros. Esta estimación utiliza el concepto de w-frecuencia del siguiente modo

$$\begin{aligned} \alpha_{ij} &= wf_h(B_j / A_i) \\ \beta_{ij} &= wg_h(B_j / A_i) \end{aligned} \quad (14)$$

Para poder utilizar estas fórmulas necesitamos dos nuevos conceptos. En primer lugar una definición de frecuencia sobre el producto cartesiano y en segundo lugar una definición de frecuencia condicional.

Con respecto a la primera cuestión, si suponemos que la importancia relativa $I_r(.)$ se asigna a un ejemplo completo $e = (e_1, e_2, \dots, e_M)$, y no a cada una de sus componentes, entonces la frecuencia de un vector de conjuntos difusos puede calcularse fácilmente utilizando la siguiente definición de grado de compatibilidad

$$\Pi_{(B_1, B_2, \dots, B_M)}(e) = \prod_i \Pi_{B_i}(e_i) \quad (15)$$

y la definición general de w-frecuencia.

Con respecto al segundo problema, la definición de una frecuencia condicional puede obtenerse a partir de las diferentes definiciones de condicionamiento para medidas de evidencia propuestas en la literatura. Por ejemplo, podemos usar cualquiera de las medidas condicionales de creencia y plausibilidad que han sido propuestas en la literatura

$$\begin{aligned} \text{Pl}_D(A/B) &= \frac{\text{Pl}(A, B)}{\text{Pl}(B)} \\ \text{Bel}_D(A/B) &= \frac{\text{Pl}(B) - \text{Pl}(\neg A, B)}{\text{Pl}(B)} \end{aligned} \quad (16)$$

or

$$\begin{aligned} \text{Pl}_U(A/B) &= \frac{\text{Pl}(A, B)}{\text{Pl}(A, B) + \text{Bel}(\bar{A}, B)} \\ \text{Bel}_U(A/B) &= \frac{\text{Bel}(A, B)}{\text{Bel}(A, B) + \text{Pl}(\bar{A}, B)} \end{aligned} \quad (17)$$

entre otras alternativas (ver [9] para justificar estas propuestas).

Una vez que el algoritmo de aprendizaje ha procesado un fichero de ejemplos, la salida es un conjunto de reglas cada una afectada por una evidencia de tipo intervalar. Este intervalo puede interpretarse como un límite inferior y superior de la evidencia que nos han proporcionado los ejemplos acerca de la "habilidad" de la regla para representar a los datos. La causa por la que obtenemos un intervalo en vez de un único número (una probabilidad) es la vaguedad de la representación del dominio. El algoritmo calcula en el límite superior, la adecuación entre la regla y los ejemplos, y en el límite inferior procesa la información acerca de la mejor adecuación con otras reglas de igual antecedente pero diferente consecuente.

El grado final que genera el algoritmo de aprendizaje puede considerarse una mixtura entre incertidumbre (un antecedente implica a varios posibles consecuentes) y vaguedad (un ejemplo puede estar "bien representado" por diferentes reglas difusas).

4. Estabilidad de la evidencia asociada a una regla

El método de aprendizaje de reglas difusas presentado en la sección anterior tiene como resultado final un conjunto de reglas, cada una afectada con un grado de incertidumbre. El problema que nos planteamos en esta sección es examinar la información final que nos proporciona este grado de incertidumbre sobre la propia regla. Recordemos que el método de aprendizaje está basado en el cálculo de ciertas frecuencias sobre algunos patrones asociados a la regla. Este método puede traernos problemas cuando algunas evidencias están soportadas por muy pocos ejemplos. Consideremos el dominio difuso

$$D=\{HN,MN,SN,Z,SP,MP,HP\} \quad (18)$$

compuesto por los siguientes números difusos trapezoidales

$$\begin{aligned} HN=(-1000,-50,1,25) \quad MN=(-25,-25,25,15) \quad SN=(-10,-10,15,10) \quad Z=(0,0,10,10) \\ SP=(10,10,10,15) \quad MP=(25,25,15,25) \quad HP=(50,1000,25,1) \end{aligned}$$

representados en la figura 1.

En este caso, utilizando el algoritmo propuesto en la sección anterior y el siguiente ejemplo

$$x1=11, x2=49, y=0 \quad (19)$$

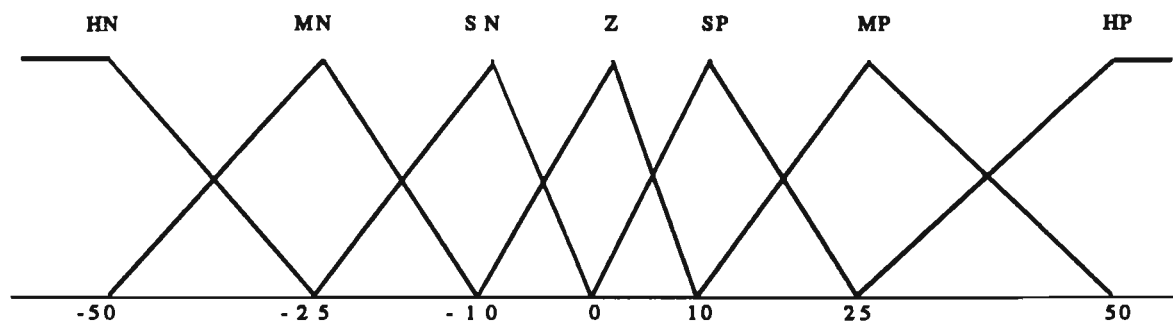


Fig.1. Dominio compuesto por siete conjuntos difusos

se obtienen las siguientes asignaciones de creencia

- R1: if X1 is SP and X2 is HP then Y is Z [1,1]
 R2: if X1 is MP and X2 is HP then Y is Z [1,1] (20)
 R3: if X1 is SP and X2 is MP then Y is Z [1,1]

Las tres reglas R1, R2 y R3 se han considerado como completamente ciertas, y para llegar a esta afirmación se ha tenido en cuenta tan sólo un único ejemplo. Ahora bien, si examinamos este ejemplo vemos que queda muy bien representado por la regla R1, y tan sólo escasamente representado por las otras dos. ¿Por qué el algoritmo asigna igual creencia a las tres? La razón es que sin más información que procesar, el algoritmo detecta, por ejemplo, que el ejemplo es positivo a la regla R2, y que como no hay ninguna información en contra, la regla se debe aceptar como cierta. El problema es bien conocido, cualquier método frecuentista necesita estar soportado por un número suficiente de ejemplos para ser creíble, y para que las asignaciones de incertidumbre sean mínimamente estables.

Una solución que podemos manejar cuando no se dispone de suficientes ejemplos para cubrir todas las reglas posibles (ya sea positiva o negativamente) es la de introducir el concepto de fiabilidad sobre la asignación de creencia a la regla o de estabilidad de esta asignación. Este concepto seguramente no es fácil de establecer, o de llegar a definir, pero desde luego si sabemos para que lo queremos, una regla será estable si su evidencia está apoyada por un número suficiente de ejemplos. Además, si incluimos el concepto de importancia relativa de un ejemplo, introducido en la sección segunda, podríamos matizar la anterior caracterización, diciendo que una regla es fiable si su evidencia está apoyada por un número suficiente de ejemplos importantes.

Para llegar a materializar estas ideas será necesario establecer un concepto nuevo, el de número suficiente de ejemplos. En el caso más simple podríamos tomar un número real que fijará este concepto para cada problema, pero también podemos intentar

flexibilizar en lo posible un concepto que ya en su propia naturaleza es vago. Si conocemos que para poder afirmar que el número de ejemplos es suficiente es indispensable que sea superior a $n1$ y es recomendable que sea superior a $n2$, con $n1 \leq n2$, podemos admitir que una representación adecuada para el número de ejemplos suficientes, es la correspondiente a la función de pertenencia de la figura 2 y que notaremos por $NS(n1, n2)$.

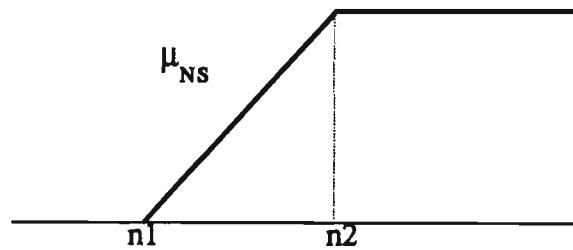


Fig.2. Representación del conjunto difuso número de ejemplos suficiente

Utilizando este concepto, necesitamos ahora evaluar el número de ejemplos que apoyan la evidencia de una regla. Este número debe estar relacionado con aquellos ejemplos que se han utilizado para el cálculo de la evidencia. Este número no debe ser simplemente el cardinal de un subconjunto de $E=\{e_1, e_2, \dots, e_h\}$, ya que no todos los ejemplos tienen la misma influencia a la hora de calcular la evidencia de la regla, y además deberíamos incluir la importancia relativa de cada regla en este cálculo. Por todo ello, dada la regla

$$R_{ij}: \text{if } X \text{ is } A_i \text{ then } Y \text{ is } B_j \quad (21)$$

proponemos la siguiente medida para determinar el número de ejemplos que apoyan la evidencia de la regla

$$\sum_j \Pi_{A_i}(e_j) I(e_j) \quad (22)$$

que obviamente coincide con la w-frecuencia del antecedente de la regla $gw(A_i)$. Así, en definitiva una forma de evaluar la estabilidad de la evidencia de una regla es mediante el siguiente cálculo

$$E(R_{ij}) = \mu_{NS}(gw(A_i)) \quad (23)$$

Curiosamente, todas las reglas con el mismo antecedente tienen la misma asignación $R(R_{ij})$, lo que el lógico puesto que todas esas reglas utilizan los mismos ejemplos, para distribuir la creencia sobre los diferentes valores del consecuente.

Si retomamos el ejemplo anterior con este nuevo concepto, y tomamos los parámetros $n1=1$ y $n2=4$, tenemos la siguiente información adicional, después del procesar el mismo anterior ejemplo ($x1=11$, $x2=49$, $y=0$) pero ahora dos veces

$$\begin{aligned} R1: & \text{if } X1 \text{ is SP and } X2 \text{ is HP then } Y \text{ is Z } [1,1] & E(R1) &= \mu_{NS}(2) = 0.33 \\ R2: & \text{if } X1 \text{ is MP and } X2 \text{ is HP then } Y \text{ is Z } [1,1] & E(R2) &= \mu_{NS}(0.14) = 0 \\ R3: & \text{if } X1 \text{ is SP and } X2 \text{ is MP then } Y \text{ is Z } [1,1] & E(R3) &= \mu_{NS}(0.08) = 0 \end{aligned} \quad (24)$$

tomando la importancia del ejemplo como normal ($I(e)=1$). La primera regla tiene aún poca estabilidad, pero desde luego las otras dos ninguna.

Obviamente este proceso es muy dependiente de la selección de la función de pertenencia de NS, es decir, de los parámetros $n1$ y $n2$. Pero una vez fijados estos valores obtenemos una información muy clara sobre el soporte sobre el que se ha calculado esta evidencia sobre cada regla, y en definitiva información sobre la estabilidad y fiabilidad de la misma. En la práctica reglas con evidencia poco fiable pueden ser rechazadas. Por ejemplo, con un umbral de estabilidad de 0.3 el algoritmo filtra las reglas mostradas en (24), y sólo admite como regla para el sistema, con la información disponible, la regla R1.

Referencias

- [1] Binaghi, E., Empirical learning for fuzzy knowledge acquisition, Proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks, Iizuka'92, 245-251, 1992.
- [2] Campos, L.M. de, González, A., A fuzzy inference model based on an uncertainty forward propagation approach, IFSA'91 Artificial Intelligence (Eds. R. Lowen, M. Roubens) 13-16, 1991.
- [3] Campos, L.M. de, Moral S., Learning rules for a fuzzy inference model, Proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks, Iizuka'92, 253-256, 1992.
- [4] Delgado, M., González, A., A frequency model to assess Dempster/Shافر's evidences, Tech. Report n. 92-1-1 Department of Computer Science and Artificial Intelligence (DECSAI), Universidad de Granada, 1992, también sometido en International Journal of Approximate Reasoning.
- [5] Delgado, M., González, A., An inductive learning procedure to identify fuzzy systems, aceptado para su publicación en International Journal for Fuzzy Sets and Systems.

- [6] King P.J., Mandani E.H., The application of fuzzy control systems to industrial processes, *Automatica*, 13 (3), 235-242, 1977.
- [7] Lee, C.C., A self-learning rule-based controller employing approximate reasoning and neural net concepts, *International Journal of Intelligent Systems* 6, 71-93, 1991.
- [8] Leung K.S., Leung Y., So L. Yam K.F., Rule learning in expert systems using genetic algorithm:1, concepts, *Proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks, Iizuka'92*, 201-204, 1992.
- [9] Moral, S., Campos, L.M. de, Updating uncertain information, *Uncertainty in Knowledge bases* (B. Bouchon-Meunier, R.R. Yager, L.A. Zadeh (eds.)), *Lectures Notes in Computer Science* 521, Springer-Verlag, 58-67, 1991.
- [10] Mukaidono M., Yamaoka M., A learning method of fuzzy inference rules with neural networks and its applications, *Proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks, Iizuka'92*, 185-187, 1992.
- [11] Pedricz, W., An identification algorithm in fuzzy relational systems, *Fuzzy Sets and Systems* 13, 153-167, 1984.
- [12] Peng, X., Wang, P., On generating linguistic rules for fuzzy models, *Lectures Notes in Computer Science* 313, *Uncertainty and Intelligent Systems* (B.Bouchon, L. Saitta, R.R. Yager (eds.)), 185-192 (1988).
- [13] Sestito, S., Dillon T., Using single-layered neural networks for the extraction of conjunctive rules and hierarchical classifications, *Journal of Applied Intelligence* 1, 157-173, 1991.
- [14] Sugeno M., Nishida M., Fuzzy control of model car. *Fuzzy Sets and Systems*, 16, 110-113, 1985.
- [15] Sugeno, M., Tanaka, K., Successive identification of a fuzzy model and its applications to prediction of a complex system, *Fuzzy Sets and Systems* 42, 315-334, 1991.
- [16] Takagi, T., Sugeno, M., Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. on Systems, Man and Cyber.*, vol. SMC-15, no. 1, 116-132, 1985.
- [17] Yue Z., Fengchi L., Fen S., Shounan B., Yunxiang P., A fuzzy production rule based expert system, *Fuzzy Sets and Systems* 44, 391-403, 1991.

EXPERIENCIAS EN EL RECONOCIMIENTO DE CARACTERES MEDIANTE UN SISTEMA BASADO EN UNA RED NEURAL Y UN AUTOMATA FUZZY

R. Reina, J.R. Gonzalez de Mendivil, J. Echanobe, J.R. Garitagoitia.
Dpto. de Electricidad y Electrónica; Universidad del Pais Vasco; Facultad de Ciencias;
Apdo. 644. 48080 Bilbao, España.

Resumen - Este trabajo presenta los resultados de una experiencia realizada en el campo del reconocimiento de caracteres escritos. Se ha utilizado un Sistema de Reconocimiento Optico de Caracteres basado en una Red Neural, al que se ha incorporado la capacidad de utilizar el contexto, con el fin de mejorar las tasas de reconocimiento cuando los caracteres están altamente degradados. El Sistema utiliza un Autómata Fuzzy para definir el contexto lexicográfico. Se han realizado numerosas pruebas con textos de imprenta degradados y con caracteres manuscritos, y se ha comprobado la gran efectividad del Autómata para recuperar caracteres erróneamente reconocidos por la Red. Finalmente, la introducción de una función "media" asociada a la función de salida del Autómata añade una nueva mejora respecto a las tasas obtenidas con la función de salida clásica.

Palabras Clave - Reconocimiento de Caracteres, Redes Neuronales, Autómatas Fuzzy, Contexto Lexicográfico.

1. Introducción.

En los últimos años se han desarrollado Sistemas de Reconocimiento de Caracteres basados en Redes Neuronales, como los propuestos en [1,2,3], los cuales presentan, en general, buenas tasas de reconocimiento. Sin embargo, cuando los caracteres están seriamente distorsionados en relación al patrón, bien sea por malas reproducciones o incluso por estar escritos a mano, las tasas de reconocimiento caen significativamente. Varios autores [4,5,6,7] han propuesto la utilización del contexto como medio para intentar paliar este problema.

En este trabajo se presenta un Sistema de Reconocimiento de Caracteres basado en una Red Neural que incorpora un diccionario que constituye el contexto lexicográfico bajo el cual son analizados los caracteres. Tipicamente, el contexto lexicografico ha sido introducido mediante la utilización de Autómatas de Estados Finitos [8]. Además, una ventaja añadida, es que existen métodos simples y eficaces para su construcción a partir de un diccionario de palabras [9]. Debido a la naturaleza de la información que va a suministrar la Red Neural, se utiliza un Autómata Fuzzy [10] que va a aceptar cadenas de caracteres fuzzy suministradas por la Red Neural de acuerdo a las secuencias establecidas por un diccionario.

Se han empleado dos funciones de salida en el Automata Fuzzy, la primera es la clásica, basada en la utilización de las funciones "max-min", y la segunda basada en la utilización de una función media. Se han realizado numerosas pruebas con textos de imprenta degradados y con caracteres manuscritos cuyos resultados presentamos más adelante. Con objeto de comprobar la utilidad del Autómata Fuzzy se ha medido la tasa de reconocimiento tanto a la salida de la Red Neural como a la salida del Sistema. Los resultados obtenidos han puesto en evidencia que hay una gran cantidad de caracteres que la Red Neural no puede reconocer y que fueron recuperados por el Autómata Fuzzy.

El resto del artículo está organizado de la siguiente manera: en el Apartado II se realiza una revisión del sistema, explicando las partes fundamentales de que consta. En el Apartado III se

detallan las pruebas realizadas y los resultados obtenidos. Por último, en el Apartado IV se presentan las conclusiones derivadas de las experiencias.

2. Revisión del Sistema.

En este apartado se hace una descripción del Sistema de Reconocimiento de Caracteres propuesto. Los componentes principales del Sistema (figura 1) son los siguientes: (a) *Escanner-Normalizador*, realiza la adquisición de la imagen, la localización de los caracteres del texto y la generación de un *carácter vector*; (b) *Red Neural*, clasifica los caracteres vector de acuerdo con un conjunto de caracteres patrón y genera *caracteres fuzzy*; (c) *Autómata Fuzzy*, acepta cadenas de *caracteres fuzzy* de acuerdo con un diccionario, generando finalmente un fichero de texto con el resultado del reconocimiento.

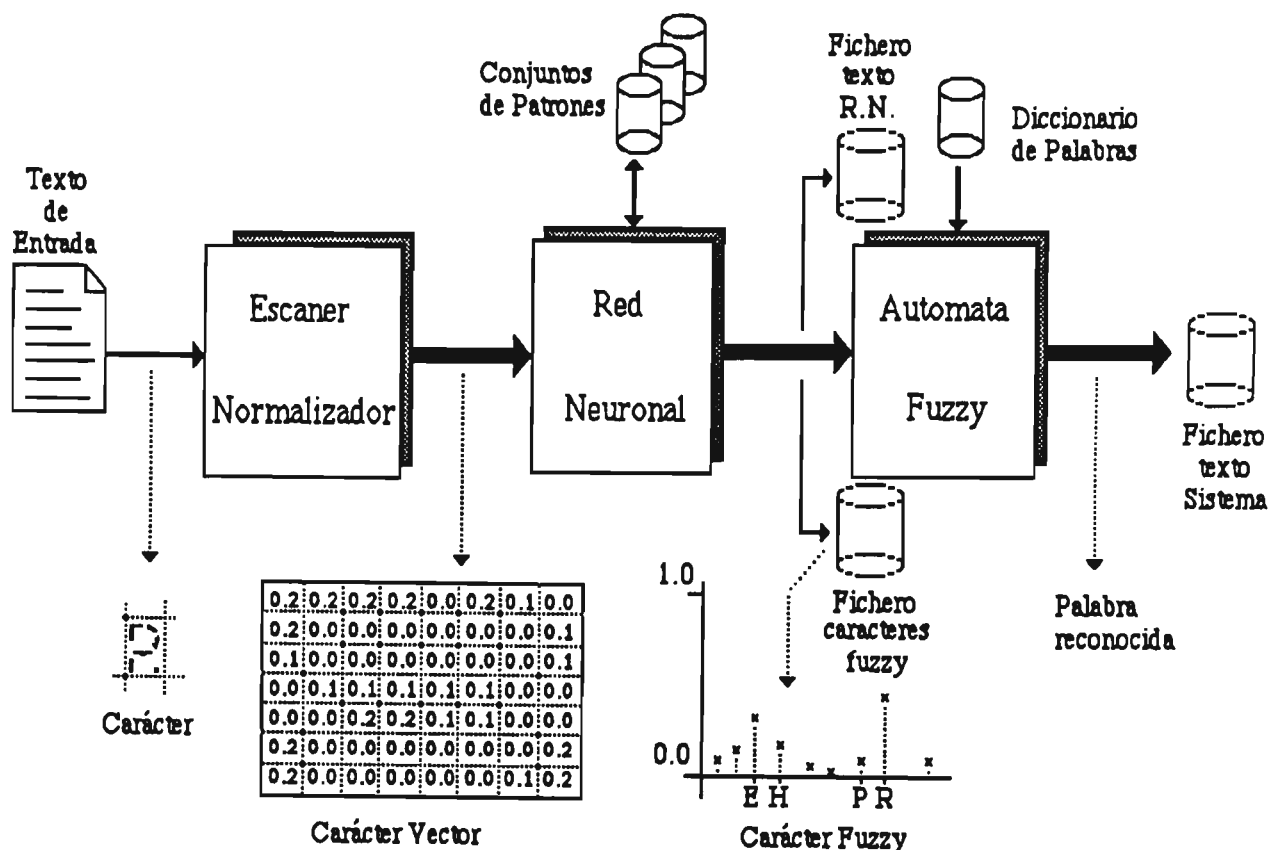


Figura 1: Esquema general del sistema.

La principal novedad con respecto a un Sistema de Reconocimiento de Caracteres basado en una Red Neural es la incorporación de un Autómata Fuzzy que permite representar el contexto. A continuación se realiza una descripción más detallada de los principales componentes.

2.1. Red Neural.

Se ha realizado una Red Neural basada en el modelo del Perceptron Multicapa [11,12]. Si bien, inicialmente se utilizó una Red con tres capas, en estos momentos se utiliza una Red sin capa oculta,

lo que no ha tenido efecto negativo sobre la capacidad de reconocimiento del Sistema. La capa de entrada posee 64 unidades que corresponden a las componentes del carácter vector. La capa de salida tiene un número variable de unidades que se ajustan de acuerdo con el número de patrones a reconocer. El algoritmo utilizado para el aprendizaje de la Red Neural es el algoritmo de "Back-Propagation" [13].

La entrada a la Red Neural la constituyen los caracteres vector suministrados por el Escaner-Normalizador. Este proceso toma una imagen del texto, identifica sus líneas y realiza la localización y aislamiento de los caracteres en la línea, de izquierda a derecha secuencialmente.

El área de un carácter aislado se divide en 64 zonas rectangulares iguales. Para cada una de estas zonas se calcula el cociente entre el número de pixels a uno y el número total de pixels, lo que permite representar el carácter como una matriz, $M[8 \times 8]$, de números reales con valores entre cero y uno. Finalmente, el carácter vector se construye ordenando la matriz M por filas y se procede a su normalización.

El aprendizaje de la Red Neural se realiza con un conjunto de caracteres patrón definidos previamente. Para cada carácter patrón se elige una unidad de la capa de salida que tomará el valor uno, mientras que el resto de las unidades tomará el valor cero. De esta forma cada unidad de la capa de salida se identifica con un patrón. La salida del proceso de aprendizaje se puede almacenar en un fichero en disco, lo que permite tener aprendidos distintos repertorios de caracteres patrón.

En el proceso de clasificación, cuando un carácter es presentado a la entrada de la Red Neural, el carácter reconocido es el que se corresponde con el patrón cuya unidad de la capa de salida asociada tiene el mayor valor. Este valor puede interpretarse como la "proximidad" del carácter al patrón (uno representa un total acuerdo, cero representa un total desacuerdo). Los valores de todas las unidades de la capa de salida representan, de la misma manera, la "proximidad" del carácter a cada uno de los patrones asociados a dichas unidades.

Es posible definir un Conjunto Fuzzy [14], denotado X , asociado al carácter de entrada en el universo de caracteres patrón Σ de la siguiente manera:

$$X = \{ (p, \mu_X(p)) \mid p \in \Sigma, \mu_X(p) = y_p \in [0, 1] \}$$

donde y_p es el valor de la unidad de la capa de salida asociada al patrón $p \in \Sigma$. La función μ_X de Σ en $[0, 1]$ es la función de pertenencia del conjunto fuzzy X .

Así pues, la Red Neural va a construir un conjunto fuzzy que denominamos carácter fuzzy, para cada carácter vector de entrada. Este método nos permite arrastrar toda la información obtenida por la Red Neural y posponer la decisión de identificar el carácter hasta no haber considerado el contexto.

2.2. Autómata Fuzzy.

El contexto lexicográfico de un carácter se define mediante un diccionario constituido por palabras o cadenas de caracteres. Es conocido que si el lenguaje definido por el diccionario es regular [9] es posible construir un reconocedor para las cadenas del lenguaje basado en un Autómata de Estados Finito. Existen métodos que permiten la construcción automática de un Autómata de Estados Finito Determinista a partir de la definición del diccionario [9].

Un Autómata de Estados Finitos Determinista [8] es una quintupla de elementos $\{Q, \Sigma, \delta, q_0, F\}$ siendo Q un conjunto finito de estados, Σ el alfabeto de entrada, δ la función de transición ($\delta: Q \times \Sigma \rightarrow Q$), $q_0 \in Q$, el estado inicial y $F \subset Q$, el conjunto de estados finales.

La entrada del Autómata es la salida de la Red Neural. Puesto que el carácter suministrado por la Red Neural es un carácter fuzzy, el Autómata es, en realidad, un Sistema Deformado [15]. Un Sistema Deformado es un complejo $\{(\Sigma, X), (Q, S), (W, T), \delta, \sigma, q_0\}$ siendo Σ , Q y W los conjuntos de entradas, estados y salidas que están limitando la borrosidad establecida por los conjuntos fuzzy X , S y T respectivamente. δ es la función de transición ($\delta: \Sigma \times Q \rightarrow Q$) cuyo dominio está limitado por $X \times S$ y su rango por S . σ es la función de salida ($\sigma: Q \rightarrow Y$) cuyo dominio está limitado por S y su rango por T . q_0 es el estado inicial no fuzzy. Las ecuaciones que definen la transición de estado en un Sistema Deformado son:

$$q_{i+1} = \delta(p_i, q_i) \quad w_i = \sigma(q_i)$$

donde δ y σ satisfacen:

$$\mu_S(q_{i+1}) \geq \min\{\mu_X(p_i), \mu_S(q_i)\} \quad \mu_T(w_i) \geq \mu_S(q_i)$$

Partiendo del estado inicial se realizan las transiciones para los caracteres fuzzy de entrada obteniéndose conjuntos fuzzy de estados del Autómata. Cuando se alcanza el fin de palabra se selecciona de entre el conjunto fuzzy de salida alcanzado $T = \{(w, \mu_T(w)) / w \in W\}$, aquella palabra w con el mayor valor de la función de pertenencia en T .

Si todas las w tienen sus $\mu_T(w)$ nulas o muy bajas entonces la palabra no es reconocida como perteneciente al diccionario.

Otra aproximación se basa en definir la $\mu_T(w)$ como la media obtenida a partir de los valores de pertenencia de los componentes de los caracteres fuzzy que etiquetan el camino desde el estado inicial hasta el estado q , siendo $w = \delta(q)$:

$$\mu_T(w) = \frac{1 + \sum_{k=0}^{|w|} \mu_X(p_k)}{|w|+1}$$

siendo p_k , el componente que etiqueta el camino del k -ésimo carácter fuzzy de la cadena de entrada.

En este caso también se selecciona desde el estado fuzzy de salida alcanzado, aquella palabra w con el mayor valor de la función de pertenencia.

3. Descripción del experimento.

Con objeto de probar el sistema construido se han realizado tres tipos de pruebas sobre caracteres mayúsculas. El primer conjunto de pruebas consistió en reconocer caracteres de imprenta de distintos tipos y tamaños. El segundo consistió en reconocer textos manuscritos de distintas personas. Por último, el tercer tipo de pruebas consistió en reconocer caracteres de imprenta

deteriorados. El contexto utilizado en todas ellas ha sido el conjunto de palabras reservadas del Pascal.

Con objeto de evaluar el efecto de la incorporación del contexto, en todas las pruebas realizadas se ha medido la tasa de reconocimiento a la salida de la Red Neural, considerada como el cociente entre el número de caracteres reconocidos y el número total de caracteres de la prueba, y la tasa de reconocimiento a la salida del Sistema, considerada como el cociente entre el número de palabras reconocidas y el número total de palabras de la prueba. A continuación se describen las pruebas realizadas.

Para el reconocimiento de caracteres de imprenta de distintos tipos y tamaños fueron preparados textos con las palabras reservadas del Pascal impresas en letras mayúsculas con cinco tipos de letra diferentes y cada uno de ellos con cinco tamaños. Sin embargo la Red Neural sólo fue entrenada con los 26 caracteres mayúsculas correspondientes a un tamaño de cada tipo. Las tasas de reconocimiento obtenidas, en tantos por ciento, sobre los textos anteriores tanto por la Red Neural como por el Sistema se muestran en la tabla 1.

	Times			Helvética			Avant-Garde			Bookman			Palatino		
9	93.89	100	100	98.69	100	100	98.69	100	100	97.38	100	100	93.01	89.58	100
10	98.25	100	100	99.56	100	100	99.56	100	100	96.51	100	100	93.01	91.67	100
12	99.13	100	100	100	100	100	100	100	100	100	100	100	100	100	100
14	100	100	100	99.56	100	100	100	100	100	98.69	100	100	98.69	100	100
18	100	100	100	98.25	91.67	100	98.25	100	100	98.25	100	100	96.07	100	100

Tabla 1: Resultados de reconocimiento de textos impresos.

En cada prueba la primera columna corresponde a la tasa de reconocimiento de la Red Neural, la segunda a la del Sistema cuando el Autómata Fuzzy opera con la función de salida clásica y la tercera cuando el Autómata opera tomando como función de salida la función media ya descrita en apartados anteriores.

A continuación se procedió al reconocimiento de caracteres manuscritos, para lo cual fueron realizadas una serie de pruebas sobre textos escritos por diez personas diferentes, obteniéndose así una cierta variedad de estilos de escritura a reconocer. Para este tipo de pruebas la Red Neural fue previamente entrenada con los 26 caracteres mayúsculas del alfabeto correspondientes a cada estilo de escritura.

Las tasas de reconocimiento obtenidas para los textos manuscritos, tanto por la Red Neural como por el Sistema se muestran en la tabla 2. La interpretación de las columnas es similar a la tabla 1.

I.D.C.	75.55	91.67	100	R.M.	49.78	77.08	93.75
R.R.I.	59.83	75.00	89.58	G.B.	63.32	95.83	100
H.A.S.	50.66	87.50	93.75	I.S.	66.81	81.25	100
J.R.	56.77	81.25	97.92	J.R.P.	41.48	39.58	85.42
J.R.G.	53.28	47.92	91.67	B.M.	55.02	68.75	85.42

Tabla 2: Resultados de reconocimiento de textos manuscritos.

Por último, se procedió al reconocimiento de caracteres de imprenta con distinto grado de deterioro. Para ello, la Red Neural fue previamente entrenada con los 26 caracteres mayúsculas del alfabeto impresos por un MacIntosh con el tipo Times en tamaño de 12 puntos. El deterioro de un carácter consiste en un borrado aleatorio de los pixels del carácter. Es posible definir distintos factores de probabilidad de borrado de pixels para obtener diferentes niveles de deterioro (máximo nivel 100%).

Las tasas de reconocimiento obtenidas tanto por la Red Neural como por el Sistema para tres factores de deterioro, 90, 95 y 97%, se muestran en la tabla 3.

Deterioro: 90%			Deterioro: 95%			Deterioro: 97%		
86.20	98.75	100	66.20	94.17	97.92	40.87	83.33	90.84

Tabla 3: Resultados de reconocimiento de textos impresos deteriorados.

Las palabras que contiene el diccionario pueden tener efecto sobre la tasa de reconocimiento del sistema. Si definimos la distancia entre dos palabras (de la misma longitud) como el número mínimo de letras que es necesario cambiar en una de ellas para obtener la otra, entonces con un diccionario con distancias bajas entre palabras se pueden obtener tasas de reconocimiento inferiores a las dadas anteriormente.

4. Resultados y conclusiones.

El Sistema mantiene unas buenas tasas de reconocimiento incluso cuando el grado de deterioro de los caracteres aumenta fuertemente. Este resultado es debido a la capacidad que tiene el Automata Fuzzy de rectificar una decisión tomada por la Red Neural. Si se utilizara una Red Neural más efectiva sería posible disminuir la diferencia entre las tasas de reconocimiento obtenidas por la Red Neural y por el Sistema. Incluso si esta diferencia fuera nula, la incorporación del contexto proporciona una mayor robustez al Sistema reconocedor.

El Automata Fuzzy utilizado constituye una herramienta sencilla y eficaz de implementar el contexto lexicográfico. Además permite la aplicación de los métodos ya conocidos para su construcción automática a partir del diccionario.

El esquema propuesto se puede utilizar en otras aplicaciones en las que el contexto se pueda expresar como un lenguaje regular. Este contexto lexicográfico puede hacer referencia, tanto a ligaduras espaciales como a ligaduras temporales.

Una posible ampliación de este trabajo es la incorporación del contexto sintáctico, utilizando el Automata Fuzzy para la generación de unidades sintácticas borrosas.

5. Referencias.

- [1] K. Fukushima, S. Miyake, T. Ito, "Neocognitron: A Neural Network for a Mechanism of Visual Pattern Recognition", IEEE Trans. Syst., Man and Cybernetics, vol. SMC-13, no.5, pags. 826-834, Sep.-Oct. 1983.
- [2] Abstract of the First Annual INNS Meeting, vol. 1, Supp. 1, Boston, 1988.
- [3] A. Rajavelu, M. T. Musavi, M. V. Shirvaikar, "A Neural Network Approach to Character Recognition", Neural Networks, vol. 2, pags. 387-393, 1989.

- [4] D.J. Burr, "Designing a Handwriting Reader", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, pags. 554-559, Sept. 1983.
- [5] K. Abend, "Compound Decision Procedures for Pattern Recognition", Proc. NEC 22, pags. 777-780, 1966.
- [6] R. O. Duda, P. E. Hart, "Experiments in the Recognition of Hand-Printed Text: part II - Context Analysis", Proc. FJCC, San Francisco, California, 1968. Thompson, Washington D. C., 1968
- [7] J. Raviv, "Decision Making in Markov Chains applied to the Problem of Pattern Recognition", IEEE Trans. Info. Theory, vol. 13, no. 4, pags. 536-551, 1967.
- [8] J. E. Hopcroft, J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley Publishing Company, Reading Massachusetts, 1979.
- [9] A. V. Aho, J. D. Ullman, "Principles of Compiler Design", Addison-Wesley Publishing Company, Reading Massachusetts, 1978.
- [10] D. Dubois, H. Prade, "Fuzzy Sets and Systems: Theory and Applications", Academic Press, New York, 1980.
- [11] R. Rosenblatt, "The Perceptron: a probabilistic model for information storage and organization in the brain", Psychological review, 65, pags. 383-408, 1958.
- [12] D.B. Parker, "Learning Logic", Invention report S81-64, file-1, Office of Technology Licensing, Stanford University, 1982.
- [13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Back- Propagating Errors", in Parallel Distributed Processing, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, ch. 8, 1986.
- [14] L. Zadeh, "Fuzzy Sets", Inform. and Contr., vol. 8, pags. 338-353, 1965.
- [15] C. V. Negoitia, D. A. Ralescu, "Fuzzy Systems and Artificial Intelligence", Kybernetes, vol. 3, pags. 173-178, 1974.

LA R-FN : UN PROCESADOR ELEMENTAL DE INFORMACION DIFUSA

Ignacio Requena - Miguel Delgado

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION E INTELIGENCIA ARTIFICIAL

FACULTAD DE CIENCIAS. UNIVERSIDAD DE GRANADA.

18071. GRANADA (ESPAÑA)

Resumen: En este trabajo presentamos un modelo de neurona difusa. La R-FN tiene una arquitectura parecida a la neurona artificial clásica, pero proporciona salidas difusas, tras procesar entradas difusas con pesos reales.

Conectando varias R-FN podemos obtener redes, que permiten procesar información difusa presentada en las entradas. Este procesamiento puede hacerse con operadores sobre números difusos, o bien con operaciones sobre números reales que representen a los valores difusos.

Utilizamos para ello, un modificación de la máquina de Boltzmann. Finalmente damos un ejemplo de aplicación.

Palabras clave: Información difusa, redes neuronales artificiales, neurona difusa, máquina de Boltzmann

1.- Introducción.

El cerebro humano procesa generalmente y muy bien además, información imprecisa. Frases como "Es medianamente alto, con ojos tirando a negros, casi rubio y con facciones agradables" son perfectamente procesadas en nuestro cerebro, y no lo son en un ordenador digital convencional.

Esta capacidad del cerebro de procesar información no precisa, es una de las aspiraciones a conseguir con máquinas inteligentes.

La teoría de los Conjuntos Difusos introducida por Zadeh [23] y la lógica difusa, son las herramientas que tenemos actualmente para procesar información difusa. Las Redes Neuronales Artificiales (RNA) pretenden modelizar funciones como-el-cerebro lo hace, y son altamente flexibles por sus funciones de aprendizaje. La fusión de ambos presupuestos puede y debe dar herramientas de procesamiento flexibles y poderosas.

Ambas aproximaciones, conjuntos y lógica difusos (CLD) y RNA, tienen el mismo objetivo final: Procesar información tal como lo hace el cerebro biológico. Podríamos decir que en un caso se modeliza la información a procesar, y en otro caso, se intenta modelizar el procesador.

El acoplamiento de Lógica Difusa y RNA se puede hacer por agregación de características de cada teoría, o bien uniendolos por similitudes entre ambos. Los primeros intentos tratan de incorporar información difusa en el proceso de una RNA. J. Keller-D. Hunt [8], B. Kosko [10,11] y W.R. Taber [19] hacen las primeras aportaciones.

Se han propuesto diversos modelos sobre la unidad elemental de una red neuronal artificial difusa (RNAD), es decir, la neurona difusa. Algunos se presentan para aplicaciones concretas. L.C. Shiue - R.O. Grondin [18], A. F. Rocha - R. Machado [12], R.R. Yager [21], P. O'Hagan [14], T. Yamakawa - S. Tomoda [22] o J.B. Kiszka - M.M. Gupta [9] entre otros, proponen diferentes modelos.

En los últimos años (88-92), han proliferado los trabajos, y hay varias conferencias internacionales ya consolidadas sobre el tema de CLD y RNA. Nuestro propósito es procesar información difusa con RNAs.

En este trabajo presentamos un nuevo modelo de neurona difusa, la R-Neurona Difusa (R-FN), ya apuntada en ([15]), y proponemos conectar en Redes varias R-FN. Estas redes, que son totalmente conectadas, pueden tener una estructura similar a la máquina de Boltzmann [2], aunque tendremos que modificar la función de energía. El procesamiento se puede realizar con operadores sobre cantidades difusas, o bien obteniendo un modelo equivalente no difuso, asociado a la red de unidades difusas.

En este trabajo utilizamos la segunda forma de procesamiento, proponiendo una modificación al algoritmo de la Máquina de Boltzmann (MB). Podremos utilizar esta máquina de Boltzmann modificada para procesar la información difusa que circula por la red. Finalmente, y como ejemplo, aplicamos el modelo a resolver el problema del mayor conjunto independiente de nodos, en un grafo con valoraciones difusas en los nodos.

El desarrollo del trabajo es como sigue. En el apartado 2 presentamos el procesador elemental difuso, indicando sus características, y en el 3 explicamos la conexión en Red de este modelo de procesador.

En el apartado 4, desarrollamos el algoritmo de aprendizaje, e indicamos que este modelo se puede utilizar también como optimizador. En el apartado 5, vemos la aplicación al problema del mayor conjunto independiente de nodos, en un grafo con valores difusos en los nodos.

2.- La R-Neurona artificial difusa.

Presentamos ahora el nuevo modelo de procesador elemental difuso (PED), R-FN. La estructura o arquitectura del modelo es similar al modelo clásico ([13]).

Las entradas son números difusos, mientras que los pesos asociados a cada entrada son valores crisp. La neurona se activa cuando la excitación (difusa) supera un umbral difuso propio. (figura 1).

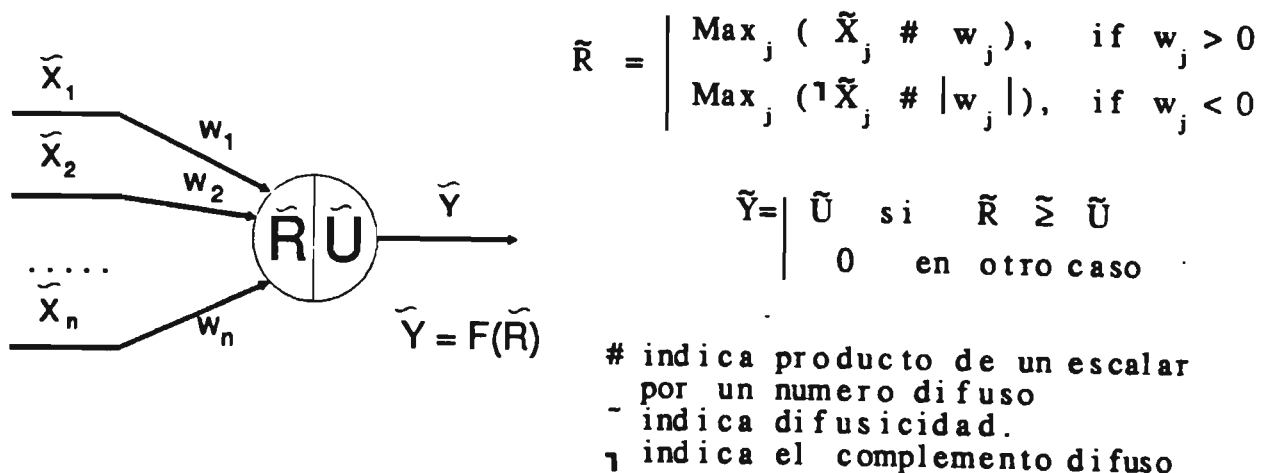


Figura 1.- La neurona difusa R-FN

Suponemos que las entradas difusas están en el rango $[0,1]$ y que las ponderaciones w_j pertenecen al intervalo $[-1,1]$ para que los valores difusos siempre tengan como rango $[0,1]$. Max se entiende como el operador de la lógica difusa.

La propagación del umbral a través del axon de la neurona es novedoso como activación. Sin embargo, y de acuerdo con el problema concreto, la activación de la R-FN puede ser diferente, por ejemplo, proporcionando una salida difusa (diferente al umbral), cuando la excitación supera el umbral difuso, salida que puede ser la misma o no, para diversas neuronas conectadas en red. Incluso puede ser una activación continua, componiendo adecuadamente la función de pertenencia de la excitación difusa con la función de activación, para obtener la función de pertenencia de la salida de la neurona.

Parece claro que no todas las neuronas biológicas son del mismo tipo. Podemos presuponer neuronas cuya tarea sea algún tipo de control, con una estructura parecida a nuestro modelo, con fuerzas sinápticas crisp (o prácticamente crisp) o también difusas. Por ahora consideramos en nuestro modelo, pesos crisp, activación por umbral y salida un valor difuso prefijado.

Para la activación del umbral difuso, tenemos desarrolladas RNA clásicas capaces de comparar números difusos trapezoidales, usando el algoritmo de Retropropagación ([17]), que incluso son capaces de aprender criterios de comparación de decisores reales (personas).

Estas unidades R-FN son muy apropiadas para modelizar variables con dos valores posibles. Si la variable está en reposo, toma el valor 0, y si está "activada", toma un valor concreto a . La variable se activa al cumplirse alguna condición (modelizada por el hecho de que la excitación de la R-FN supere el umbral), que en la neurona se establece en términos de los pesos.

Este tipo de variables son frecuentes en problemas con información difusa. Por ejemplo, en un grafo difuso cada nodo tiene un grado de pertenencia (a) al grafo, y en un grafo con valoraciones difusas (en nodos o arcos), para muchos problemas un arco pertenece a la solución (variable activada con valor a) o no pertenece.

3.- Una red neuronal artificial difusa.

Con estos PED (R-FN) podemos formar redes con una o más capas que pueden permitir el procesamiento difuso apropiado. En principio no hay restricciones en cuanto a la estructura y topología de la Red.

Tenemos dos problemas iniciales que resolver, respecto del modelo que procesa valores crisp (MacCulloch-Pitts [13]):

- a) La asignación de umbrales a cada unidad de la Red
 - b) El mecanismo de activación, que conlleva la comparación de dos valores difusos.
- a) La asignación de umbrales:

La asignación de Umbrales a los PEDs se realiza mediante conocimiento de expertos sobre el problema concreto, y debe ser previo al entrenamiento de la Red. Esto estaría en sintonía con el aprendizaje por la "experiencia" que parece claro sucede en las neuronas biológicas. Para este proceso, podemos usar otros sistemas como los Mapas Cognitivos Difusos ([10]), o las Redes Combinatorias ([12]).

Es por tanto una asignación previa a la utilización del modelo de red, y los valores asignados dependerán bastante del problema concreto. A cada unidad se le

asignará un umbral propio, aunque puede ser el mismo valor para todas las unidades, en función del problema al que se quiera aplicar.

b) El mecanismo de activación:

La comparación de los valores de excitación y de umbral la realizaremos con un modelo que hemos desarrollado, de RNA clásicas capaces de aprender el criterio de un decisor para comparar números difusos trapezoidales, usando el algoritmo de retropropagación, (ver [4, 5]). Como hemos citado allí, esto se puede extender a números difusos cualesquiera usando representaciones aproximadas basadas en α -cortes.

De todas formas, en un planteamiento teórico, para la activación de las unidades al superarse el umbral difuso, podemos usar alguno de los métodos de comparación de números difusos.

Una Red de PED del tipo estudiado antes, estará formada por una serie de ellos conectados entre sí.

Queremos proponer aquí una estructura de red, totalmente conectada, con conexiones simétricas, e incluso con conexiones autobias o autobucles. No hay por tanto capas en la Red, en el sentido de las redes de propagación hacia adelante. Solamente podremos distinguir en el conjunto de las unidades del modelo, algunos PEDs de Entrada, y otros de Salida, considerando todos los demás como Escondidos u Ocultos. Para aplicaciones concretas, algunas conexiones pueden tener peso asociado nulo, que es equivalente a eliminarlas del modelo. Figura 2.

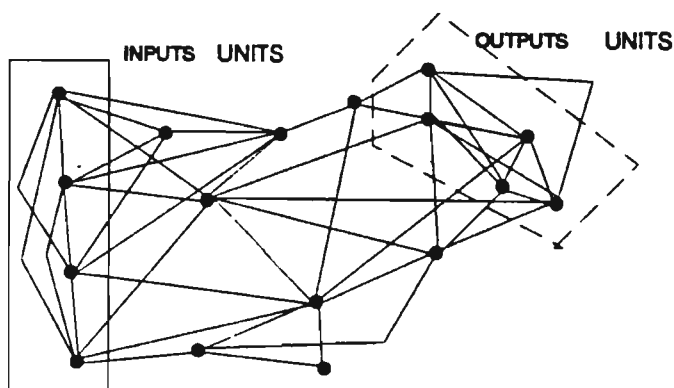


Figura 2. Una Red de R-FN totalmente conectada.

Habrà que tomar la decisión, y será también en función del problema concreto, sobre si la salida de cada unidad será un valor prefijado igual para todas las unidades, o si cada unidad propagará un valor diferente (p.e., su propio umbral).

Para procesar la información subyacente en la red, obtenemos un modelo asociado, sustituyendo los valores difusos por los valores reales que el método de comparación de números difusos del Índice Promedio ([3, 6]) asigna a estos valores. El procesamiento directo con operadores de CLD lo dejamos para próximos trabajos.

Al sustituir los valores difusos por sus índices respectivos, tenemos un sistema en que las unidades son de dos estados. Si no están activadas, la salida (estado) es 0, y al activarse, es el valor del índice asignado a la salida difusa inicial.

Un procedimiento de entrenamiento adecuado para tales redes, puede obtenerse, modificando la Máquina de Boltzmann [2], de forma que la información difusa procesada intervenga en la función de energía del sistema, y así, en el ajuste de los pesos.

Como veremos en el apartado siguiente, podremos utilizar este modelo, tanto para problemas de aprendizaje, como problemas de optimización, de forma similar al uso de la MB en estos tipos de problemas.

4.- Algoritmo modificado de la máquina de Boltzmann.

El algoritmo de la máquina de Boltzmann se aplica directamente a situaciones donde las variables que intervienen son binarias, es decir, solo pueden tomar los valores $\{0, 1\}$. Consideramos una arquitectura como la MB, totalmente conectada, con la única diferencia en los valores de las variables que intervienen en la función de energía de la MB.

Cuando procesamos información difusa en cada unidad, el valor de salida lo podemos tomar como el Índice Promedio (IP) del valor difuso que esta unidad concreta procesa. A esta salida Y la llamaremos valor de activación de la unidad k .

$$Y_k = \begin{cases} 0 & \text{si } s(k) = 0 \\ \text{IP}(\tilde{c}_k) & \text{si } s(k) = 1 \end{cases},$$

donde $s(k)$ indica la activación de la unidad k .

Entonces, los elementos necesarios en la estructura de la MB son :

Vector de activación y configuraciones.

En cada momento de la evolución del sistema, asignamos a cada unidad i un valor $s(i)$, que será un 1 si la unidad está activada, y un 0 si no lo está. Definimos entonces, un vector binario N dimensional $s = (v_1, v_2, \dots, v_N)$, que llamaremos vector de activación, y donde v_i indica la activación de la unidad i -ésima.

Si tenemos N unidades en la red, llamamos configuración global, o simplemente configuración, a un vector binario N -dimensional $c = (g_1, g_2, \dots, g_N)$, donde g_i indica el estado (valor de activación) de la unidad i -ésima, para esa configuración c , y que viene dado por la expresión

$$g_i = \begin{cases} 0 & \text{si } s(i) = 0 \\ \text{IP}(\tilde{Y}_i) & \text{si } s(i) = 1 \end{cases} \quad (1)$$

Cuando distinguimos unidades de entrada y de salida, que en conjunto llamaremos unidades visibles o del entorno, la parte del vector configuración formado por estas unidades, le llamaremos, configuración de entrada, de salida o del entorno (visible) respectivamente. Al vector de las unidades ocultas se le llama configuración interna. Hablamos también, de vectores de activación de entrada, salida o interno.

Cada conexión (i,j) tiene un peso asociado $w(i,j)$ con valores reales. Si el peso es positivo, hay una deseabilidad para que la conexión esté activada, y decimos que es una conexión excitadora. Si el peso es negativo, hablamos de conexiones inhibitorias. En general, los pesos imponen local y globalmente restricciones sobre las configuraciones. Toda configuración mantiene activadas diferentes conexiones, con pesos diferentes.

La función de energía del sistema.

Cada configuración tiene asignada una energía global determinada, dada por la función de energía del sistema $E: \mathcal{C} \rightarrow \mathbb{R}$, que definimos como

$$E(s) = - \sum w_{ij} g_i g_j, \quad (2)$$

donde g_i viene definida por (1).

La función de energía depende siempre de los parámetros libres del sistema, es decir de los pesos de las conexiones, y también de la información procesada por el mismo, vinculada a los estados de las unidades.

El objetivo a alcanzar es obtener configuraciones de mínima energía. Para ello vamos a usar el algoritmo del Annealing Simulado (AS) ([1]) en el modelo.

Transiciones de estados.

Dada una configuración c , podemos llegar mediante una transición a las configuraciones que se diferencian de c en que se ha cambiado la activación (y el valor de activación por tanto) de una unidad r .

Si modificamos el estado de una unidad r ($c \rightarrow c_r$), se produce un cambio en la energía global del sistema ΔE .

Siguiendo con el AS, se aceptan las modificaciones de estados locales (unidades de la red) que conlleven reducción de la función de energía ($\Delta E < 0$). Con el fin de evitar caer en mínimos locales, se pueden aceptar transiciones malas con una cierta probabilidad. Esta probabilidad tiene en cuenta la modificación del algoritmo propuesta.

Distribución de probabilidad de las configuraciones.

De forma similar a la MB, la probabilidad de que el sistema esté en una determinada configuración viene expresada por la distribución de Boltzmann - Gibbs:

$$p(s) = \frac{e^{-(1/T) E(s)}}{Z(T)}, \quad \text{con} \quad Z(T) = \int_{\text{todos estados}} e^{-(1/T) E(s)} ds \quad (3)$$

donde $E(s)$ es la energía que hemos definido en (3), y T es la temperatura ficticia o parámetro de control.

Cuando la red está sin entrenar, los pesos iniciales (generalmente aleatorios) proporcionan una distribución de probabilidad sobre los diferentes estados posibles. El sistema realiza el conjunto de ejemplos de aprendizaje, si al fijar las entradas y salidas sucesivamente para cada ejemplo, las unidades escondidas ajustan sus estados para obtener configuraciones de mínima energía.

El aprendizaje consiste en ajustar los pesos de forma que la distribución de probabilidad del sistema se acerque progresivamente a la distribución subyacente en los ejemplos.

La convergencia del algoritmo modificado sigue siendo asintótica en probabilidad, lo que se comprueba fácilmente siguiendo un camino paralelo a la demostración para la MB, ya que la distribución de probabilidad sobre los estados posibles sigue siendo del tipo de Boltzmann-Gibbs (3).

Adaptación de los pesos durante el aprendizaje.

La adaptación de pesos ha de procurar la disminución de alguna función que mida la diferencia entre $q(s)$ y $p(s)$. Una medida de discrepancia entre los ejemplos (p) y el modelo (q), viene dada por la entropía relativa, o ganancia de información

$$G(p,q) = - \int_V p(v) \lg (q(v) / p(v)) dv$$

La solución óptima corresponde al conjunto de parámetros que dan una distribución del modelo más próxima en medida, a la distribución de los datos. El óptimo minimiza esta entropía relativa.

La minimización de $G(p,q)$, se hace mediante el gradiente descendente, y el aprendizaje mediante el ajuste de pesos en la forma:

$$\Delta w_{ij}(n) = -\eta \left(\frac{\partial G(w)}{\partial w_{ij}} \right)_{w(n)}$$

Siguiendo un camino paralelo a la MB, llegamos a que

$$\frac{\partial G}{\partial w_{ij}} = H_{ij}^+ - H_{ij}^-$$

donde H_{ij}^+ es el valor esperado de la energía local con respecto a la condicional de Boltzmann-Gibbs y a la distribución verdadera de los datos, y H_{ij}^- es el valor esperado de la energía local con respecto a la distribución completa sobre todo el sistema. Es decir,

$$H_{ij}^+ = \langle g_i g_j(v, h) \rangle = \int_v p(v) \left\{ \int_h q(h/v) g_i g_j(v, h) dh \right\} dv$$

$$H_{ij}^- = \langle g_i g_j(v, h) \rangle = \int_h \int_v g_i g_j(v, h) q(v, h) dv dh$$

Entonces

$$\Delta w_{ij} = \eta (H_{ij}^+ - H_{ij}^-) \quad (4)$$

donde η deberá ser pequeña para realizar un buen gradiente descendente.

Generalmente estos valores esperados se aproximan numéricamente por métodos tipo Montecarlo. Al igual que en la MB, usamos el algoritmo Metrópolis :

$$\langle g_i g_j(s) \rangle \approx \frac{1}{R} \sum_{m=1}^R g_i g_j(s_m),$$

donde los estados globales s_m se seleccionan con probabilidad proporcional a la distribución de Boltzmann-Gibbs.

Así, a la hora de ajustar los pesos en el algoritmo, habría que calcular las medias $\langle g_i g_j \rangle$ sobre todas las muestras de entrenamiento. Esta media se mide siguiendo el mismo proceso que en la MB para calcular $\langle s(i) s(j) \rangle$, es decir mediante una estimación numérica.

Como hemos dicho anteriormente, en la práctica no se puede hacer un número de iteraciones suficientes para acercarnos a la convergencia asintótica. Por este motivo, es necesario tomar aproximaciones finitas del algoritmo, especificando claramente la configuración del AS.

En definitiva, en cada iteración hay que reducir la diferencia entre la distribución del sistema, y la distribución determinada por los ejemplos, aplicando el ajuste de pesos indicado en (4).

Para establecer esta diferencia, hacemos :

- Introducimos cada ejemplo en la red (fijando entradas y salidas) y aplicamos el A.S. hasta alcanzar el equilibrio en cada muestra de entrenamiento. Con las configuraciones obtenidas en cada ejemplo, calculamos las medias $\langle g_i g_j \rangle$ sobre el total de muestras de aprendizaje. Esto nos proporciona una estimación de las medias buscadas respecto de la distribución que conforman los ejemplos de aprendizaje.
- Ahora introducimos solo la entrada de cada muestra, y aplicamos de nuevo el A.S. hasta alcanzar el equilibrio. Con las configuraciones obtenidas, calculamos las mismas medias sobre el total de muestras. Esto nos proporciona una estimación de las medias respecto de la distribución que realiza libremente el sistema.

- c) La diferencia entre ambas estimaciones la usamos (multiplicada por la tasa de aprendizaje), para el ajuste correspondiente de los pesos.

En definitiva, el algoritmo sigue siendo convergente asintóticamente en probabilidad. Por tanto, tenemos que seguir utilizando aproximaciones en tiempo finito. La modificación propuesta afecta a las probabilidades de aceptación, y al ajuste de los pesos en el aprendizaje.

De la misma forma que la MB ([1]), este algoritmo modificado puede utilizarse como identificador de modelos (aprendizaje), como optimizador, e incluso como memoria asociativa.

5.- Aplicaciones a problemas de optimización.

En este trabajo nos centramos en la aplicación de este algoritmo modificado de la MB, como optimizador. Teniendo en cuenta la modificación introducida en la función de energía del sistema, el camino iterativo que hay que seguir es paralelo al indicado para la MB (ver en [1]), es decir:

- 1) Plantear, con variables bivaloradas, un problema asociado y equivalente al problema de optimización inicial.
- 2) Al problema con variables bivaloradas, le asociamos una MB con la función de energía modificada. Para ello, asignamos pesos a las conexiones de la red, de forma que el sistema resultante sea consistente con el problema previo.

Ser consistente significa que la función de energía mejora cuando pasamos de configuraciones no factibles a factibles (que corresponden a soluciones factibles del problema previo), y que además preserva el orden con respecto a la función objetivo del problema de optimización asociado. Los pesos de las conexiones se fijan previamente de forma que se cumplan las restricciones del problema de optimización inicial.

De acuerdo con lo anterior, el algoritmo para optimización quedará en forma esquemática:

Algoritmo de optimización.

- 1.- Asignar valores a los pesos w_{ij} , $\forall i,j$, de forma que la Función de Energía

$$E(s) = - \sum_{(i,j)} w_{ij} g_i g_j$$

sea consistente.

- Fijar el valor inicial de Temperatura $T(1) = T_0$
 - Fijar la configuración inicial $s = s_0$, y calcular el valor de la energía.
- 2.- Se aplica el AS hasta el equilibrio, y obtenemos la configuración de mínima energía.
 - 3.- Trasladamos los valores binarios de la configuración obtenida a las variables del problema original (POC).

5.1.- El problema del mayor conjunto independiente en un grafo con valoraciones difusas en los nodos.

Vamos a analizar una extensión del problema de obtener el mayor subconjunto independiente de un grafo $G(V, A)$, consistente en asumir que los nodos del problema tienen asociado un valor difuso.

Con el planteamiento anterior, tratamos de obtener el conjunto independiente de nodos del grafo, con valor máximo, es decir, si $\tilde{v}_i$ es el valor difuso asociado al

nodo i , $i = 1, \dots, p$, hallar el conjunto W tal que

$$\text{maximice } \sum_{i \in W} \tilde{v}_i,$$

con la condición de que $\forall i, j \in W$, el arco $(i, j) \notin A$.

Para aplicar el algoritmo modificado de la MB, tenemos que asociar al problema original, un problema equivalente con variables bivaloradas.

$$\text{Max } \sum \tilde{v}_i \text{ es equivalente a } \text{Max } \sum IP(v_i)$$

ya que el IP es una función ordenadora lineal de números difusos.

Ahora, a cada nodo i le asocio una variable

$$x_i = \begin{cases} IP(\tilde{v}_i) & \text{si } i \in W \\ 0 & \text{en otro caso} \end{cases}$$

y entonces, el problema equivalente lo formulamos como :

$$\begin{aligned} &\text{Max } \sum x_i \\ &\text{sujeto } x_i x_j d_{ij} = 0, \quad i, j = 1, \dots, p \\ &d_{ij} = \begin{cases} 1 & \text{si } (i, j) \in A \\ 0 & \text{si no existe el arco } (i, j). \end{cases} \\ &x_i = \{ 0, IP(\tilde{v}_i) \} \end{aligned} \quad (5)$$

Este es el problema que vamos a resolver con una MB modificada, que construimos de la siguiente forma :

- A cada variable x_i le asocio una unidad elemental, con un valor de activación

$$g_i = \sqrt{x_i}$$

- En la red, incluimos las conexiones correspondientes a los arcos del grafo original, y todas las autoconexiones o bias. Es decir, (i, j) es una conexión de la MB asociada, si y solo si el arco $(i, j) \in A$ en $G(V, A)$, o bien, $i = j$.
- Como es un problema de maximización, considero como función de energía,

$$E = \sum w_{ij} g_i g_j,$$

Necesitamos calcular ahora los pesos de las conexiones, para que la MB asociada, sea consistente con el problema de optimización (5). Para ello, nos interesa que los pesos de las autoconexiones sean positivos y grandes, para intentar activar el mayor número de unidades posibles. Por otro lado, interesa que ninguna otra conexión esté activada en la solución final, ya que esto indica obviamente soluciones no factibles en el problema. Por tanto, los pesos de las conexiones (i, j) con $i \neq j$ han de ser negativos.

Para que una configuración sea factible, es necesario que todas las conexiones inhibitorias (pesos negativos) estén desactivadas. Suponemos que asignamos a todas las autoconexiones un peso $a > 0$, y a todas las conexiones inhibitorias, un peso $-b$ con $b > 0$. Entonces, para que la MB asociada sea consistente, podemos calcular fácilmente que a y b han de verificar la condición

$$b > \frac{a}{h} \quad \text{con} \quad h = \min_j g_j.$$

EJEMPLO .-

Sea el grafo siguiente. En cada nodo se indica un valor difuso trapezoidal (en el intervalo $[0, 10]$ por comodidad), y el valor asignado al mismo por el IP, dividido por 10 (para que el índice esté en $[0, 1]$).

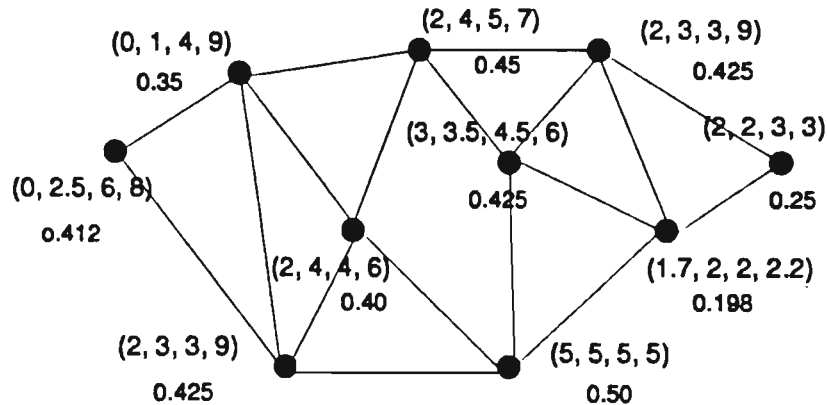


Figura 3.- Grafo con valores difusos trapezoidales en los nodos

Tratamos de obtener el conjunto independiente de nodos, que maximiza la suma de sus valoraciones difusas. Esto equivale a resolver

$$\begin{aligned}
 &\text{Max} \quad \sum x_i \\
 &\text{sujeto} \quad x_i x_j d_{ij} = 0, \quad i, j = 1, \dots, p \\
 &\quad x_i = \{ 0, \text{IP}(v_i) \} \\
 &\quad d_{ij} = \begin{cases} 1 & \text{si } (i, j) \in A \\ 0 & \text{si no existe el arco } (i, j). \end{cases}
 \end{aligned}$$

Para ello, le asociamos la siguiente máquina de Boltzmann modificada.

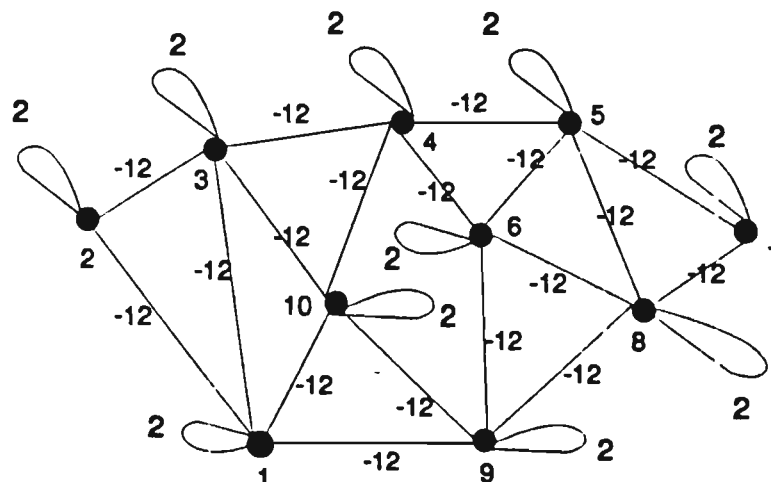


Figura 4.- Esquema de MB modificada asociada al grafo de la fig. 3

Los pesos de esta MB modificada, los hemos asignado, de acuerdo con lo indicado anteriormente.

Teniendo en cuenta que $h = 0.198$, tomamos :
 - pesos autoexcitatorios $a = 2$,
 - pesos inhibitorios $b = -12$.

Hemos implementado con un programa en C, este algoritmo modificado, y la solución obtenida en este ejemplo está formada por los nodos

$$S = \{ 2, 4, 7, 9 \}$$

El valor de la función objetivo para la solución óptima es

$$f(x) = 1.612$$

Si queremos tener el valor difuso del problema original, bastará con sumar los valores difusos de los nodos que proporcionan la solución, y en nuestro caso es:

$$Z = (9, 13.5, 19, 23)$$

5. 2.- Aplicación del algoritmo modificado a otros problemas.

Este algoritmo modificado se puede aplicar también a otros problemas.

- a) Los problemas de optimización booleana con costes difusos y la función objetivo de la forma que sigue, se pueden resolver utilizando la versión optimizadora del modelo presentado en el apartado anterior.

$$f(x) = \sum_{j=1}^n \tilde{c}_i \tilde{c}_j x_i x_j \quad x_i \in \{0, 1\}$$

- b) Se puede aplicar también a algunos problemas de optimización sobre grafos difusos. Aquí, las valoraciones de los nodos que consideramos no son números difusos, sino directamente los grados de pertenencia de cada nodo al grafo difuso.

Referencias

- [1] Aarts, E.H.L.- Korst, J.H.M. (1990). Simulated Annealing and Boltzmann Machine. Wiley & Sons. Chichester.
- [2] Ackley, D.H.- Hinton, G.E.- Sejnowski, T.J. (1985). A Learning Algorithm for Boltzmann Machines. Cognitive Science vol 9. pag 147-169
- [3] Campos, L.M.- González, A. (1989). A subjective approach for ranking fuzzy numbers. Fuzzy Sets and Systems, 29. pg 145-153.
- [4] Cano, J.E. - Delgado, M. - Requena, I. (1991). Using artificial neural networks to aid decision making process. Proc. IWANN'91. Granada (Spain). Springer Verlag. pg 461-469.
- [5] Cano, J.E. - Delgado, M. - Requena, I. (1991). Sistemas neuronales artificiales en la toma de decisiones en ambiente difuso. Actas IV reunión técnica AEPIA. Madrid. pg 30-45.

- [6] González, A. (1987): Métodos subjetivos para la comparación de números difusos. Tesis Doctoral, Universidad de Granada.
- [7] Kaufmann, A. (1975): A introduction to theory of fuzzy subsets. A. Press, N. Y.
- [8] Keller, J.- Hunt, D. (1985). Incorporating fuzzy membership functions into the Perceptron Algorithm. IEEE Transac. PAMI- 7. pag 693-699.
- [9] Kiszka, J.B. - Gupta, M.M. (1989). Fuzzy Logic Model of single neuron. Fuzzy Logic Neural Network. Busefal 1989. Pag 98-108
- [10] Kosko, B.(1986). Fuzzy Cognitive Maps. I. Journal Man-Machine Studies. 24. 65-75
- [11] Kosko, B. (1987). Fuzzy Associative Memories. Fuzzy Expert Systems. Kaendel, A. (editor). Addison-Wesley. MA..
- [12] Machado, R.J. -Rocha, A.F. (1990). The Combinatorial Neural Network: a Connectionist Model for Knowledge Based Systems. En [1]. pp. 9-11
- [13] McCulloch, W. - Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. 7. pag 115-133.
- [14] O'Hagan, M. (1990). A Fuzzy Neuron based on Maximun Entropy O. W. A.. En Proc. 3th Int. C. Infor. Proc. Manag. Uncert. in Knowledge-based Systems. Paris 1990
- [15] Requena, I. (1991). Redes Neuronales Artificiales: Algunas Ideas Básicas. En Algunas Cuestiones sobre Tratamiento de Información en Incertidumbre (D.C.C.I.A. editor). Publicaciones de la Universidad de Granada. pg 110-130.
- [16] Requena, I.- Delgado, M. (1992). R-FN: A model of fuzzy neuron. Proc. 2nd Int. C. FL & NN (IIZUKA'92). pg 793-796.
- [17] Rumelhart, D.- Hinton, G.- Williams, R..(1986). Learning internal representations by backpropagation errors. Nature, 323. pg 533-536
- [18] Shiue, L.C. -Grondin, R.O. (1987). Proc First IEEE ICNN. VOL. II. pg 299-304.
- [19] Taber, W.R. (1989). The Evolving Neuron and Fuzzy Logic. En Verdegay, J.L.- Delgado, M. (Edit.). (1989). Aproximate Reasoning tools for Artificial Intelligence. pg 254-271. Verlag Tuv Rheinland.
- [20] Wasserman, P.D.(1989). Neural Computing: Theory and practice. Van Nost. Rh.-N.Y.
- [21] Yager, R. (1988). On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisions. IEEE Trans. On S.M.C.. 8.1. Pag 183-190.
- [22] Yamakawa, T - Tomoda, S. (1989). A Fuzzy Neuron and its Application to Pattern Recognition. Proc. Third IFSA Congress. Seattle 1989. Pag 30-38.
- [23] Zadeh, L.A. (1965). Fuzzy Sets. Information and Control, 8. pg. 338-353.
- [24] Zadeh, L.A. (1975): The concept of a linguistic variable and its application to approximate reasoning. Information Science, 8, 199 (I); 8, 301 (II); 9,43 (III).

Perceptrón borroso

Autores: Julio Rives, Rafael Pereira y J. Antonio Díaz

INISEL
Sistemas de Mando y Control y Tráfico Aéreo
Apdo. de Correos 294
Polígono Industrial 1 de San Fernando de Henares
28850 MADRID
Teléfono n.: 34 1 6756300
Fax n.: 34 1 6561773
e-mail: jrives@inisel.es
e-mail: rpereira@inisel.es

Resumen: En este artículo se revisa el paradigma de las redes neuronales desde la perspectiva de la lógica "fuzzy". Se define una nueva familia de operadores de agregación triangulares denominados operadores de agregación sigmoidales. Se redefine el modelo de neurona y se describe una metodología para diseño de redes multicapa que, aparte de la capa de entrada, fuerza el uso cuando más de 3 capas, una que realiza la operación de unión, otra la media generalizada y otra la intersección. Es más, se demuestra que esta arquitectura puede aprender teóricamente cualquier función y, más aun, que la propagación clásica por adición promediada de actividades es un caso particular de nuestro modelo. Finalmente, se deduce el algoritmo de *retropropagación borrosa*.

Palabras Clave: Conorma triangular, media generalizada, norma triangular, agregaciones sigmoidales, neurona borrosa, red multicapa borrosa, retropropagación borrosa.

1. Introducción

No son pocos los investigadores de inteligencia artificial que están trabajando en los campos de la lógica borrosa y de las redes neuronales. El mercado de ambos está en plena ebullición con una ligera ventaja del primero, sobre todo debido a que la lógica borrosa actualmente está preparada para el diseño, desarrollo y explotación. Sin embargo, lejos de rivalizar, estos dos mundos deberían irremediablemente coexistir en un futuro cercano y alcanzar un mayor grado de integración. La prometedora sinergia entre ambas tecnologías presenta indicios de estar materializándose ya en Japón, Europa y Estados Unidos a través de proyectos concertados, congresos y artículos especializados [Caudill 90; Rives & Pereira 92].

En este artículo vamos a revisar el paradigma de las redes neuronales, aproximándonos al problema desde una perspectiva borrosa. El nuevo planteamiento incorpora plenamente a cada neurona de la red agregaciones triangulares como medidas borrosas que combinan las actividades de sus entradas. El conocimiento se disemina a través de la red de una forma genuinamente difusa. Redefinimos el modelo de neurona y describimos las bases de una metodología para diseño de redes multicapa borrosas. Demostramos cómo la arquitectura propuesta puede aprender teóricamente cualquier función y cómo la propagación clásica por adición promediada de actividades es un caso particular de nuestro modelo.

2. Operadores de agregación

Hay 3 tipos de operadores de agregación básicos (definidos entre $[0,1]^n$ y $[0,1]$) en orden decreciente de magnitud [Bonissone & Decker 86]: los de disyunción, los operadores intermedios y los de conjunción. Los operadores de agregación más utilizados para la disyunción o unión y para la conjunción o intersección, son las conormas y normas triangulares respectivamente. Cada operador de unión tiene su dual en el sentido de DeMorgan en un operador de intersección. Las agregaciones más utilizadas para las operaciones intermedias son las medias generalizadas.

Estas generalizaciones juegan un papel vital en el manejo de la incertidumbre en la vertiente cognoscitiva de la inteligencia artificial, concretamente en los sistemas expertos. Nuestra intención es convencer al lector de que también pueden desempeñar un papel clave en la vertiente conductista de la inteligencia artificial, es decir, acoplarse a los sistemas conexionistas de modo que se obtengan artefactos productivos, sistemas difusos de procesamiento masivo paralelo.

En general, los operadores de agregación cumplen los axiomas de monotonía creciente, conmutatividad y asociatividad (tabla 1). Las conormas triangulares, medias generalizadas y normas triangulares se particularizan mediante la definición de un axioma más para condiciones límite (tabla 2).

a- Monotonía creciente	$a_1 < b_1 \dots a_n < b_n \Rightarrow O(a_1, \dots, a_n) < O(b_1, \dots, b_n)$
b- Conmutatividad	$O(x_1, \dots, x_n) = O(x_n, \dots, x_1)$
c- Asociatividad	$O(x_1, \dots, x_{n-1}, x_n) = O(O(x_1, \dots, x_{n-1}), x_n)$

Tabla 1. Axiomas que cumplen todos los operadores de agregación triangulares.

	conormas triangulares	medias generalizadas	normas triangulares
d- Condiciones límite	$O(0, \dots, 0, x, 0, \dots, 0) = x$ $O(1, \dots, 1) = 1$	$O(x, \dots, x, \dots, x) = x$	$O(1, \dots, 1, x, 1, \dots, 1) = x$ $O(0, \dots, 0) = 0$

Tabla 2. El axioma de condiciones límite particulariza los operadores de agregación triangulares.

Una de las más populares familias parametrizadas de conormas y normas, en base al operador de negación $N(x) = 1 - x$, es la de Yager. En ella, la más optimista de las conormas se obtiene para $p = 0^+$, mientras que la más conservadora, la menor, es el "máximo" y se obtiene para $p = \infty$. De las normas de Yager, la más optimista, la mayor, es el "mínimo" y se obtiene para $r = \infty$, mientras que la más conservadora se obtiene para $r = 0^+$. Por otra parte, de los operadores intermedios de agregación, una familia que cubre todo el espectro de borrosidad entre el "máximo" y el "mínimo" es la de las medias generalizadas (ver figuras 1 y 2).

La conorma $\min(1, x_1 + \dots + x_n)$ ($p = 1$) es una de las conormas más altas y menos selectiva con sus entradas: contempla el caso mejor, se fija más en el conjunto que en las individualidades y, en comparación con el máximo, generaliza más a prisa. Por el contrario, su dual $\max(0, x_1 + \dots + x_n - 1)$ ($r = 1$), es una de las normas más bajas y estrictas: apropiada para agregar conjuntamente valores mínimos o incuestionables de evidencia, se fija más en las individualidades que en el conjunto y, en comparación con el mínimo, generaliza más despacio.

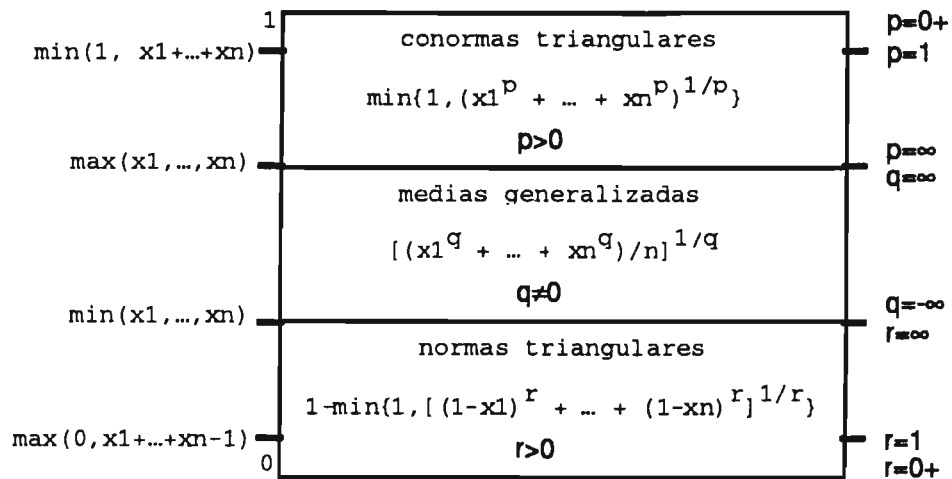


Figura 1. Conormas y normas parametrizadas de la familia de Yager y medias generalizadas.

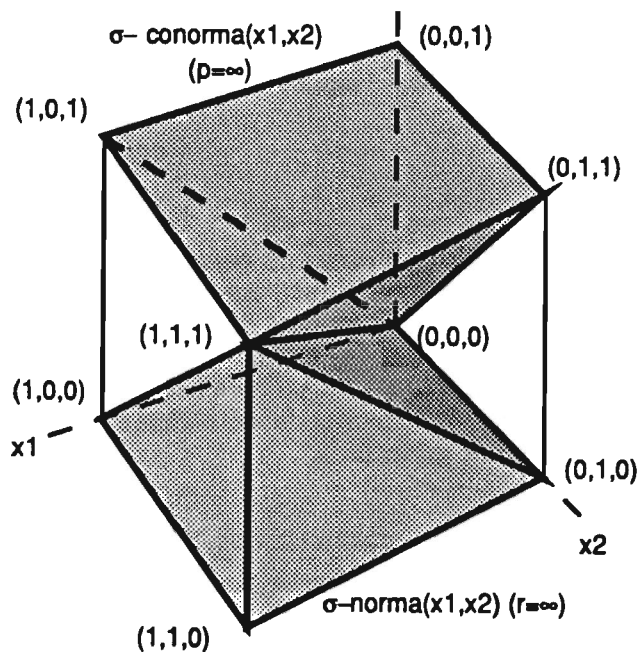


Figura 2. Conorma para $p=\infty$ (la menor) y norma para $r=\infty$ (la mayor) con 2 entradas. El espacio restante del cubo unitario es cubierto por las medias generalizadas.

3. Neurona tradicional

Un nodo tradicional de una red conexionista puede verse como un procesador matemático con n entradas y una sola salida. Cada entrada I_i tiene asociado un peso W_i que indica el factor de multiplicación que recibirá la "excitación" que llegue por ella, y la salida O puede verse como una función, generalmente no lineal, de la "excitación total". Consiguientemente, la salida de una neurona tradicional suele hallarse según la fórmula $O=f(\sum W_i \cdot I_i)$, donde f es la llamada función de transferencia de la neurona. Su derivabilidad es deseable por lo que concierne al aprendizaje, según veremos más adelante (ver figura 3).

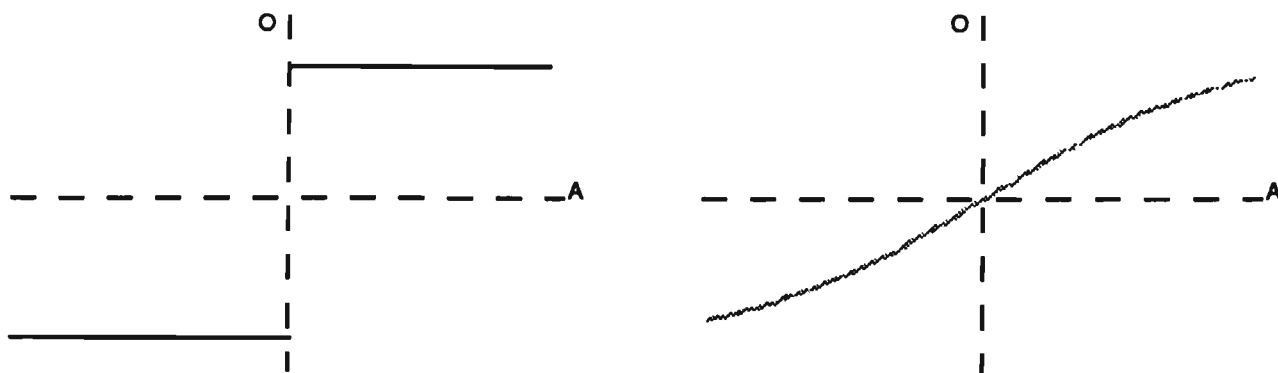


Figura 3. El escalón y la tanh como ejemplos de funciones de transferencia no derivables y derivables.

Llama la atención cómo, hasta la fecha, los investigadores de redes neuronales han analizado el efecto de modificar las I_i , las W_i y f , pero no la expresión de la excitación total. La excitación total según la fórmula $A = \sum W_i * I_i$ compuesta con una función de saturación a la salida de valor máximo 1, es un caso muy particular de agregación que, concretamente, corresponde a la operación realizada por la conorma de Yager $\min(1, (\sum W_i^p * I_i^p)^{1/p})$ para $p=1$. Así pues, la motivación fundamental de nuestro estudio consiste en preguntarnos ¿por qué no podemos usar la aritmética borrosa como función de agregación de una neurona?

4. Neurona borrosa

Es obviamente deseable que los axiomas de conmutatividad y asociatividad que cumplen los operadores de agregación triangulares los verifiquen también las entradas de una neurona sea cual sea el método de propagar la actividad de la red. El axioma de monotonía creciente es útil para que la relación entre la entrada y la salida sea biunívoca. En lo que respecta a los axiomas de condiciones límite, restringen tanto los pesos como las entradas y la salida a oscilar en el rango $[0,1]$, aunque teóricamente podrían definirse en el rango $[-\infty, \infty]$. Esto puede entenderse como una operación de modelado o normalización, y no supone pérdida de generalidad.

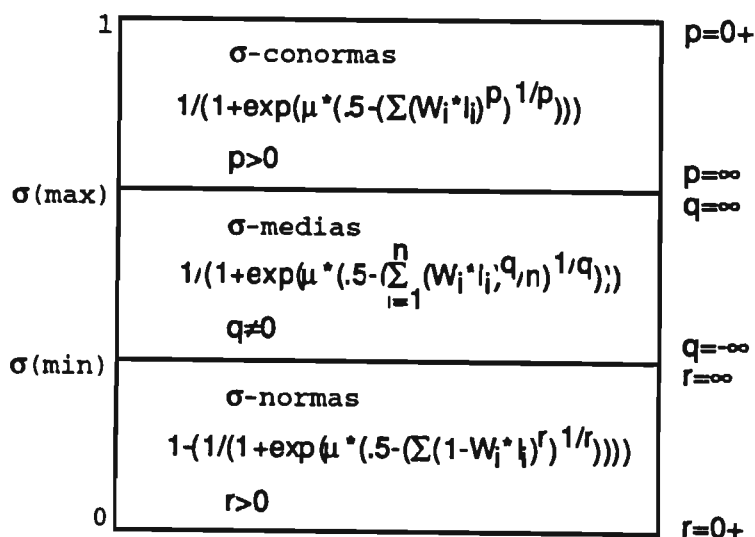


Figura 4. Conormas, medias y normas sigmoideas parametrizadas.

Fijémonos en las conormas de Yager. Sustituuyamos la función no derivable "mínimo", que satura la salida, por la función derivable "sigmoide" $\sigma(A_i)=1/(1+e^{\mu(0.5-A_i)})$ que tiene la misma finalidad, donde μ es un factor de ganancia y $A_i=(\sum(W_i*I_i)^p)^{1/p}$. Se genera así una nueva familia de conormas triangulares, aunque con distintas condiciones límite, que denominaremos "sigmoidales". Las medias sigmoidales se definen aplicando la función sigmoide a las medias generalizadas. Las normas sigmoidales se generan complementando a 1 las entradas $(W_i*I_i)^p$ y la salida final (ver figura 4). Los operadores de agregación sigmoidales cumplen los axiomas de conmutatividad, asociatividad, monotonía y verifican las condiciones límite que muestra la tabla 3.

	σ -conormas	σ -medias	σ -normas
d- Condiciones límite	$O(0,\dots,0,x,0,\dots,0) = \sigma(x)$ $O(1,\dots,1) = \sigma(1)$	$O(x,\dots,x,\dots,x) = \sigma(x)$	$O(1,\dots,1,x,1,\dots,1) = \sigma(x)$ $O(0,\dots,0) = \sigma(0)$

Tabla 3. El axioma de condiciones límite para los operadores de agregación sigmoidales.

Para que las condiciones límite se parezcan lo más posible a las de las agregaciones triangulares, una buena elección para el factor de ganancia es $\mu \geq 4.4$, ya que produce $O(0) \leq 0.1$ y $O(1) \geq 0.9$ (ver figuras 5 y 6).

Nótese que las redes clásicas son un caso particular del modelo propuesto cuando estamos agregando disyuntivamente las entradas con $O(A_i)=1/(1+e^{\mu(0.5-A_i)})$ y $p=1$ (ver curva gruesa de la figura 6). A medida que aumenta p , la neurona suma las entradas cada vez con más atención puesta en las entradas individuales, hasta que en $p=\infty$ sólo deja pasar el máximo, produciendo $\sigma(\max)$. Por el contrario, según disminuimos p , las diferencias entre las entradas se difuminan más y más, hasta que en $p=0^+$ la neurona produce $\sigma(1)$ a su salida para toda entrada.

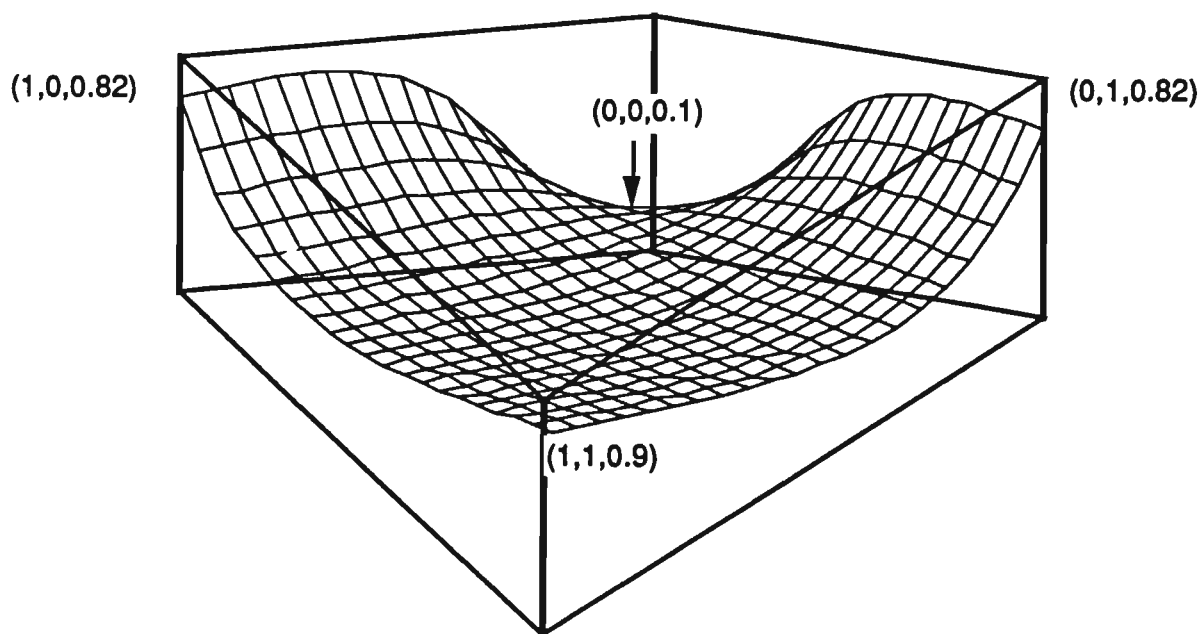


Figura 5. Media sigmoideal para 2 entradas con $\mu=4.4$ y $q=4$.

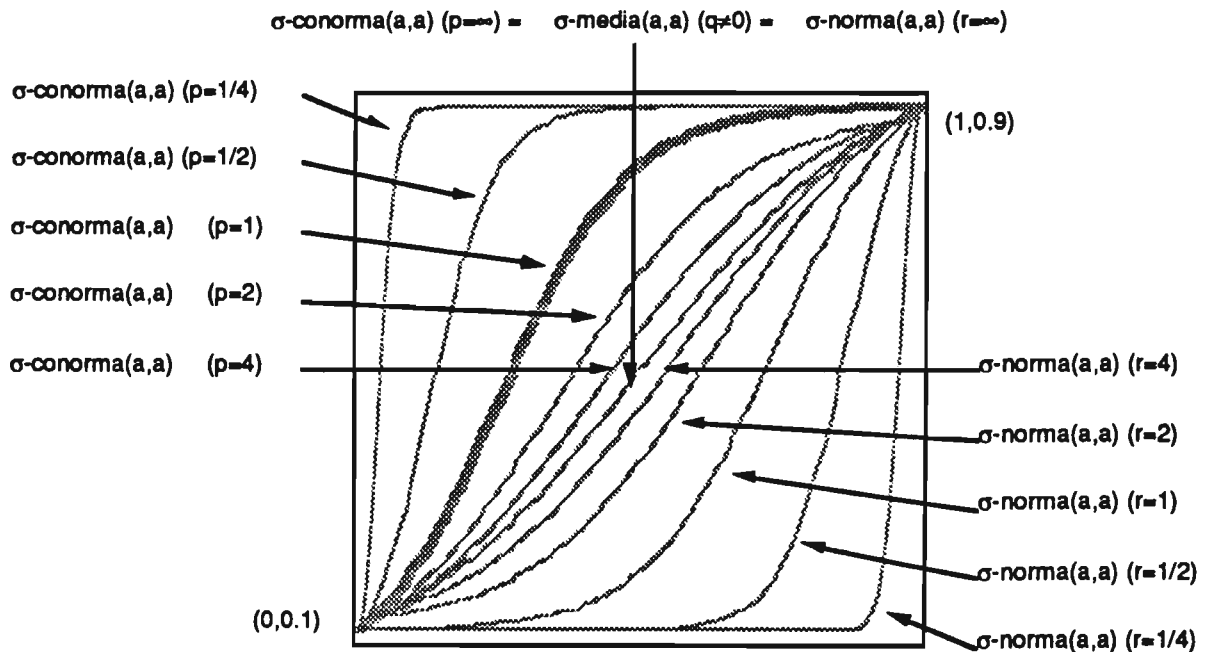


Figura 6. Corte del cubo unitario de conormas, medias y normas sigmoideas con el plano $x_1=x_2$.

5. Red multicapa

Desde la perspectiva tradicional, lo que hemos hecho es transformar la neurona de manera que, fijada la función de transferencia en $O(x)=1/(1+e^{\mu(0.5-x)})$, variando p en las agregaciones disyuntivas, variando q en las agregaciones intermedias y variando r en las agregaciones conjuntivas podemos barrer todas y cada una de las alinealidades en el hipercubo unitario. Como el problema planteado en el hipercubo $[(0,\dots,0),(1,\dots,1)]$ es isomórfico al del hipercubo $[(-\infty,\dots,-\infty),(\infty,\dots,\infty)]$, podemos afirmar que 1 capa sola de neuronas borrosas tiene capacidad para resolver 1 no linealidad simple.

Cuando, por el contrario, debemos diseñar una red con una funcionalidad cualquiera, necesitaremos considerar cualquier posible tipo de combinación no lineal. Así, un modelo de red multicapa borrosa genérico se compondrá de 3 capas (además de la entrada) colocadas en cualquier orden, una que realiza uniones, otra medias, y otra intersecciones. Las capas están interconectadas en un sólo sentido desde la capa de entrada hacia la de salida. En definitiva, lo que estamos proponiendo aquí es un procedimiento tal que, sintonizando adecuadamente los parámetros p , q y r en sus capas respectivas, podemos resolver teóricamente cualquier función, lineal o no lineal (ver figura 7).

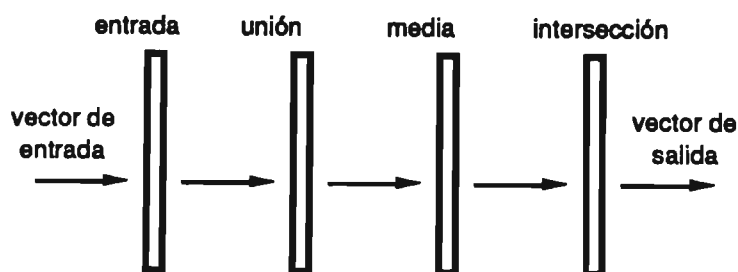


Figura 7. Un modelo de perceptrón borroso multicapa.

Una consecuencia crucial de nuestro análisis es que la red multicapa tradicional con 1 sola capa oculta no tiene capacidad para simular cualquier función no lineal. Ahora bien, una red multicapa tradicional con 2 capas ocultas, más la de salida, puede cubrir todas las posibles funciones no lineales variando aleatoriamente la función de transferencia. Sin embargo, este método es anárquico y habitualmente vano. ¿Cómo podemos ir variando metódicamente la función de transferencia sin que nos dejemos fuera ninguna agregación no lineal?, ¿con qué criterio cambiamos la función de transferencia?...

Se presenta a continuación el algoritmo de aprendizaje correspondiente al modelo propuesto, la *retropropagación borrosa* ("fuzzy gradient back propagation" o *FGBP*), que engloba la retropropagación convencional.

6. Retropropagación borrosa

El algoritmo de aprendizaje para redes multicapa borrosas que vamos a desarrollar, la retropropagación borrosa, es análogo al clásico de retropropagación del gradiente (GBP) para una red multicapa tradicional, y además lo generaliza. Recordemos al lector que para que el FGBP funcione, tanto los pesos como el vector de entrada y el de salida se inscriben en el rango $[0,1]$. Todos los pesos entre neuronas pueden cambiar durante el aprendizaje, de modo que la red va a ser capaz de extraer las regularidades encontradas en los ejemplos.

El algoritmo de retropropagación borrosa se deduce de la siguiente manera:

- el valor de la salida para cualquier neurona h de la red es x_h según aparece en la tabla 4, en donde v es un parámetro genérico, m es el número de elementos de la capa de la media generalizada, y donde W_{hg} es el peso de conexión desde la neurona g a la h (ver figura 8).

capa	$x_h = \sigma(A_h)$	A_h	Φ_h
unión	$1/(1+e^{\mu(0.5-A_u)})$	$(\Phi_u)^{1/v}$	$\sum_e (W_{ue} * x_e)^v$
media	$1/(1+e^{\mu(0.5-A_m)})$	$(\Phi_m)^{1/v}$	$(\sum_u (W_{mu} * x_u)^v)/m$
intersección	$1/(1+e^{\mu(0.5-A_i)})$	$(\Phi_i)^{1/v}$	$\sum_m (1-W_{im} * x_m)^v$

Tabla 4. Salida x_h de una neurona cualquiera del perceptrón borroso.

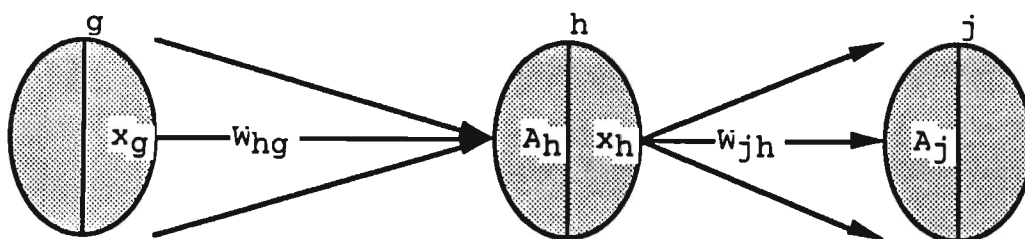


Figura 8. Factores que intervienen en la modificación de los pesos W_{hg} con cada nuevo ejemplo.

Lo que queremos es que la red asocie para cada vector de entrada un vector de salida deseado "Y". Pero cuando la red está desentrenada, cuando recibe una entrada la propaga a través de sus capas obteniendo una salida calculada "X", con i componentes. Lo que tenemos que hacer es calcular el costo, C_k , como una distancia (i.e., el error cuadrático medio) entre la salida calculada y la salida deseada para cada ejemplo k , donde $k=1, \dots, n$ y n es el número total de ejemplos que van a entrenar a la red:

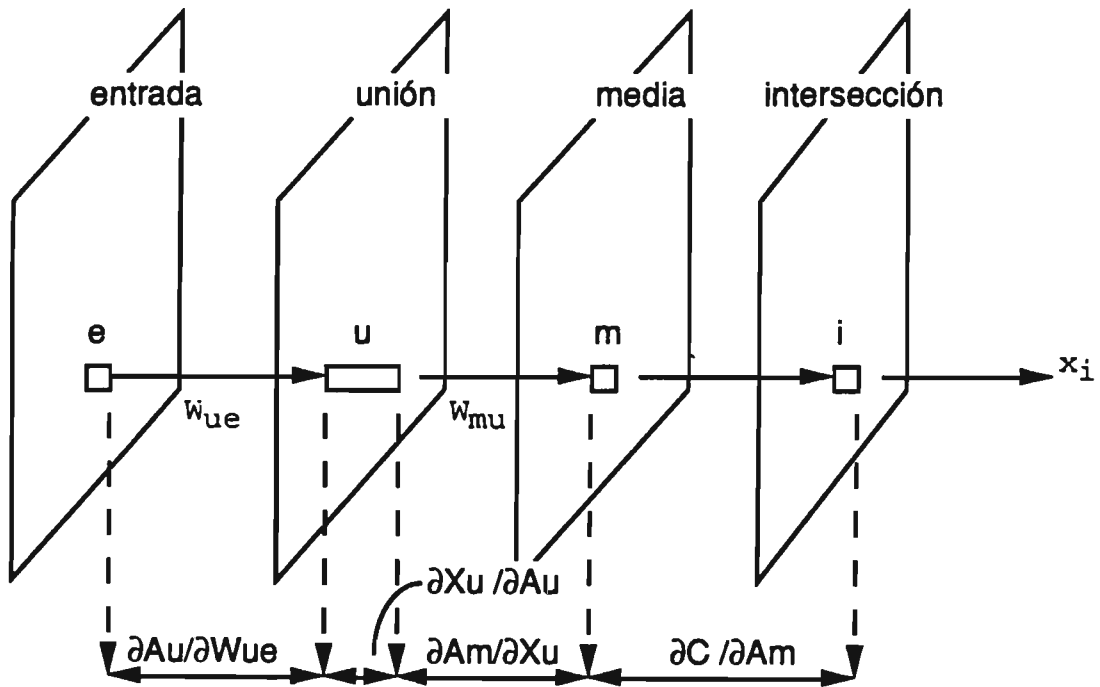


Figura 9. Descomposición de la red multicapa en zonas que afectan a la variación de los pesos W_{ue} .

- $C_k = \|X - Y\| = ((\sum_i (x_i - y_i)^2)/i)^{1/2}$, en donde el costo está normalizado entre 0 y 1.

Minimizando respecto de la variación de los pesos usando la regla de Widrow y Hoff obtenemos:

- $W_{hg}(k+1) = W_{hg}(k) + \beta(k+1) * \partial C_k / \partial W_{hg}$,

donde β es un coeficiente de aprendizaje en cada ejemplo. En otras palabras, se cambian los pesos locales de acuerdo a la talla y sentido del gradiente de la superficie de error. Ahora bien (ver figura 9),

- $\partial C_k / \partial W_{hg} = \partial C_k / \partial A_h * \partial A_h / \partial W_{hg}$.

Para una neurona de la capa de salida o intersección ($h=i$ y $g=m$)

- $\partial C_k / \partial A_i = \pi_i = [1/(i * C_k)] * (x_i - y_i) * \partial x_i / \partial A_i = [1/(i * C_k)] * (x_i - y_i) * \sigma'(A_i)$,
- $\partial A_i / \partial W_{im} = \partial[(\Phi_i)^{1/p}] / \partial W_{im} = (A_i / \Phi_i) * (W_{im})^{v-1} * (-1) * x_m^v$;

para una neurona de la capa de la media generalizada ($h=m$ y $g=u$)

- $\partial C_k / \partial A_m = \pi_m = [\sum_i (\partial C_k / \partial A_i * \partial A_i / \partial x_m)] * \partial x_m / \partial A_m = [\sum_i (\pi_i * (A_i / \Phi_i) * (W_{im})^v)] * x_m^{v-1} * \sigma'(A_m)$,
- $\partial A_m / \partial W_{mu} = \partial[(\Phi_m)^{1/p}] / \partial W_{mu} = (A_m / \Phi_m) * (W_{mu})^{v-1} * (1/m) * x_u^v$;

y para una neurona de una capa de entrada o unión ($h=u$ y $g=e$)

- $\partial C_k / \partial A_u = [\sum_m (\partial C_k / \partial A_m * \partial A_m / \partial x_u)] * \partial x_u / \partial A_u = [\sum_m (\pi_m * (A_m / \Phi_m) * (W_{mu})^v)] * x_u^{v-1} * \sigma'(A_u)$,
- $\partial A_u / \partial W_{ue} = \partial[(\Phi_u)^{1/p}] / \partial W_{ue} = (A_u / \Phi_u) * (W_{ue})^{v-1} * x_e^v$.

- el valor de la derivada de la salida se calcula según la tabla 5.

capa	$\partial x_h / \partial A_h$
unión	$\mu * \sigma(A_h) * (\sigma(A_h) - 1)$
media	$\mu * \sigma(A_h) * (\sigma(A_h) - 1)$
intersección	$\mu * \sigma(A_h) * (1 - \sigma(A_h))$

Tabla 5. Derivadas $\partial x_h / \partial A_h$ de la función de transferencia respecto de la salida para cada capa.

En resumen, para cada ejemplo y empezando por las últimas capas se incrementarán los pesos de acuerdo a la tabla 6, añadiendo al valor actual del peso el producto de (ver figuras 8 y 9):

- un factor de aprendizaje dependiente de la iteración, $\beta(k)$;
- por un factor dependiente del error $[1/(i * C_k)] * (x_i - y_i)$ en la capa de salida, o dependiente del error, actividad y pesos de la capa siguiente $[\sum_j (\pi_j * (A_j / \Phi_j) * (W_{jh})^v)] * x_h^{v-1}$;
- por la derivada de su función de transferencia, $\mu * \sigma(A_h) * (\sigma(A_h) - 1)$;
- por un factor dependiente de la excitación total de la neurona destino de la conexión, A_h / Φ_h ;
- por un factor dependiente del peso de la conexión bajo estudio, $(W_{hg})^{v-1}$; y
- por la actividad de la neurona origen de la conexión, x_g^v .

actualización de los pesos ΔW
$\Delta W_{im}(k) = \beta(k) * [1/(i * C_k)] * (x_i - y_i) * \mu * \sigma(A_i) * (1 - \sigma(A_i)) * (A_i / \Phi_i) * (W_{im})^{v-1} * x_m^v$
$\Delta W_{mu}(k) = \beta(k) * [\sum_i (\pi_i * (A_i / \Phi_i) * (W_{im})^v)] * x_m^{v-1} * \mu * \sigma(A_m) * (\sigma(A_m) - 1) * (A_m / \Phi_m) * (W_{mu})^{v-1} * (1/m) * x_u^v$
$\Delta W_{ue}(k) = \beta(k) * [\sum_m (\pi_m * (A_m / \Phi_m) * (W_{mu})^v)] * x_u^{v-1} * \mu * \sigma(A_u) * (\sigma(A_u) - 1) * (A_u / \Phi_u) * (W_{ue})^{v-1} * x_e^v$

Tabla 6. Actualización de los pesos locales por cada ejemplo.

Obsérvese que si $v \rightarrow \infty$, entonces se tendrá considerablemente en cuenta el peso de la conexión, mientras que si $v \rightarrow 0+$ el incremento es inversamente proporcional al peso actual.

Nótese también que cuando $v \neq 1$, la excitación total de entrada no es lineal, de modo que podría ser perfectamente válido aplicar una función de transferencia lineal, como por ejemplo la $O(x)=x$. Así el incremento de computación que supone hacer la agregación borrosa queda contrarrestado por la desaparición de los factores " $\mu * \sigma(A_h) * (\sigma(A_h) - 1)$ " en las fórmulas de la tabla 6.

Independientemente de si estamos calculando los nuevos pesos de una conexión a una capa de salida o a una oculta, puede verse que la actividad del nodo origen de la conexión no es un parámetro tan determinante como en el GBP [Fogelman & Mejia 88], pues la variación de un peso también es función de su valor actual, de la excitación total de la neurona destino y de la salida de ésta, factores que en la retropropagación clásica no aparecían. Desaparece de este modo la necesidad de empujar la conexión ("jogging"), inicializarla o añadir un pequeño desplazamiento de la derivada de la sigmoide cuando se produce saturación, esto es, cuando deja de aprender por haber entrado en un mínimo local.

Un último punto a destacar de la retropropagación borrosa es que puede aplicarse igualmente a las versiones modificadas de la retropropagación convencional [NeuralWare Inc. 91], como la actualización acumulativa de pesos, la retropropagación con "momentum", la retropropagación "rápida", o la retropropagación con cambios que afectan a la función de error.

7. Conclusiones

Mientras que la aproximación tradicional forzaba el uso de sólo 2 capas porque, al estar basada en modelos binarios, bastaba una capa para realizar el OR y otra para el AND, mediante un planteamiento "fuzzy" nosotros hemos descrito una metodología para diseño de redes multicapa que fuerza el uso de 3 capas (una por agregación fundamental) más la de entrada. Asimismo, hemos redefinido el modelo de neurona arguyendo que lo importante es realizar en cada nodo una función no lineal, por ejemplo una agregación no lineal, en lugar de la composición clásica de una agregación lineal con una función no lineal.

Además, como organización conexionista y como sistema difuso, el perceptrón borroso tiene las siguientes características:

- no es un modelo de caja negra puro, puesto que se sabe lo que hace en cada capa de procesamiento;
- puede aprender teóricamente cualquier función que transforma una entrada en una salida si ésta es conmutativa y asociativa respecto de sus entradas;
- marca las pautas para generalizar de forma análoga otras arquitecturas de red con sus correspondientes algoritmos de aprendizaje.
- procesa la información de forma absolutamente distribuida;
- es muy tolerante al fallo y extremadamente insensible al ruido;

Las líneas de trabajo actual y futuro presentan 3 vertientes:

- analizar cómo nuestro modelo repercute en el número de neuronas de una capa;
- estudiar cómo nuestro modelo afecta a la base de entrenamiento y viceversa;
- realizar aplicaciones prácticas que corroboren nuestro modelo teórico.

8. Referencias

[Bonissone & Decker 86] P. P. Bonissone & K. S. Decker. Selecting Uncertainty and Granularity: An Experiment in Trading-off Precision and Complexity. "Uncertainty in AI". Kanal & Lemmer (eds); pp. 217-247. Elsevier Science Publishers, North Holland. 1986.

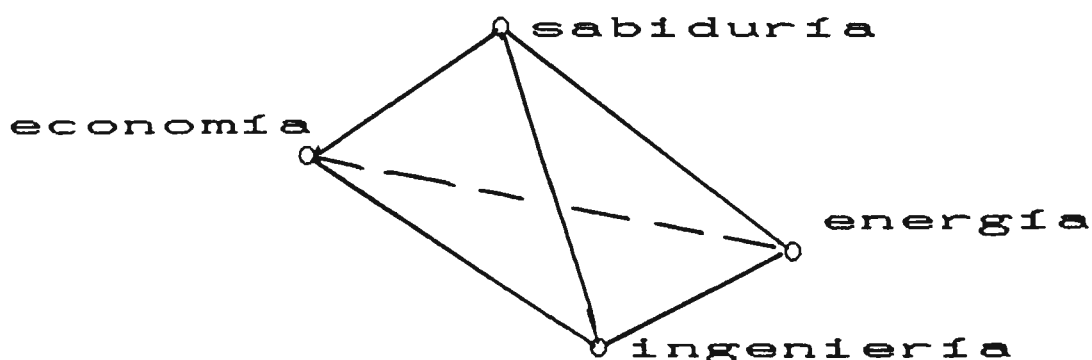
[Caudill 90] Caudill, Maureen. Fuzzy Cognitive Maps. AI Expert. April 1990.

[Fogelman & Mejia 88] Fogelman, F. & Mejia, C. Redes Conexionistas para el Aprendizaje. Third International Symposium on Knowledge Engineering, October 17-18, Madrid, Spain. 1988.

[NeuralWare Inc. 91] NeuralWare Inc. "Neural Computing". NeuralWare Incorporation, Technical Publications Group. 1991.

[Rives & Pereira 92] Rives, J. & Pereira, R. Very fuzzy cognitive Maps. Second International Conference in Fuzzy Logic & Neural Networks, Iizuka 92, Japan.

Culturergia:
un modelo económico difuso



Autor

Manuel Luna de Toledo
Profesor de proyectos de energía
Departamento de ingeniería de la producción
Escuela Técnica Superior de Ingenieros Industriales
Universidad Politécnica de Madrid

Resumen:

El mercado, de cualquier tipo que sea, es incapaz de resolver los problemas sociales para los miembros de cualquier colectividad porque usa métodos unificadores. Su principio del coste no es para igualar entes heterogéneos, sino para lograr que actividad culmine en la máxima facilidad de acceso, porque se usan la menor cantidad de monedas. Nada más.

La aparición de la lógica difusa será el gran trampolín que consiga proporcionarle, verdaderamente, la gran herramienta de la libertad de acción y de la potencialidad de servicio. Hay que investigar estos campos.

Este trabajo es el resumen de quince años de actividad realizada en solitario. Hay varios amplios textos en preparación. Y esta es la segunda vez que se presenta en un Congreso científico. Todo lo que se escribe es original, no encontrarán nada parecido en lugar alguno.

Palabra clave: Culturergia

EL sistema

Busca las dialécticas conceptuales mediante las sinergias interactivas de un metodología que siempre es pentavalente, pues se realiza en modelos tetraédricos cuyos vértices se ocupan, uno a uno, por conceptos interactivos para definir al general del tetraedro. Cuatro más una, cinco sinergias.

Parte de la distinción entre crecimiento, desarrollo, satisfacción e integración, que serán explicados mediante el ejemplo de la energía, rama de la ingeniería de la que sé un un poquito.

El nombre

Le he llamado culturergia o generación de acción contenida en la cultura, pues realmente busca demostrar que todas las acciones de relación con el ámbito son económicas y así, de verdad, plantear que nuestro único problema es buscar los análisis por consecuencias para reconvertirlos en mercado. En un mercado 100 % clásico, pues la competencia es necesaria.

La cultura es para la antropología la relación del hombre con su ámbito y desde ahí quiero conseguir que todas las relaciones culturales sean, a la vez, económicas y que sirvan para crear dinero, pues el dinero es la mejor forma de materializar la riqueza y transmitirla a donde se desee.

Decidí buscar métodos de análisis para la economía de mercado desde la acción, por ella misma, independientemente de los costes, buscando sus consecuencias para llegar a generar sinergias interactivas que justificasen las inversiones por sí mismas o por sus consecuencias de acción. Creo que he logrado esquemas dialécticos publicables y que ya deberían ser discutidos, rechazados o completados.

El producto y sus distinciones

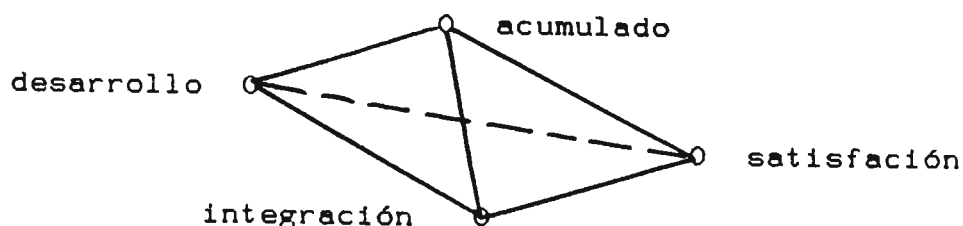
El producto, PIB, se define como suma de valores de mercado de la acción de un país entre dos fechas dadas. Tiene dos formas clásicas, el crecimiento y el desarrollo, de muy difícil distinción para muchos, casi todos, que los equiparan. Les añadí la satisfacción y la integración.

El crecimiento, o aumento monetario del producto, es indispensable porque mide las consecuencias que el aumento de productividad han provocado, por ello el gradiente del ratio de crecimiento ha de ser positivo y, como mínimo igual, al gradiente del ratio de productividad o sino la economía ha perdido actividad. La economía no dispone de otras rápidas y fáciles maneras para saber la actividad real en un período.

Existen dos productos, producto total o sea todo lo realizado y producto de desarrollo, o sea del total, lo que se usó, la diferencia entre ambos son los deshechos, contaminación, basuras, sobrantes, etc. A la sociedad sólo le importa el producto de desarrollo; pues es lo que usa.

Al producto total, PIB, le llamaré acumulado, al producto realmente usado, le llamaré desarrollo. La economía clásica no sabe distinguir entre ambos y no puede medir la diferencia, pues los métodos monetarios son poco apropiados. El desarrollo siempre es menor que el acumulado, su diferencia es el desperdicio, que se podría medir sectorialmente.

A la sociedad le interesa, por una parte, que el gradiente del desarrollo sea superior al gradiente del acumulado, para darle la mayor eficacia a la acción financiera de la economía, y por otra, lograr el mayor beneficio en el uso que haga de los activos para el desarrollo, ya que son los que recibe en las manos. Lo llamaré satisfacción. Finalmente, todo se relaciona con todo y las acciones de cada producto de desarrollo provocan unas consecuencias en cualquier lugar, la teoría del caos. Lo llamaré integración. Con estos cuatro conceptos construyo un tetraedro sinérgico e interactivo.



El fin de la economía es optimizar este tetraedro y se verá obligada a asumir acciones nuevas que complementen a las productivas y mercantiles. Además, la acción de la economía ha de hacerse desde todas las posibilidades que todo saber le entregue. Ha de saber monetarizar muchas más relaciones con los ámbitos que ahora, pues la moneda o módulo de asunción y transmisión de la riqueza, es indispensable.

El crecimiento se mide financieramente y le corresponde a los economistas. El desarrollo se mide en unidades propias de producto y le corresponde a los ingenieros. Las satisfacción e integración se miden con métodos varios, de indicadores sociales, de producto, etc, y le corresponden a los políticos, sociólogos, pedagogos, etc. Todo en grandes rasgos.

Nada pertenece a nada exclusivamente, por ello los índices a crear se basarán en la matemática de la lógica difusa, métodos **fuzzy y análisis cluster**, de reciente aplicación a la tecnología e iniciados por Lotfi Zadeh, hace treinta años.

La Modelización

Con estas bases se crean los sistemas sinérgicos interactivos que sean precisos para ir buscando, analizando, decidiendo y resolviendo. Siempre con modelos borrosos, en similitud con las dinámicas vitales de las sociedades, se buscan métodos que integren el error como una variable existente y necesaria y por ello se opina que los esquemas de la

lógica difusa han de ser los óptimos para ir creando la serie de indicadores autoreorganizables, que la experiencia vaya pidiendo.

El objetivo es llegar a una modelización propia y que sea coherente con cada realidad, que lógicamente será distinta de otras modelizaciones. Con esta idea básica diseñé una amplia serie de modelos, iniciados para la energía y continuados culturalmente, sólo expongo algunos y linealmente.

- a/ para la búsqueda de acciones
coherencia cultural, consistencia social, participación holística e integración sistémica
- b/ para el mercado
de competencia, de competencia, de compartición y de compañía. * Una competencia derivada de competir, o calidad; y de competir, o costes; ambos simultáneamente.
- c/ para los activos
permanentes, obsolescentes, circulantes y trascendentes
- e/ para la eficacia
rentabilidad, usualidad, accesibilidad e integralidad
- e/ para la energía
eficacia, poder, acción y virtud
- f/ para el consumo
necesario, suficiente, respetado y apropiado
- g/ para el disfrute de la economía
consumo, uso, disposición y relación
- h/ para la realización de la economía
trabajo, empleo, ocupación y actividad
- i/ para los fines de la economía
bienes, servicios, procesos y sistemas
- j/ para las dinámicas económicas
acción, reacción, difusión y absorción
- k/ para la circunstancia económica
incidencia, admitancia, pertenencia y consecuencia

Recomendación

Sugiero al lector que dibuje un tetraedro y que lo use para pensar de forma sinérgica e interactiva, es un modelo que obliga a una disciplina mental muy profunda y se logran sorprendentes resultados. En este mismo trabajo se usa muy frecuentemente, he aquí los existentes hasta este momento:

a/ Métodos económicos: empresariales, productivos, comerciales y crediticios para crear mercado mediante las actividades válidas para ello.

b/ Saberes: antropológicos, tecnológicos, naturales y sociales para definir un amplio campo de conocimientos creadores de actividad y consecuentemente de economía.

c/ crecimiento, desarrollo, satisfacción e integración como diferentes formas del disfrute de la riqueza.

d/ buscando, analizando, decidiendo y resolviendo como un proceso completo de pensamiento.

Y así existen bastantes más tetraedros en todo el trabajo.

Los conceptos/actividades colocados en los vértices han de ser resueltos simultáneamente sin preferencia alguna ni tampoco con equivalencias pues en cada situación han de dar las respuestas que sean apropiadas. Veamos un ejemplo.

El tetraedro de eficacia (rentabilidad, ususalidad, accesibilidad e integralidad) busca resolver la rentabilidad: es decir, que los actos mercantiles necesarios para disponer de un bien, en su sentido más amplio, siempre sean realizables, en su modelo de oportunidad; que la accesibilidad al bien se logre no sólo por los planteamientos desde los costes competitivos, sino con modelos más completos, sean: la formación permanente en los productos obtenibles, leyes de calidad mínima, etc; que la ususalidad que el poseedor del bien consiga con él, tiene consecuencias que a todos nos interesan y por lo tanto, la acción de la economía no se culmina con la compra del bien, sino que muchas veces puede ser hasta el inicio de aquella, ya que se buscará la permanencia de resultados, sea el caso de la energía; que la integralidad o consecuencias del bien, siempre habrán de prevenirse y buscarse universal y generalísticamente, y por métodos ad-hoc.

En estos casos, todo habrá de considerarse y por la rentabilidad no habrá de perjudicarse la calidad, ni por la ususalidad se perjudique el mercado, ni nadie por nadie. Estos modelos quieren la completa satisfacción en la disposición de los bienes, sin olvidarse nunca del mercado, al que acabarán potenciando ya que surgirán nuevas actividades dentro de él. Pero un mercado cuya finalidad es lograr la comunicación y la libertad de todas las actividades, un mercado inconcreto, difuso, que no imponga a nadie el modelo que un ámbito haya sido beneficioso. Un mercado de funciones interactivas.

Funciones interactivas

Economías de uso, consumo, relación y disposición; mercados de competencia, compartición y compañía, sin olvidar al de siempre "la competencia"; riquezas de bienes, servicios, procesos y sistemas; bienes permanentes, obsolescentes, circulantes y trascendentes; etc. Pensándolo todo para dar paso a nuevos entes de actividad en la empresa y a nuevos tipos de empresa para realizar cualquier actividad que sea justificable por el conocimiento.

Entropía

La termodinámica enseña que un sistema, para funcionar, ha de entregar inevitablemente, al exterior, una cantidad de energía, entropía, que siempre se pensó que era desorden, pero Prigogine, que fue premio Nobel por este descubrimiento, puso en evidencia la capacidad que tienen los sistemas termodinámicos de autoorganizarse por zonas parciales usando energías que de otra forma serían desorden, si se hace trabajar a los sistemas para que actúen como cerrados/abiertos y lejos del equilibrio.

Esta es la gran lección que tiene pendiente la economía de mercado, reconvertirse en un sistema cerrado, para su eficacia interna, y abierto a los sistemas sociales, económicos, técnicos y naturales e imitando a los sistemas disipativos autoorganizarse parcialmente, sin más contemplaciones.

Si los exégetas de la economía de mercado sólo competitiva supiesen termodinámica, intentarían imitar, por encima de todo, en los modelos financieros de la energía a los sistemas disipativos de Prigogine. Naturalmente esta es una propuesta harto difícil e ¿irrealizable?; es una consideración para introducir un poco más de cordura al usar el mercado.

En el SIMO, años 85 y 90, he sido uno de los ponentes del de la mesa de debate que organiza el colegio de decoradores y arquitectos de interior de Madrid. Una vez el tema a tratar era la ergonomía y el otro fue libre, pero me pidieron que se orientase a concienciar a los decoradores sobre las responsabilidad técnica que adquirirían en su trabajo. O sea, otra vez ergonomía.

La primera vez presenté al concepto de empresa como un modelo de decisión económica cuya objetivo era provocar las dinámicas y los estudios precisos para llegar a definir las empresas, que realizarían las actividades que inicialmente no eran encajables en el modelo competitivo, pero que eran necesarias. La decoración está llena de ellas. Presenté a la ergonomía como algo más trascendente que la simple adaptación de un hombre a su puesto de trabajo, la señalé como una necesidad a definir, una ergonomía social/cultural para conseguir la adecuación del hombre a todos sus ámbitos y desde, ahí crear actividad.

Mi propuesta en ambos SIMOS fue la presentación, sin decirlo, del tetraedro formado por empresa, empresa, ergonomía y caos sencillamente pensados para aplicar a los sistemas sociales lo que Prigogine vió en los sistemas termodinámicos. Los tres primeros componentes ya se han señalado ligeramente, el cuarto, el caos, lo constituye el ámbito sobre él que se realiza la actividad de la empresa, ente pensado para estudiar las consecuencias de una acción, que al generar empresas se realizarán en el caos. Este ente, la empresa, no existe y el estado debería inventarlo a todo correr para disponer de una herramienta para la economía política mucho más eficaz que el PIB y el IPC juntos, porque sabría por donde le iban a salir las palomas del ojo.

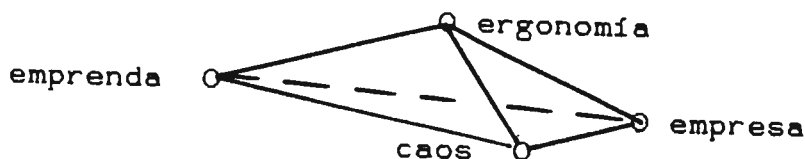
Esto querría, con Prigogine y siempre con Prigogine, decir que no debería haber miedo a estabilizar una acción determinada si existe un ente capaz de estudiar su integralidad y su acción borrosa en el conjunto. Pero no una acción que siempre sea la misma y que sea la única que define a todo el sistema, que en el mercado es el coste, como elemento de decisión.

Siguiendo a Prigogine, hemos de buscar como valorar cada acción del mercado en su propio ámbito, sin querer que esta valoración sea la misma que para el conjunto. En una palabra, hacer un acumulado colectivo financiero único y simplificador es la más importante causa de la pérdida de acción en el mercado. Los submercados parciales para cualquier actividad estarían magníficamente amparadas por la economía de mercado si se les aplicase el concepto de subsistema a equilibrarse independientemente del conjunto total. ¿Qué esperamos para fijarnos en los modelos termodinámicos y aplicarlos a la economía?

Prigogine indica que en una reacción química se puede lograr que en algún sector o subsistema se realice su autoorganización con independencia del equilibrio total del sistema, que puede no estar equilibrado. Prigogine dice que la entropía perdida por el sistema total, sirve ahora para que los subsistemas parciales se autoorganicen, lo que equivaldría a decir que las consecuencias obtenibles por aquellas son el argumento en su favor, aunque sólo se reflejen en su propio ámbito. Aplicar este concepto a la economía es pedir que los subsistemas parciales de acción los autoorganicemos nosotros, sin preocuparnos del total. Y es darle un golpe de muerte a medir los sistemas por un único índice, cualquiera que sea, aunque sea la peseta.

Ergonomía

La búsqueda y resolución de los actos apropiados a cada diferente realidad ha de ser practicada de tal manera que aquellos se efectúen en plena coherencia consu ámbito, para ellos se plantea el siguiente tetraedro,



En el tetraedro anterior, ergonomía representa todas las acciones coherentes con su ámbito, aquellas que vamos a realizar por ellas mismas. Cada mercado, creador de flujos monetarios por causa de realizar todas y cada de las actividades obtenidas desde el conocimiento, y cada forma de usar cada bien habrá de buscar en la ergonomía su plena justificación.

La empresa es la clásica de mercado que efectuará las actividades que la empresa le organice y provoque, porque la empresa sólo es el estudio para conseguir ese resultado y el diseño de las leyes y directrices para hacer posible cada acto ergómico, un ejemplo es construir cúpulas semiesféricas en vez de tejados de las casas de lugares tórridos.

La empresa busca no homogeneizar acciones por un motivo ajeno a ellas. La empresa busca el hacer, todo hacer que sea optable, todo hacer que sea provocable. Cualquier política económica es emprendarial, busca que los todos demás puedan actuar, el problema está en que este modelo han de ser posibles todos los modelos de decisión que nos ayuden a resolver más y más problemas, cada vez.

El caos es ámbito usado como elemento para entregarle respuestas parciales que una a una sean las coherentes con cada diferenciada situación, sin pretender que un modelo único sea útil todos. Este caos es para que en parcelas aisladas autororganice su equilibrio, es un caos que vive como el sistema inestable de Prigogine y bien mirado se parece muchísimo al ámbito general de la geografía económica.

Esto es bastante más poderoso que el actual modelo de acumulados financieros para lograr que cualquier conocimiento pueda entregar su bien a los hombres que lo precisaren.

Este modelo que es la síntesis de muchos trabajos, de muchos años, sólo podrá ser realizado si se disponen de métodos de análisis matemático que imiten a la propia vida y a su difusa forma de hacer.

Métodos de lógica difusa.

Modelos de razonamiento posibilístico: análisis comparativo*

Francesc Esteva, Pere Garcia-Calves, Lluís Godo

Centre d'Estudis Avançats de Blanes (CSIC)

Camí de Sta. Bàrbara s/n

17300 Blanes, Girona, Spain

e-mails: {esteva, pere, godo}@ceab.es

Resumen: Dos son los principales modelos de razonamiento posibilístico que han sido propuestos en la literatura: el modelo introducido por Dubois y Prade y el modelo introducido por Ruspini. Dubois y Prade utilizan una relación de preferencia en el conjunto de los mundos posibles para dar semántica a su modelo, mientras que Ruspini utiliza una noción de similitud o semejanza entre mundos. En este trabajo se comparan los dos modelos, estableciéndose una relación entre sus semánticas, planteándose también el problema de la validez de las reglas de inferencia de cada modelo en el otro.

Palabras clave: Lógica posibilística, necesidad, posibilidad, relación de similitud.

1. Introducción

El propósito de este trabajo es relacionar la formalización semántica de dos modelos para el razonamiento posibilístico. El primero de ellos fue propuesto y desarrollado por Dubois y Prade et al. en una serie de trabajos publicados desde mediados de los 80 (ver por ejemplo Dubois y Prade, 88) y su semántica esta basada en una relación de preferencia entre mundos, mientras que el introducido por Ruspini en (Ruspini 91a, 91b) esta basado en una relación de similitud o de semejanza entre mundos posibles. Aunque con diferentes motivaciones, ambos trabajos tienen como objetivo mostrar que la naturaleza de la información incompleta, que el razonamiento posibilístico pretende modelizar, difiere de la modelizada por otros tipos de razonamiento como el probabilístico o el evidencial. A nivel semántico las principales diferencias entre los dos modelos son las siguientes:

- en el modelo de Dubois y Prade, conocido también como Lógica Posibilística (LP), la descripción incompleta de un estado se hace por medio de un conjunto difuso de mundos posibles, o equivalentemente por una distribución de posibilidad del conjunto de mundos

* Este trabajo ha sido psrcialmente financiado por los proyectos: DGICYT nº PB91-0334-C03-03 y por el Esprit Basic Research Action nº 3085 DRUMS.

posibles, la cual induce un preorden total sobre el conjunto de los mundos, normalmente llamado orden de preferencia .

- en el modelo de Ruspini, al que llamaremos Lógica de la Similitud (LS), la incompletitud al describir un estado del sistema es modelizada mediante una relación de similitud sobre el conjunto de los mundos posibles, además de un subconjunto de mundos posibles, llamado conjunto evidencial, que modeliza el conocimiento que se tiene sobre el sistema. De esta forma la naturaleza del razonamiento posibilístico se basa en nociones métricas más que en nociones de preferencia.

Sin embargo, ambos modelos pueden relacionarse estableciendo una correspondencia entre relaciones de similitud y de preferencia. Así, en (Niiniluoto, 87) se define una noción no standard de conjunto difuso a partir de una relación de similitud. Este conjunto difuso interpretado como distribución de posibilidad permite definir un modelo de Dubois y Prade a partir de un modelo de Ruspini. Por otro lado, el teorema de representación de las T-indistinguibilidades (Valverde 85) permite obtener una relación de similitud generada por una distribución de posibilidad. Dicha similitud nos da un grado de similitud entre mundos que sólo tiene en cuenta el grado con que estos son preferidos.

Este trabajo está estructurado de la siguiente forma. En la segunda sección son presentadas las nociones básicas de los dos modelos. Además, las nociones básicas del modelo de Ruspini son reformuladas desde el punto de vista de la teoría de la medida y de los conjuntos difusos. Ello permite entender mejor la relación entre los modelos. En la tercera sección se estudia con detalle la correspondencia entre los dos modelos. Se demuestra que el modelo de Ruspini subsume el modelo de Dubois y Prade en el sentido que existe un subconjunto de enunciados y de modelos en la LS que preservan la relación de satisfacción entre enunciados y modelos de la LP.

En lo que sigue usaremos las convenciones siguientes:

- L representará un lenguaje proposicional.
- W un conjunto de interpretaciones clásica de L , W^* un conjunto de interpretaciones de L valoradas sobre $[0,1]$
- Para cada $w \in W$, $w \models p$ representará que p es cierto en la interpretación w , y dada una proposición p , $[p]$ representará el conjunto $\{ w \in W \mid w \models p \}$ de modelos donde la proposición p es cierta.
- T representará una T-norma continua, y R_T la implicación generada por residuación a partir de T , i.e., $R_T(x,y) = \sup\{c \in [0,1] \mid T(x,c) \leq y\}$.

2. Dos aproximaciones al razonamiento posibilístico.

2.1. Relación de preferencia entre mundos posibles: Lógica posibilística

El modelo desarrollado por Dubois, Prade y colegas a partir de los 80 tiene como característica básica la representación de la información incompleta sobre el estado real de las cosas por medio de una distribución de posibilidad $\pi: W \rightarrow [0,1]$, sobre el conjunto W de las

posibles interpretaciones (mundos posibles). Dicha distribución es a menudo interpretada como un preorden total de preferencia entre los mundos, es decir, w_1 es preferido a w_2 si $\pi(w_1) \geq \pi(w_2)$. Cada distribución de posibilidad π induce un par de medidas una de posibilidad y otra de necesidad, duales entre sí, sobre el álgebra de Boole de las proposiciones definidas por:

$$\text{Pos}_\pi, \text{Nec}_\pi : L \rightarrow [0,1]$$

$$\text{Pos}_\pi(p) = \text{Sup} \{ \pi(w) \mid w \models p \}$$

$$\text{Nec}_\pi(p) = 1 - \text{Pos}_\pi(\neg p)$$

Es fácil comprobar que estas funciones satisfacen los axiomas de las medidas de posibilidad y necesidad respectivamente, es decir, se verifican las siguientes propiedades:

- 1) $\text{Pos}_\pi(T) = \text{Nec}_\pi(T) = 1$, y $\text{Pos}_\pi(\perp) = \text{Nec}_\pi(\perp) = 0$, siendo T una tautología and $\perp$ una antitautología
- 2) Pos_π y Nec_π son monótonas respecto al orden reticular: $p \leq q$ si $\neg p \vee q = T$
- 3) $\text{Pos}_\pi(p \vee q) = \max(\text{Pos}_\pi(p), \text{Pos}_\pi(q))$ and $\text{Nec}_\pi(p \wedge q) = \min(\text{Nec}_\pi(p), \text{Nec}_\pi(q))$

Dichas medidas son la base de la denominada lógica Posibilística, que mediante fórmulas de la lógica proposicional valoradas mediante un grado de necesidad y otro de posibilidad permite el tratamiento de la incertidumbre. Las principales características de dicha lógica son resumidas a continuación.

Lenguaje: Dada una álgebra de Boole de proposiciones L , el lenguaje de la Lógica Posibilística esta formado por el siguiente conjunto de formulas:

$$PL_L = \{ (p, \Pi\alpha), (p, N\alpha) \mid p \in L, \alpha \in [0,1] \}$$

Semántica: un modelo es una distribución de posibilidad π sobre W . La relación de Satisfacción está definida por:

$$\pi \models (p, \Pi\alpha) \text{ si, y sólo, si } \text{Pos}_\pi(p) \geq \alpha$$

$$\pi \models (p, N\alpha) \text{ si, y sólo si, } \text{Nec}_\pi(p) \geq \alpha$$

Inferencia: La inferencia en la Lógica Posibilística está basada en el principio de resolución generalizado sobre las clausulas possibilísticas.

$$\frac{(p \vee q, v_1) \quad (\neg p \vee r, v_2)}{(q \vee r, v_1 \otimes v_2)}$$

siendo $\otimes$ la operación definida por:

$$N\alpha \otimes N\beta = N\min(\alpha, \beta) \quad \Pi\alpha \otimes \Pi\beta = \Pi 0 \quad N\alpha \otimes \Pi\beta = \begin{cases} \Pi\beta, & \text{if } \alpha + \beta > 1 \\ \Pi 0, & \text{if } \alpha + \beta \leq 1 \end{cases}$$

2.2. Relación de similitud entre mundos posibles: Fundamentos semánticos de la lógica difusa según Ruspini.

El modelo de Ruspini [Ruspini, 91a] para el razonamiento posibilístico se basa en la introducción de una relación S en el conjunto de los mundos posibles W que asigna un número entre 0 y 1 a cada par de mundos posibles. Dicho número cuantifica su similitud desde el punto de vista del problema particular que se considera. Para que dicha relación pueda atrapar la noción de similitud, se le requieren las tres propiedades siguientes:

$$\text{Reflexiva: } S(w, w) = 1$$

$$\text{Simétrica: } S(w, w') = S(w', w)$$

$$\text{T-transitiva: } S(w, w') \geq T(S(w, w''), S(w'', w'))$$

donde T es una t-norma continua. Estas relaciones son conocidas como relaciones de T-indistinguibilidad. Por razones de simplicidad supondremos además que S satisface: $S(w, w') = 1$ iff $w = w'$.

Los α -cortes de una función de similitud S definen una familia anidada de relaciones de accesibilidad R_α sobre el conjunto de los mundos posibles: $R_\alpha(w, w')$ si, y sólo si, $S(w, w') \geq \alpha$. Para cada $\alpha \in [0,1]$, la relación de accesibilidad R_α define una Lógica Modal Brouweriana cuyos operadores modales son $\Diamond_\alpha$ y $\Box_\alpha$, y cuya relación de satisfacción se define de la forma siguiente:

$$w \models \Diamond_\alpha p, \text{ ssi existe un mundo } w' \text{ tal que } S(w, w') \geq \alpha \text{ y } w' \models p$$

$$w \models \Box_\alpha p, \text{ ssi para cada mundo } w' \text{ tal que } S(w, w') \geq \alpha \text{ satisface } w' \models p$$

La relación de similitud S sobre W permite a Ruspini generalizar la relación de deducción entre enunciados en términos de una medida de proximidad a un subconjunto de mundos posibles definiendo los denominados *grado de Implicación* y *grado de Consistencia* como:

$$I(p \mid q) = \inf_{w' \models q} \sup_{w \models p} S(w, w') \quad C(p \mid q) = \sup_{w' \models q} \sup_{w \models p} S(w, w')$$

Estas dos medidas también pueden ser expresadas a través de los operadores modales como:

$$I(p \mid q) = \sup \{ \alpha \in [0,1] \mid q \models \Diamond_\alpha p \} \quad C(p \mid q) = \sup \{ \alpha \in [0,1] \mid q \models \Box_\alpha \neg p \}$$

De estas definiciones se deduce directamente que $C(p \mid q) \geq I(p \mid q)$ y el grado de consistencia C es una medida simétrica mientras que el grado de implicación I no lo es, pero conserva la T-transitividad de la función de similitud S , i.e. $I(p \mid q) \geq T(I(p \mid r), I(r \mid q))$ para cualquier proposición p, q y r .

La información sobre el estado de un sistema se modeliza mediante un subconjunto E de mundos posibles, denominado *conjunto evidencial*, es decir, el conjunto de mundos posibles que son compatibles con la evidencia. Dado un conjunto evidencial E , Ruspini introduce las llamadas *distribuciones de posibilidad y necesidad (condicionadas y no condicionadas)*. las cuales, como veremos más adelante, tienen un significado que no coincide plenamente con la definición standard de distribuciones de posibilidad y la necesidad "clásicas".

1) Distribuciones no condicionadas

Sea E un conjunto evidencial. Funciones $Nec : L \rightarrow [0,1]$ y $Pos : L \rightarrow [0,1]$ se dirán respectivamente de necesidad no condicionada y de posibilidad no condicionada respecto a E si:

$$\begin{aligned} Nec(p) &\leq I(p \mid E) \\ Pos(p) &\geq C(p \mid E) \end{aligned}$$

2) Distribuciones condicionadas

Sea E un conjunto evidencial. Funciones $Nec : L \rightarrow [0,1]$ y $Pos : L \rightarrow [0,1]$ se dirán de necesidad condicionada y de posibilidad condicionada respecto a E si:

$$\begin{aligned} Nec(p \mid q) &\leq \inf_{w \mid E} \{ I_T (I(q \mid w), I(p \mid w)) \} \\ Pos(p \mid q) &\geq \sup_{w \mid E} \{ I_T (I(q \mid w), I(p \mid w)) \} \end{aligned}$$

3) Modus Ponens

i) *para funciones de necesidad*: dado un conjunto de proposiciones $P = \{q_i \mid i \in I\}$ tal que para cada $q_i \in P$ se cumple

$$\begin{aligned} Nec(q_i) &\leq I(q_i \mid E) \\ Nec(p \mid q_i) &\leq \inf_{w \models E} \{ R_T (I(q_i \mid w), I(p \mid w)) \}, \end{aligned}$$

entonces también se cumple

$$\sup_{i \in I} \{ T(Nec(q_i), Nec(p \mid q_i)) \} \leq I(p \mid E)$$

ii) *para funciones de posibilidad*: dado un conjunto de proposiciones $P = \{q_i \mid i \in I\}$ tal que para cada $w \in E$ existe $q_i \in P$ tal que $I(q_i \mid w) = 1$, y para cada $q_i \in P$ se cumple

$$\begin{aligned} Pos(q_i) &\geq C(q_i \mid E) \\ Pos(p \mid q_i) &\geq \sup_{w \models E} \{ R_T (I(q_i \mid w), I(p \mid w)) \}, \end{aligned}$$

entonces también se cumple

$$\sup_{i \in I} \{ T(Pos(q_i), Pos(p \mid q_i)) \} \geq C(p \mid E)$$

Por tal de poder comparar más fácilmente dicho modelo con el modelo de Dubois y Prade introducimos, en la siguientes subsecciones, una reinterpretación del modelo de Ruspini desde el punto de vista de la teoría de la medida y de los conjuntos difusos.

2.2.1. Medidas de Implicación y Consistencia

Dada una relación de similitud S y un conjunto evidencial E , introducimos sobre el conjunto de proposiciones el par de medidas incondicionales $(I_{S,E}, C_{S,E})^1$ definidas por:

$$I_{S,E}(p) = \inf_{w' \in E} \sup_{w \models p} S(w, w')$$

$$C_{S,E}(p) = \sup_{w' \in E} \sup_{w \models p} S(w, w')$$

Es importante notar que $C_{S,E}$ es una medida clásica de posibilidad, mientras que $I_{S,E}$ no es una medida de necesidad ya que en general la igualdad $I_{S,E}(p \wedge q) = \min(I_{S,E}(p), I_{S,E}(q))$ no se satisface. Es inmediato comprobar que las dos medidas están relacionadas por las siguientes desigualdades:

$$\begin{array}{ccccc} C_{S,E}(p) & \geq & I_{S,E}(p) & \geq & 1 - C_{S,E}(\neg p) \\ C_{S,E}(p) & \geq & 1 - I_{S,E}(p) & \geq & 1 - C_{S,E}(\neg p) \end{array}$$

Demostración.

La desigualdad $C_{S,E}(p) \geq I_{S,E}(p)$ se deduce de la definición. Para demostrar que $I_{S,E}(p) \geq 1 - C_{S,E}(\neg p)$, obsérvese que $S(w, w) = 1$ para todo w , entonces para todo w' se satisface $\sup_{w \models p} S(w, w') + \sup_{w \models \neg p} S(w, w') \geq 1$, de lo que se deduce:
 $I_{S,E}(p) = \inf_{w' \in E} \sup_{w \models p} S(w, w') \geq \inf_{w' \in E} (1 - \sup_{w \models \neg p} S(w, w')) = 1 - C_{S,E}(\neg p)$.

A continuación definimos las siguientes medidas condicionadas:

$$I_{S,E}(p | q) = \inf_{w \in E} \{ R_T(I_{S,w}(q), I_{S,w}(p)) \}$$

$$C_{S,E}(p | q) = \sup_{w \in E} \{ R_T(I_{S,w}(q), I_{S,w}(p)) \}$$

Nótese que en particular, cuando q es una tautología T , i.e., cuando $[q] = W$, entonces se recuperan las medidas incondicionales : $I_{S,E}(p | T) = I_{S,E}(p)$ y $C_{S,E}(p | T) = C_{S,E}(p)$. Nótese también que el grado de implicación y el grado de consistencia entre dos proposiciones definidos por Ruspini pueden ser expresados como

$$I(p | q) = I_{S,[q]}(p) \qquad C(p | q) = C_{S,[q]}(p).$$

Finalmente, dada una proposición p y una familia de proposiciones $\{q_i | i \in I\}$, las propiedades entre una t -norma T y su implicación residuada R_T permiten expresar las siguientes generalizaciones del Modus Ponens para las medidas de implicación y consistencia :

¹Ruspini nota estas funciones por $I(. | E)$ y $C(. | E)$ respectivamente.

$$\begin{array}{c|c}
\begin{array}{l} I_{S,E}(q_i) \geq \alpha_i \\ I_{S,E}(p \mid q_i) \geq \beta_i \end{array} & \begin{array}{l} C_{S,E}(q_i) \leq \alpha_i \\ C_{S,E}(p \mid q_i) \leq \beta_i \end{array} \\
\hline
I_{S,E}(p) \geq \sup_{i \in I} T(\alpha_i, \beta_i) & C_{S,E}(p) \leq \inf_{i \in I} T(\alpha_i, \beta_i),
\end{array}$$

siendo la segunda regla válida si para cada $w \in E$ existe $i \in I$ tal que $I_{S,w}(q_i) = 1$.

2.2.2. Similitudes y conjuntos difusos

La noción de similitud y la de conjunto difuso pueden relacionarse entre sí. Por ejemplo, en (Niiniluoto, 87) una noción de conjunto difuso fue introducida como una generalización de la extensión de los conceptos vagos o mal definidos donde los grados de pertenencia son calculados a partir de una relación difusa. Mas específicamente, supongamos que un concepto A es aplicado sobre un conjunto de casos paradigmáticos $A_p = \{a_1, a_2, \dots, a_n\}$ de un universo U , el método propuesto por Niiniluoto afirma que el concepto A puede aplicarse a otros casos si estos son suficientemente similares a alguno de los de A_p . Supongamos por tanto que tenemos definida una función de similitud S entre elementos de U , entonces una generalización natural de la función característica de A puede ser definida para cada elemento de U de la siguiente forma::

$$\mu_A(x) = \sup \{ S(x, a_i) \mid a_i \in A_p \}$$

Niiniluoto no va mas allá en su propuesta, y no da una definición formal ni condiciones sobre la relación difusa. Usando dicha noción de conjunto difuso nosotros reinterpretaremos los principales elementos del modelo de Ruspini.

1) Dada una relación de similitud S , asociaremos a cada proposición p un conjunto difuso del conjunto de los mundos posibles, que notaremos por $\tilde{p}$, y cuya función característica será:

$$\mu_{\tilde{p}}(w) = \sup \{ S(w, w') \mid w' \in [p] \}$$

Obsérvese que se verifica $\mu_{\tilde{p}}(w) = I(p \mid w) = C(p \mid w)$, i.e. el valor máximo de α tal que $w \models \prod_{\alpha} p$ o equivalentemente $w \not\models N_{\alpha} \neg p$. Con esta definición de conjunto difuso, la unión coincide con la clásica pero no es el caso de la intersección y el del complementario. Más concretamente, sólo tenemos en general las relaciones siguientes :

$$\mu_{(p \vee q)}(w) = \sup \{ S(w, w') \mid w' \in [p \vee q] \} = \max(\mu_{\tilde{p}}(w), \mu_{\tilde{q}}(w))$$

$$\mu_{(p \wedge q)}(w) = \sup \{ S(w, w') \mid w' \in [p \wedge q] \} \leq \min(\mu_{\tilde{p}}(w), \mu_{\tilde{q}}(w))$$

$$\mu_{(\neg p)}(w) = \sup \{ S(w, w') \mid w' \in [\neg p] \} \geq 1 - \mu_{\tilde{p}}(w)$$

Es interesante observar que si nombramos FS a la función de L al conjunto de las partes difusas de W que asigna a cada proposición p el conjunto difuso $\tilde{p}$, entonces el álgebra $(FS(L), \vee, \wedge, \neg)$, cuyas operaciones están definidas por

$$\tilde{p} \vee \tilde{q} = (p \vee q)^\sim \quad \tilde{p} \wedge \tilde{q} = (p \wedge q)^\sim \quad \neg \tilde{p} = (\neg p)^\sim$$

es una álgebra de Boole. Recuérdese que el álgebra introducida por Zadeh para los conjuntos difusos, i.e., $([0,1]^L, \max, \min, 1-\text{Id})$ no es una álgebra de Boole.

Como veremos más adelante, usando esta notación, la formulación del Modus Ponens generalizado hace uso del condicional $\tilde{p} \rightarrow \tilde{q}$, cuya función característica viene definida como $\mu_{\tilde{p} \rightarrow \tilde{q}}(w) = R_T(\mu_{\tilde{q}}(w), \mu_{\tilde{p}}(w))$, en vez de definirla como $(\neg p \vee q)^\sim$. Nótese que $\tilde{p} \rightarrow \tilde{q}$ no ha de pertenecer necesariamente a $F_S(L)$. Esta situación es muy similar a la que se da en el marco del razonamiento probabilístico donde los condicionales son a menudo modelizados mediante probabilidades condicionadas y no como probabilidades sobre la implicación material.

2) Utilizando la definición anterior de conjunto difuso, las medidas no condicionadas $I_{S,E}$ y $C_{S,E}$ pueden ser reformuladas como

$$I_{S,E}(p) = I(p | E) = \inf_{w \in E} \{ \mu_{\tilde{p}}(w) \}$$

$$C_{S,E}(p) = C(p | E) = \sup_{w \in E} \{ \mu_{\tilde{p}}(w) \}$$

o en otras palabras, el intervalo $[I_{S,E}(p), C_{S,E}(p)]$ representa el rango de los valores de verdad de la proposición difusa $\tilde{p}$ cuando el conjunto evidencial E se toma como referencia (ver figura 1).

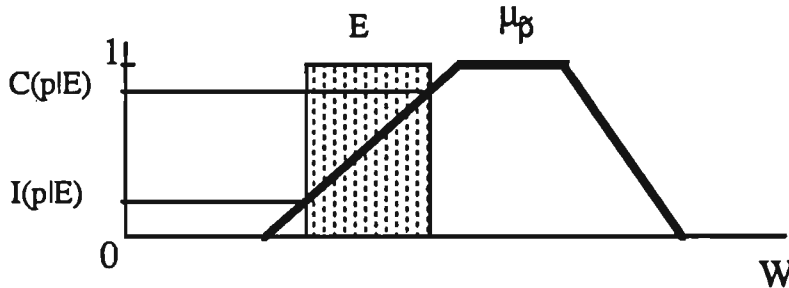


Figura 1.

Estas dos medidas coinciden como funciones con las medidas de necesidad y posibilidad de un conjunto difuso $\tilde{p}$ dado E como referencia definidas por Zadeh, a saber:

$$\text{Pos}(\tilde{p} ; E) = \sup_{w \in \Omega} \min(\mu_E(w), \mu_{\tilde{p}}(w)) = C_{S,E}(p)$$

$$\text{Nec}(\tilde{p} ; E) = \inf_{w \in \Omega} \max(1 - \mu_E(w), \mu_{\tilde{p}}(w)) = I_{S,E}(p)$$

Ahora bien, téngase en cuenta que, aunque en el álgebra de conjuntos difusos definida por Zadeh dichas medidas Pos y Nec generalizan a las medidas de posibilidad y necesidad, en el álgebra de Boole $F_S(L)$ definida anteriormente, $C_{S,E}$ continúa siendo una medida de posibilidad pero $I_{S,E}$ no es una medida de necesidad, tal como ya se había comentado anteriormente.

3) Dados dos conjuntos difusos $\tilde{p}$ y $\tilde{q}$ definimos el conjunto difuso de un condicional $\tilde{q} \rightarrow \tilde{p}$ como:

$$\mu_{\tilde{q} \rightarrow \tilde{p}}(w) = RT(\mu_{\tilde{q}}(w), \mu_{\tilde{p}}(w))$$

Esto es una técnica habitual en Lógica difusa donde el condicional del tipo "si q entonces p " son modelizados mediante funciones de implicación residuadas. Con esta definición, las medidas condicionadas $I_{S,E}(.|.)$ y $C_{S,E}(.|.)$ pueden ser expresadas de la siguiente forma:

$$I_{S,E}(p | q) = \inf_{w \in E} \{ \mu_{\tilde{q} \rightarrow \tilde{p}}(w) \}$$

$$C_{S,E}(p | q) = \sup_{w \in E} \{ \mu_{\tilde{q} \rightarrow \tilde{p}}(w) \}$$

4) Como consecuencia directa de las anteriores definiciones, los esquemas del Modus Ponens generalizado, para el caso de que el conjunto de P tenga una sola proposición q , pueden ser formulados como:

$$\inf_{w \in E} \{ \mu_{\tilde{q}}(w) \} \geq \alpha$$

$$\sup_{w \in E} \{ \mu_{\tilde{q}}(w) \} = 1$$

$$\inf_{w \in E} \{ \mu_{\tilde{q} \rightarrow \tilde{p}}(w) \} \geq \beta$$

$$\sup_{w \in E} \{ \mu_{\tilde{q} \rightarrow \tilde{p}}(w) \} \leq \beta$$

$$\inf_{w \in E} \{ \mu_{\tilde{p}}(w) \} \geq T(\alpha, \beta)$$

$$\sup_{w \in E} \{ \mu_{\tilde{p}}(w) \} \leq \beta,$$

siendo el segundo de ellos válido solamente si existe $w \in E$ tal que $\mu_{\tilde{q}}(w) = 1$.

3. Correspondencia entre el modelo de Dubois-Prade y el de Ruspini

A partir de los resultados anteriores y para facilitar la comparación entre los dos modelos daremos una presentación formal de la LS.

Primeramente observemos que para introducir el Modus Ponens generalizado como una regla de inferencia de nuestra lógica tenemos que ampliar el lenguaje definido anteriormente para la LP con condicionales de la forma $p|q$ para cada par de proposiciones. De esta forma tendremos:

Lenguaje: Dada una álgebra de Boole de proposiciones L , el lenguaje de LS se define como:

$$SL_L = \{ (p, [\alpha, \beta]), (p | q, [\alpha, \beta]), \text{ tal que } p, q \in L, \alpha, \beta \in [0, 1], \alpha \leq \beta \}$$

Semántica: Un modelo para LS es un par (S, E) , siendo S una relación de similitud respecto T sobre el conjunto de los mundos posibles W o interpretaciones de L , y E un subconjunto de W denominado conjunto evidencial. La relación de satisfacción entre enunciados y modelos se define de la forma siguiente:

Para cualquier $p \in SL_L$, $(S, E) \models (p, [\alpha, \beta])$ sí, y sólo sí, $\alpha \leq I_{S,E}(p)$ y $C_{S,E}(p) \leq \beta$

Regla de inferencia: Modus Ponens generalizado.

$$\frac{(p_i, [\alpha_i, \beta_i])_{i \in I} \quad (q \mid p_i, [\delta_i, \phi_i])_{i \in I}}{(q, [\sup_{i \in I} T(\alpha_i, \delta_i), \sup_{i \in I} T(\beta_i, \phi_i)])}$$

siendo la regla válida si para todo $w \in W$ existe $i \in I$ tal que $C_{S,w}(p_i) = 1$.

En lo que resta de sección describiremos como algunos de los elementos de la Lógica posibilitista pueden ser descritos dentro de la LS y viceversa.

3.1. Relación a nivel semántico.

3.1.1. De la LS a la LP

Para transformar un modelo de la LS en un modelo de la LP usaremos el concepto, introducido por Niiniluoto, que permite obtener un conjunto difuso a partir de una relación de similitud. Concretamente, dado un modelo (S, E) de la LS, consideramos el subconjunto difuso de W cuya función característica está definida por:

$$\mu_E(w) = \sup \{ S(w, w') \mid w' \in E \}$$

y por tanto la distribución de posibilidad $\pi_{S,E}(w) = \mu_E(w)$ inducida en W será usada como modelo de la LP. De este modo, dicha distribución de posibilidad, permite definir las medidas de posibilidad y necesidad sobre las proposiciones de L , de la forma siguiente:

$$\begin{aligned} \text{Pos}_{\pi_{S,E}}(p) &= \sup \{ \pi_{S,E}(w) \mid w \models p \} = \sup_{w' \models E} \sup_{w \models p} S(w, w') \\ \text{Nec}_{\pi_{S,E}}(p) &= 1 - \text{Pos}_{\pi_{S,E}}(\neg p) = \inf_{w' \models E} \inf_{w \models \neg p} 1 - S(w, w') \end{aligned}$$

Notemos la coincidencia entre grados de posibilidad en $\pi_{S,E}$ y los grados de consistencia incondicional en $\pi_{S,E}$, i.e.

$$\text{Pos}_{\pi_{S,E}}(p) = C_{S,E}(p)$$

se satisface para cada proposición p . Sin embargo no se puede establecer ninguna relación entre los grados de necesidad en $\pi_{S,E}$ y los de implicación en $\pi_{S,E}$. Ello es debido a que el grado de implicación de una proposición p no está relacionado con el grado de consistencia de la proposición negada $\neg p$. Además, la anterior correspondencia entre modelos induce una relación entre las relaciones de satisfacción de las dos lógicas, tal como establece la siguiente proposición.

Proposición. Para cada modelo (S, E) se satisfacen las siguientes propiedades:

- i) Si $(S, E) \models (p, [\alpha, \beta])$ entonces $\pi_{S,E} \models (\neg p, N_{1-\beta})$ y $\pi_{S,E} \models (p, \Pi_\alpha)$
- ii) Si $\pi_{S,E} \models (p, N_\alpha)$ entonces $(S, E) \models (\neg p, [0, 1-\alpha])$ y $(S, E) \models (p, [\alpha, 1])$

Demostración.

i) Si $(S, E) \models (p, [\alpha, \beta])$, entonces por definición $I_{S,E}(p) \geq \alpha$ y $C_{S,E}(p) \leq \beta$, pero por la anterior proposición tenemos $\text{Pos}_{\pi_{S,E}}(p) = C_{S,E}(p) \geq I_{S,E}(p) \geq \alpha$ y $\text{Nec}_{\pi_{S,E}}(\neg p) = 1 - C_{S,E}(p) \leq 1 - \beta$, de donde tenemos $\pi_{S,E} \models (p, \prod_{\alpha})$ y $\pi_{S,E} \models (\neg p, N_{1-\beta})$.

ii) Si $\pi_{S,E} \models (p, N_{\alpha})$, entonces $1 - C_{S,E}(\neg p) = \text{Nec}_{\pi_{S,E}}(p) \geq \alpha$, de donde $(S, E) \models (\neg p, [0, 1 - \alpha])$. Por otro lado $I_{S,E}(p) \geq \text{Nec}_{\pi_{S,E}}(p) \geq \alpha$ y por lo tanto $(S, E) \models (p, [\alpha, 1])$.

3.1.2. De la LP a la LS.

Para tratar el paso de LP a LS, hace falta de entrada un método para definir una similitud a partir de una distribución de posibilidad. Para ello, utilizaremos un resultado básico del "fuzzy clustering", que de forma sucinta describimos a continuación. Sea U un conjunto de objetos, A un conjunto de atributos que se pueden predicar de los objetos de U , y supongamos conocida una modelización del grado de satisfacción de cada una de las proposiciones "el objeto x posee el atributo a ". El problema es cómo obtener una clasificación del conjunto U a partir de estos grados. Concretando, el problema es cómo obtener una relación de similitud sobre U , a partir de una familia $\{h_a\}_{a \in A}$ de subconjuntos difusos de U , cada uno de los cuales modelizando el atributo a para todos los objetos de U . La forma más general viene dada por el siguiente teorema de representación de similitudes (Valverde, 85)

Teorema. Una función binaria S , que hace corresponder a cada par de elementos del universo U un elemento de $[0, 1]$ es una relación de T -similitud si, y sólo si, existe una familia H de conjuntos difusos de U tal que:

$$S(w, w') = \inf_{h \in H} \{ \min(R_T(h(w), h(w')), R_T(h(w'), h(w))) \}$$

para todo w y w' de U . Cada elemento h de la familia H se llama un generador de S .

De acuerdo con este teorema, dada una distribución de posibilidad π , la T -similitud generada por π viene definida por:

$$S_{\pi}(w, w') = \min(R_T(\pi(w), \pi(w')), R_T(\pi(w'), \pi(w)))$$

La elección de un conjunto evidencial E_{π} asociado a π parece natural hacerla dentro del conjunto de mundos que son más preferidos con respecto a π , i.e., cualquier subconjunto del $\text{Core}(\pi) = \{ w \in W \mid \pi(w) = 1 \}$. Además, eligiendo cualquier subconjunto no vacío de $\text{Core}(\pi)$ como conjunto evidencial se verifican las propiedades siguientes:

Proposición. Sea π una distribución de posibilidad sobre W , y S_{π} la T -similitud definida anteriormente. Entonces para cada $E_{\pi} \subset \text{Core}(\pi)$ y $E_{\pi} \neq \emptyset$ se satisfacen las propiedades siguientes:

$$(i) \pi_{S_{\pi}, E_{\pi}} = \pi.$$

(ii) $IS_{\pi,E\pi}(p) = CS_{\pi,E\pi}(p) = Pos_{\pi}(p)$, para cada proposición p .

Demostración.

$$\begin{aligned}
 (i) \pi_{S_{\pi,E\pi}}(w) &= \text{Sup}\{ S_{\pi}(w, w') \mid w' \in E_{\pi} \} = \\
 &= \text{Sup}\{ \text{Min}(R_T(\pi(w), \pi(w')), R_T(\pi(w'), \pi(w))) \mid w' \in E_{\pi} \} = \\
 &= \text{Min}(R_T(\pi(w), 1), R_T(1, \pi(w))) = \pi(w). \\
 (ii) CS_{\pi,E\pi}(p) &= C(p \mid E_{\pi}) = \text{Sup}_{w' \models E_{\pi}} \text{Sup}_{w \models p} S_{\pi}(w, w') = \\
 &= \text{Sup}_{w' \models E_{\pi}} \text{Sup}_{w \models p} \text{Min}(R_T(\pi(w), \pi(w')), R_T(\pi(w'), \pi(w))) = \\
 &= \text{Sup}_{w \models p} \text{Min}(R_T(\pi(w), 1), R_T(1, \pi(w))) = \text{Sup}_{w \models p} \pi(w) = Pos_{\pi}(p) \\
 IS_{\pi,E\pi}(p) &= I(p \mid E_{\pi}) = \text{Inf}_{w' \models E_{\pi}} \text{Sup}_{w \models p} S_{\pi}(w, w') = \\
 &= \text{Inf}_{w' \models E_{\pi}} \text{Sup}_{w \models p} \text{Min}(R_T(\pi(w), \pi(w')), R_T(\pi(w'), \pi(w))) = \\
 &= \text{Sup}_{w \models p} \text{Min}(R_T(\pi(w), 1), R_T(1, \pi(w))) = \text{Sup}_{w \models p} \pi(w) = Pos_{\pi}(p)
 \end{aligned}$$

La igualdad $\pi_{S_{\pi,E\pi}} = \pi$ pone de manifiesto que se recupera la distribución de posibilidad original π a partir de la relación de similitud S_{π} y de cualquier conjunto evidencial $E_{\pi} \subset \text{Core}(\pi)$.

Por otra parte, la igualdad $IS_{\pi,E\pi}(p) = CS_{\pi,E\pi}(p) = Pos_{\pi}(p)$ muestra que los grados de implicación y consistencia inducidos por (S_{π}, E_{π}) coinciden y son iguales a la posibilidad inducida por π . Esto es una consecuencia directa de hacer indistinguibles todos los elementos de $\text{Core}(\pi)$ a través de S_{π} . Otra consecuencia es que la función característica de $\bar{p}$, $\mu_{\bar{p}}(w) = \text{Sup}\{ S_{\pi}(w, w') \mid w' \in [p] \}$, tiene valor constante e igual a $Pos_{\pi}(p)$ sobre $\text{Core}(\pi)$ (ver figura. 2).

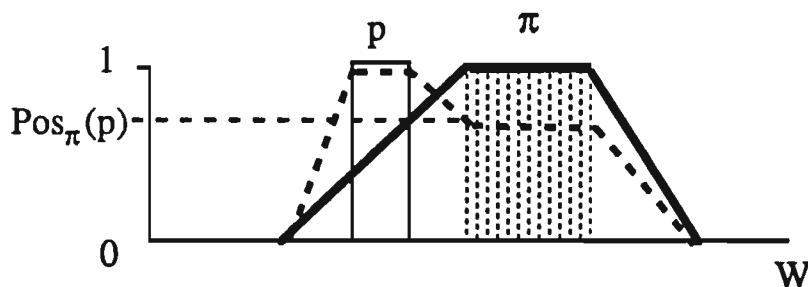


Figura 2. (La línea segmentada corresponde a la función $\mu_{\bar{p}}(w) = \text{Sup}\{ S_{\pi}(w, w') \mid w' \in [p] \}$)

El siguiente teorema muestra como se puede describir la semántica de la LP en términos de la semántica de la LS.

Teorema. Dado un modelo π en la LP, se verifican las siguientes condiciones:

- i) $\pi \models (p, \Pi_{\alpha})$ ssi $(S_{\pi}, E_{\pi}) \models (p, [\alpha, 1])$
- ii) $\pi \models (p, N_{\alpha})$ ssi $(S_{\pi}, E_{\pi}) \models (\neg p, [0, 1-\alpha])$
- iii) $(S_{\pi}, E_{\pi}) \models (p, [\alpha, \beta])$ ssi $\pi \models (p, \Pi_{\alpha})$ y $\pi \models (\neg p, N_{1-\beta})$

Los resultados dados anteriormente pueden sintetizarse como sigue. A partir de los resultados de §3.1.1 y del apartado actual, definimos las funciones F y G como:

$$F(S, E) = \pi_{S, E}$$

$$G(\pi) = (S\pi, \text{Core}(\pi)^2)$$

Del resultado dado en (Jacas, 88) sobre generadores de indistinguibilidades, resulta sencillo probar que $\pi_{S, E}$ es un generador de la relación de similitud S , y por lo tanto la similitud $S\pi_{S, E}$ generada por $\pi_{S, E}$ es siempre mayor o igual que S para cualquier E . Esta propiedad, junto con la igualdad $\pi_{S\pi, E\pi} = \pi$ de la proposición anterior, nos permite enunciar la proposición siguiente:

Proposición. Si Id_{SL} y Id_{PL} representan las funciones identidad sobre los modelos de la LP y de la LS respectivamente, entonces las funciones F y G satisfacen las propiedades siguientes:

(i) $\text{GoF} \geq \text{Id}_{SL}$ y $\text{FoG} = \text{Id}_{PL}$,³

(ii) GoF es un operador clausura sobre los modelos de la LS, y además los modelos (S, E) son cerrados respecto este operador si, y sólo si, S es una relación de similitud unidimensional (i.e. puede ser generada por un sólo generador) y $E = \text{Core}(\pi_{S, E})$.

Finalmente es sencillo comprobar que, aunque $\text{GoF} \geq \text{Id}_{SL}$, el grado de consistencia se preserva mediante dicha transformación, i.e., $C_{S, E}(p) = C_{\text{GoF}(S, E)}(p)$ se satisface para todo p .

3.2. Relación a nivel sintáctico.

Como se ha visto, la inferencia en los dos modelos se plantea de forma distinta. En la LS se formula a través de medidas condicionadas, mientras que en la LP los enunciados condicionales son modelizados a través de la implicación material.

Para el estudio de la relación entre los mecanismos de inferencia propuestos en la LP y en la LS tenemos pues dos posibilidades: definir un esquema de resolución en la LS o considerar medidas condicionadas en la LP.

1) Dado que $C_{S, E}$ es una medida de posibilidad, e $I_{S, E}$ no es una medida de necesidad, la única regla de inferencia en la LS correspondiente al principio de resolución en la LP es la forma dual del mismo principio. Teniendo en cuenta que la fórmula $(p, N\alpha)$ de la LP se corresponde con la fórmula $(\neg p, [0, \alpha])$ en la LS, el siguiente esquema de inferencia es válido en la LS:

$$\frac{(p \wedge q, [0, \alpha]) \quad (\neg p \wedge r, [0, \beta])}{(q \wedge r, [0, \max(\alpha, \beta)])}$$

² Nosotros elegimos aquí el conjunto evidencial maximal por razones de simplicidad, aunque otras elecciones son posibles.

³Un par de funciones (F, G) entre conjuntos ordenados verificando (i) es conocido como una conexión de Galois..

2) Condicionales:

- En la LP, se ha definido una medida de posibilidad condicionada $\Pi(q|p)$ (ver Dubois-Prade 91b) como la solución más específica a la ecuación $\Pi(p \wedge q) = T(\Pi(q|p), \Pi(p))$, siendo T una t -norma. Se demuestra que dicha posibilidad condicionada es:

$$\Pi(q|p) = R_T(\Pi(p), \Pi(p \wedge q)).$$

Por ejemplo, cuando $T = \text{mínimo}$ o $T = \text{producto}$ se obtienen las posibilidades condicionales siguientes:

$$\Pi_{\min}(q|p) = \begin{cases} 1, & \text{if } \Pi(p \wedge q) = \Pi(p) \\ \Pi(p \wedge q), & \text{otherwise} \end{cases} \quad \Pi_{\text{prod}}(q|p) = \begin{cases} 1, & \text{if } \Pi(p \wedge q) = \Pi(p) \\ \frac{\Pi(p \wedge q)}{\Pi(p)}, & \text{otherwise} \end{cases}$$

- La medida de los condicionales usados en el Modus Ponens como mecanismo de inferencia en la LS, es decir $C_{S,E}(I)$, no es una medida de posibilidad condicionada del tipo dado por Dubois-Prade ya que sólo satisface:

$$C_{S,E}(p \wedge q) \leq T(C_{S,E}(q|p), C_{S,E}(p)).$$

Por lo tanto el Modus Ponens usado como mecanismo de inferencia por Ruspini en la LS no puede trasladarse en la LP usando las posibilidades condicionales ya definidas. El estudio de qué medidas condicionales se tendrían que definir en LP de forma que se correspondieran con las definidas en LS, sobrepasa los objetivos del presente trabajo.

4. Conclusiones y perspectivas.

En este trabajo hemos relacionado dos modelos para el razonamiento posibilístico. Para ello hemos partido de la definición formal de las lógicas que se derivan de dichos modelos: la LP y la LS. Se han comparado dichas lógicas desde el punto de vista semántico, i. e., comparando sus relaciones de satisfacción. Desde este punto de vista, la LS subsume a la LP en el sentido de que existe un subconjunto de enunciados y de modelos en la LS que preservan la relación de satisfacción entre enunciados y modelos de la LP. En contrapartida, se demuestra que desde el punto de vista sintáctico su relación no resulta tan directa, dado que mientras la LP utiliza la resolución como regla de inferencia, la LS utiliza un Modus Ponens con enunciados condicionales evaluadas mediante medidas condicionadas.

5. Referencias

- Dubois et al., 91 Dubois D., Lang J., Prade H. **Handling Uncertainty, Context, Vague Predicates, and Partial Inconsistency in Possibilistic Logic.** In *proceedings of the IJCAI'91. Fuzzy logic in AI. Workshop*, Australia.

- Dubois-Prade, 88 Dubois D., Prade H. **An Introduction to Possibility Theory and Fuzzy Logic.** In *Non-standard Logics for Automated Reasoning*, Smets et al. eds., Academic Press, New York, pp. 287-326
- Godo et al., 91a Godo L., Esteve F., Garcia P., Agusti-Cullell J. **A Formal Semantical Approach to Fuzzy Logic.** In *Proceedings of 21th International Symposium on Multiple-Valued Logic*. IEEE computer Society Press. Pp. 72-79.
- Jacas, 88 Jacas J. **On the generators of T-Indistinguishability operators.** In *Stochastica*, Vol. 12, 1, pp. 49-63.
- Niiniluoto, 87 Niiniluoto I. **Truthlikeness.** Syntheses Library, Vol 185. Riedel, Holland.
- Ruspini, 91a Ruspini E. H. **On the Semantics of Fuzzy Logic.** In *International Journal of Approximate Reasoning*, 5, pp. 45-88.
- Ruspini, 91b Ruspini E. H. **Truth as Utility: A Conceptual Synthesis.** In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann. pp. 316-322.
- Valverde, 85 Valverde L. **On the Structure of F-indistinguishability operators.** *Fuzzy Sets & Systems*, 17, pp. 313-328.
- Zadeh, 78 Zadeh L.A. **Fuzzy Sets as a Basis for a Theory of Possibility.** In *Fuzzy sets and Systems*, 1, pp. 3-28.

VALIDACION DE DATOS MEDIANTE LOGICA BORROSA EN SISTEMAS EXPERTOS EN TIEMPO REAL

Juan A. Aguilar, José M. Domínguez, Erik de Pablo y Xavier Alamán



Instituto de Ingeniería del Conocimiento
UAM Cantoblanco. Módulo C-XVI, planta 4
28049 MADRID
E-mail: ALAMAN at EMDCCII1.EARN

Resumen

Uno de los problemas que aparecen al desarrollar sistemas expertos en tiempo real on-line es el de la validación de datos procedentes de sensores. En este artículo se presenta un tratamiento del problema mediante técnicas de lógica borrosa, originado durante el desarrollo del proyecto MIP, un sistema experto en tiempo real para la ayuda al control de procesos industriales descrito en [Alamán et al. 92]. Este sistema experto está en funcionamiento desde marzo de 1991 en una planta petroquímica de INH-REPSOL en Tarragona. Como complemento al sistema se implementó un módulo específico de lógica borrosa para la validación de datos, cuya tarea es asignar coeficientes de confianza a las medidas procedentes de sensores.

1. Descripción del problema

Uno de los problemas más importantes que aparecen al desarrollar sistemas expertos en tiempo real on-line es la frecuente obtención de medidas erróneas debido a múltiples razones técnicas. Los expertos comprueban cuidadosamente la validez de todas las medidas antes de tomar cualquier decisión. Si un sistema experto ha de realizar el mismo trabajo, la validación es esencial: los datos erróneos han de ser detectados con rapidez y exactitud.

Otros tratamientos del problema de la validación de sensores pueden encontrarse en [Alexander et al. 89], [Chendrasekaran y Punch 87], [Fox et al. 83], [Leinweber y Gidwani 86], o [Rowan 89]. En todos ellos se mencionan los siguientes aspectos a tener en cuenta:

- La historia de cada variable ha de ser almacenada y procesada de alguna manera para obtener información sobre el grado de confianza que merece una determinada medida.
- La redundancia en la información es uno de las claves a la hora de validar medidas. Esto puede conseguirse usando varios sensores para medir la misma variable física o midiendo a la vez distintas variables físicas numéricamente relacionadas entre sí.
- Es importante distinguir entre comportamientos normales y anormales en cada sensor. Por ejemplo, si un sensor está midiendo la temperatura de una gran cantidad de materia, no serían normales variaciones bruscas en las medidas. Por otra parte, si el sensor está midiendo un flujo a través de una válvula y las medidas no cambian, se puede sospechar que son erróneas.

Además, hay dos fuentes de incertidumbre involucradas en el proceso de la medición: por una parte, la imprecisión en las medidas debido a las limitaciones técnicas de los sensores, y por otra, la inevitable presencia de fallos de los sensores o de la circuitería asociada debido a la naturaleza física del proceso (mal funcionamiento, degradación, etc.).

Teniendo en cuenta todas estas conclusiones se implementó en el sistema experto MIP un módulo específico para tratar el problema de la validación de datos mediante el empleo de lógica borrosa, lo que se explica a continuación.

2. Lógica borrosa y validación de datos

¿Por qué elegir lógica borrosa para tratar el problema de la validación de datos? Las razones para ello son las siguientes:

- El razonamiento borroso interpreta de forma natural la imprecisión en las medidas. Por ejemplo, los conceptos de "medidas parecidas" o una medida "que no cambia" son conceptos típicamente borrosos.
- Es fácil construir reglas sobre la confianza merecida por las medidas basándose en conceptos como "parecidas", "distintas", "evolucionan a la par", etc. Es importante señalar la importancia de establecer una correspondencia no lineal entre los comportamientos de las variables y la confianza que merecen. Por ejemplo, si una medida "cambia poco" presenta un comportamiento normal, pero si "no cambia absolutamente nada" entonces presenta un comportamiento anormal. Este tipo de situaciones serían intratables con correspondencia lineal.
- Existen motores de inferencia basados en esta clase de lógica en el mercado que cumplen los requisitos de eficiencia exigidos por las aplicaciones en tiempo real. En caso de necesidad, se podrían utilizar estos mismos motores implementados en hardware.
- Los sistemas basados en lógica borrosa son fáciles de ajustar. Esta característica es fundamental para sistemas que han de ser conectados y empleados en situaciones reales.

En nuestro sistema se ha utilizado una arquitectura de tipo modular. En lugar de tratar la validación como una parte más del trabajo del sistema experto, un módulo específico preprocesa los datos y determina coeficientes de confianza (números entre 0 y 1) para cada medida. El módulo ha sido implementado mediante un entorno de desarrollo de sistemas expertos basados en lógica borrosa que genera código C. Este código es absolutamente portable, logrando una perfecta integración del módulo con el resto del sistema.

Por tanto, el sistema experto toma como entrada no sólo las variables, sino que además toma simultáneamente sus coeficientes de confianza. De este modo cualquier razonamiento posterior puede tener en cuenta estos coeficientes de confianza. Esta arquitectura presenta una serie de ventajas, como:

- El problema de la diagnosis (fallos en el proceso) se separa claramente del problema de la validación (fallos en los sensores). Al tratar estos dos problemas de forma separada se mejora considerablemente la comunicación con los expertos en las reuniones de adquisición de conocimiento, lo que permite una representación más clara del conocimiento sobre la diagnosis del proceso.
- Estos dos pasos pueden llevarse a cabo simultáneamente, en paralelo, obteniendo mayor rapidez en el sistema. Esto es fundamental para un sistema que ha de trabajar en tiempo real.
- Este esquema es muy parecido al utilizado por los expertos: se comprueba primero la validez de las medidas y entonces se usan estos datos para continuar el razonamiento.
- Se facilita el mantenimiento del sistema.
- Cualquier otro módulo del sistema además del módulo experto puede utilizar estos coeficientes de confianza. Por ejemplo, un simulador puede elegir entre varias variables según sean sus coeficientes de confianza para efectuar un cálculo determinado.

Por otra parte, una desventaja del tratamiento propuesto es que la confianza de las medidas se calcula en términos de las propias medidas (y de su historia) pero no se tiene en cuenta ninguna evidencia procedente de cualquier razonamiento adicional. Por ejemplo, si el sistema llega a aseverar que cierta conclusión es errónea, parecería razonable sospechar de los datos que conducen a esa conclusión. Esta posibilidad no está contemplada en el sistema. Sin embargo, según nuestra experiencia en casos reales, en la mayor parte de las ocasiones los expertos emplean un esquema de razonamiento similar al propuesto en nuestro modelo.

3. Validación de sensores mediante lógica borrosa

Como se explicó en la sección anterior, hay un módulo que complementa al sistema MIP preprocesando los datos procedentes de sensores y asignando coeficientes de confianza a cada medida. Para calcular estos coeficientes, el módulo considera por un lado el comportamiento individual de cada sensor (cómo van evolucionando sus medidas) y por otro el comportamiento colectivo de conjuntos de sensores relacionados entre sí (grupos de sensores que miden la misma variable física o que miden distintas variables numéricamente relacionadas entre sí). Finalmente se funde toda la información conseguida en un sólo número, llamado coeficiente de confianza total, que queda asociado a la medida.

3.1. Comportamiento individual

Se procesa la historia de cada sensor, analizando diversas características de su comportamiento. Por ejemplo, qué valores toma, las diferencias de primer orden, etc. Para cada una de las características se construye un histograma y de éste se obtiene una distribución de posibilidad ([Dubois y Prade 89]). Estas distribuciones son interpretadas como la "definición borrosa de normalidad" para cada una de las características (véase fig. 1). Para analizar el comportamiento del sensor se calcula el grado de "normalidad" de cada una de las características anteriores, usando las distribuciones de posibilidad. De los diversos grados obtenidos, se llega a un coefi-

ciente de confianza individual final mediante una t-norma (por ejemplo, el mínimo). Este coeficiente de confianza individual representa la "normalidad" del comportamiento del sensor.

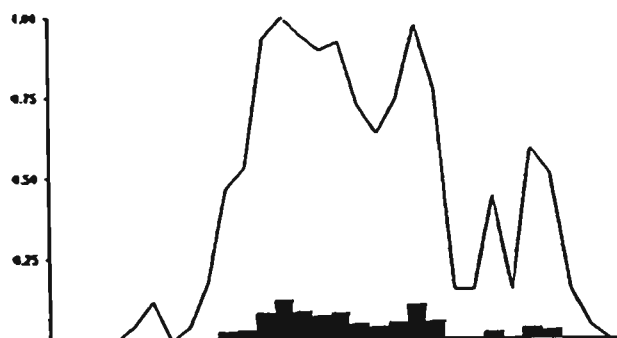


Fig. 1. Distribución de posibilidad construida a partir de un histograma.

¿Qué período de tiempo se ha de tener en cuenta para la obtención de las distribuciones de posibilidad? MIP genera automáticamente históricos de 96 horas de duración para cada medida. Las distribuciones de posibilidad obtenidas a partir de dos históricos consecutivos son muy parecidas (véase fig. 2). Si se toman históricos de cada mes se puede observar la evolución de las distribuciones a lo largo del tiempo (fig. 3). La solución adoptada ha sido tomar un histórico de 96 horas para calcular unas distribuciones iniciales y posteriormente ir las renovando cada 96 horas. Además, este proceso es llevado a cabo en paralelo con la asignación de coeficientes de confianza.

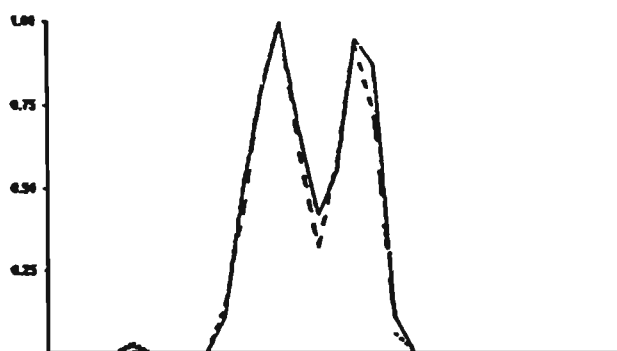


Fig. 2. Distribuciones de posibilidad para dos históricos consecutivos.



Fig. 3. Distribuciones de posibilidad en enero, febrero y marzo de 1992.

3.2. Comportamiento colectivo

Así como en el apartado anterior se tenían en cuenta solamente las medidas que procedían de un sensor, ahora se consideran conjuntos de sensores que miden variables relacionadas entre sí. Estas relaciones pueden ser empleadas para asignar más o menos confianza a las medidas, dependiendo del comportamiento "en conjunto" de todas ellas.

Supóngase que se tienen varios sensores, todos ellos midiendo la misma variable física. En este caso, se comparan las medidas por parejas y dependiendo del "grado de coincidencia" en sus valores y en la evolución de sus valores, se determinan coeficientes de confianza colectivos o de comparación. Este proceso puede repetirse para ternas, cuaternas, etc.

Por otra parte, supóngase que se tienen varios sensores midiendo a la vez distintas variables físicas relacionadas numéricamente entre sí. Ahora, se analiza la evolución de las diferentes medidas relacionadas y, dependiendo del "grado de concordancia" entre esta evolución y las relaciones entre las diversas variables medidas, se asigna un coeficiente de confianza colectivo a cada medida.

En ambos casos se define un conjunto de reglas para asignar los coeficientes de confianza colectivos empleando "grado de coincidencia" y "grado de concordancia" como variables lingüísticas con valores definidos por conjuntos borrosos ("parecida", "distinta", "constante", ...). Estas reglas representan a su vez el conocimiento disponible a partir del experto sobre el comportamiento colectivo de los sensores donde se tiene en cuenta las distintas características de precisión y sensibilidad de cada sensor. Por ejemplo, en el primer caso las reglas podrían ser:

- Si la medida A es parecida a la medida B, entonces la confianza colectiva es alta.
- Si la medida A es distinta a la medida B, entonces la confianza colectiva es baja.
- etc.

En el segundo caso, serían:

- Si la medida A aumenta y la medida B disminuye, la confianza colectiva para A es alta y para B es alta.
- Si las medidas A y B permanecen constantes, la confianza colectiva para A es baja y para B es media.
- etc.

3.3. Coeficiente de confianza final

Como resultado de lo anteriormente expuesto, se obtienen diversos coeficientes de confianza: uno individual y varios colectivos. El siguiente paso es refundir esta evidencia en un coeficiente de confianza final. Esto puede hacerse usando la siguiente función:

$$\sqrt{w_1\sigma_1^2 + w_2\sigma_2^2 + \dots + w_n\sigma_n^2}$$

donde α_i son los coeficientes de confianza obtenidos, y w_i son los pesos de cada coeficiente en la fusión final ($0 < w_i < 1$, $w_1 + \dots + w_n = 1$). En este paso se usa el criterio del experto, permitiendo a las peculiares características de cada tipo de sensor tener un impacto distinto en la fusión final: habrá sensores para los que el coeficiente más importante será el individual y habrá otros para los que importen más los coeficientes colectivos. Además empíricamente se prueba que es muy fácil ajustar la influencia de cada coeficiente de confianza en el resultado final.

La elección de esta función ha sido motivada porque presenta variaciones más suaves en su valor que una media ponderada lineal. Cuanto más cercanos a uno son los distintos coeficientes de confianza, más cercano a uno es el coeficiente final. Este valor se asocia a la medida, y puede así ser utilizado entonces por cualquier módulo del sistema en cualquier proceso posterior.

4. Ejemplos

En este apartado se van a estudiar dos casos reales tomados de la planta petroquímica para los que se ha llevado a la práctica todo lo explicado anteriormente. Además, ha de señalarse que estos ejemplos pueden ser fácilmente generalizados a otros conjuntos de medidas procedentes de sensores.

4.1. Exceso de oxígeno

Este ejemplo ilustra el caso de varios sensores que miden a la vez una misma variable física: el exceso de oxígeno producido en el reactor de la planta petroquímica.

El sistema experto MIP dispone de tres medidas del exceso de oxígeno. Esta es una variable crítica para el razonamiento. Dos de las medidas proceden de dos sensores con diferentes factores de sensibilidad y precisión (A y B) y otra medida procede de un análisis automático llevado a cabo por un espectrómetro de masas (C).

Para cada medida se obtienen tres coeficientes de confianza; uno individual y dos de comparación. El coeficiente de confianza individual se obtiene al comparar el comportamiento actual del sensor con su comportamiento anterior, como se explicó anteriormente.

Los coeficientes de confianza de comparación se obtienen al comparar dos sensores y decidiendo su "coincidencia". Si se "parecen", la confianza es ALTA. Si son "diferentes" la confianza es BAJA. El caso "no muy diferentes" necesita mayor elaboración. En este caso, se usan las diferencias de primer orden de los valores por medio de un conjunto de reglas tales como:

- Si el sensor A NO CAMBIA y el sensor B SUBE MUCHO, entonces la confianza de comparación AB es BAJA y la confianza de comparación BA es MEDIA.
- Si el sensor A SUBE y el sensor B BAJA, entonces la confianza de comparación AB es BAJA y la confianza de comparación BA es BAJA.

- etc.

Las reglas se diseñan de tal manera que cubren todos los casos, considerando siempre el conocimiento del experto sobre sensibilidad, precisión, etc. de cada sensor.

La fusión final de los coeficientes se hace mediante la función que aparecía en la sección anterior.

4.2. Evaporador de amoniaco

Este ejemplo ilustra el caso de varias medidas de variables numéricamente relacionadas entre sí. En el evaporador de amoniaco de la planta petroquímica considerada en el proyecto MIP hay tres medidas (temperatura, presión y nivel de amoniaco) que están relacionadas como sigue:

- Si el nivel desciende mucho, la temperatura tiene que aumentar mucho y la presión tiene que aumentar (el recíproco no es cierto).
- La evolución de la temperatura y la presión ha de ser similar.
- La temperatura debe cambiar al menos en la segunda cifra decimal.
- etc.

Este conocimiento se codifica en forma de reglas, representando con conjuntos borrosos conceptos como "desciende mucho", "aumenta", etc. Por ejemplo:

- Si el nivel DESCIEDE MUCHO y la temperatura AUMENTA MUCHO y la presión AUMENTA, entonces el coeficiente de confianza colectivo del nivel es ALTO, el de la temperatura es ALTO y el de la presión es ALTO.
- Si la temperatura DESCIEDE y la presión DESCIEDE, entonces el coeficiente de confianza colectivo de la temperatura es ALTO y el de la presión es ALTO.
- etc.

Como en el caso anterior, para cada sensor se calcula un coeficiente de confianza individual (por medio de las distribuciones de posibilidad para las distintas características de comportamiento) y un coeficiente de confianza colectivo (usando el conocimiento codificado en las reglas), obteniendo el coeficiente de confianza final de forma similar a la empleada en el ejemplo precedente.

5. Resultados y conclusiones

El método descrito en este artículo ha resultado ser bastante efectivo en todos los casos estudiados. Dos características que han influido decisivamente en su éxito son la redundancia de información y el fácil ajuste del sistema final, lo que permite corregir cualquier comportamiento no deseado fácil y rápidamente. Este método puede aplicarse a sensores que miden la misma variable o a sensores que miden variables numéricamente relacionadas.

En una primera etapa el módulo fue instalado en la versión off-line de MIP en julio de 1991, con el fin de probar su funcionamiento. Los resultados logrados fueron muy satisfactorios y el módulo se instaló en la planta en enero de 1992. En esta fase de validación en planta se han añadido otros conjuntos de variables. Lo obtenido en este proyecto se aplicará en el proyecto HINT, un proyecto ESPRIT que combinará diferentes técnicas de Inteligencia Artificial como redes neuronales, lógica borrosa, razonamiento basado en modelos, etc. dentro de una arquitectura integrada con el fin de mejorar la operación de control de procesos industriales.

Como trabajo a realizar en el futuro se puede señalar lo siguiente:

- ¿Es necesario filtrar los datos erróneos antes de calcular las distribuciones de posibilidad? Si así fuera, ¿cómo se podrían identificar automáticamente estos datos erróneos?
- ¿Sería interesante considerar también valores de utilidad? En otras palabras, si una medida es crítica, ¿cómo afecta este hecho al cálculo del coeficiente de confianza?

Actualmente se trabaja en la solución de estos problemas.

Referencias

[Alaman et al. 92] Xavier Alarnán, Soledad Romero, Carlos Aguirre, Pedro Serrahima, Rafael Muñoz, Vicente López, Jose Dorronsoro, Erik de Pablo, "Knowledge-Based Systems for Real-Time Process Control: The MIP Project". 1992 IFAC/IFIP/IMACS International Symposium on Artificial Intelligence in Real-Time Control, Delft, The Netherlands, June 1992.

[Alexander et al. 89] Alexander S. M., Vaidya C. M. and Kamel K. A., "An Architecture for Sensor Fusion in Intelligent Process Monitoring" *Computers Ind. Eng. Vol 16, No. 2*, pp. 307-311 1989.

[Chendrasekaran y Punch 87] Chendrasekaran, B. and Punch W. F., "Data Validation During Diagnosis, A Step Beyond Traditional Sensor Validation" *AAAI 1987*, pp. 778-782 1987.

[Dubois y Prade 89] Dubois, D. and Prade, H., "Fuzzy Sets, Probability and Measurement", *European Journal of Operational Research* 40, 135-154, 1989.

[Fox et al. 83] Mark S. Fox, Simon Lowenfeld and Pamela Kleinosky, "Techniques for Sensor-Based Diagnosis", *IJCAI 83*, pp. 158-163 1983.

[Leinweber y Gidwani 86] David Leinweber and Kumar Gidwani, "Real-Time Expert System Development Techniques and Applications", *Western Conference on K.B. Engineering and E.S.*, pp. 69-77, 1986.

[Rowan 89] Duncan A. Rowan, "On-Line Expert Systems in Process Industries", *AI EXPERT, Miller Ferman Publications*, August 1989.

ESTRUCTURAS ALGEBRAÍCAS PARA CONJUNTOS INTUICIONISTAS FUZZY

P. BURILLO LOPEZ; H. BUSTINCE SOLA

Departamento de Matemática e Informática, Universidad Pública de Navarra,
Pamplona, España.

Palabras clave: Conjunto fuzzy, conjunto intuicionista fuzzy, t-norma y t-conorma

0. Introducción

Como generalización de la noción del concepto de conjunto fuzzy, FS , propuesta por algunos investigadores en la década de los setenta ([4], [6]), se introduce el de conjunto fuzzy intervalo-valorado, $IVFS$. K. Atanassov dio en [1] la definición de conjunto fuzzy intuicionista, IFS , y posteriormente en [2], estableció que, en cierto modo, los $IVFS$ son una versión de los IFS , introduciendo en el mismo artículo el concepto de conjunto fuzzy intuicionista intervalo-valorado $IVIFS$.

En este trabajo proponemos un conjunto de aplicaciones entre los conjuntos: FS , IFS , $IVFS$ e $IVIFS$. Estas aplicaciones nos permitirán concluir que los IFS son un caso particular de los $IVIFS$ y, en cierto sentido, estos últimos comprenden a los demás.

Comenzaremos recordando algunas operaciones con subintervalos cerrados del intervalo $[0, 1]$. Posteriormente definiremos las t-normas intervalo-valoradas, que serán utilizadas para generalizar algunas operaciones en los distintos conjuntos, estudiando a la vez la estructura de estos últimos. Terminaremos dando algunas aplicaciones entre los conjuntos y los resultados que se obtienen de ellas.

I. Intervalos

Sea $D[0,1]$ el conjunto de todos los subintervalos cerrados del intervalo $[0,1]$; representaremos a los elementos de este conjunto con letras mayúsculas, $A, B, C...$ siendo $A = [A_L, A_U]$, donde A_L y A_U son el extremo inferior y superior respectivamente, teniéndose ([3]) $A = B \iff A_L = B_L \text{ y } A_U = B_U$ y $A \leq B \iff A_L \leq B_L \text{ y } A_U \leq B_U$ y denotaremos con $\bar{A}$ al intervalo *complementario* de A , es decir $\bar{A} = 1 - A = [1 - A_U, 1 - A_L]$, con 1 al intervalo $[1, 1]$ y con 0 al intervalo $[0, 0]$.

Sambuc R. ([2]) y C.Ponsard ([3]) definen en $D[0,1]$ las siguientes operaciones:

$$\forall A, B \in D[0,1], \wedge(A, B) \equiv [\wedge(A_L, B_L), \wedge(A_U, B_U)]$$

$$\forall A, B \in D[0,1], \vee(A, B) \equiv [\vee(A_L, B_L), \vee(A_U, B_U)]$$

Con lo cual $\{D[0,1], \wedge, \vee\}$ tiene estructura ([3]) de retículo distributivo, acotado y no complementado, en donde se satisfacen las leyes de Morgan.

También consideraremos en $D[0,1]$ otras operaciones:

$$\text{Suma probabilística, } A \hat{+} B \equiv [A_L \hat{+} B_L, A_U \hat{+} B_U]$$

$$\text{Producto ([2]), } A \cdot B \equiv [A_L \cdot B_L, A_U \cdot B_U]$$

$$\text{Suma ([2]), } A + B \equiv [\wedge(A_L + B_L, 1), \wedge(A_U + B_U, 1)]$$

$$A \sqcap B \equiv [\vee(0, A_L + B_L - 1), \vee(0, A_U + B_U - 1)]$$

Se comprueba fácilmente que estas operaciones son internas y cumplen las propiedades conmutativa y asociativa.

II. T-normas y t-conormas intervalo-valoradas.

Como consecuencia de las bien conocidas t-normas y t-conormas triangulares en $[0,1]$ damos la siguiente definición.

Definición 1. Llamaremos t-norma intervalo-valorada a toda aplicación

$$T_{iv} : D[0,1] \times D[0,1] \longrightarrow D[0,1]$$

satisfaciendo las propiedades

- i) Condiciones frontera, $T_{iv}(A, 1) = A$ y $T_{iv}(A, 0) = 0 \quad \forall A \in D[0, 1]$
- ii) Monotonía, $T_{iv}(A, B) \leq T_{iv}(C, D)$ si $A \leq C$ y $B \leq D$
- iii) Conmutativa, $T_{iv}(A, B) = T_{iv}(B, A) \quad \forall A, B \in D[0, 1]$
- iv) Asociativa, $T_{iv}(T_{iv}(A, B), C) = T_{iv}(A, T_{iv}(C, D)) \quad \forall A, B, C \in D[0, 1]$

Dada una t-norma intervalo-valorada T_{iv} , podemos considerar la aplicación

$$S_{iv} : D[0, 1] \times D[0, 1] \longrightarrow D[0, 1]$$

$$S_{iv}(A, B) \equiv 1 - T_{iv}(1 - A, 1 - B)$$

a S_{iv} la llamaremos t-conorma intervalo-valorada (o dual de T_{iv}).

Obviamente S_{iv} satisface las propiedades

- i) $S_{iv}(A, 0) = A$, $S_{iv}(A, 1) = 1$
- ii) $S_{iv}(A, B) \leq S_{iv}(C, D)$ si $A \leq C$ y $B \leq D$
- iii) $S_{iv}(A, B) = S_{iv}(B, A)$
- iv) $S_{iv}(S_{iv}(A, B), C) = S_{iv}(A, S_{iv}(B, C))$

Los ejemplos más significativos de t-normas intervalo-valoradas y su correspondiente t-conorma dual son los siguientes

$$i) T_{iv}(A, B) = \wedge(A, B)$$

$$S_{iv}(A, B) = 1 - T_{iv}(1 - A, 1 - B) = 1 - \wedge(1 - A, 1 - B) =$$

$$= 1 - \wedge([1 - A_U, 1 - A_L], [1 - B_U, 1 - B_L]) =$$

$$= 1 - ([\wedge(1 - A_U, 1 - B_U), \wedge(1 - A_L, 1 - B_L)])$$

$$= [1 - \wedge(1 - A_L, 1 - B_L), 1 - \wedge(1 - A_U, 1 - B_U)], \text{ considerando las posibles desigualdades}$$

entre A_L, A_U y B_L, B_U resulta que $S_{iv}(A, B) = \vee(A, B)$

$$ii) T_{iv}(A, B) = A.B$$

$$S_{iv}(A, B) = 1 - T_{iv}(1 - A, 1 - B) = 1 - T_{iv}([1 - A_U, 1 - A_L], [1 - B_U, 1 - B_L]) =$$

$$1 - [(1 - A_U)(1 - B_U), (1 - A_L)(1 - B_L)] = [1 - (1 - A_L)(1 - B_L), 1 - (1 - A_U)(1 - B_U)] =$$

$$= [A_L + B_L - A_L B_L, A_U + B_U - A_U B_U] = [A_L \hat{+} B_L, A_U \hat{+} B_U] = A \hat{+} B$$

De la misma forma se puede comprobar que

$$\text{iii) } T_{iv}(A, B) = A + B$$

$$S_{iv}(A, B) = \sqcap(A, B)$$

son t-norma y t-conorma dual, respectivamente.

III. Conjuntos, operaciones y estructura.

Describiremos a continuación los diferentes conjuntos que utilizaremos, dando además algunos resultados operacionales con ellos.

Definición 2. ([1]). Conjunto intuicionista fuzzy

Sea $E \neq \emptyset$ un conjunto dado. Un conjunto intuicionista fuzzy sobre E es una expresión A dada por

$$A = \{(x, \mu_A(x), \nu_A(x)) | x \in E\}$$

donde

$$\mu_A : E \longrightarrow [0, 1]$$

$$\nu_A : E \longrightarrow [0, 1]$$

verifican $\mu_A(x) + \nu_A(x) \leq 1 \quad \forall x \in E$

Las funciones μ_A y ν_A reciben el nombre de funciones de pertenencia y no pertenencia respectivamente. Denotaremos con IFS al conjunto de todos los conjuntos intuicionistas fuzzy sobre E .

$\forall A, B \in IFS$ se definen ([1])

1. $A \leq B \iff \mu_A(x) \leq \mu_B(x) \text{ y } \nu_A(x) \geq \nu_B(x) \quad \forall x \in E$
2. $A = B \iff A \leq B \text{ y } B \leq A$

$$3. A_c = \{(x, \nu_A(x), \mu_A(x)) | x \in E\}$$

Teorema 1: Sean T y S t-norma y t-conorma dual en $[0, 1]$, construimos las expresiones

$$T(A, B) = \{(x, T(\mu_A(x), \mu_B(x)), S(\nu_A(x), \nu_B(x))) | x \in E\}$$

$$S(A, B) = \{(x, S(\mu_A(x), \mu_B(x)), T(\nu_A(x), \nu_B(x))) | x \in E\} \quad \forall A, B \in IFS.$$

Entonces se verifica

a) ([1]) Si $S = \vee$ y $T = \wedge$ entonces $\{IFS, \wedge, \vee\}$ es un retículo distributivo, acotado, no complementado que satisface las leyes de Morgan.

b) Para S y T cualesquiera, se satisfacen las propiedades conmutativa, asociativa y $S(A_c, B_c) = (T(A, B))_c$, $T(A_c, B_c) = (S(A, B))_c$.

Demostración. Probamos b). La conmutativa y asociativa para $T(A, B)$ y $S(A, B)$ son evidentes por la propia definición de T y S . Además $S(A_c, B_c) = (T(A, B))_c$ pues

$$S(A_c, B_c) = \{(x, S(\nu_A(x), \nu_B(x)), T(\mu_A(x), \mu_B(x))) | x \in E\}$$

$$T(A, B) = \{(x, T(\mu_A(x), \mu_B(x)), S(\nu_A(x), \nu_B(x))) | x \in E\}$$

$$(T(A, B))_c = \{x, S(\nu_A(x), \nu_B(x)), T(\mu_A(x), \mu_B(x)) | x \in E\} = S(A_c, B_c)$$

De manera análoga se demuestra que $T(A_c, B_c) = (S(A, B))_c$. c.q.d.

Definición 3. ([3]) Conjuntos fuzzy intervalo-valorados.

Sea $E \neq \emptyset$ un conjunto dado. Un conjunto fuzzy intervalo-valorado sobre E es una expresión A dada por

$$A = \{(x, M_A(x)) | x \in E\}$$

donde

$$M_A : E \longrightarrow D[0, 1]$$

es una función llamada de pertenencia. Denotaremos con $IVFS$ al conjunto de todos los conjuntos fuzzy intervalo-valorados sobre el mismo E .

$\forall A, B \in IVFS$ se definen ([3])

1. $A \leq B \iff M_{AL}(x) \leq M_{BL}(x) \text{ y } M_{AU}(x) \leq M_{BU}(x) \forall x \in E$
2. $A = B \iff M_{AL}(x) = M_{BL}(x) \text{ y } M_{AU}(x) = M_{BU}(x) \forall x \in E$
3. $A_c = \{(x, \bar{M}_A(x)) | x \in E\}$

Teorema 2. Sean T_{iv} y S_{iv} t-norma y t-conorma dual dadas en la definición 1, construimos las expresiones

$$T_{iv}(A, B) \equiv \{(x, T_{iv}(M_A(x), M_B(x))) | x \in E\}$$

$$S_{iv}(A, B) \equiv \{(x, S_{iv}(M_A(x), M_B(x))) | x \in E\} \quad \forall A, B \in IVFS.$$

Entonces se verifica:

a) Si $T_{iv} = \wedge$ y $S_{iv} = \vee$ entonces $\{IVFS, \wedge, \vee\}$ es un retículo distributivo, acotado no complementado que satisface las leyes de Morgan.

b) Para T_{iv} y S_{iv} cualesquiera, se satisfacen las propiedades conmutativa, asociativa y $T_{iv}(A_c, B_c) = (S_{iv}(A, B))_c$, $S_{iv}(A_c, B_c) = (T_{iv}(A, B))_c$.

Demostración. La parte a) es consecuencia directa de que $\{D[0, 1], \wedge, \vee\}$ es un retículo distributivo, acotado, no complementado que satisface las leyes de Morgan.

Para probar el apartado b), basta recordar que T_{iv} y S_{iv} son operaciones conmutativas y asociativas: además

$$S_{iv}(A, B) = \{(x, S_{iv}(M_A(x), M_B(x))) | x \in E\}$$

$$(S_{iv}(A, B))_c = \{(x, 1 - S_{iv}(M_A(x), M_B(x))) | x \in E\} =$$

$$= \{(x, T_{iv}(1 - M_A(x), 1 - M_B(x))) | x \in E\} =$$

$$= \{(x, T_{iv}(\bar{M}_A(x), \bar{M}_B(x))) | x \in E\} = T_{iv}(A_c, B_c).$$

De forma similar se comprueba que $S_{iv}(A_c, B_c) = (T_{iv}(A, B))_c$. c.q.d.

Definición 4. ([2]). Conjuntos fuzzy intuicionistas intervalo-valorados.

Sea $E \neq \emptyset$ un conjunto dado. Un conjunto fuzzy intuicionista intervalo-valorado sobre E es una expresión dada por

$$A = \{(x, M_A(x), N_A(x)) | x \in E\}$$

donde

$$M_A : E \longrightarrow D[0, 1]$$

$$N_A : E \longrightarrow D[0, 1]$$

verifican $M_{AU}(x) + N_{AU}(x) \leq 1 \quad \forall x \in E$

Las funciones M_A y N_A reciben el nombre de funciones de pertenencia y de no pertenencia respectivamente. Denotaremos con *IVIFS* al conjunto de todos los conjuntos fuzzy intuicionistas intervalo-valorados sobre el mismo E .

$\forall A, B \in IVIFS$, se definen ([2])

1. $A \leq B \iff M_A(x) \leq M_B(x) \text{ y } N_A(x) \geq N_B(x) \quad \forall x \in E$
2. $A = B \iff M_A(x) = M_B(x) \text{ y } N_A(x) = N_B(x) \quad \forall x \in E$
3. $A_c = \{(x, N_A(x), M_A(x)) | x \in E\}$

Teorema 3. Sean T_{iv} y S_{iv} t-norma y t-conorma dual dadas en la definición 1, construimos

$$T_{iv}(A, B) = \{(x, T_{iv}(M_A(x), M_B(x)), S_{iv}(N_A(x), N_B(x))) \in E\}$$

$$S_{iv}(A, B) = \{(x, S_{iv}(M_A(x), M_B(x)), T_{iv}(N_A(x), N_B(x))) \in E\}. \quad \forall A, B \in IVIFS$$

Entonces se verifica

a) Si $T_{iv} = \wedge$ y $S_{iv} = \vee$ entonces, $\{IVIFS, \wedge, \vee\}$ es un retículo distributivo acotado y no complementado que satisface las leyes de Morgan.

b) Para T_{iv} y S_{iv} cualesquiera, se satisfacen las propiedades conmutativa, asociativa y $T_{iv}(A_c, B_c) = (S_{iv}(A, B))_c$, $S_{iv}(A_c, B_c) = (T_{iv}(A, B))_c$

Demostración. El apartado a) es fácilmente comprobable.

La demostración de b) es análoga a las realizadas en los teoremas 1 y 2. c.q.d.

IV. Algunas aplicaciones entre los conjuntos: $FS, IFS, IVFS$ e $IVIFS$

K. Atanassov demostró en ([2]) que la aplicación

$$f : IVFS \longrightarrow IFS$$

$$A = \{(x, M_A(x)) | x \in E\} \longrightarrow f(A) = \{(x, M_{AL}(x), 1 - M_{AU}(x)) | x \in E\}$$

es un isomorfismo. Nuestro objetivo es profundizar en la consideración de otras aplicaciones entre los conjuntos $FS, IFS, IVFS$ e $IVIFS$.

Desde luego, a cada elemento $g \in D[0, 1]^{D[0,1] \times D[0,1]}$ se le puede asociar una aplicación θ_g de $IVIFS$ en $IVFS$ dada por

$$\theta_g : IVIFS \longrightarrow IVFS$$

$$A = \{(x, M_A(x), N_A(x)) | x \in E\} \longrightarrow \theta_g(A) = \{(x, g(M_A(x), N_A(x))) | x \in E\}$$

En lo sucesivo utilizaremos las funciones g siguientes

$$g_1([a, b], [c, d]) = [a, 1 - d]$$

$$g_2([a, b], [c, d]) = [a, 1 - c]$$

$$g_3([a, b], [c, d]) = [b, 1 - d]$$

$$g_4([a, b], [c, d]) = [b, 1 - c]$$

$\forall [a, b], [c, d] \in D[0, 1]$ y sus correspondientes θ_g inducidas.

Notemos que g_j con $j = 1, \dots, 4$, verifican dos sencillas propiedades que más adelante justificarán su elección.

1. Con el orden en $D[0, 1] \times D[0, 1]$ inducido por el orden en $D[0, 1]$ dado en la primera parte, g_j ($j = 1, \dots, 4$) son crecientes.

2. Si $a, c \in D[0, 1]$ entonces $g_j([a, a], [c, c]) = [a, 1 - c]$, $j = 1, \dots, 4$

Observemos que las funciones asociadas θ_g se describen de la forma siguiente

$$\theta_{g_1}(A) = \{(x, [M_{AL}(x), 1 - N_{AU}(x)]) | x \in E\}$$

$$\theta_{g_2}(A) = \{(x, [M_{AL}(x), 1 - N_{AL}(x)]) | x \in E\}$$

$$\theta_{g_3}(A) = \{(x, [M_{AU}(x), 1 - N_{AU}(x)]) | x \in E\}$$

$$\theta_{g_4}(A) = \{(x, [M_{AU}(x), 1 - N_{AL}(x)]) | x \in E\}$$

$$\forall A \in IVIFS$$

Teorema 4. Si $\forall A, B \in IVIFS$ tomamos $T_{iv}(A, B)$ y $S_{iv}(A, B)$ como sigue

$$\begin{aligned} T_{iv}(A, B) &= \{(x, T_{iv}(M_A(x), M_B(x)), S_{iv}(N_A(x), N_B(x))) | x \in E\} = \\ &= \{(x, [T(M_{AL}(x), M_{BL}(x)), T(M_{AU}(x), M_{BU}(x))], \\ &\quad [S(N_{AL}(x), N_{BL}(x)), S(N_{AU}(x), N_{BU}(x))]) | x \in E\} \\ S_{iv}(A, B) &= \{(x, S_{iv}(M_A(x), M_B(x)), T_{iv}(N_A(x), N_B(x))) | x \in E\} = \\ &= \{(x, [S(M_{AL}(x), M_{BL}(x)), S(M_{AU}(x), M_{BU}(x))], \\ &\quad [T(N_{AL}(x), N_{BL}(x)), T(N_{AU}(x), N_{BU}(x))]) | x \in E\} \end{aligned}$$

donde T y S son t-norma y t-conorma en $[0, 1]$. Entonces

$$i) \theta_{g_j}(S_{iv}(A, B)) = S_{iv}(\theta_{g_j}(A), \theta_{g_j}(B)) \quad .j = 1, \dots, 4$$

$$ii) \theta_{g_j}(T_{iv}(A, B)) = T_{iv}(\theta_{g_j}(A), \theta_{g_j}(B)) \quad .j = 1, \dots, 4$$

es decir, θ_{g_j} es un homomorfismo entre las estructuras $IVIFS$ e IFS con las operaciones dadas.

Demostración. La desarrollamos para $j = 2$, los demás casos se hacen de forma similar.

i)

$$\begin{aligned} \theta_{g_2}(S_{iv}(A, B)) &= \{(x, [S(M_{AL}(x), M_{BL}(x)), 1 - T(N_{AU}(x), N_{BU}(x))]) | x \in E\} = \\ &= \{(x, [S(M_{AL}(x), M_{BL}(x)), S(1 - N_{AU}(x), 1 - N_{BU}(x))]) | x \in E\} \end{aligned}$$

$$\theta_{g_2}(A) = \{(x, [M_{AL}(x), 1 - N_{AU}(x)]) | x \in E\}$$

$$\theta_{g_2}(B) = \{(x, [M_{BL}(x), 1 - N_{BU}(x)]) | x \in E\}$$

$$S_{iv}(\theta_{g_2}(A), \theta_{g_2}(B)) = \{(x, [S(M_{AL}(x), M_{BL}(x)), S(1 - N_{AU}(x), 1 - N_{BU}(x))])\}$$

ii)

$$\theta_{g_2}(T_{iv}(A, B)) = \{(x, [T(M_{AL}(x), M_{BL}(x)), 1 - S(N_{AU}(x), N_{BU}(x))]) | x \in E\}$$

$$= \{(x, [T(M_{AL}(x), M_{BL}(x)), T(1 - N_{AU}(x), 1 - N_{BU}(x))]) | x \in E\}$$

$$T_{iv}(\theta_{g_2}(A), \theta_{g_2}(B)) = \{(x, T_{iv}([M_{AL}(x), 1 - N_{AU}(x)], [M_{BL}(x), 1 - N_{BU}(x)]))\}$$

$$= \{(x, [T(M_{AL}(x), M_{BL}(x)), T(1 - N_{AU}(x), 1 - N_{BU}(x))]) | x \in E\} = \theta_{g_2}(T_{iv}(A, B))$$

Desde luego, a cada elemento $g \in D[0, 1]^{D[0, 1] \times D[0, 1]}$ se le puede asociar una aplicación θ_g de *IVIFS* en *IVFS* dada por

$$\Psi\psi : IFS \longrightarrow FS$$

$$A = \{(x, \mu_A(x), \nu_A(x)) | x \in E\} \longrightarrow \Psi\psi(A) = \{(x, \psi(\mu_A(x), \nu_A(x)) | x \in E\}$$

En lo sucesivo utilizaremos las funciones ψ siguientes

$$\psi_1(a, b) = a \quad a, b \in [0, 1]$$

$$\psi_2(a, b) = 1 - b \quad a, b \in [0, 1]$$

y sus correspondientes Ψ_ψ inducidas.

Observemos que las funciones Ψ_ψ se describen por

$$\Psi_{\psi_1}(A) = \{(x, \mu_A(x)) | x \in E\}$$

$$\Psi_{\psi_2}(A) = \{(x, 1 - \nu_A(x)) | x \in E\}$$

Teorema 5.

$$i) \Psi_{\psi_j}(S(A, B)) = S(\Psi_{\psi_j}(A), \Psi_{\psi_j}(B))$$

$$ii) \Psi_{\psi_j}(T(A, B)) = T(\Psi_{\psi_j}(A), \Psi_{\psi_j}(B))$$

con $j = 1, 2$ y S, T t-conorma y t-norma en $[0, 1]$.

Luego Ψ_{ψ_j} es un homomorfismo entre las estructuras IFS y FS con los operadores S y T .

Demostración. Tomaremos $j = 2$, teniendo en cuenta que para $j = 1$ la demostración se hace de la misma forma.

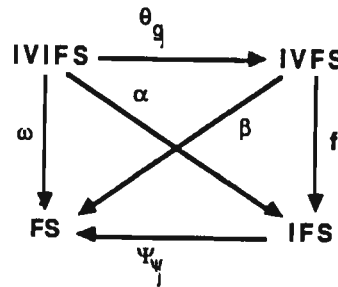
i)

$$\begin{aligned} S(A, B) &= \{(x, S(\mu_A(x), \mu_B(x)), T(\nu_A(x), \nu_B(x))) | x \in E\} = \\ &= \{(x, S(\mu_A(x), \mu_B(x)), 1 - S(1 - \nu_A(x), 1 - \nu_B(x))) | x \in E\} \\ \Psi_{\psi_2}(S(A, B)) &= \{(x, S(1 - \nu_A(x), 1 - \nu_B(x))) | x \in E\} = S(\Psi_{\psi_2}(A), \Psi_{\psi_2}(B)) \end{aligned}$$

ii)

$$\begin{aligned} T(A, B) &= \{(x, T(\mu_A(x), \mu_B(x)), S(\nu_A(x), \nu_B(x))) | x \in E\} = \\ &= \{(x, T(\mu_A(x), \mu_B(x)), 1 - T(1 - \nu_A(x), 1 - \nu_B(x))) | x \in E\} \\ \Psi_{\psi_2}(T(A, B)) &= \{(x, T(1 - \nu_A(x), 1 - \nu_B(x))) | x \in E\} = T(\Psi_{\psi_2}(A), \Psi_{\psi_2}(B)) \end{aligned}$$

Como conclusión a los teorema 4 y 5 se obtiene el diagrama conmutativo



en donde $\theta_{g_j}, \omega, \alpha, \beta, \Psi_{\psi_j}$, son homomorfismos y f isomorfismo.

V.Observaciones

1. $IVFS$ e IFS son estructuras isomorfas.
2. La elección de g_j ($j = 1, \dots, 4$), permite escribir

$$\alpha([a, a], [b, b]) = \{(x, a, b) | x \in E\}$$

expresión que indica que los IFS son un caso particular de los $IVIFS$. Esta observación justifica, en nuestro contexto, la elección hecha para g_j , puesto que, en definitiva, permite escribir $IFS \subset IVIFS$.

3. Se observa que $FS \subset IVFS$ sin más que considerar $f(FS) \subset IFS$.

En definitiva se obtiene

$$f(FS) \subset IFS \subset IVIFS$$

pudiendo afirmarse que el concepto de $IVIFS$ es el más amplio y comprende, en cierto sentido, a todos los demás.

Bibliografía:

- [1] ATANASSOV K., *Intuitionistic fuzzy sets*, Fuzzy Sets and Systems. 1986.
- [2] ATANASSOV K., *Interval valued intuitionistic fuzzy sets*, Fuzzy Sets and Systems. 1989.
- [3] GORZALCZANY M., *A method of inference in approximate reasoning based on interval-valued fuzzy sets*, Fuzzy Sets and Systems 21, 1-17, 1987.
- [4] GRATTAN-GUINNESS, *Fuzzy membership mapped into intervals and many-valued quantities*, Ztsch.math.Logik, Grundlehren Math., 22. 149-160, 1979.
- [5] PONSARD C., *Hirarchie des places centrales et graphes ϕ -flous*. Environment and planning A, 1977.
- [6] SAMBUC R., *Functions ϕ -flous. Application a l'aide au diagnostic en pathologie thyroïdienne*. Thèse. Université de Marseille, 1975.

ALGORITMOS GENETICOS CON FUNCION DE EVALUACION DIFUSA Y APLICACIONES

J.L. Castro, M. Delgado,
F. Herrera, J.L. Verdegay

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071-Granada, España

jcastro@ugr.es, mdelgado@ugr.es,
fherrera@ugr.es, jverdegay@ugr.es

Resumen: En el presente trabajo se realiza un estudio sobre la aplicación de los algoritmos genéticos a ciertos problemas cuando los individuos de la población están evaluados mediante números difusos. Se propone un método para la obtención de las probabilidades de selección a partir de la evaluación difusa utilizando funciones ordenadoras lineales sobre números difusos. Finalmente se presentan algunas aplicaciones de los algoritmos genéticos con función de evaluación difusa sobre las que se discute su uso.

Palabras clave: Algoritmos genéticos, números difusos, funciones ordenadoras.

1. Introduccion

Los algoritmos genéticos (AG) fueron desarrollados por John Holland, quien proporcionó los fundamentos teóricos de los mismos en su libro "Adaptation in Natural and Artificial Systems" (1975) [13]. Son procedimientos adaptativos que encuentran soluciones en espacios complejos inspirados por la evolución biológica, esto es, por un proceso de evolución basado en la selección natural.

Existe una gran cantidad de problemas, entre los que se incluyen algunos de optimización, para los cuales no existe un buen algoritmo que resuelva los mismos. En muchos de estos casos sería deseable disponer de métodos de resolución que nos proporcionen "buenas" soluciones si éstas pueden ser obtenidas con rapidez. Los AG nos permiten obtener buenas soluciones para muchos de estos problemas.

Un AG parte de una población aleatoria inicial de candidatos a la solución optimal y evoluciona hacia la mejor solución en forma de selección. En cada generación las soluciones relativamente buenas se reproducen, y las soluciones relativamente malas mueren para ser reemplazadas por otras con las características de buenas soluciones. Para distinguir entre las soluciones se utiliza una función de evaluación o función objetivo que permita distinguir las soluciones relativamente buenas de las malas para el problema tratado. Son muy simples desde el punto de vista computacional y potentes en la búsqueda del óptimo, no estando limitados por restricciones en el espacio de búsqueda. [12,13].

Los AG han sido muy utilizados para resolver problemas de optimización. Cuando tratamos con problemas de optimización difusa la evaluación o valoración de una solución factible puede ser un número difuso. Si queremos utilizar AG para resolver este tipo de problemas, parece razonable utilizar funciones de evaluación difusas. El propósito de este trabajo es estudiar el uso de éstas funciones en AG, así como su aplicación a algunos problemas.

Comenzamos con una introducción a los algoritmos genéticos. A continuación se definen las funciones de evaluación difusas para los mismos, y por último se estudian algunas aplicaciones de los AG con función de evaluación difusa.

2. Fundamentos de los Algoritmos Genéticos

Aunque existen muchas posibles variantes de AG, los mecanismos fundamentales bajo los cuales funcionan son: operar sobre una población de individuos, que inicialmente se genera de forma aleatoria, y cambiar en cada iteración la población atendiendo a los tres siguientes pasos: (1) evaluación de los individuos de la población, (2) selección de un nuevo conjunto de individuos, reproducción, sobre la base de su relativa conveniencia o adaptación y (3) recombinación para formar una nueva población a partir de los operadores de cruce y mutación. Los individuos resultantes de estas tres operaciones forman la siguiente población, iterando este proceso hasta que el sistema cese de mejorar.

Generalmente, cada individuo de la población se representa mediante una cadena binaria de longitud fija que codifica los valores de las variables que intervienen en el problema, con lo cual las representaciones de los datos (cadenas de bits), y las operaciones sobre ellas pueden manipularse para generar cadenas de bits que están bien adaptadas al problema a resolver.

Durante la iteración t , el AG mantiene una población $P(t)$ de soluciones $x_1^t, \dots, x_N^t$ (el tamaño de la población permanece fijo, N). Cada solución, x_1^t se evalúa por una función E , $E(x_1^t)$ que nos da una medida de conveniencia de la misma. En la siguiente iteración ($t+1$) se forma una nueva población en base a las operaciones (2) y (3).

En la figura 1 se muestra la estructura de un algoritmo genético y en las figuras 2.1, 2.2., 2.3 se ilustran las operaciones básicas de reproducción, cruce y mutación:

Procedimiento Algoritmo Genético

```

Comienzo (1)
   $t = 0$ 
  inicializar  $P(t)$ 
  evaluar  $P(t)$ 
  Mientras (no se de la condición de parada) hacer
    Comienzo (2)
       $t = t + 1$ 
      seleccionar  $P(t)$  desde  $P(t-1)$ 
      recombinar  $P(t)$ 
      evaluar  $P(t)$ 
    Final (2)
  Final (1)
  
```

Fig.1.: Algoritmo Genético

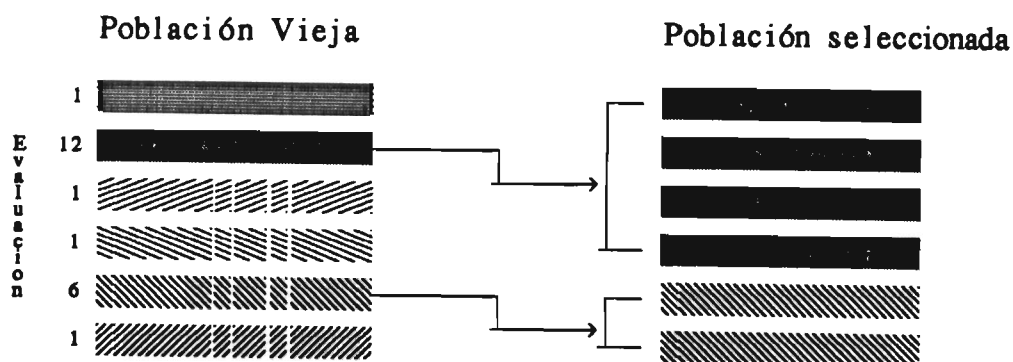


Fig.2.1.: Generación y Reproducción

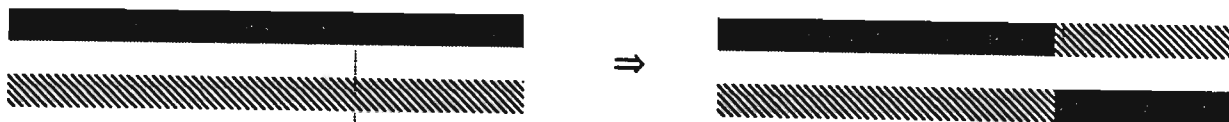


Fig. 2.2.: Cruce sobre un punto

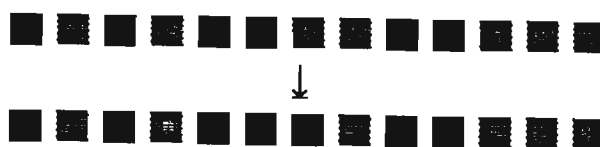


Fig. 2.3.: Mutación

Asociadas a los operadores genéticos de cruce y mutación existen la probabilidad de cruce, P_c , y la probabilidad de mutación, P_m . La primera es la probabilidad de aplicar el operador de cruce sobre parejas de cadenas obtenidas en la reproducción, la segunda es la probabilidad de aplicar el operador de mutación sobre los caracteres de cada cadena.

Generalmente se acepta que un AG debe tener las siguientes 5 componentes:

1. Una representación genética de las soluciones del problema, es decir, representación de las variables que intervienen en cadenas de bit,
2. una forma de crear una población inicial de soluciones,
3. una función de evaluación en términos de conveniencia de la solución evaluada,
4. operadores genéticos que cambien la composición de los descendientes,
5. valores para los parámetros utilizados por los AG (N , P_c , P_m , etc.).

Un esquema es un patrón de similitud que describe un subconjunto de cadenas con similitudes en ciertas posiciones de las cadenas. Aunque los AG solo procesan N estructuras en cada generación, también procesan N^3 esquemas. Esta es una importante propiedad que recibe el nombre de paralelismo implícito [12,13].

3. Algoritmos Genéticos con Funciones de Evaluación Difusas

Sea S una población de N individuos,

$$S = (s_1, s_2, \dots, s_N). \quad (1)$$

La función de evaluación $E(\cdot)$ para cada individuo s_i nos da un valor de conveniencia o adaptación de dicho individuo, $E(s_i)$, utilizado para determinar una probabilidad de selección P_{s_i} que se expresa de la siguiente forma

$$P_{s_i} = E(s_i) / \left(\sum_{i=1}^N E(s_i) \right), i = 1, \dots, N. \quad (2)$$

Como anteriormente hemos indicado, en muchos problemas podemos tener falta de precisión asociada a los elementos del mismo. Esta imprecisión en el problema puede llevar asociada imprecisión en la valoración de un individuo. Así, la conveniencia se puede describir adecuadamente como una propiedad difusa, y ser representada mediante un número difuso. Y la función de evaluación estaría definida desde S al conjunto de números difusos $F(R)$, $E: S \longrightarrow F(R)$, $E(s_i) \in F(R)$.

A partir de estos números difusos es necesario definir la probabilidad de

selección asociada a cada individuo para obtener una nueva población. Podríamos considerar una función $f:F(R) \longrightarrow R$, que nos permitiese definir la probabilidad de selección a partir de la función de evaluación difusa de acuerdo con (2), de la siguiente forma:

$$P_{s_i} = \frac{f(E(s_i))}{f(\sum_{i=1}^N E(s_i))} \quad (3)$$

seleccionando la función $f:F(R) \longrightarrow R$ tal que el conjunto $\{P_{s_i}\}_{i=1,\dots,N}$ verifique que es una distribución de probabilidad, y además sea una probabilidad de selección que permita reproducir las cadenas mas fuertes. Para ello vamos a introducir las siguientes condiciones sobre la función f :

1) Que preserve la suma

$$f\left(\sum_{i=1}^N E(s_i)\right) = \sum_{i=1}^N f(E(s_i)) \quad (4)$$

condición necesaria para que $\sum_{i=1}^N P_{s_i} = 1$.

2) Que preserve el orden

$$E(s_k) \leq E(s_j) \iff f(E(s_k)) \leq f(E(s_j)) \quad (5)$$

condición necesaria para que sea una probabilidad de selección que permita la reproducción de las cadenas más fuertes, ya que con esta propiedad, tenemos que los individuos más fuertes, es decir, mejor adaptados al problema, tienen mayor probabilidad de selección asociada,

$$E(s_k) \leq E(s_j) \iff P_{s_k} \leq P_{s_j}.$$

Una importante clase de funciones que verifican estas condiciones son las funciones ordenadoras lineales [4,5,7]. Dado $A \in F(R)$ y notando por $A^\alpha = [a_1(\alpha), a_2(\alpha)]$, $\alpha \in [0,1]$, un α -corte del número difuso A , una función ordenadora lineal se puede obtener de las siguientes formas:

a) Funciones ordenadoras basadas en un solo α -corte.

Dados $A \in F(R)$, $\lambda, \alpha \in [0,1]$. Se define

$$f(A) = \lambda a_1(\alpha) + (1-\lambda)a_2(\alpha).$$

La definición contiene dos parámetros, los cuales depende del decisor: α , que es un grado de comparación, y λ , que es un nivel de optimismo-pesimismo en la comparación.

b) Funciones ordenadores basadas en todos los α -cortes.

Dados $A \in F(R)$, $\lambda \in [0,1]$ y $P(\cdot)$ una medida aditiva sobre $[0,1]$. Se define

$$f(A) = \int_0^1 (\lambda a_1(\alpha) + (1-\lambda)a_2(\alpha)) dP(\alpha).$$

Como puede verse, solo tiene un parámetro, λ , el cual actúa igualmente como un grado de optimismo-pesimismo.

Esta definición general de $f(A)$ fué propuesto en [7,11], donde también se mostro que es una función ordenadora lineal. En particular, para números difusos triangulares $f(A)$ tiene la siguiente expresión:

$$f(A) = a_1(1) + (a_2(1) - a_1(1)) - (a_1(1) - a_1(0)) / (r+1) + \lambda(a_2(0) - a_1(0)) / (r+1)$$

donde r es un parámetro de acuerdo al cual $f(A)$ toma valores cercanos a los valores modales ($r > 1$) o cercanos a los valores sobre el soporte ($r < 1$).

4. Aplicaciones

Muchos problemas reales pueden ser representados mediante la optimización de una función de forma que todas las características y restricciones del problema puedan ser recogidas en la misma. Ahora bien, la optimización de tal función puede ser un problema polinomial de orden elevado o NP, como es el caso de muchos problemas de optimización en redes y grafos, problemas de decisión, problemas de satisfacibilidad, etc. En muchos problemas de este tipo ya ha sido mostrada la utilidad de los AG para la resolución de los mismos.

De entre los muchos posibles hemos seleccionado dos problemas para mostrar el uso de las funciones de conveniencia difusas en los AG.

a) *Máximo flujo en una red con capacidades difusas*, [1,6]

Se pretende estudiar el cálculo del flujo máximo en una red con capacidades difusas. Este problema se plantea como sigue: Considérese una red $S = (N, A)$ con m nodos y n arcos a través de la cual fluye un tipo de líquido o gas desde un nodo origen n_o hasta un nodo final n_f . Se suponen capacidades difusas en los arcos de la red. N representa el conjunto de vértices de la red y $A \subset N \times N$ el conjunto de arcos. El problema es encontrar el flujo máximo desde n_o hasta n_f donde asociado a cada arco (i,j) se tiene una cota inferior de $l_{ij} = 0$ y una cota superior imprecisa, capacidad, $u_{ij} \in F(R)$.

A continuación se introduce el concepto de corte en una red que será básico para el desarrollo del problema.

Definición 1. (Separación del nodo n_o del nodo n_f) [1]

Sea N_1 cualquier conjunto de nodos en la red tal que N_1 contiene el nodo n_o pero no al nodo n_f . Sea $N_2 = N - N_1$. Entonces $(N_1, N_2) = \{(i,j): n_i \in N_1, n_j \in N_2\}$ se llama un corte que separa al nodo n_o del nodo n_f .

De forma paralela al caso real, se define la capacidad difusa como:

Definición 2.

La capacidad difusa de un corte (N_1, N_2) es un conjunto difuso definido como

$$\tilde{C}(N_1, N_2) = \sum_{(N_1, N_2)} \tilde{u}_{ij} = \sum_{\substack{n_i \in N_1 \\ n_j \in N_2}} \tilde{u}_{ij} \quad (6)$$

entendiendo la operación de suma como la suma extendida de números difusos.

Teorema 1. (Teorema de flujo máximo y corte mínimo, [10])

El valor del flujo máximo en S , de n_o hacia n_f es igual a la capacidad del mínimo corte en S que separa n_o y n_f .

Así el problema queda reducido a como obtener una partición del conjunto de nodos N en dos subconjuntos N_1, N_2 , que separen n_o y n_f y tales que la suma de las capacidades de los arcos con origen en N_1 y extremos en N_2 sea mínima.

Para ello, se definen las variables x_i asociadas a los nodos $x_i \in N$, con valor 1 si $n_i \in N_1$, y con valor 0 si $n_i \in N_2$. Entonces el problema se reduce a minimizar

$$g(x) = \sum_{i=1}^m \sum_{j=i+1}^m \tilde{u}_{ij} [x_i(1-x_j) + (1-x_i)x_j]. \quad (7)$$

con lo que se tiene un problema que consiste en minimizar una función booleana sin restricciones. Para resolver el problema desde el punto de vista de los AG vamos a introducir los siguientes elementos: La función de evaluación $E(\cdot)$ que se define como

$$E(x) = \sum_{i=1}^m \sum_{j=i+1}^m \tilde{u}_{ij} - g(x) \quad (8)$$

de modo que se traslada el problema de minimización de $g(x)$ a un problema de maximización de la función de evaluación $E(\cdot)$, que son equivalentes. Y la representación genética de las soluciones del problema, que consistirá en

considerar directamente una cadena de bits donde la subcadena asociada a la posición i corresponde al valor asociado a la variable x_i . La probabilidad de selección para cada individuo de la población se obtendría utilizando funciones ordenadoras lineales tal como ha sido propuesto en la sección anterior, (3).

Nota: Como se sabe, [8], si consideramos una red con varios nodos iniciales $\{n_{o1}, \dots, n_{ok}\}$ y varios nodos finales $\{n_{f1}, \dots, n_{fp}\}$, el problema de encontrar el máximo flujo desde todos los nodos iniciales hacia todos los nodos finales se puede convertir en un problema de cálculo del máximo flujo en una red simple introduciendo un vértice inicial artificial n_o y un nuevo vértice final n_f . Se introducen arcos desde n_o hacia los vértices $\{n_{o1}, \dots, n_{ok}\}$ con capacidad infinita y desde los vértices $\{n_{f1}, \dots, n_{fp}\}$ hasta el vértice n_f igualmente con capacidad infinita, Fig. 3., con lo que esta generalización sería resoluble de la misma forma que el caso simple anterior.

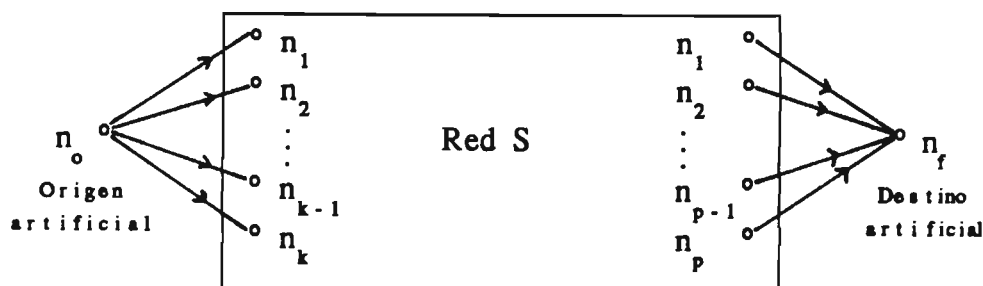


Fig. 3.: Red con varios orígenes y destinos

b) El problema de la asignación con etiquetas lingüísticas

El enfoque lingüístico que consiste en suponer que las variables que intervienen en el problema toman valores lingüísticos en lugar de numéricos, [16], es particularmente útil para abordar muchos problemas, ya que permite una representación de la información que poseen los expertos más directa y adecuada en los casos en que sean incapaces de expresarse con exactitud.

Un caso entre ellos es el que estudiamos a continuación: Supóngase que se tienen m individuos y m trabajos, y tenemos una valoración lingüística sobre la asignación del individuo i al trabajo j , establecida en términos de $L = \{\text{imposible, extremadamente inadecuado, muy inadecuado, inadecuado, posible, ...}\}$. Esta valoración sería una etiqueta lingüística que nos indicaría la adecuación del individuo i al trabajo j . Así sería necesario trabajar con un conjunto de etiquetas lingüísticas que nos discretizarían la valoración sobre la asignación.

En [3] se ha estudiado el empleo de conjuntos de términos de cardinal impar donde el término medio representa una probabilidad de "aproximadamente 0.5", y donde los restantes se colocan simétricamente alrededor de él, encontrándose el límite de discernimiento (máxima granularidad) en 11 o a lo sumo 13. Nosotros consideraremos un conjunto con 9 etiquetas E_i , $i = 1, \dots, 9$, para la discretización del espacio de valoraciones posibles por un experto. Este conjunto ha sido estudiado en [3], lo notaremos por L y los términos del mismo que vamos a considerar son los siguientes:

$$L = \{\text{imposible, extremadamente inadecuado, muy inadecuado, inadecuado, (8) posible, adecuado, muy adecuado, extremadamente adecuado, perfecto}\}$$

Notemos que estos términos pueden ser modificados de acuerdo al criterio del decisor.

La regla semántica de una variable lingüística asocia a cada etiqueta E_i un subconjunto difuso de $[0,1]$, N_i , que se caracteriza mediante una función de pertenencia, μ_{N_i} . Teniendo en cuenta que los datos lingüísticos son aproximaciones, que dan los expertos y aceptan los usuarios porque ó no hay ó no interesa emplear información más precisa, parece lógico pensar en representar las etiquetas mediante funciones de pertenencia poco complicadas. Vamos a considerar funciones triangulares o trapezoidales para traducir la vaguedad inherente en las valoraciones lingüísticas. La pérdida de información en que se incurre al aproximar subconjuntos difusos convexos por funciones de pertenencia triangulares o trapezoidales, es despreciable frente a la imprecisión inherente al propio concepto difuso.

Las funciones de pertenencia trapezoidales se pueden representar paramétricamente mediante los cuatro valores que definen el soporte y la moda, $(a_i, b_i, \alpha_i, \beta_i)$. La figura 4. ilustra la función de pertenencia de N_i , μ_{N_i} .

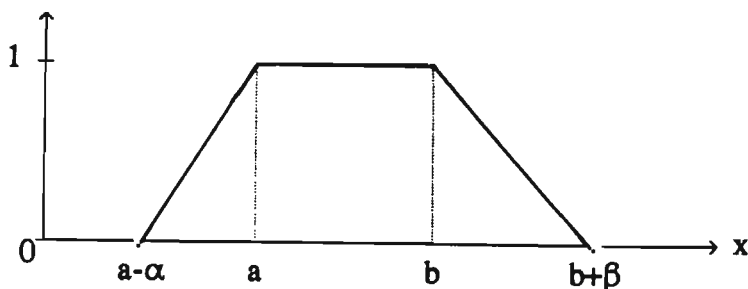


Fig. 4.: Función de pertenencia de N_i

Los valores de los números difusos asociados con cada etiqueta del conjunto de términos que proponemos fueron obtenidos de la adaptación de los resultados de experimentos psicológicos sobre el uso de probabilidades lingüísticas [2]. La tabla 1, indica la semántica del conjunto de etiquetas propuesto, [3],

imposible	(0 0 0 0)
extremadamente inadecuado	(0 .02 0 .05)
muy inadecuado	(.1 .18 .06 .05)
inadecuado	(.22 .36 .05 .06)
posible	(.41 .58 .09 .07)
adecuado	(.63 .80 .05 .06)
muy adecuado	(.78 .92 .06 .05)
extremadamente adecuado	(.98 1 .05 0)
perfecto	(1 1 0 0)

Tabla 1: Semántica de las nueve etiquetas del conjunto L

y la figura 5. la distribución de las funciones de pertenencia de los elementos de L.

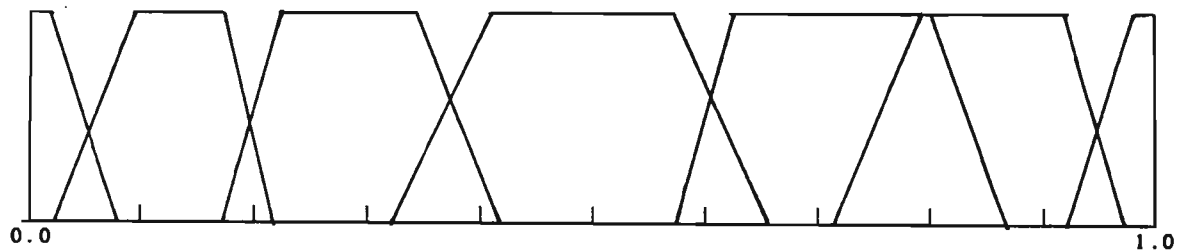


Fig. 5.: Distribución de las funciones de pertenencia de los elementos de L

De cara a resolver el problema, puede formularse el siguiente modelo matemático:

$$\begin{aligned}
 &\text{Max: } \sum_{i=1}^m \sum_{j=1}^m n_{ij} x_{ij} \\
 &\text{s. a. } \sum_{j=1}^m x_{ij} = 1 \quad i = 1, \dots, m, \\
 &\quad \sum_{i=1}^m x_{ij} = 1 \quad j = 1, \dots, m, \\
 &\quad x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, m.
 \end{aligned} \tag{9}$$

A continuación, es necesario plantearse la interpretación del objetivo $\sum_{i=1}^m \sum_{j=1}^m n_{ij} x_{ij}$ donde n_{ij} es el número difuso asociado a la etiqueta $e_{ij} \in L$ que nos da la valoración sobre la asignación del individuo i al trabajo j . Básicamente tendremos dos formas de abordar el problema.

La primera considerando el objetivo como la suma de los números difusos asociados a las etiquetas e_{ij} , de modo que tendríamos un objetivo difuso que sería utilizado como función de evaluación difusa.

La segunda combinando las etiquetas para obtener otra etiqueta que nos valore la combinación individuos-trabajos en cada asignación de las variables.

Existen básicamente dos formas de abordar este problema de la agregación y combinación de informaciones imprecisas. La primera consiste en establecer métodos para el cálculo directo sobre etiquetas, mientras que la segunda responde a la idea de realizar los cálculos empleando las funciones de pertenencia asociadas a los términos, utilizando para ello el principio de Extensión de Zadeh como herramienta básica. La mayoría de las técnicas de agregación que pueden encontrarse en la literatura pertenecen a la segunda categoría. Ahora bien, los cálculos dan como resultado subconjuntos difusos que no corresponden a ninguna etiqueta de los conjuntos de términos originales. Si finalmente se quiere obtener una etiqueta asociada se necesitará una aproximación lingüística [16,17]. Para la cual emplearemos el siguiente criterio: calcular el número difuso medio $(\sum_{i=1}^m \sum_{j=1}^m n_{ij} x_{ij})/m$ y a continuación asociar a cada número difuso un punto del plano de coordenadas (x, y) , x = centro de gravedad, y = area, y aproximar cada número no etiquetado por el valor lingüístico que minimice la distancia euclídea entre los puntos correspondientes [3,9], obteniéndose para cada asignación una valoración lingüística.

Siguiendo las ideas expuestas en la sección 3, la probabilidad de selección se obtendrá bien a partir del objetivo difuso obtenido como suma de los números difusos asociados a las etiquetas, con la función de evaluación

$$E(x) = \sum_{i=1}^m \sum_{j=1}^m n_{ij} x_{ij}$$

o a partir del número difuso asociado a la valoración lingüística de las soluciones factibles, con la función de evaluación $E(\cdot)$, $E: S \longrightarrow F(L)$, donde $F(L)$ denota el conjunto de números difusos que representan la semántica del conjunto de etiquetas.

En [14] Michalewicz describe el algoritmo Genetic-2 utilizado en [15] para resolver el problema del transporte. La representación genética del problema se puede diseñar de acuerdo al mismo, de modo que se puede utilizar el algoritmo Genetic-2 para la resolución de este problema, pero con la modificación correspondiente a la obtención de las probabilidades de selección a partir de la función de evaluación difusa.

Referencias

- [1] M. S. Bazaraa, J.J. Jarvis, *Linear Programming and Network flows*. John Wiley & Sons, 1977.
- [2] R. Beyth-Marom, How Probable is Probable? A Numerical Taxonomy Translation of Verbal Probability Expressions. *Journal of Forecasting* 1 (1982) 257-269.
- [3] P.P. Bonissone, K.S. Decker, Selecting Uncertainty Calculy and Granularity: An Experiment in Trading-off Precision and Complexity. KBS Working Paper, General Electric Corporate Research and Development Center. Schenectady, New York (1985).
- [4] G. Bortolan, R. Degani, A Review of some Methods for Ranking Fuzzy Subsets. *Fuzzy Sets and Systems* 15 (1985) 1-20.
- [5] L. Campos, A. Gonzalez, M.A. Vila, On the use of the Ranking Function approach to solve Fuzzy Matrix Games in a direct way. *Fuzzy Sets and Systems* 49 (1992) 193-203.
- [6] S. Chanas, Fuzzy Optimization in Networks. En: J. Kacprzyk and S. Orlovski, Eds., *Optimization Models using Fuzzy Sets and Possibility Theory*. Reidel P. Co., Boston, 1987, 303-327.
- [7] L.M. de Campos, A. Gonzalez, A Subjective Approach for Ranking Fuzzy Numbers. *Fuzzy Sets and Systems*, 29 (1989) 145-153.
- [8] N. Christofides, *Graph Theory. An Algorithmic Approach*. Acedemic Press, 1975.
- [9] M. Delgado, J.L. Verdegay, M.A. Vila, Lingtistic Assessments in Decision Making Problems. *Technical Report* N. 90-1-2, December 5, 1990.
- [10] L.R. Ford, D.R. Fulkerson, Maximal flow through a network. *Canad. J. Math* 8 (1956) 399-404.
- [11] A. González, A study of the Ranking Function approach through Mean Values. *Fuzzy Sets and Systems* 35, (1990), 29-41.
- [12] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [13] J.H. Holland, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.
- [14] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1992.
- [15] G.A. Vignaux, Z. Michalewicz, A Genetic Algorithm for the Linear Transportation Problem. *IEEE Transactions on Systems, Man and Cybernetics* 21 2 (1991) 445-452.

- [16] L. Zadeh, The Concept of a Linguistic Variable and its Applications to Approximate Reasoning. Part I, *Information Sciences* 8 199-249, Part II, *Information Sciences* 8 301-357, Part III, *Information Sciences* 9 (1975) 43-80.
- [17] L. Zadeh, Fuzzy Sets and Information Granularity. In: M.M. Gupta et al. Eds., *Advances in Fuzzy Sets Theory and Applications*. North Holland Publishing Company, New York, (1979) 3-18.

TOMA DE DECISIONES CON PROBABILIDADES LINGÜÍSTICAS

M.T.Lamata, S.Moral

Dpto.Ciencias de la Computación e I.A.
Universidad de Granada
18071-Granada, España.

Mlamata@ugr.es, Smoral@ugr.es

Resumen: *En este trabajo se estudian los problemas del cálculo con probabilidades cuando estas no son numéricas sino que vienen expresadas en términos lingüísticos. Consideramos diferentes formas de representación para estas etiquetas lingüísticas, así como cuales serían las propiedades deseables que deberían cumplir los diferentes conjuntos con que representamos las etiquetas. También estudiamos distintos criterios de decisión análogos a los de la probabilidad clásica.*

Palabras clave: Probabilidades lingüísticas, decisión.

1.-Introducción

Cuando nos encontramos ante un fenómeno incierto y tratamos de evaluarlo se nos presentan varios problemas; el primero de ellos está directamente relacionado con el tipo de información, de esta forma, una primera pregunta que nos debemos plantear es: ¿Cual será o como representar la incertidumbre del problema en cuestión?

Así cuando consultamos un texto médico podemos leer expresiones como las siguientes:

_La incidencia de la enfermedad E en la población P es elevada.

_Si observamos el síntoma S entonces es muy posible que se presente la enfermedad E.

Este tipo de conocimiento aún siendo de naturaleza incierta o probabilística no se debe tratar como tal, parece más adecuado desarrollar herramientas diferentes a las probabilísticas y que sean capaces de operar con este tipo de incertidumbre. En este sentido nos planteamos distintas cuestiones:- Formas de

Este trabajo ha sido financiado por la DGICYT bajo el proyecto PS89-0152

incertidumbre. En este sentido nos planteamos distintas cuestiones:- Formas de representar la incertidumbre, propiedades que deben cumplir las posibles representaciones, ¿Qué sucede cuando se pierde alguna de las propiedades exigidas a las medidas?; Cuales podrían ser los criterios de ayuda a la decisión ?.

Basicamente existen tres formas de abordar este problema: Una primera consistiría en establecer métodos para el cálculo directo con etiquetas, la segunda se apoyaría en los conjuntos difusos, así, Zadeh (1979) introduce el uso de cuantificadores numéricos representados por n^{os} difusos, y la tercera cuyos exponentes podrían ser Amarger et al (1991) consideran la representación por medio de intervalos.

Nuestro trabajo irá dirigido a tratar y desarrollar cálculos para las posibles representaciones de etiquetas lingüísticas.

Si queremos expresar la información "la probabilidad del suceso A es .75" no tendremos ningún problema ya que nuestro trabajo consistirá en operar dentro de la teoría de la probabilidad. Ahora bien, si nuestra información viene dada en términos más imprecisos como por ejemplo "la probabilidad del suceso A es aproximadamente .75" o "la probabilidad del suceso A es alta" tenemos que introducir otras teorías distintas a las probabilísticas.

Las anteriores informaciones sería posible traducirlas a:

L1.- Intervalos de probabilidad

Supone que el verdadero valor de la probabilidad del suceso está dentro del intervalo de extremos a,b. $\rightarrow [a,b]$

L2.- Números difusos

Es similar al anterior, salvo que para definir el número difuso necesitamos cuatro parámetros a,b, α , β $\rightarrow [a,b,\alpha,\beta]$ en el caso de n^{os} difusos trapezoidales.

L3.-Conjunto finito de valores lingüísticos

Se trata de operar con valores simbólicos de probabilidad directamente, y para ello, tendremos que enumerar los posibles valores que pueden tomar las probabilidades y su orden.

Este trabajo está dividido en tres secciones principales: En la primera, estudiamos la estructura de los conjuntos de etiquetas que nos será necesario para generalizar los cálculos probabilísticos. En la segunda sección se estudian las probabilidades que toman valores en el conjunto finito de valores lingüísticos, introduciendo conceptos como el de distribución de probabilidad, marginalización,

etc. Y en el último, hacemos un estudio del problema de decisión, estudiando algunos de los criterios clásicos para la toma de decisiones.

Nuestro paso siguiente será definir la relación de orden para L_1 , L_2 y L_3 así como la definición de las operaciones de suma y producto necesarias para operar con probabilidades.

2.-Cálculos con etiquetas lingüísticas

Sea L el conjunto de las posibles etiquetas con que nos referimos a las probabilidades de los elementos de un determinado conjunto. Cuando definimos las probabilidades y operamos con ellas necesitamos dos operaciones: La SUMA para poder hallar las probabilidades de conjuntos disjuntos y el PRODUCTO para encontrar las probabilidades bidimensionales a partir de las marginales y de las condicionadas.

Operaciones que deberían verificar las siguientes propiedades:

1.- *Commutativa*

$$\forall l_1, l_2 \in L \quad l_1 + l_2 = l_2 + l_1 \quad l_1 \times l_2 = l_2 \times l_1$$

2.- *Asociativa*

$$\forall l_1, l_2, l_3 \in L \quad (l_1 + l_2) + l_3 = l_1 + (l_2 + l_3) \quad (l_1 \times l_2) \times l_3 = l_1 \times (l_2 \times l_3)$$

3.- *Elemento neutro*

$$\exists 0 \in L \text{ tal que } \forall l \in L \quad 0 + l = l$$

$$\exists 1 \in L \text{ tal que } \forall l \in L \quad 1 \times l = l$$

4.- *Distributiva*

$$\forall l_1, l_2, l_3 \in L \quad l_1 \times (l_2 + l_3) = (l_1 \times l_2) + (l_1 \times l_3)$$

5.- *Complementario con respecto a uno*

$$\forall l \in L \quad \exists l' \in L \text{ tal que } l + l' = 1$$

6.- *Elemento inverso para la normalización*

$$\forall l_1, l_2 \in L \text{ si } \exists l_3 \in L \text{ tal que } l_1 + l_3 = l_2 \text{ entonces } \exists l_4 \in L \text{ tal que } l_2 \times l_4 = l_1 \text{ y este } l_4 \text{ es único.}$$

Veamos que problemas se nos plantean cuando operamos con; intervalos, números

difusos o etiquetas.

Ejemplo1

Sea $L_1 = \{[a, b] \text{ tal que } a, b \in \mathbb{R}; a < b; a, b \geq 0\}$

Con las operaciones suma y producto definidas por.

$$[a_1, b_1] + [a_2, b_2] = [a_1 + a_2, b_1 + b_2]$$

$$[a_1, b_1] \times [a_2, b_2] = [a_1 \times b_1, a_2 \times b_2]$$

Las operaciones anteriores que en este caso no se cumplen son:

- No existe el complementario con respecto a 1, ya que

$$[a, b] + [., 7] = 1 \Rightarrow a = .8 \text{ y } b = .3 \text{ con lo que } a \text{ sería mayor que } b$$

- Tampoco existe elemento inverso

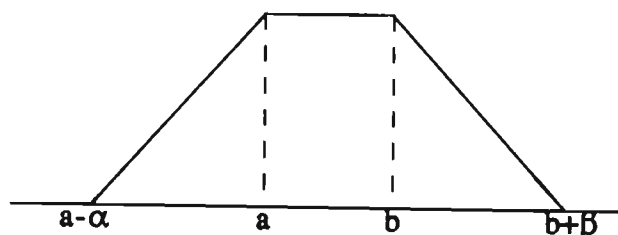
$$[., 2] + [., 7] = [., 9]$$

$$[., 9] \times [a, b] = [., 2] \Rightarrow a = 1/3 \text{ y } b = 2/9$$

Ejemplo2

Sea $L_2 = \{(a, b, \alpha, \beta) \text{ tal que } 0 \leq a \leq b; \alpha, \beta \geq 0\}$

donde (a, b, α, β) representa el número difuso trapezoidal



con las operaciones de suma y producto

$$(a_1, b_1, \alpha_1, \beta_1) + (a_2, b_2, \alpha_2, \beta_2) = (a_1 + a_2, b_1 + b_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)$$

$$(a_1, b_1, \alpha_1, \beta_1) \times (a_2, b_2, \alpha_2, \beta_2) = (a_1 a_2, b_1 b_2, a_1 \alpha_2 + a_2 \alpha_1 - \alpha_1 \alpha_2, b_1 \beta_2 + b_2 \beta_1 - \beta_1 \beta_2)$$

Al ser estos números una generalización de los anteriores, los problemas que se nos plantean son análogos.

Ejemplo3

Supongamos que $L = \{0, CI, PP, mR, R, MR, MP, CS, 1\}$ es un conjunto finito de

etiquetas.

donde:

0 $\rightarrow$ Imposible

CI $\rightarrow$ Casi imposible

PP $\rightarrow$ Poco probable

mR $\rightarrow$ Algo menor que regular

R $\rightarrow$ Regular

MR $\rightarrow$ Algo mayor que regular

MP $\rightarrow$ Muy probable

CS $\rightarrow$ Casi seguro

1 $\rightarrow$ Seguro

Las operaciones de suma y producto entre dos etiquetas se describen posteriormente por medio de tablas; para ello se pueden seguir dos enfoques :

El primero consistiría en asignar un número difuso a cada una de las etiquetas, operar con esos números difusos y al resultado asignarle la etiqueta correspondiente , pero esto no siempre es posible, por ello se le asignará la etiqueta más próxima.

Otro método propuesto por Delgado, Verdegay y Vila (1988) y López de Mántaras, Godó y Sangüesa (1990) consiste en la construcción directa de las tablas.

+	0	CI	PP	mR	R	MR	MP	CS	1
0	0	CI	PP	mR	R	MR	MP	CS	1
CI	CI	CI	PP	mR	R	MR	MP	1	1
PP	PP	PP	mR	R	MR	MR	1	1	1
mR	mR	mR	R	MR	MP	1	1	1	1
R	R	R	MR	MP	1	1	1	1	1
MR	MR	MR	MP	1	1	1	1	1	1
MP	MP	MP	1	1	1	1	1	1	1
CS	CS	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Tabla 1: Suma

Podemos observar como en la tabla de la suma, por ejemplo $1+1=1$, lo cual es consecuencia de la necesidad de cerrar la operación en el conjunto de las probabi-

lidades no siendo esto del todo extraño ya que no podemos tener dos conjuntos disjuntos con probabilidad 1 cada uno de ellos.

x	0	CI	PP	mR	R	MR	MP	CS	1
0	0	0	0	0	0	0	0	0	0
CI	0	CI	CI	CI	CI	CI	CI	CI	CI
PP	0	CI	CI	CI	PP	PP	PP	PP	PP
mR	0	CI	CI	PP	PP	PP	PP	mR	mR
R	0	CI	PP	PP	PP	mR	mR	R	R
MR	0	CI	PP	PP	mR	R	R	MR	MR
MP	0	CI	PP	PP	mR	R	MR	MP	MP
CS	0	CI	PP	mR	R	MR	MP	CS	CS
1	0	CI	PP	mR	R	MR	MP	CS	1

Tabla 2: Producto

Además, es necesario agregar que algunas de las propiedades anteriormente citadas son difíciles de verificar y por lo tanto tendremos que prescindir de ellas desde el principio. Veamos algunos problemas

Inversa con respecto al producto

Si consideramos el orden natural definido en el conjunto de las etiquetas, vemos que

$$0 < CI < PP < mR < R < MR < MP < CS < 1$$

entonces, el resultado de multiplicar cualquier etiqueta distinta de $\{0\}$ ó $\{1\}$ por el resto nos daría etiquetas repetidas, por ejemplo, asociada a MR obtenemos:

$$MR \longrightarrow 0, CI, \boxed{PP, PP}, mR, \boxed{R, R}, \boxed{MR, MR} \quad (1)$$

Se observa que como consecuencia de ser L finito, hay algunas etiquetas repetidas con lo cual no es posible obtener la inversa de PP con respecto a MR ya que PP no es única.

Una posible solución a este problema sería el proponer métodos para normalizar distribuciones de probabilidad después de condicionar.

Distributiva

Sucede algo parecido

Vemos que $CI + CI = CI$ con lo que $CI + \overset{i \text{ veces}}{\dots} + CI = CI$, mientras que para cualquier otra etiqueta $l + \overset{i \text{ veces}}{\dots} + l = 1$, esto implica que la propiedad distributiva no se verifique en todos los casos, ya que

$$lx(l + \overset{i \text{ veces}}{\dots} + l) = lx1 = 1$$

y

$$(lxl) + \overset{i \text{ veces}}{\dots} + (lxl) = CI + \overset{i \text{ veces}}{\dots} + CI = CI \quad \text{si } l=CI$$

Formalmente, una posible solución a este problema sea el tomar:

$$l + l' = \max(l, l')$$

$$l \times l' = \min(l, l')$$

expresión que tendrá sentido en el caso de posibilidades pero no en el de probabilidades.

3. Probabilidades con etiquetas lingüísticas

Supongamos una variable X que toma sus valores sobre un conjunto finito U . Una distribución de probabilidad acerca de X con respecto al conjunto de etiquetas L , está definida por medio de la aplicación p

$$p: U \longrightarrow L$$

verificando

$$\sum_{u \in U} p(u) = 1$$

donde 1 es el elemento neutro para la multiplicación en L .

Y teniendo en cuenta que toda distribución de probabilidad define una medida de probabilidad, esta vendrá dada por:

$$P: P(U) \longrightarrow L$$

$$P(A) = \sum_{a \in A} p(a)$$

verificando las siguientes propiedades

1.- $P(\emptyset) = 0$

2.- $P(U) = 1$

3.- Si $A \cap B = \emptyset \Rightarrow P(A \cup B) = P(A) + P(B)$

En lo sucesivo, supongamos una variable n -dimensional $X = (X_1, X_2, \dots, X_n)$, donde cada una de las variables X_i toma sus valores sobre un conjunto U_i . Si $J \subseteq \{1, \dots, n\}$, denotamos por X_J al conjunto de variables $(X_j)_{j \in J}$ y por U_J al conjunto $\prod_{j \in J} U_j$.

Si $J, K \subseteq \{1, \dots, n\}$ y los conjuntos de variables son disjuntos, esto es, $J \cap K = \emptyset$, llamaremos probabilidad condicional de X_J dado X_K a la aplicación

$$p^{J/K}: U_{J \cup K} \longrightarrow \mathbb{L}$$

tal que

$$\forall v \in V \sum_{u \in U_J} p^{J/K}(u, v) = 1$$

Bidimensional

Si tenemos una distribución de probabilidad p^K para X_K y una distribución condicional $p^{J/K}$ para X_J dado X_K , entonces estas distribuciones definen la distribución global $p^{J \cup K}$ sobre $U_{J \cup K}$ dada por:

$$p^{J \cup K}(u, v) = p^K(v) \times p^{J/K}(u, v) \quad (2)$$

Marginal

Si tenemos una distribución de probabilidad p definida sobre $U_{J \cup K}$, llamamos marginal de p sobre U_J a la distribución

$$p^{\perp J}: U_J \longrightarrow \mathbb{L}$$

dada por

$$p^{\perp J}(u) = \sum_{v \in U_K} p(u, v)$$

El primer problema que se plantea es que si alguna de las anteriores propiedades no se verifican, entonces, la probabilidad "a priori" p^J y la marginal no tienen por que coincidir, como se pone de manifiesto en el siguiente ejemplo.

Ejemplo4

Sean X_1 y X_2 dos variables que toman sus valores sobre $U_1=\{u_1, v_1\}$ y $U_2=\{u_2, v_2\}$ respectivamente, con distribución de probabilidad p sobre U , dada por:

$$p(u_1)=[.2, .5] \quad p(v_1)=[.5, .8]$$

suponiendo que la condicional de X_2 sobre X_1 sea

$$p^{2/1}(u_2 | u_1)=[.1, .2] \quad p^{2/1}(v_2 | u_1)=[.8, .9]$$

$$p^{2/1}(u_2 | v_1)=[.6, .7] \quad p^{2/1}(v_2 | v_1)=[.3, .4]$$

entonces la probabilidad global $p^{(1,2)}$ se calcula por medio de (2)

$$p^{(1,2)}(u_1 | u_2)=[.02, .1] \quad p^{(1,2)}(u_1 | v_2)=[.16, .45]$$

$$p^{(1,2)}(v_1 | u_2)=[.3, .56] \quad p^{(1,2)}(v_1 | v_2)=[.15, .32]$$

Si ahora calculamos la marginal sobre U_1 , obtenemos

$$(p^{(1,2)})_{\perp^1}(u_1)=[.18, .55]$$

$$(p^{(1,2)})_{\perp^1}(v_1)=[.45, .88]$$

Observamos como estos intervalos son menos precisos que los de partida, ello es consecuencia de no haber impuesto la condición de que la suma de las condicionadas sea cero.

En el caso de L_4 también tendríamos problemas; no coincidirían la "a priori" y la marginal como consecuencia de la explicación dada en (1).

4.-Modelo de decisión

Cualquier individuo ante un problema de decisión, deberá elegir una acción de entre el conjunto de alternativas posibles. Su problema estriba en que todo proceso de decisión lleva consigo un riesgo, riesgo que el decisor tratará de minimizar.

De esta forma, un problema de decisión consistirá en: " Escoger la mejor alternativa de entre un conjunto de alternativas posibles D ", para lo cual tendremos que ordenar el conjunto de las decisiones D con algún criterio que el

decisor emplee para su consecución.

En una situación real, cuando el decisor elija la acción $d_i \in D$ recibirá una recompensa $r(d_i, e_j)$, siendo e_j el verdadero estado de la naturaleza.

Si por $E = \{e_1, e_2, \dots, e_j, \dots, e_n\}$ denotamos el conjunto de estados de la naturaleza y $D = \{d_1, d_2, \dots, d_i, \dots, d_m\}$ el de las posibles decisiones, las ganancias o recompensas vendrán expresadas por la aplicación

$$I : D \times E \longrightarrow U$$

donde U es el conjunto de las utilidades.

$D \setminus E$	e_1	e_2		e_j		e_n
d_1	$r(d_1, e_1)$	$r(d_1, e_2)$		$r(d_1, e_j)$		$r(d_1, e_n)$
d_2	$r(d_2, e_1)$	$r(d_2, e_2)$		$r(d_2, e_j)$		$r(d_2, e_n)$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
d_i	$r(d_i, e_1)$	$r(d_i, e_2)$		$r(d_i, e_j)$		$r(d_i, e_n)$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
d_m	$r(d_m, e_1)$	$r(d_m, e_2)$		$r(d_m, e_j)$		$r(d_m, e_n)$

- Para nosotros, este conjunto U será el conjunto de términos lingüísticos L , que como ya hemos indicado está ordenado.

- La información acerca de los estados de la naturaleza será también lingüística

$$I : E \longrightarrow L$$

y por lo tanto, $I(E)$ representará la creencia que se tiene sobre la presencia de un determinado estado.

- Al decisor lo supondremos coherente y por ello, supuesto cualquier criterio como ayuda a la decisión, diremos que una decisión es preferida a otra si y solo si su vector de recompensas asociado es mayor.

En la mayoría de las situaciones reales, cuando un decisor se enfrenta al proceso de la toma de decisiones el problema básico que se le plantea es su desconocimiento total o parcial sobre el estado de la naturaleza que se presenta.

Según sea el grado de desconocimiento o incertidumbre acerca de los $e_j \in E$, nos encontraremos con distintos criterios para poder tomar una decisión acertada.

5.-Criterios de decisión

Dominancia.

Diremos que la decisión d_i domina o es preferida a la d_k si y solo si

$$d_i > d_k \iff \forall i, k \quad r(d_i, e_j) \geq r(d_k, e_j) \quad \text{con } j=1, 2, \dots, n$$

Según este criterio todas aquellas acciones que sean dominadas por otras serán desechadas.

Riesgo.

Como el decisor conoce las probabilidades de los estados e_j dados en términos lingüísticos, el criterio consistiría en calcular para cada acción posible, el valor medio esperado por medio de una tabla en la que se multipliquen probabilidades y recompensas y otra con la que se sumen recompensas. Al final, se elegiría aquella decisión cuya etiqueta con la relación de orden ya definida fuese mayor.

$$d_i > d_k \iff OR \left(\sum_{e_j \in E} r(d_i, e_j) \times p(e_j) \right) \geq OR \left(\sum_{e_j \in E} r(d_k, e_j) \times p(e_j) \right)$$

donde OR es el número de orden que ocupa la etiqueta en el conjunto de valores lingüísticos.

Ignorancia.

Nuestro desconocimiento sobre los estados de E es total, no tenemos información alguna encontrándonos en el caso de incertidumbre; la forma de decidir dependerá de la actitud del decisor ante el riesgo.

Para cada decisión d_i , hallamos la recompensa cuya etiqueta sea de orden mayor;

$$d_i \longrightarrow \text{Max-}OR(r(d_i, e_j)) \quad (3)$$

y aquella con menor orden

$$d_i \longrightarrow \text{Min-}OR(r(d_i, e_j)) \quad (4)$$

Max-Max

Consistiría en elegir aquella decisión que maximice (3). Es un criterio propio de personas optimistas.

$$\underset{d_i}{\text{Max-OR}} \underset{e_j}{\text{Max-OR}} (r(d_i, e_j))$$

Max-Min

Este criterio tiene un marcado sentido pesimista, el decisor trata de asegurarse una ganancia, por ello, tratará de maximizar la expresión (4)

$$\underset{d_i}{\text{Max-OR}} \underset{e_j}{\text{Min-OR}} (r(d_i, e_j))$$

Mixto

Supuesto que un decisor no va a ser totalmente optimista ni totalmente pesimista, sería lógico hacer una mixtura de los dos criterios anteriores en la forma siguiente:

$$\alpha \times \underset{d_i}{\text{Max-OR}} \underset{e_j}{\text{Max-OR}} (r(d_i, e_j)) + (1-\alpha) \times \underset{d_i}{\text{Max-OR}} \underset{e_j}{\text{Min-OR}} (r(d_i, e_j))$$

donde α representa el grado de optimismo por parte del decisor; en principio α debe de ser una etiqueta lingüística con la condición de que $\alpha + (1-\alpha) = 1$, como existirá más de una etiqueta que cumpla esta condición, nosotros elegiremos $(1-\alpha)$ como la etiqueta que cumpliendo la condición anterior tenga el orden menor.

Es decir

$$\alpha + \text{Min-OR}(1-\alpha) = 1$$

Orden lexicográfico

Para cada decisión d_i deberemos calcular los valores máximos y mínimos correspondientes al vector n -dimensional $\vec{r}_i$. Con ello, mediante la aplicación g nos situaremos en un espacio bi-dimensional donde será posible decidir por la primera o bien por la segunda de las componentes.

$$\begin{array}{ccc} g: \mathbb{L}^n & \longrightarrow & \mathbb{L}^2 \\ \vec{r}_i & \longrightarrow & [l_{i1}, l_{i2}] \end{array}$$

siendo:

$$l_{i1} = \text{Min-OR} (r(d_i, e_j))$$

$$l_{12} = \text{Max-OR } (r(d_i, e_j))$$

Directo

Consistirá en elegir aquella decisión cuyo extremo superior del intervalo asociado sea máximo, y en el caso de haber dos decisiones en las mismas condiciones decidiremos teniendo en cuenta el extremo inferior asociado a esas acciones.

Inverso

Es simétrico del anterior, primero se decide con la componente inferior para en caso de igualdad tener en cuenta la superior.

Si representamos los valores de $[l_{1i}, l_{2i}]$ con $i=1,2,\dots,m$ sobre los ejes coordenados, podríamos obviar todas aquellas alternativas que fuesen dominadas, quedándonos exclusivamente con la envolvente convexa, que nos daría el conjunto de decisiones admisibles.

Referencias

- R.Bellman y L.A.Zadeh**, Decision Making in a Fuzzy Environment. Management Science, 17, 141-164 (1970).
- P.Bonissone, S.Decker**, Selecting uncertainty calculi and granularity: an experiment in trading-off precision and complexity. En Uncertainty in Artificial Intelligence (L.N.Kanal and J.F.Lemmer, eds) Amsterdam, North-Holland, 217-247 (1986).
- M.Delgado, J.L.Verdegay y M.A.Vila**, Ranking Linguistic Outcomes under Fuzzyness and Randomness. Proceedings of the Eighteenth International Symposium on Multiple-Valued Logic, Palma de Mallorca, Spain, Computer Society Press, 352-356 (1988).
- D.Dubois, H.Prade, L.Godó, R.López de Mántaras**, A Symbolic Approach to Reasoning with Linguistic Quantifiers. Uncertainty in Artificial Intelligence. Proceedings of the Eighth Conference, Morgan & Kaufmann, San Mateo. 74-92 (1992).
- P.Jain, A.M.Agónino**, Stochastic sensitivity analysis using fuzzy influence diagrams. En Uncertainty in Artificial Intelligence, (R.D.Shacter et al, eds)

Nort-Holland, Amsterdam, 79-92 (1990).

M.T.Lamata, S.Moral, Calculus with Linguistic Probabilities and Beliefs. Por aparecer en *Advances in the Dempster-Shafer Theory of Evidence* (M.Fedrizzi et al, eds) John Wiley, New-york.

R.López de Mántaras, L.Godó, R.Sangüesa, Connective operators elicitation for linguistic term sets. *Proc. Inter. Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, 729-733 (1990).

R.R.Yager, Optimal Alternative Selection in the face of Evidential Knowledge. *Optimization Models using Fuzzy Sets and Possibility Theory*. (J. Kacprzyk y S.A. Orłowski eds), *Optimization Models using Fuzzy Sets and Possibility Theory*, Reidel Publishing Company, 123-140, (1987).

M.Tong y P.Bonissone, Linguistic Solutions to Fuzzy Decision Problems. *TIMS/Studies in the Management Sciences*, 20, 323-334 (1984).

L.A.Zadeh, Fuzzy Sets and Information Granularity. En (M.M.Gupta et al.eds.), *Advances in Fuzzy Sets Theory and Applications*, North Holland Publishing Company, Nueva York, 3-19 (1979).

ESTUDIO DE UN SISTEMA INTERACTIVO DE AYUDA A LA DECISION EN PROBLEMAS DE OPTIMIZACION DIFUSA

Jose Manuel Cadenas[†], Francisco Herrera[‡]

[†] Dpto. de Informática y Automática
Universidad de Murcia, 30071-Espinardo, Murcia

[‡] Dpto. de Ciencias de la Computación e I.A.
Universidad de Granada, 18071-Granada

Resumen: En este trabajo se presenta un estudio sobre el análisis y diseño de un sistema interactivo de ayuda a la decisión en problemas de optimización difusa, que permite al decisor tomar una decisión de forma racional de acuerdo a sus preferencias de actuación con respecto al modelo asociado. Se analiza el sistema, presentando los módulos que componen el mismo.

Palabras clave: Sistema de ayuda a la decisión, programación lineal difusa.

1. Introducción

Un Sistema de Ayuda a la Decisión (SAD) tiene como meta "ayudar a un decisor a encontrar *la mejor solución ó la más preferida* para su problema de decisión". Una definición bastante general de un SAD fue dada por Sprague y Carlson [7] englobando los aspectos claves de los mismos. Definen un SAD como un sistema basado en el computador, que ayude a los decisores, que haga frente a problemas que a veces están vagamente definidos y que actúe a través de una interacción directa: usuario, datos y análisis de modelos. Esta definición proporciona un amplio conjunto de SAD en una gran variedad de áreas.

Cuando el criterio y el conjunto de alternativas son conocidas, la solución más preferida bajo un comportamiento racional del decisor se puede definir como una solución no dominada, preferida por el decisor sobre todas las demás soluciones factibles ó alternativas.

En muchos casos este comportamiento de selección puede ser modelado

mediante una función de n variables $f(x_1, \dots, x_n)$, $n \geq 1$, donde x_i , $i = 1, \dots, n$ son variables que representan los elementos que intervienen en el problema de decisión. Normalmente, están sujetas a unas ciertas restricciones, esto es, $g_i(x_1, \dots, x_n) \leq b_i$, $i = 1, \dots, m$. Estas funciones tienen usualmente un significado real (por ejemplo, total de coste o ganancias, restricciones de capacidad o demanda, etc.).

Los problemas de optimización difusa son problemas de este tipo donde el decisor tiene un conocimiento impreciso sobre alguno o varios elementos del problema, estando la imprecisión representada mediante funciones de pertenencia asociadas a restricciones difusas ó a coeficientes imprecisos.

Existen en la literatura muchos modelos y métodos para la resolución de estos problemas [3,4,5,11]. Se obtienen soluciones puntuales en algunos casos, y soluciones difusas en otros, más acordes con el planteamiento impreciso de los problemas. Estas últimas recogen un conjunto de buenas alternativas, que engloban a las soluciones puntuales obtenidas en otros métodos. El decisor debe tomar la decisión final para su problema.

El propósito de este trabajo es presentar el análisis y diseño de un sistema interactivo de ayuda a la decisión en problemas de optimización difusa, que permite al decisor tomar una decisión de forma racional de acuerdo a sus preferencias de actuación con respecto al modelo asociado. Para ello en la sección 2 introducimos el problema general de optimización y los distintos modelos asociados. En la sección 3 presentamos el sistema interactivo de ayuda a la decisión.

2. Problema General de Optimización Difusa

Vamos a trabajar con problemas de optimización donde se suponen las funciones f y g_i , $i = 1, \dots, m$, que intervienen en el problema, funciones lineales. Por tanto, hablaremos de problemas de programación lineal difusa (PLD). Recordemos que en PLD los distintos modelos que se plantean, desde un punto de vista atomizado, son los siguientes:

- a) Problemas con restricciones difusas, en los que el decisor permite que el cumplimiento de tales restricciones sea flexible, en el sentido de estar dispuesto a tolerar cierto margen de violación que él mismo establece.
- b) Problemas con un objetivo difuso, caracterizados porque la información que se tiene sobre los coeficientes de la función objetivo es de tipo impreciso,

lo que conduce a considerarlos definidos mediante números difusos, y

c) Problemas con coeficientes imprecisos, lo que conduce a considerar números difusos, tanto en la matriz tecnológica como en el lado de la derecha, con un origen similar al del caso anterior.

Naturalmente, se permiten todas las combinaciones posibles de estas tres situaciones. El problema más general englobando los tres modelos anteriores puede escribirse como

$$\begin{aligned} \text{Max: } & \sum_{j=1}^n c_j x_j \\ \text{s. a. } & \sum_{j=1}^n a_{ij} x_j \leq b_i, i \in M = \{1, \dots, m\} \\ & x_j \geq 0, j \in N = \{1, \dots, n\} \end{aligned} \quad (1)$$

donde se consideran los siguientes elementos:

a) para cada fila, restricción,

$$\exists \mu_i \in F(\mathbb{R}) \text{ tal que } \mu_i: \mathbb{R} \longrightarrow [0,1], i = 1, \dots, m$$

la cual define el número difuso de la parte derecha,

b) para cada $i = 1, \dots, m$ y $j = 1, \dots, n$,

$$\exists \mu_{ij} \in F(\mathbb{R}) \text{ tal que } \mu_{ij}: \mathbb{R} \longrightarrow [0,1]$$

definiendo los números difusos en la matriz tecnológica,

c) para cada $j = 1, \dots, n$

$$\exists \mu_j \in F(\mathbb{R}) \text{ tal que } \mu_j: \mathbb{R} \longrightarrow [0,1]$$

definiendo los números difusos en la función objetivo, y por último

d) para cada fila de (1)

$$\exists \mu^i \in F[F(\mathbb{R})]: \mu^i: F(\mathbb{R}) \longrightarrow [0,1]$$

dando para cada $x \in \mathbb{R}^n$, el grado de cumplimiento de cada número difuso

$$\mu_{i1}x_1 + \mu_{i2}x_2 + \dots + \mu_{in}x_n, i = 1, \dots, m,$$

con respecto a la restricción i -ésima, esto es, la adecuación entre este número y el correspondiente b_i con respecto a la i -ésima restricción.

A continuación pasamos a estudiar el análisis y diseño del SAD interactivo.

3. Análisis y Diseño del SAD interactivo

Se pretende hacer un análisis y diseño bajo la idea de que un SAD interactivo debe proporcionar al usuario un fácil acceso al mismo, soportar la interacción con los usuarios no técnicos, así como actuar con problemas mal definidos.

El sistema que proponemos es un modelo dependiente del decisor, que resuelve de una forma eficiente los problemas planteados con métodos acordes al mismo. Interacciona con el decisor para la selección de los métodos adecuados, hasta que la solución obtenida le satisfaga.

El sistema debe analizar el problema, y distinguirá tres tipos, los atómicos anteriormente relatados a) y b) y el modelo general (1) que englobará en su resolución al modelo atómico c). Estos tipos tendrán asociados los tres submódulos RES-DIF, COS-DIF y GEN-DIF respectivamente, figura 1.

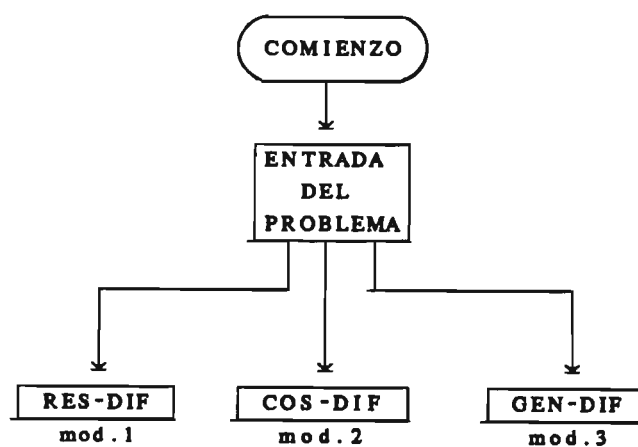


Fig. 1. Esquema general

A continuación pasamos a analizar estos tres módulos.

Mod. 1. RES-DIF. Cuando las restricciones son difusas el problema se puede escribir como

$$\begin{aligned} \text{Max: } z &= \sum_{j=1}^n c_j x_j \\ \text{s. a. } &\sum_{j=1}^n a_{ij} x_j \lesssim b_i, i \in N \\ &x_j \geq 0, j \in M \end{aligned} \quad (2)$$

donde para cada restricción, $i = 1, \dots, n$, $\exists \mu^i$,

$$\mu^i: \mathbf{R} \longrightarrow (0,1]$$

función de pertenencia asociada. Además, se puede tener una meta z_0 , tal que , $z \geq z_0$ o bien una meta difusa $z \gtrsim z_0$, con su correspondiente función de pertenencia asociada [6,8,10].

Ahora, supongamos que el problema de entrada corresponde al problema del tipo a), el sistema ejecuta el módulo 1, que comienza preguntando al decisor si la función objetivo tendrá una meta, y en caso afirmativo si esta es o no difusa. Se pasa a resolver el problema y se obtiene una solución difusa del mismo. A continuación interacciona con el decisor para seleccionar la solución puntual que se adecue al problema planteado por él mismo con el mayor grado posible. En todo momento se podrá modificar el problema o la meta de acuerdo a la información obtenida por el decisor sobre la solución difusa. El organigrama del mismo se presenta en la figura 2.

Mod. 2. COS-DIF. Cuando los coeficientes de la función objetivo son números difusos, el problema se puede escribir como

$$\begin{aligned} \text{Max: } & \sum_{j=1}^n c_j x_j \\ \text{s. a. } & \sum_{j=1}^n a_{ij} x_j \leq b_i, i \in N \\ & x_j \geq 0, j \in M \end{aligned} \quad (3)$$

donde para cada $j = 1, \dots, n$, $\exists \mu_j \in F(\mathbb{R})$

$$\mu_j: \mathbb{R} \longrightarrow [0,1]$$

funciones de pertenencia asociadas a los números difusos de la función objetivo.

Al ejecutarse este módulo, se plantearían los distintos métodos utilizados para resolver el problema: haciendo uso del teorema de representación ó de los métodos de comparación de números difusos [1,4].

Se entrará en una fase interactiva con el decisor hasta obtener una solución que satisfaga al mismo. El decisor en todo momento podrá cambiar de método, y dentro de cada uno de ellos, elegir la aproximación (multiobjetivo, posibilística, estratificada a trozos, intervalar, distintos índices de comparación ...) que más se adecue a su comportamiento racional de selección. El organigrama del mismo se presenta en la figura 3.

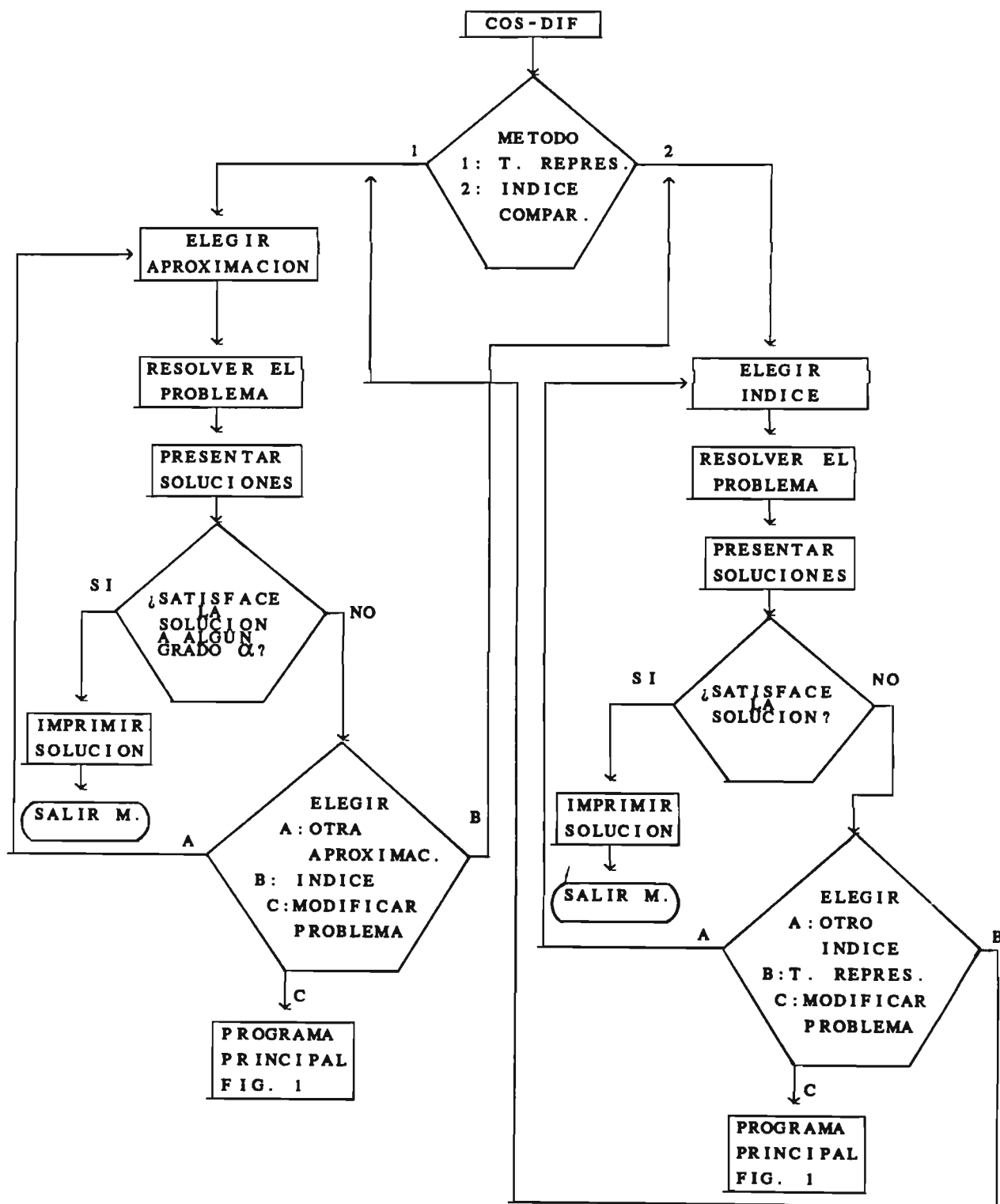


Fig. 2. Módulo COS-DIF

Mod. 3. GEN-DIF. Como ya hemos indicado en este módulo se considera el problema (1). Se comenzará pidiendo al decisor el índice de comparación a aplicar, obteniendo la solución difusa asociada a este índice. El decisor debe estudiar la satisfacción de la solución a nivel α . En caso negativo se le oferta al decisor elegir otro índice de comparación hasta obtener una solución que le satisfaga o finalmente volver al programa principal, fig. 1.

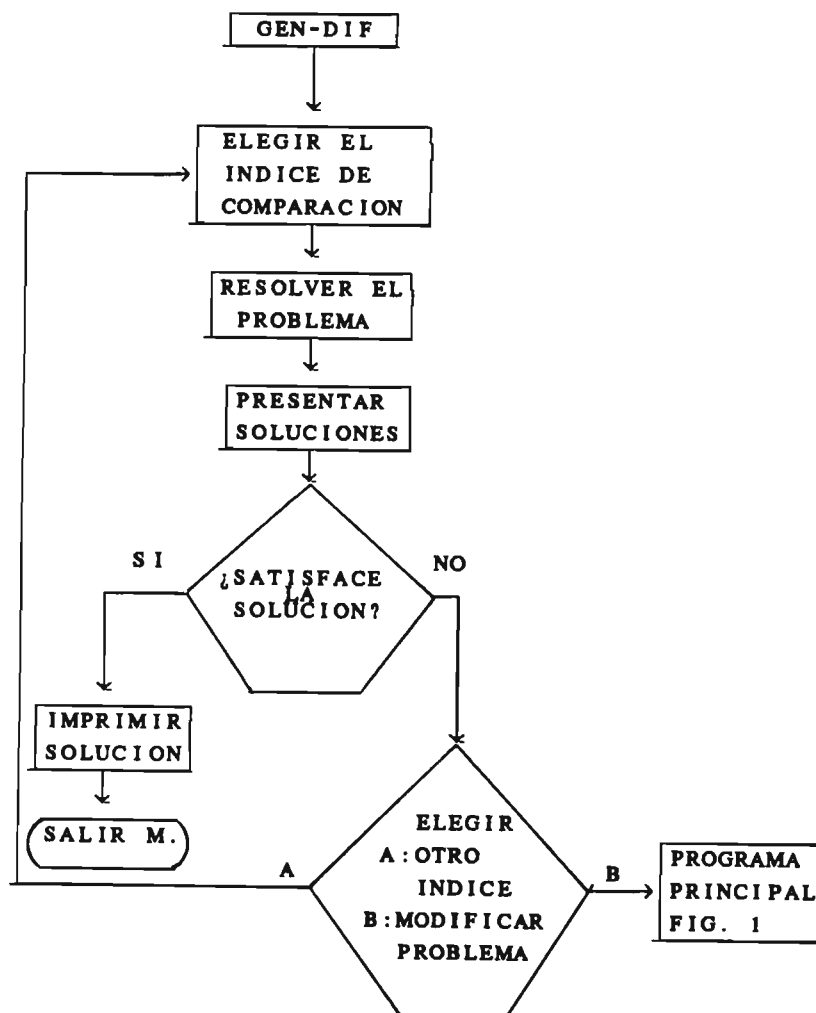


Fig. 4. Módulo GEN-DIF

Referencias

- [1] G. Bortolan, R. Degani, A review of some methods for ranking fuzzy subsets. *Fuzzy Sets and Systems* 15 (1985) 1-19.
- [2] M. Delgado, J.L. Verdegay y M.A. Vila, Imprecise costs in mathematical programming problems. *Control and Cybernetics* 16 (1985) 2 113-121.

- [3] M. Delgado, J.L. Verdegay, M.A. Vila, A General Model for Fuzzy Linear Programming. *Fuzzy Sets and Systems* 29 (1989) 21-29.
- [4] M. Delgado, J.L. Verdegay, M.A. Vila, Relating different approaches to solve linear programming problems with imprecise costs. *Fuzzy Sets and Systems* 37 (1990) 33-42.
- [5] M. Fedrizzi, J. Kacprzyk, J.L. Verdegay, A Survey of Fuzzy Optimization and Fuzzy Mathematical Programming. In: M. Fedrizzi, J. Kacprzyk, M., Roubens, Eds., *Interactive Fuzzy Optimization*. (Springer Verlag, Berlin, 1991).
- [6] Y. Lai, C. Hwang, Interactive fuzzy linear programming. *Fuzzy Sets and Systems* 45 (1992) 169-183.
- [7] R. H. Sprague, E.D. Carlson, *Building Effective Decision Support Systems*. (Prentice-Hall, 1982).
- [8] J.L. Verdegay, Fuzzy Mathematical Programming. In: M.M. Gupta and E. Sanchez, Eds., *Fuzzy Information and Decision Processes*. (North-Holland, Amsterdam, 1982) 231-237.
- [9] H.P Williams, *Model Building Mathematical programming*. (Wiley and Sons, New York, 1990)
- [10] H.J. Zimmermann, Description and Optimization of Fuzzy Systems. *International Journal of General Systems* 2 (1976) 209-215.
- [11] H.J. Zimmermann, *Fuzzy Sets, Decision Making, and Expert Systems* (Kluwer Academic Publishers, Boston, 1987).

HERENCIA DIFUSA

Ricardo Conejo Muñoz
José Luis Pérez de la Cruz
Rafael Morales Bueno
Francisco A. Triguero Ruiz

Universidad de Málaga.
Facultad de Informática
Departamento de Lenguajes y Ciencias de la Computación
E-Mail : conejo@ctima.uma.es

RESUMEN En los sistemas basados en marcos u objetos, los valores de los atributos pueden heredarse de padres a hijos (herencia descendente); de hijos a padres (herencia ascendente); o pasar de una colección de hermanos a otro (herencia colateral). Se define un modelo que integra dentro de la teoría de la posibilidad todos estos casos de herencia, considerando distribuciones de posibilidad para los valores para los atributos. Esto resulta especialmente útil para resolver las colisiones que se presentan en el caso de herencia descendente múltiple, tan problemático en el caso nítido. Este modelo puede integrarse en sistemas basados en razonamiento aproximado difuso impreciso e incierto.

Palabras Clave: Fuzzy Inheritance; Approximate Reasoning; Fuzzy Expert Systems

1. Introducción

Los sistemas de representación del conocimiento basados en marcos, cuyo origen se remonta a los trabajos de Minsky (Minsky 75), son ya una herramienta clásica dentro de la Inteligencia Artificial y la Ingeniería del Conocimiento. En los últimos años, se han impuesto también los sistemas de programación orientados a objetos, en los que desde otro punto de vista se emplean conceptos semejantes. En ambos casos es especialmente importante el papel que la herencia representa como vehículo de transmisión del conocimiento entre diversos marcos o clases. Las ventajas más sobresalientes del mecanismo de herencia son dos:

A) por una parte, al definir las propiedades de los objetos y conceptos más específicos en términos de otros más genéricos, permite una mejor y más compacta estructuración de la información, reduciendo así el tamaño de la memoria necesaria para almacenar los hechos. De esta manera, la herencia se reduce a una cuestión de implementación (Brachman 83).

B) por otra parte, implementa de manera directa y computacionalmente sencilla una forma de razonamiento por defecto no monótono, permitiendo asignar valores plausibles o probables en ausencia de información más específica. Es esta facilidad la que ha originado la conocida polémica suscitada por los marcos y redes semánticas en la comunidad de la IA a partir de las críticas logicistas (Hayes 79), (Brachman 85), que han dado lugar últimamente a formulaciones más teóricas del concepto de herencia, como las de (Touretzky 86) o (Etherington 88).

Por otra parte, los formalismos de Inteligencia Artificial y las aplicaciones de Sistemas Expertos han tenido que afrontar el problema de la representación y el manejo de la información imprecisa e incierta. Uno de los paradigmas más utilizados para realizar esta tarea es el que se sigue de la teoría de conjuntos difusos, creada por Zadeh (Zadeh 65).

El objeto del presente trabajo es plantear un modelo computacional de herencia aplicable a los sistemas basados en razonamiento aproximado difuso. En lo sucesivo y salvo que se especifique lo contrario, se supone que los valores de los atributos son conjuntos

difusos definidos sobre un dominio real. A efectos prácticos, consideraremos estos conjuntos como intervalos difusos definidos por funciones de pertenencia trapezoidales.

2. Herencia descendente

Tradicionalmente suele entenderse por herencia la *herencia descendente*, es decir, la transmisión de valores o procedimientos de una clase padre (o antepasado) a una clase hijo (o descendiente); una de las pocas generalizaciones de esta idea viene dada por la de *regularidad* (Mili y Rada 90). En el marco-padre se definen una serie de atributos que le son propios, e igualmente se hace en el marco-hijo. Si se desea realizar una inferencia para la cual es necesario conocer algún atributo del marco-hijo no definido en él, pero definido en el marco-padre, se toma éste como valor por defecto para el razonamiento, siempre y cuando no sea posible deducir por otro camino un valor contrario. En el caso nítido, estos valores son elementos de un cierto dominio. Razonemos sobre el ejemplo de la figura 1., que intenta representar mediante un sistema de marcos el enunciado "los jugadores de baloncesto suelen ser altos":

```
marco Jugador-de-Baloncesto
{
  es-un Hombre;
  altura = 2.0;
}
marco Juan
{
  es-un Jugador-de-Baloncesto;
}
```

fig.1.

Si en el proceso de razonamiento es necesario realizar cualquier inferencia que tenga en consideración la terna (objeto-atributo-valor) correspondiente a la altura de Juan, por defecto se optaría por (Juan,altura,2.0). El razonamiento subyacente en esta herencia es el siguiente:

La altura de todo Jugador-de-Baloncesto es 2.0 metros
Juan es un Jugador-de-Baloncesto

Luego la altura de Juan es 2.0 metros

Esta formalización es inapropiada por varias razones:

A) Por una parte, la premisa no es completamente cierta: no se ha dicho que todos los jugadores de baloncesto midan 2.0 metros. Asimismo, el valor nítido que se le atribuye a la altura no es el mas acorde con nuestro conocimiento, que se expresaría como un valor impreciso cercano a los 2.0 metros, es decir, el valor del atributo sería representado por su distribución de posibilidad en el dominio.

B) Por otra parte, la conclusión no es segura, ya que se trata de un valor por defecto, y no de un valor asignado o calculado expresamente para JUAN, cuyo grado de pertenencia a la clase JUGADOR-DE-BALONCESTO puede ser incierto.

Por todo ello, en un sistema de razonamiento aproximado difuso el Modus Ponens anterior debe reemplazarse por el Modus Ponens Generalizado, en el sentido de Zadeh (Zadeh 78). Así, el mecanismo de la herencia se reduciría a la *deducción difusa*, y sería fácilmente adaptable a los sistemas expertos de razonamiento aproximado difuso que incorporan el tratamiento conjunto de imprecisión e incertidumbre (vd. el cap. 3 de (Dubois y Prade 88), o de (López de Mántaras 90)).

Centremos nuestra atención por el momento en la imprecisión, y por tanto supongamos que las premisas correspondientes al razonamiento deductivo asociado a la herencia son completamente ciertas, con lo que el valor heredado por el marco-hijo coincide exactamente con el valor del atributo del marco-padre. Generalizando, la solución que se plantea es sustituir el valor nítido de un atributo por una distribución de posibilidad definida sobre el dominio de variación de éste:

$$\pi_p : X \rightarrow [0,1]$$

donde:

$\pi_p(x)$ se interpreta como la posibilidad que tiene la variable p de tomar el valor x .

$\pi_p(x) = 0$, se interpreta como “ p no puede tomar el valor x ”, o “ x es imposible”

$\pi_p(x) = 1$, se interpreta como “ x es perfectamente posible”

Cuando no sepa nada acerca de los valores que puede tomar p , tomaremos:

$$\forall x \in X \quad \pi_p(x) = 1$$

y si se sabe que p toma valores correspondientes a un conjunto difuso A , tomaremos:

$$\forall x \in X \quad \pi_p(x) = \mu_A(x)$$

Se dice que una distribución de posibilidad está *normalizada* si al menos un valor del dominio es perfectamente posible para el atributo, en adelante supondremos que todas las distribuciones de posibilidad están normalizadas, es decir:

$$\exists x \in X \quad \pi_p(x) = 1$$

Se dice que una distribución de posibilidad π_p es mas *específica* que otra π_p' si se cumple:

$$\forall x \in X \quad \pi_p(x) \leq \pi_p'(x)$$

Cuando esto ocurre, la información de π_p' es redundante y puede ser desechada en favor de la mas específica π_p .

Desde el punto de vista práctico, la herencia de atributos difusos, representados por distribuciones de posibilidad, tiene el inconveniente de ser, por lo general, poco específica. Pensemos, volviendo al ejemplo anterior, que la existencia de algunos jugadores de baloncesto no muy altos hace perfectamente posibles valores no muy altos en la distribución de posibilidad, como veremos en el siguiente epígrafe. El valor de herencia por defecto, en este caso, sería poco informativo, disminuyendo así la efectividad del mecanismo. Por ello proponemos que en los sistemas basados en el conocimiento se utilicen dos valores difusos para los atributos sobre los que actúe la herencia. A estos valores les denominaremos, “ámbito” y “prototipo”. El primero de ellos corresponde al valor definido anteriormente, como la distribución de posibilidad para cada elemento del dominio. El “prototipo” será el conjunto difuso $\mu_T(x)$, correspondiente a un valor representativo o típico de la clase. Evidentemente ha de cumplirse la inclusión del prototipo en el ámbito.

$$\mu_T(x) \leq \pi_p(x)$$

Usando este par de valores, ámbito y prototipo, podemos modificar el mecanismo de herencia de acuerdo con informaciones parciales difusas, y en general permitir el uso de los modificadores lingüísticos habituales, como por ejemplo las que se siguen de las sentencias: (véase figura 2),:

“Juan es un Jugador de Baloncesto”

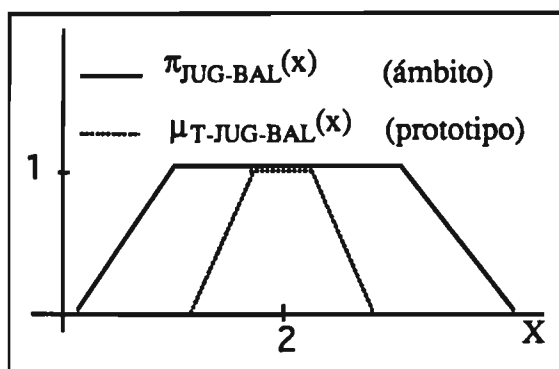
$$\pi_{\text{JUAN.h}}(x) = \pi_{\text{JUG-BAL.h}}(x)$$

“Juan es el típico Jugador de Baloncesto”

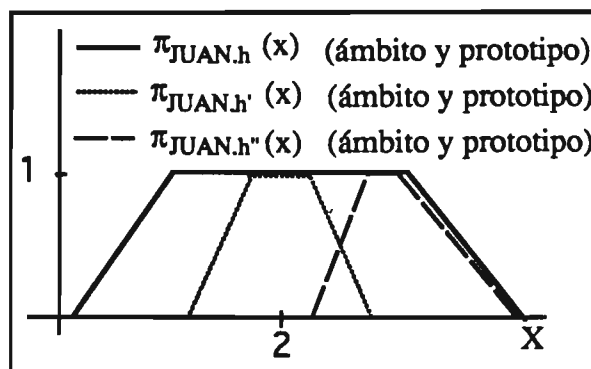
$$\pi_{\text{JUAN.h}}(x) = \mu_{\text{T-JUG-BAL.h}}(x)$$

“Juan es un Jugador de Baloncesto alto”

$$\pi_{\text{JUAN.h}}(x) = \begin{cases} \pi_{\text{JUG-BAL}}(x) - \mu_{\text{T-JUG-BAL}}(x), & x \geq x_{\max} \\ 0, & x < x_{\max} \end{cases} / \mu_{\text{T-JUG-BAL}}(x_{\max}) = 1$$



Valor del atributo h del marco-padre



Valores del atributo h del marco-hijo

fig.2.

Con este modelo se alcanza un mayor poder expresivo. Por ejemplo, al indicar que Juan es un Jugador de Baloncesto alto, el sistema será capaz de extraer el máximo de información de esta sentencia, ya que gracias al mecanismo de herencia y a la existencia de un prototipo, es capaz de determinar una distribución mas específica. Además, es interesante resaltar que un mismo modificador lingüístico aplicado a un mismo dominio, puede dar lugar a interpretaciones diferentes de acuerdo con el contexto o ámbito en el que sea utilizado. Así, se recoge una característica pragmática inherente al lenguaje y al conocimiento humano: no es lo mismo decir que "Juan es un Jugador de Baloncesto alto", que decir que "Juan es un japonés alto". (Si Juan mide alrededor de 1.80 el primer enunciado no parece que sea cierto, pero si el segundo).

Resumiendo, en ausencia de otra información el mecanismo de herencia nos proporciona el ámbito de los atributos del marco-hijo a partir del ámbito y el prototipo de los correspondientes atributos del marco-padre. El valor por defecto del prototipo correspondiente a un atributo del marco-hijo sobre el que no se tiene ninguna información se hará coincidir con su ámbito. De esta forma, en aquellos casos en los que la herencia se realice en cadena, el efecto será el mismo que si se realizara una herencia directa de la clase superior, sobre la que existe información completa.

3. Herencia ascendente

Un caso menos tratado en relación con la herencia, aunque igualmente interesante, se presenta cuando los valores de una clase antepasado se determinan a partir de los valores de sus descendientes (*inducción* o herencia ascendente). Supongamos que deseamos obtener el valor del atributo del marco-padre h, a partir de los valores $h_1...h_n$, correspondientes a los atributos de los marcos-hijos. En los sistemas basados en atributos nítidos, la inducción puede realizarse de varias formas:

- eligiendo un marco-hijo al azar, y considerándolo como representante del resto para la herencia ascendente
- extrapolando el valor del atributo h a partir de valores estadísticos como la media, la moda o la mediana.

Sin embargo, ninguno de estos métodos es enteramente satisfactorio, ya que se pierde información sobre la variabilidad del atributo.

En el caso de que los atributos $h_1...h_n$ tomen valores difusos (definidos mediante ámbito y prototipo), el ámbito del atributo del marco-padre puede inferirse como la unión de

los ámbitos de los atributos de los marcos-hijos, es decir, si los ámbitos de los atributos de los marcos-hijos vienen definidos por las distribuciones de posibilidad $\pi_1 \dots \pi_n$, el ámbito del atributo del marco-padre se obtiene, considerando la t-norma/t-conorma min/max, y aplicando su propiedad asociativa :

$$\pi_h(x) = \max\{\pi_i(x)\}$$

Esta definición es coherente con la idea intuitiva de posibilidad de que el atributo h tome el valor x, como la "envolvente" de los ámbitos de los atributos de los marcos descendientes de él. De esta forma, los razonamientos en los que intervenga el ámbito de los atributos del marco-padre serán menos específicos que si hubiésemos empleado cualquier representante de entre los marcos-hijos, pero, por el contrario tienen la ventaja de considerar todos los valores de la clase.

En general, desde un punto de vista práctico, es deseable poder contar con un prototipo, o valor típico para un atributo. Esto ayuda a obtener conclusiones mas específicas y por tanto mas precisas. Al igual que en el caso nítido, el prototipo puede obtenerse tomando una instancia cualquiera, o bien, hallando un estadístico que sirva como representante. Para ello podemos emplear una media generalizada (Dyckhoff y Pedrycz 84), o una particularización de esta como la media armónica o media aritmética difusa definida de la siguiente forma:

Sean N cantidades difusas $Q_1 \dots Q_N$ definidas mediante sus correspondientes funciones de pertenencia $\mu_1(w_1) \dots \mu_N(w_N)$. La media de estas cantidades se define como la cantidad difusa:

$$\bar{Q} = \frac{\sum_{i=1}^N Q_i}{N}$$

Según el principio de extensión (Zadeh 75) la función de pertenencia asociada puede hallarse mediante la expresión:

$$\mu_{\bar{Q}}(x) = \sup \left\{ \min\{\mu_i(w_i)\} \middle/ \frac{\sum_{i=1}^N w_i}{N} = x \right\}$$

A diferencia de lo que ocurre con otras medias, si las funciones de pertenencia de las cantidades difusas se aproximan mediante funciones trapezoidales definidas por cuatro puntos, $Q_i = (\alpha_i, \beta_i, \gamma_i, \delta_i)$, se demuestra fácilmente que el valor de la media armónica de N cantidades difusas trapezoidales será otra cantidad difusa trapezoidal definida por

$$\bar{Q} = (\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\delta}) \quad \text{en donde} \quad \bar{\alpha} = \frac{\sum_{i=1}^N \alpha_i}{N}, \quad \bar{\beta} = \frac{\sum_{i=1}^N \beta_i}{N}, \quad \bar{\gamma} = \frac{\sum_{i=1}^N \gamma_i}{N}, \quad \bar{\delta} = \frac{\sum_{i=1}^N \delta_i}{N}$$

El prototipo del atributo del marco-padre, o valor típico medio para este atributo puede ser obtenido como la media de los valores de los ámbitos de los atributos de los marcos-hijos. Además, según la definición realizada se cumple la necesaria inclusión del prototipo en el ámbito para el marco-padre, (vease Dyckhoff y Pedrycz 84), ya que:

$$\mu_{T,h}(x) = \bar{\pi}_{h_1 \dots h_n}(x) \leq \max\{\pi_i(x)\} = \pi_h(x)$$

La herencia ascendente supone en cierta forma un aprendizaje del sistema, ya que a partir de la información de que dispone abstrae ciertas conclusiones que pueden ser utilizadas posteriormente.

4. Herencia colateral

Establecida la herencia descendente y ascendente, es fácilmente implementable la herencia colateral, según el esquema de la figura 3., mediante la composición de la herencia ascendente y descendente. Esto proporciona al sistema un tipo de razonamiento por analogía. Efectivamente, si conocemos los valores de los atributos $h_1 \dots h_n$ para N instancias o marcos herederos de un mismo marco, podemos aproximar por defecto el ámbito y el prototipo del atributo h_c de un nuevo marco o instancia heredero del mismo padre, a partir del ámbito y prototipo del correspondiente atributo paterno h .

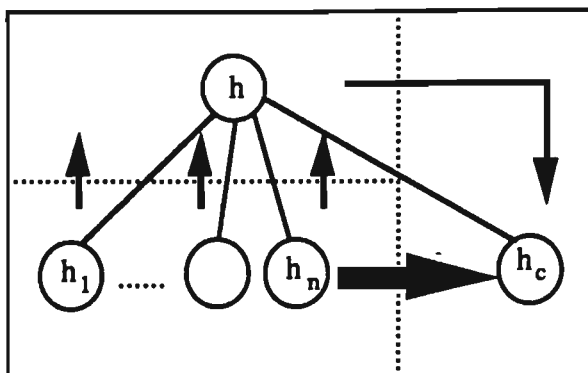


fig.3.

En la herencia colateral tiene especial relevancia el uso de modificadores lingüísticos que comparan el individuo con otros de su misma clase. De esta forma se puede modular el valor de herencia por defecto en condiciones en las que no sepamos exactamente cuál es el valor del atributo, pero tengamos una indicación vaga sobre su relación con los demás. Sentencias como "La altura de Juan es la normal para un jugador de baloncesto", "Juan es bajo para ser un jugador de baloncesto", "Juan es un poco mas alto que la media" contrastan con la escueta sentencia "Juan es un jugador de baloncesto". Por tanto, aunque en ninguno de los tres casos se pueda aplicar un valor de altura por defecto completamente exacto, con el mecanismo de herencia controlado por ámbito y prototipo difusos podemos establecer distribuciones mas específicas si contamos con mayor información.

5. Herencia descendente múltiple

La herencia múltiple en los sistemas de IA ha suscitado fundamentalmente dos críticas desde el punto de vista logicista:

A) El valor heredado depende estrechamente del algoritmo particular que implementa la herencia, y de los datos concretos a los que se aplica. En efecto, la opción mas frecuente es establecer una prioridad entre los antecesores, eligiendo el valor por defecto correspondiente al primero de ellos según un orden estática o dinámicamente determinado.

B) Es imposible combinar las propiedades heredadas por diferentes caminos. Consideremos por ejemplo el (retorcido) ejemplo de (Patel-Schneider 91): sean las dos clases ELEFANTES-CON-MAS-DE-3-PATAS y ELEFANTES-CON-MENOS-DE-5-PATAS. Si creamos una clase descendiente común de ambas, los sistemas habituales no serán capaces de sintetizar el valor concreto que ha de tener el atributo "número de patas"

En el caso difuso, habiendo definido los atributos mediante un ámbito y un prototipo, la herencia múltiple no plantea ninguno de los dos problemas, ya que no se establece un único valor para una clase u objeto, sino una distribución de posibilidad para cada uno de los valores que pueden tomar los atributos de los descendientes (ámbito). La herencia múltiple se resuelve en función del principio de mínima especificidad (Dubois y Prade 86), (López de Mántaras 91), que establece que, en ausencia de otra información, debemos ser tan imprecisos como sea posible. En este caso, el principio se enunciará así:

Cuando disponemos de información procedente de varias fuentes que merecen total confianza, la distribución de posibilidad resultante es la menos específica de las distribuciones que satisface las restricciones impuestas por todas las fuentes, es decir:

$$\forall x \in X \quad \pi_p(x) = \sup_{i=1, \dots, n} (w \in [0,1] / w \leq \pi_p^i(x))$$

Por tanto, en el caso general, el valor de herencia se obtiene mediante la intersección de los ámbitos de los valores de los atributos de los marcos-padres.

Haciendo uso del prototipo y sus modificadores podemos construir distribuciones de posibilidad mas específicas heredadas por defecto, como puede verse en el siguiente ejemplo que aparece en las fig.4.

Pueden presentarse situaciones en los que al aplicar el mecanismo de herencia múltiple obtengamos una distribución de posibilidad nula, o que no este normalizada. Estas situaciones denotan una incoherencia en nuestra base de conocimientos, ya que indicarían que ningún valor es completamente posible para los descendientes comunes de ciertos marcos, o al menos, que no lo son bajo las circunstancias que se indican en la base de conocimientos.

6. Conclusión

En los sistemas de razonamiento aproximado basados en conjuntos y lógica difusa, un mecanismo de herencia entre marcos con atributos difusos es fácilmente implementable, y computacionalmente eficiente. El valor difuso al que denominamos ámbito,

asociado a un atributo, se interpreta como la distribución de posibilidad para cada uno de los valores nítidos que puede tomar dicho atributo. El razonamiento por defecto clásico en los mecanismos de herencia queda así sustituido por un razonamiento impreciso sobre todos los valores posibles. Como ventaja adicional, la herencia múltiple no produce conflicto entre distintos valores por defecto, permitiendo la detección de incoherencias en la base de conocimientos.

Por otra parte, si el sistema de razonamiento permite el tratamiento conjunto de imprecisión e incertidumbre, se puede controlar el grado de incertidumbre introducido al calificar un objeto como perteneciente a una determinada clase, usando para ello el mismo algoritmo de cálculo del Modus Ponens Generalizado, utilizado por el propio motor de inferencias.

El uso de prototipos mejora el rendimiento práctico del sistema incorporando el máximo de información disponible sobre un determinado objeto y permitiendo el uso de modificadores lingüísticos incorporables fácilmente a la sintaxis de un lenguaje de representación del conocimiento basado en marcos.

```
marco Jugador-de-Baloncesto
{
  es-un Hombre;
  altura = (1.7,1.9,2.2,2.4);
}
marco Japonés
{
  es-un Hombre;
  altura = [ambito (1.4,1.6,2.0,2.2)
             prototipo (1.5,1.6,1.7,1.8)];
}
marco Tanaka
{
  es-un Jugador-de-Baloncesto;
  es-un Japonés (altura > prototipo);
}
```

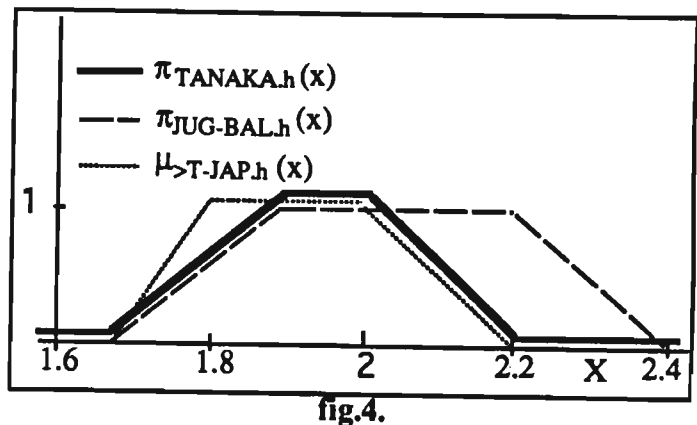


fig.4.

Referencias

- Brachman,R.J. (1983): What IS-A is and isn't: an analysis of taxonomic links in semantic networks. *Computer* 16(10) 30-36.
- Brachman,R.J. (1985): "I lied about the trees", or defaults and definitions in knowledge representation. *AI Magazine* 6(3) 80-93.
- Dubois, D. y H. Prade (1986): The principium of minimun specificity as a basis for evidential reasoning. *Proc. Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris*.
- Dyckhoff, H. y W. Pedrycz (1984): Generalizad means as a model of compensative conectives. *Fuzzy Sets and Systems* 14 143-154.
- Etherington, D,W, (1988): *Reasoning with Incomplete Information*. Londres, Pitman.
- Hayes,P. (1979): The logic of frames. En Metzging, D. (ed.): *Frame Conceptions and text understanding*, Berlín, Walter de Gruyter. Reimpreso en Brachman,R.J. y H.J. Levesque: *Readings in Knowledge Representation*, San Mateo, Ca., Morgan Kaufman, 1985.
- López de Mántaras,R. (1990): *Approximate Reasoning Models*. Chichester, Ellis Horwood.
- Mili, H. y R. Rada (1990): Inheritance Generalizad to Fuzzy Regularity. *IEEE Trans. on Systems, Man and Cybernetics* 20(5) 1184-1198.
- Minsky, M. (1975): A framework for representing knowledge. Memo 306 MIT AI Lab. Reimpreso en Haugeland,J.(ed.): *Mind Design*, Cambridge, Ma., MIT Press, 1981.
- Patel-Schneider,P.F. (1991): What's Inheritance got to do with Knowledge Representation. En Lenzerini et al. (eds.): *Inheritance Hierarchies in Knowledge Representation and Programming Languages*, Chichester, Wiley.
- Sombé, L. (1990): *Reasoning under incomplete information in Artificial Intelligence*. Special issue of the *International Journal of Intelligent Systems* 5(4)
- Touretzky, D.S. (1986): *The Mathematics of Inheritance Systems*. Londres, Pitman.
- Zadeh, L. (1965): Fuzzy Sets. *Information & Control* 8 338-353.
- Zadeh, L. (1975): The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences* 8 199-249, 8 301-357, 9 43-80.
- Zadeh, L. (1978): PRUF: A meaning representation language for natural languages. *International Journal of Man-Machine Studies* 10 395-460.

MODIFICACIONES SOBRE ID3 PARA EL TRATAMIENTO DE INCERTIDUMBRE

Autor: Luis I. Arranz Ramos
Área de Informática
DISEL S.A.
Madrid (España)

Resumen: Este trabajo presenta una modificación realizada sobre el algoritmo de clasificación ID3 para que funcione en dominios en los que existe ambigüedad. El método aquí descrito combina técnicas de inducción con el uso de factores de certeza. Además, este tratamiento permite que no sea imprescindible conocer con exactitud la filiación de los ejemplos de partida, sino que es suficiente conocer su grado de pertenencia a una clase. Esto posibilita el uso de conjuntos borrosos, por lo que el algoritmo podría ser llamado FID3 (Fuzzy ID3).

Palabras Clave: Clasificación, Aprendizaje, Inducción, Incertidumbre.

1 Presentación del ID3

1.1 Objetivo del algoritmo

El ID3 [1] es un algoritmo inductivo que permite clasificar objetos pertenecientes a dos clases distintas, que denominaremos + y -. La extensión a un número superior de clases es inmediata.

A partir de un conjunto de objetos $\mathcal{R}$ de los que se sabe cuáles pertenecen a + y cuáles a -, ID3 genera un árbol de decisión que clasifica correctamente a todos los objetos de $\mathcal{R}$ y que se puede utilizar para clasificar otros objetos. Si los ejemplos tomados en $\mathcal{R}$ son significativos, esta última clasificación tendrá éxito.

Todos los objetos considerados vienen caracterizados por una serie de atributos A, B, C, ..., con dominios finitos (y reducidos) de valores $\{A_1, A_2, \dots, A_n\}$, $\{B_1, B_2, \dots, B_m\}$, $\{C_1, C_2, \dots, C_l\}$, ... Si el dominio de algún atributo es continuo, debe ser discretizado. Si es muy grande puede resultar conveniente transformarlo en otro más pequeño en el que cada valor corresponde a un subconjunto de valores del dominio original. El tamaño de estos dominios es un factor muy importante en la complejidad del algoritmo, de ahí el interés en reducirlos razonablemente. Algunos algoritmos inductivos (Assistant 86 [2], por ejemplo) los dicotomizan en una fase previa.

Cada objeto viene descrito por los valores que toman en él los distintos atributos y por la clase (+ ó -) a la que pertenece. Por tanto, el conjunto $\mathcal{R}$ será definido del siguiente modo:

$$\mathcal{R} = \{A_i B_j C_k \dots +, A_r B_s C_t \dots -, \dots\}$$

Las hojas del árbol de decisión generado corresponden a las clases + ó -. Los nodos no hoja corresponden a atributos. De cada uno de estos nodos saldrán tantas ramas como valores tenga el atributo asociado. La clasificación de un nuevo objeto se realiza recorriendo el árbol desde la raíz hasta alcanzar alguna hoja. En cada nodo se toma el camino correspondiente al valor del atributo asociado que tenga el objeto a clasificar. La clase que se encuentra en la hoja alcanzada es la que se asigna al objeto en cuestión.

1.2 Descripción del algoritmo

El algoritmo ID3 parte de la raíz del árbol en la que se encuentran todos los objetos de $\mathcal{R}$. Con éste y con todos los nodos que se generen se pueden presentar tres casos:

1. Todos los objetos de $\mathcal{R}$ pertenecen a la misma clase. Esto significa que al llegar a este nodo ya se conoce la clase sin necesidad de observar ningún atributo más. Por lo tanto, éste será un nodo hoja etiquetado con la clase a la que pertenecen todos sus objetos.
2. El conjunto $\mathcal{R}$ asociado está vacío. Esto se produce cuando no se dispone de ningún ejemplo con los valores de los atributos que conducen hasta este nodo. Dicho nodo será una hoja y se etiquetará aleatoriamente con $+$ ó con $-$.
3. $\mathcal{R}$ contiene objetos de ambas clases. La decisión no se puede realizar todavía, por lo que se elige (en el siguiente apartado se explica cómo) un atributo T y se crean tantos nodos descendientes como valores pueda tomar T . El conjunto $\mathcal{R}$ se divide en $\mathcal{R}_1, \mathcal{R}_2, \dots$; de tal modo que en cada $\mathcal{R}_i$ estén todos los objetos de $\mathcal{R}$ que toman el valor T_i para el atributo T . Cada rama corresponde a un T_i y el nodo al que conduce está asociado al correspondiente $\mathcal{R}_i$.

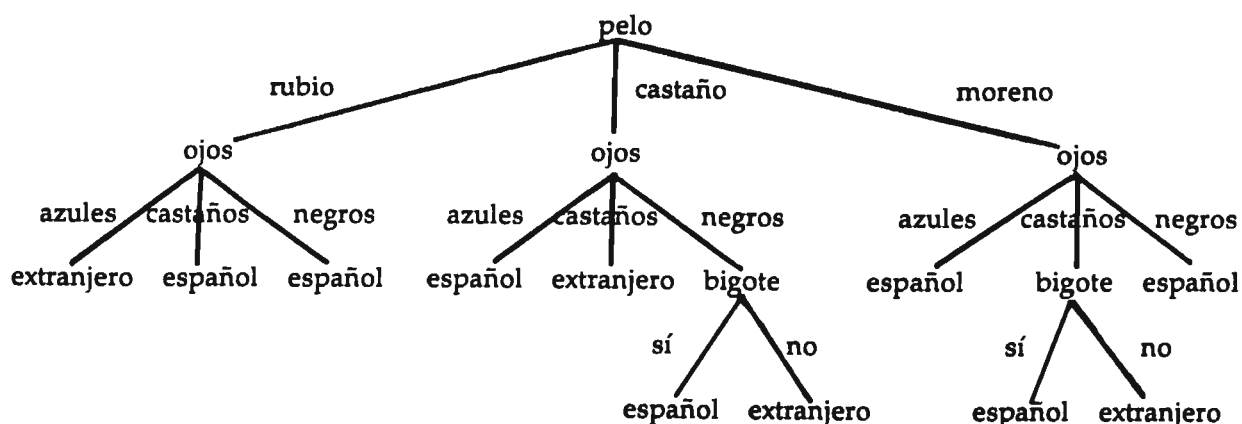
En los dos primeros casos el nodo tratado es una hoja y, por lo tanto, no se expande. En el último, por el contrario, se generan nuevos nodos (tantos como valores tenga el dominio del atributo seleccionado), que serán tratados de igual modo que su antecesor.

Este algoritmo funciona correctamente a menos que haya objetos que teniendo idénticos valores en todos sus atributos pertenezcan a clases distintas. ID3 los trataría como si fueran el mismo objeto, con la contradicción de que pertenezca a $+$ y a $-$ simultáneamente.

Como ejemplo ilustrativo consideremos que se quiere clasificar a una serie de sujetos en españoles y extranjeros atendiendo a características físicas tales como el color del pelo (rubio, castaño o moreno), el color de los ojos (azules, castaños o negros) y si tienen bigote (sí o no). Partamos del siguiente conjunto de ejemplos:

PELO	OJOS	BIGOTE	NACIONALIDAD
rubio	azules	sí	extranjero
rubio	castaños	no	español
rubio	castaños	sí	español
castaño	castaños	no	extranjero
castaño	azules	sí	español
castaño	negros	sí	español
castaño	negros	no	extranjero
moreno	castaños	sí	español
moreno	castaños	no	extranjero
moreno	negros	sí	español

Al aplicar el algoritmo anterior a este conjunto de ejemplos, seleccionando los atributos en el orden pelo, ojos, bigote, se obtiene el siguiente árbol:



1.3 Selección de atributos

Si la elección del atributo mencionada en el apartado anterior se realiza de forma aleatoria, el árbol resultante es correcto pero no óptimo. La clasificación de futuros objetos será tanto más lenta cuantos más nodos haya que visitar hasta alcanzar una hoja para un objeto dado, por lo que la optimización consistirá en minimizar el número de nodos del árbol.

Por otro lado, si la elección de atributos se hace aleatoriamente, se formarán bastantes hojas en las que no hay ningún objeto y a las que, como se ha dicho, se les asigna una clase aleatoriamente. Esto supone una fuente de imprecisión ya que los objetos que sean procesados con posterioridad y lleguen a dichas hojas, serán clasificados correctamente sólo en el 50% de los casos.

La estrategia utilizada por ID3 para la selección del atributo en cada nodo se basa en la Teoría de la Información. Esta estrategia se describe a continuación.

Consideremos un nodo en el que se tiene un conjunto de objetos $\mathcal{R}$ de los que x pertenecen a la clase + e y pertenecen a la clase -. La probabilidad de que un objeto de $\mathcal{R}$ pertenezca a la clase + se puede aproximar mediante la frecuencia relativa:

$$P(+) = \frac{x}{x+y}$$

De igual modo tomaremos que

$$P(-) = \frac{y}{x+y}$$

Evidentemente, el número de elementos de $\mathcal{R}$ es

$$\text{card}(\mathcal{R}) = x + y$$

La entropía esperada del árbol de decisión que clasifica a los objetos de $\mathcal{R}$ será

$$H(\mathcal{R}) = -P(+) \cdot \log_2 P(+) - P(-) \cdot \log_2 P(-)$$

$$H(\emptyset) = 0$$

Si $P(+)$ (o $P(-)$) es cero, el producto $P(+) \cdot \log_2 P(+)$ (o el $P(-) \cdot \log_2 P(-)$) debe ser 0, evitando calcular el logaritmo.

Si se selecciona el atributo T , las entropías esperadas en cada uno de los nodos hijos serán las $H(\mathcal{R}_i)$ y la correspondiente a todo el subárbol será:

$$H(\mathcal{R}/T) = \sum_{i=1}^n P[T = T_i] \cdot H(\mathcal{R}_i) \quad (1)$$

$P[T = T_i]$ es la probabilidad de que un objeto de $\mathcal{R}$ tome el valor T_i para el atributo T . Al aproximarla mediante la frecuencia relativa queda:

$$P[T = T_i] = \frac{\text{card}(\mathcal{R}_i)}{\text{card}(\mathcal{R})} \quad (2)$$

El criterio que sigue ID3 para seleccionar un atributo se basa en que el subárbol de decisión más sencillo será aquél que ofrezca una menor entropía. Recordemos que la entropía de un sistema (en Teoría de la Información) mide el grado de desinformación que de él se tiene. Cuanto menor sea este grado, más información se tendrá y más sencillo (en número de nodos) resultará el resto del árbol.

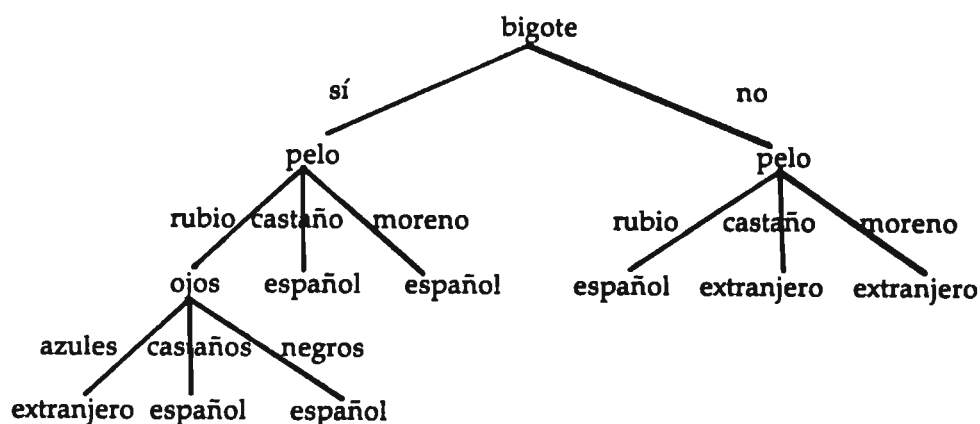
El método a seguir consistirá en calcular $H(\mathcal{R}/T)$ para todos los atributos que no hayan sido tratados en nodos ascendentes del que nos ocupa ($T=A$, $T=B$, ...) y seleccionar el atributo que minimice dicho valor.

Otra forma de expresar esto es decir que se tiene $H(\mathcal{R})$ y se va a pasar, al elegir un atributo T , a tener $H(\mathcal{R}/T)$. Como lo que interesa es que la ganancia de información sea máxima, hay que maximizar $H(\mathcal{R}) - H(\mathcal{R}/T)$. Como $H(\mathcal{R})$ no depende del atributo elegido, esto equivale a maximizar $-H(\mathcal{R}/T)$, es decir, a minimizar $H(\mathcal{R}/T)$.

Por lo tanto, en cada nodo con un conjunto de objetos $\mathcal{R}$ se elige el atributo T que minimice (sustituyendo (2) en (1))

$$\sum_{i=1}^n \frac{\text{card}(\mathcal{R}_i)}{\text{card}(\mathcal{R})} \cdot H(\mathcal{R}_i)$$

Con este criterio, el árbol de decisión generado por ID3 para el conjunto de objetos utilizado anteriormente como ejemplo es el siguiente:



Este árbol presenta dos ventajas sobre el primero que obtuvimos:

- el menor número de nodos (12 frente a 17), que reducirá el tiempo necesario para realizar posteriores clasificaciones
- el menor número de hojas con conjunto vacío; en estas hojas (marcadas en las figuras con un asterisco) se asigna aleatoriamente la clase, por lo que la probabilidad de acertar es 0.5.

1.4 Aplicaciones

Entre las diversas aplicaciones que tiene el ID3, las más importantes son la generación de reglas para un sistema experto y el reconocimiento de patrones.

1.4.1 Generación de reglas

A partir del árbol de decisión generado por el algoritmo es fácil obtener reglas útiles para un sistema experto clasificativo. Basta con recorrer todos los posibles caminos desde la raíz hasta una cualquiera de las hojas.

Supongamos que en uno de estos caminos se visitan nodos correspondientes a los atributos B, D y A, llegando a una hoja etiquetada con -. Los arcos recorridos serán B_k , D_r y A_s . Esto significa que todos los objetos que tengan valor B_k en el atributo B, D_r en D y A_s en A son clasificados asignándoles la clase -. La regla se podría escribir:

SI ($B = B_k$) Y ($D = D_r$) Y ($A = A_s$)
ENTONCES el objeto pertenece a -

ID3, por tanto, proporciona un método de adquisición de conocimiento basándose en ejemplos con una participación reducida de los expertos humanos. Dicha participación consiste en proporcionar ejemplos significativos y en determinar los atributos a estudiar, así como sus dominios de valores.

1.4.2 Reconocimiento de patrones

La instrucción de un ordenador para el reconocimiento de un patrón se puede realizar proporcionándole ejemplos positivos (que corresponden al patrón) y negativos (que no corresponden a él). Estos ejemplos se pueden categorizar incluyéndolos en una de las dos clases manejadas por ID3, con lo que el árbol de decisión resultante permite reconocer un objeto determinado.

Si de lo que se trata es de reconocer múltiples patrones, se puede realizar una pasada del algoritmo para cada uno de ellos o generalizar el método para que admita más de dos clases.

2 Modificaciones sobre ID3

2.1 El problema de la ambigüedad

En 1.2 se estableció que el algoritmo ID3 funciona correctamente si no existen objetos que pertenezcan a clases distintas pero que tengan idénticos valores para todos sus atributos. De no ser así, se tienen dos ejemplos del mismo objeto (a juzgar por los atributos considerados) que pertenecen a clases distintas.

Desgraciadamente, esta ambigüedad aparece frecuentemente cuando se tratan casos reales. En general, los dominios en los que aparece este problema se denominan dominios "con ruido". Sus principales fuentes son:

- **Discretización inadecuada.** Como ya se ha dicho, los dominios de los atributos deben ser discretos y reducidos. En el proceso de discretización se pierde información y esta pérdida puede conducir a la ambigüedad.

Si, por ejemplo, se está tratando de determinar las causas del éxito o el fracaso en una reacción química concreta, uno de los atributos a considerar podría ser la temperatura. Si la discretización se realiza de 20 en 20 grados centígrados, dos casos (ejemplos concretos) distintos en los que sólo varía la temperatura en 15 grados pasarían a ser considerados como uno solo. Si en uno hubo éxito y en el otro no, tras la discretización se tendrá que dos objetos idénticos pertenecen a clases distintas.

- **Conocimiento impreciso.** También puede suceder que no seamos capaces de distinguir dos objetos (de clases distintas) porque no se hayan tenido en cuenta atributos relevantes.

Volviendo al ejemplo anterior, quizá se pudiera observar que dos reacciones en las que las características consideradas coinciden producen resultados distintos debido a que se diferencian en un aspecto (la presión, por ejemplo) que se había pasado por alto.

- **Subjetividad de los expertos.** Existen situaciones en las que la asignación de los objetos a una u otra clase no se pueden realizar de forma inequívoca, sino que se necesita recurrir a una valoración (siempre subjetiva) de expertos humanos.

Por ejemplo, para determinar las condiciones en que una persona es aceptable para concederle un crédito, se pueden tomar casos pasados caracterizados por el salario percibido, los avales aportados, la cantidad de que se trata, etc., pero lo cierto es que la concesión o no del crédito depende, además, de un factor subjetivo, imponderable, que variará según quien sea la persona encargada de la concesión e incluso según sea su estado de ánimo en el momento de tomar la decisión.

Puede suceder, por tanto, que una misma persona tenga más suerte en una sucursal que en otra y que su caso sea clasificado como aceptable y no aceptable, respectivamente.

La ambigüedad se puede intentar evitar tratando cada uno de los problemas mencionados. Por ejemplo, se puede hacer un estudio concienzudo de los umbrales a considerar al discretizar un dominio continuo. Sin embargo, es imposible garantizar a priori la eliminación de la ambigüedad.

Otros sistemas tratan de eliminar los ejemplos erróneos [3], generan reglas a pesar de que existan contraejemplos [4], o utilizan diversos métodos de pre y post-poda [5][6].

El objetivo del presente trabajo no es esta eliminación sino la modificación del ID3 para que funcione aún cuando la premisa mencionada al principio de este apartado no se cumpla, es decir, aún cuando haya ambigüedad. La solución propuesta se desarrolla a continuación.

2.2 Solución mediante factores de certeza

Supongamos que en el proceso de generación de el árbol de decisión se crea un nodo en el que ya no falta ningún atributo por ser tratado pero que contiene elementos de ambas clases (+ y -). El procedimiento descrito en 1.2 no tiene previsto este caso, que refleja la ambigüedad a la que se ha hecho mención en el apartado anterior.

El tratamiento propuesto consiste en no asignar todos los nodos hoja categóricamente a una u otra clase sino permitir que algunos de dichos nodos pertenezcan a una clase con un determinado factor de certeza (FC). Si representamos dicho factor con un número comprendido entre 0 y 1, los valores extremos corresponden a la situación que se da en el ID3 clásico. Por ejemplo, $FC = 1$ puede significar que es un nodo + y $FC = 0$ que es un nodo -.

La diferencia es que ahora puede haber nodos con $FC = 0.9$, lo que significa que es muy probable que un objeto que corresponda a este tipo de nodos sea + (y muy poco probable que sea -). Los factores de certeza miden la confianza que se tiene en que los elementos correspondientes sean de una determinada clase, por ejemplo, de +. El valor $1 - FC$ mide la confianza que se tiene en que sean de la contraria.

Los FC 's se pueden aproximar mediante las frecuencias relativas. Consideremos que en un nodo de las características descritas al principio de este apartado hay x objetos que pertenecen a + e y objetos que pertenecen a -. El FC (para la clase +) sería

$$FC = \frac{x}{x + y}$$

Esta aproximación es consistente con lo dicho anteriormente. Si y es 0, todos los objetos son de la clase +, por lo que FC debe ser 1. Si x es 0, todos los objetos son de - y FC debe ser 0. Es fácil comprobar que estos son los resultados que arroja la fórmula.

Si tanto x como y son 0, la fórmula no es aplicable ya que se tiene un conjunto vacío y el nodo puede ser tratado como se vio en el segundo caso de 1.2, es decir, asignarle aleatoriamente $FC = 0$ (clase -) o $FC = 1$ (clase +). Sin embargo, parece más lógico asignarle siempre 0.5, ya que esto refleja la incertidumbre que se tiene sobre los objetos asociados, debido a la carencia de ejemplos de los mismos. Para cualquier otro valor (positivo) de x y de y , FC toma valores entre 0 y 1.

En resumen, si se admite que pueda existir ambigüedad en los ejemplos seleccionados, con cada nodo se pueden presentar cuatro casos. Los tres primeros coinciden con los descritos en 1.2:

1. Todos los objetos de $\mathcal{X}$ pertenecen a la misma clase. El nodo es una hoja cuyo FC será 1 si la clase es + y 0 si la clase es -.
2. El conjunto $\mathcal{X}$ asociado está vacío. El FC será 0.5.
3. $\mathcal{X}$ contiene objetos de ambas clases y quedan atributos por ser tratados. Se elige uno de estos atributos y se expande el nodo.

El cuarto caso es:

4. $\mathcal{X}$ contiene objetos de ambas clases y NO quedan atributos por ser tratados. Se calcula el factor de certeza asociado al nodo según la fórmula

$$FC = \frac{x}{x + y}$$

siendo x el número de objetos de la clase + e y el número de objetos de -.

La filosofía seguida para seleccionar atributos sigue siendo la misma, ya que la ambigüedad no influye en ella.

2.3 Flexibilización de los ejemplos

Una vez que se tiene en cuenta la consideración de los factores de certeza para la clasificación de los objetos, es posible flexibilizar la definición de esos mismos objetos admitiendo que su función de pertenencia a una clase tenga un dominio continuo entre cero y uno en lugar de considerarla bivaluada.

Esto permite aplicar ID3 a campos con conocimiento inexacto, en los que no se puede decidir si los objetos pertenecen a una clase determinada de forma categórica.

La clasificación de los ejemplos de partida ahora *no* se hace diciendo los que pertenecen a una u otra clase, si no que además se debe expresar el grado de pertenencia a la misma. La definición del conjunto $\mathcal{R}$ será por tanto:

$$\mathcal{R} = \{A_i B_j C_k \dots + \sigma_1^+, A_r B_s C_t \dots - \sigma_2^-, \dots\}$$

siendo σ_m^+ el grado de pertenencia del objeto m a la clase $+$ y σ_m^- el grado de pertenencia del objeto m a la clase $-$, $\sigma_m^+, \sigma_m^- \in [0, 1] \forall m$.

Para simplificar, se pueden expresar solamente los grados de pertenencia a una de las dos clases, a la $+$, por ejemplo. Estos grados coincidirán con los anteriores para los objetos en los que se expresa su pertenencia a $+$. En los restantes, se calculará como el complemento a 1 del grado de pertenencia a $-$, aplicando la fórmula habitual de la lógica borrosa. Es decir:

si se conoce el grado de pertenencia a $+$:

$$\mu_m = \sigma_m^+$$

si se conoce el grado de pertenencia a $-$:

$$\mu_m = 1 - \sigma_m^-$$

La definición de $\mathcal{R}$ queda:

$$\mathcal{R} = \{A_i B_j C_k \dots \mu_1 A_r B_s C_t \dots \mu_2 \dots\}$$

La única diferencia en lo que al tratamiento se refiere se encuentra en la forma de calcular $P(+)$ y $P(-)$. $P(+)$ (probabilidad de que un objeto de $\mathcal{R}$ pertenezca a $+$) se puede calcular como la media de los grados de pertenencia de los objetos de $\mathcal{R}$ a $+$:

$$P(+) = \frac{1}{\text{card}(\mathcal{R})} \cdot \sum_{\forall m \in \mathcal{R}} \mu_m$$

$P(-)$ será entonces la media de los grados de pertenencia de los objetos de $\mathcal{R}$ a $-$, es decir:

$$P(-) = \frac{1}{\text{card}(\mathcal{R})} \cdot \sum_{\forall m \in \mathcal{R}} (1 - \mu_m) = 1 - P(+)$$

Al tener los objetos grados de pertenencia a las clases, si se tienen n atributos, el árbol tendrá n niveles y siempre habrá que recorrer n nodos para realizar la clasificación a menos que se obtenga un nodo en un nivel superior con $FC = 0$ ó $FC = 1$. Esto sólo sucederá si el conjunto $\mathcal{R}$ asociado a un nodo tiene todos sus elementos con grados de pertenencia 0 o todos con grado de pertenencia 1. Si son pocos los objetos con estos valores como grados de pertenencia, es poco probable que aparezcan este tipo de nodos.

Esto hace que la elección del atributo para optimizar el árbol deje de ser crítica, ya que su tamaño raramente variará. Una mejora consiste en no expandir los nodos cuyos FC's estén muy próximos a 0 o a 1. Ya que estos FC's son frecuencias relativas, de algún modo están expresando la probabilidad de que al llegar a ese nodo el objeto que se está clasificando sea $+$. Estableciendo unos umbrales de poda (que pueden ser configurables dependiendo del dominio) se puede reducir el tamaño del árbol.

Por ejemplo, se puede evitar expandir nodos cuyos FC's sean mayores que 0.9 o menores que 0.1. En estos casos, la incertidumbre es muy pequeña y puede no merecer la pena crear un nivel más. Se utiliza aquí el principio de que, en ocasiones, es preferible encontrar una solución sencilla aproximada a encontrar la solución exacta, si ésta es mucho más compleja.

3 Conclusión

La variante del ID3 presentada en este trabajo afronta el problema de tener que tratar con dominios con ruido combinando las técnicas de inducción y manejo de incertidumbre. El resultado es un método sencillo de implementar que trata por igual a todos los ejemplos del conjunto de entrenamiento y que en ningún caso "falla" en la clasificación de posteriores objetos. La idoneidad de los resultados dependen, evidentemente de la representatividad del conjunto de ejemplos inicial.

La salida proporcionada por este algoritmo puede ser un árbol de decisión "borroso" o un conjunto de reglas de producción con factores de certeza asociados. En el caso del árbol, cada rama corresponde a un valor de un atributo y cada hoja tiene el grado de pertenencia a un conjunto (a la clase +, por ejemplo) de los objetos que presentan los valores encontrados en el camino desde la raíz hasta ella.

En el caso de las reglas de producción, sus partes izquierdas son conjunciones de selctores (atributo = valor) y sus partes derechas son la asignación a una clase determinada con un factor de certeza.

Además, este tratamiento permite que no sea imprescindible conocer con exactitud la filiación de los ejemplos de partida, sino que es suficiente conocer su grado de pertenencia a una clase. Esto posibilita el uso de conjuntos borrosos, por lo que el algoritmo podría ser llamado FID3 (Fuzzy ID3).

4 Referencias

1. Quinlan, J. (1986)
Induction of Decision Trees
Machine Learning vol 1
Kluwer
2. Bratko, I. (1987)
ASSISTANT 86: A Knowledge-Elicitation Tool for Sophisticated Users
Progress in Machine Learning
Sigma Press
3. Michalski, R. (1978)
Selection of most representative training examples and incremental generation of VL1 hypotheses
Univ. of Illinois
4. Clark, P. (1987)
Induction in Noisy Domains
Progress in Machine Learning
Sigma Press
5. Niblett, T.B. (1986)
Learning decision rules in noisy domains
Expert Systems, 86
Cambridge University Press
6. Bratko, I. (1986)
Learning diagnostic rules from incomplete and noisy data
AI Methods in Statistics
UNICOM Seminar, London

GRAMATICAS CON ATRIBUTOS: ESTUDIO DEL CASO FUZZY

Morales Bueno, Rafael(*)
 Clares Rodríguez, Buenaventura (**)
 Conejo Muñoz, Ricardo(*)
 Pérez de la Cruz Molina, José Luis(*)

(*)Dpto. Lenguajes y Ciencias de la Computación
 Facultad de Informática Universidad de Málaga.
 29071-Málaga.

(**)Dpto. Lenguajes y Sistemas Informáticos
 Facultad de Ciencias. Universidad de Granada.
 18071-Granada.

Resumen.- Las gramáticas de contexto libre difusas se pueden usar para describir lenguajes de programación con características difusas en su sintaxis. Las gramáticas con atributos permiten capturar algunos aspectos semánticos de los lenguajes de programación. Al considerar gramáticas con atributos difusos, estos atributos son útiles para expresar formalmente características semánticas difusas de lenguajes de programación.

El objetivo de esta comunicación es el estudio de las gramáticas de contexto libre (difusas o nítidas) con atributos (difusos o nítidos) y su aplicación a la descripción de la sintaxis y la semántica de un lenguaje de programación desarrollado por los autores.

Palabras clave: Fuzzy attribute grammars, Fuzzy programming.

1 NOTACION Y TERMINOLOGIA

En lo que sigue consideraremos que W es el semianillo ordenado $([0,1], \oplus, \otimes, \leq)$ donde se evalúan los grados [SAN77].

Usaremos W -conjuntos sobre $(Z^+)^n$, $n \geq 1$, con la representación estándar $A = \mu_A(a) / a + \dots + \mu_A(z) / z$. Si $a = (a_1, \dots, a_n) \in (Z^+)^n$ entonces π_i representa la i -ésima proyección $\pi_i(a) = a_i$.

Sean U y V conjuntos, $U, V \subseteq (Z^+)^n$. Consideramos que una W -función f de U en V es una función de $U \times V$ en W , y escribimos $f: U \times V \rightarrow W$, siendo $f(u,v)$ el peso o grado de que el valor de la función en u es v .

Traducimos el término inglés "crisp" por nítido.

2 INTRODUCCION

En este párrafo recordamos la definición de gramática de contexto libre difusa, según [LEE69], que denominaremos W -gramática, recordamos el concepto de W -derivación [MOR91a], introducimos el concepto de W -árbol de derivación y mostramos su equivalencia.

Definición 1.- Una gramática de contexto libre difusa (que llamaremos W -gramática) es una cuádrupla $\mathbb{G} = (N, T, \mathbb{R}, S)$ donde N (no terminales), T (terminales) y S (axioma, $S \in N$) tienen el significado habitual y $\mathbb{R}$ es un conjunto difuso sobre $N \times (N \cup T)^*$ tal que: $\forall A \in N$, $A \neq S$, $\mu_{\mathbb{R}}(A, \epsilon) = 0$ y $\mu_{\mathbb{R}}(S, \epsilon) = 1$ donde ϵ representa la cadena vacía. (Escribiremos V para $N \cup T$).

Notación: $\mu_{\mathbb{R}}(A, w) = \alpha$, con $A \in N, w \in V^*$, lo representaremos $A \xrightarrow{\alpha} w$, y diremos que $(A \xrightarrow{\alpha} w) \in \mathbb{R}$, y mas abreviadamente que $(A \xrightarrow{\alpha} w) \equiv r$ es una regla sintáctica.

Informalmente (ver el desarrollo formal en [MOR91a]) una W-derivación de u a v, siendo u y v cadenas en V^* , es una secuencia de reglas de la W-gramática que partiendo de u lleva a v, y su grado asociado se obtiene por aplicación del operador $\otimes$ a los grados de las reglas.

A continuación definimos el lenguaje (conjunto difuso) generado por una gramática difusa.

Definición 2.- Sea $\mathbb{G} = (N, T, \mathbb{R}, S)$ una W-gramática. El lenguaje generado por $\mathbb{G}$, $\mathcal{L}(\mathbb{G})$, es un conjunto difuso sobre T^* , tal que el grado de pertenencia de la cadena x es:

$$\mu_{\mathbb{G}}(x) = \sum (\mu_{\mathbb{R}}(\alpha_0, \alpha_1) \otimes \dots \otimes (\mu_{\mathbb{R}}(\alpha_m, \alpha_{m+1})),$$

donde el símbolo $\sum$, se refiere al operador $\oplus$ extendido a todas las posibles secuencias de reglas tales que $\alpha_0 = S$ (el axioma) y $\alpha_{m+1} = x$.

Se dice que dos W-gramáticas $\mathbb{G}_1$ y $\mathbb{G}_2$ son equivalentes sii $\forall x \in T^* : \mu_1(x) = \mu_2(x)$.

Citaremos a continuación algunos resultados relativos a W-gramáticas :

Proposición 1 [LEE69] .- Cada gramática $\mathbb{G}$ es equivalente a otra cuyas reglas son de la forma $A \xrightarrow{\alpha} BC$ o $A \xrightarrow{\beta} a$, $A, B, C \in N$, $a \in T$ (forma normal de Chomsky).

Proposición 2 [LEE69] .- Cada gramática $\mathbb{G}$ es equivalente a otra cuyas reglas son de la forma $A \xrightarrow{\alpha} aw$, $A \in N$, $a \in T$, $w \in T^*$ (forma normal de Greibach).

Proposición 3 [MIZ70].- Sea $\mathbb{G}$ una gramática de contexto libre difusa. Sea $\mathcal{L}(\mathbb{G})$ el lenguaje generado por $\mathbb{G}$. Entonces $\mathcal{L}(\mathbb{G}, \lambda) = \{x \in T^* \mid \mu_{\mathbb{G}}(x) > \lambda\}$ y $\mathcal{L}(\mathbb{G}, \geq, \lambda) = \{x \in T^* \mid \mu_{\mathbb{G}}(x) \geq \lambda\}$ son lenguajes nítidos de contexto libre.

A continuación presentamos, brevemente, la notación usada para W-árboles de derivación. Esta notación está adaptada de [LOE88].

D es un arbol base.

(D, e, gr) es un (S, W) -arbol, con $e: D \rightarrow S$, $gr: D \rightarrow W$, etiquetados del arbol base.

$subarb((D, e, gr), w)$ es el subarbol de (D, e, gr) en el nodo w.

$subst((D, e, gr), w, (D_1, e_1, gr_1))$ es el arbol obtenido sustituyendo, en (D, e, gr) , (D_1, e_1, gr_1) en el nodo w.

$frontera((D, e, gr))$ es la secuencia de etiquetas (e) de los nodos hoja, considerando los nodos del arbol base en orden lexicográfico.

Seguidamente vemos el concepto de W-arbol de derivación para las W-gramáticas. Este concepto es útil para posteriormente asociar las atributos a los nodos del arbol.

Definición 3.- Sea $\mathbb{G} = (N, T, \mathbb{R}, S)$ una gramática de contexto libre difusa, $V = N \cup T$; Sean $A \in N$ y $u \in V^*$. Decimos que el $(V \cup \{\epsilon\}, W)$ -arbol (D, e, gr) es un W-arbol de derivación de A a u en $\mathbb{G}$, sii

a) $e(\epsilon) = A$

b) Para cada $d \in D$ con hijos $d_1, d_2, \dots, d_k$ y $k \geq 1$:

o bien $(e(d) \xrightarrow{\alpha} e(d_1)e(d_2) \dots e(d_k)) \in \mathbb{R}$, y $e(d_i) \neq \varepsilon$, $1 \leq i \leq k$

o bien $k=1$, $e(d_1) = \varepsilon$ y $(e(d) \xrightarrow{\alpha} \varepsilon) \in \mathbb{R}$, .

y $gr(d) = \alpha$.

c) Para cada nodo hoja $d \in D$, $gr(d) = 1$.

d) frontera $((D, e, gr)) = u$.

Enunciamos, sin demostración el teorema que muestra la correspondencia entre W-derivaciones y W-arboles de derivación en el caso fuzzy.

Teorema 1.- Sea $\mathbb{G} = (N, T, \mathbb{R}, S)$ una W-gramática. Sean $A \in N$ y $u \in V^*$. Se verifican las siguientes afirmaciones:

a) Si hay una W-derivación de A a u con grado α , entonces existe un W-arbol de derivación (D, e, gr) de A a u en $\mathbb{G}$, y tal que $\alpha = \bigotimes_{d \in D} gr(d)$

b) Si hay un W-arbol de derivación (D, e, gr) de A a u en $\mathbb{G}$, entonces existe una W-derivación de A a u con grado $\bigotimes_{d \in D} gr(d)$

3 W-GRAMATICAS CON ATRIBUTOS

Los conceptos y resultados que exponemos en este párrafo son válidos para los casos nítido y difuso, por ello no haremos referencia a esta distinción. Para llegar al concepto de W-gramática con atributos, veamos en primer lugar el concepto de regla semántica.

Definición 4.- Sea $\mathbb{G} = (N, T, \mathbb{R}, S)$ una W-gramática, $V = N \cup T$. Sea Att un conjunto finito $Att = \{a_1, \dots, a_n\}$, compuesto por los atributos de los miembros de V . Para cada $X \in V$, sea $Att(X)$ un subconjunto de Att (formado por los atributos de X), y para cada $a_i \in Att$, $i=1, \dots, n$, sea D_i el dominio de a_i . Sea r una regla dada por $X_\varepsilon \xrightarrow{\alpha} X_1 \dots X_k$, con $X_\varepsilon \in N$, $X_j \in V$, $j=1, \dots, k$. Definimos una regla semántica asociada a r como un par ordenado (f, A) donde f es una función, $f: D_{i_1} \times \dots \times D_{i_p} \rightarrow D_{i_0}$, $i_0, \dots, i_p \in \{1, \dots, n\}$, $p > 0$; y $A \in (Att \times \{\varepsilon, 1, \dots, k\})^{p+1}$,

$A = \left((a_{i_0}, j_0), \dots, (a_{i_p}, j_p) \right)$, $j_0, \dots, j_p \in \{\varepsilon, 1, \dots, k\}$, tal que para cada $h = 0, \dots, p$, $a_{i_h} \in Att(X_{j_h})$; y si $p=0$, entonces $f \in D_{i_0}$

Definición 5.- Una W-gramática con atributos $\mathbb{AG}$ es una 6-tupla $\mathbb{AG} = (\mathbb{G}, Att, \mathbb{D}, \mathbb{X}, \mathbb{C}, \mathbb{F})$ donde:

$\mathbb{G} = (N, T, \mathbb{R}, S)$ es una W-gramática.

Att es un conjunto finito formado por los atributos de los elementos de V .

$\mathbb{D} = \{ D_a / a \in Att \}$ (Dominios de los atributos)

$\mathbb{X} = \{ Att(X) / X \in V, Att(X) \subseteq Att \}$ and $Att = \bigcup \{ Att(X) / X \in V \}$

$\mathbb{C} = \{ H(X), S(X) / X \in V, H(X) \cap S(X) = \emptyset \text{ y } H(X) \cup S(X) = Att(X) \}$

$\mathbb{F} = \{ \mathbb{F}_r / r \text{ regla en } \mathbb{G} \}$, tal que, si $X_\varepsilon \xrightarrow{\alpha} X_1 \dots X_k$ es una regla sintáctica en $\mathbb{G}$, entonces se verifican las dos condiciones siguientes:

1) Para cada $a \in S(X_\varepsilon)$, existe una única regla semántica $(f, A) \in \mathbb{F}_r$,

$f: D_{i_1} \times \dots \times D_{i_p} \rightarrow D_{i_0}$, $A = \left((a_{i_0}, j_0), \dots, (a_{i_p}, j_p) \right)$, tal que $a_{i_0} = a$ y $j_0 = \varepsilon$

2) Para cada $a' \in H(X_i)$, $i=1, \dots, k$, existe una única regla semántica $(f, A) \in \mathbb{F}_r$,

$f : D_{i_1} \times \dots \times D_{i_p} \rightarrow D_{i_0}, A = \left((a_{i_0}, j_0), \dots, (a_{i_p}, j_p) \right)$, tal que $a_{i_0} = a'$, y $j_0 = i$

Interpretación: Dado $X \in V$, los elementos de $H(X)$ son los atributos heredados de X y los elementos de $S(X)$ son los atributos sintetizados de X .

$\mathcal{F}$ es el conjunto de reglas semánticas.

Las dos condiciones exigidas nos dicen que hay una única regla semántica para cada atributo heredado y para cada atributo sintetizado.

Veamos como las reglas semánticas nos permiten asignar valores a los atributos de los símbolos que aparecen en un W -árbol de derivación.

Definición 6.- Sea $\mathcal{AG} = (\mathcal{G}, \text{Att}, \mathcal{D}, \mathcal{X}, \mathcal{U}, \mathcal{F})$ una W -gramática con atributos y (D, e, gr) el W -árbol de derivación de una sentencia respecto a la W -gramática subyacente $\mathcal{G}$.

Para cada $a \in \text{Att}$, sea $D(a) = \{k \in D / a \in \text{Att}(e(k))\}$.

Para cada $a \in \text{Att}$, sea $val_a: D(a) \rightarrow D_a$ una aplicación.

Sea $X \xrightarrow{\alpha} X_1 \dots X_k$ la regla r de $\mathcal{G}$ usada en el nodo q . Diremos que la regla semántica $(f, A) \in \mathcal{F}_r$ $f : D_{i_1} \times \dots \times D_{i_p} \rightarrow D_{i_0}, A = \left((a_{i_0}, j_0), \dots, (a_{i_p}, j_p) \right)$, se satisface si :

$$val_{a_{i_0}}(q_{j_0}) = f(val_{a_{i_1}}(q_{j_1}), \dots, val_{a_{i_p}}(q_{j_p}))$$

Interpretación: El conjunto $D(a)$ es el conjunto de los nodos de D que se corresponden, por e , con símbolos de la W -gramática que poseen el atributo a . La aplicación val_a asigna a cada uno de estos nodos un valor del dominio D_a del atributo a .

Observe que las aplicaciones val_a constituyen etiquetados parciales del árbol base D .

Entonces, una regla semántica (f, A) se satisface si los valores asociados a los atributos son coherentes con la evaluación de la función f para esos valores. También, la satisfacción de una regla semántica se puede ver como una forma de calcular el valor de un atributo, para un nodo, a partir de los valores de atributos para otros nodos o como una validación semántica de la construcción, en caso de que se conozcan todos los valores de los atributos que interviene en la regla semántica.

Definición 7.- Sea $\mathcal{AG} = (\mathcal{G}, \text{Att}, \mathcal{D}, \mathcal{X}, \mathcal{U}, \mathcal{F})$ una W -gramática con atributos y (D, e, gr) el W -árbol de derivación de una sentencia respecto a la W -gramática subyacente $\mathcal{G}$. Una familia de etiquetas $\{val_a: D(a) \rightarrow D_a / a \in \text{Att}\}$ del árbol D es una "anotación de valores de atributos" (en adelante ava) respecto a $\mathcal{AG}$.

Definición 8.- Sea $\mathcal{AG} = (\mathcal{G}, \text{Att}, \mathcal{D}, \mathcal{X}, \mathcal{U}, \mathcal{F})$ una W -gramática con atributos y (D, e, gr) el W -árbol de derivación de una sentencia respecto a la W -gramática subyacente $\mathcal{G}$. Una ava para el árbol (D, e, gr) es consistente respecto a $\mathcal{AG}$ si y solo si todas las reglas semánticas se satisfacen.

Considerando las reglas semánticas como ecuaciones entre los valores de los atributos en los distintos nodos de un árbol (D, e, gr) , una ava consistente es una solución de este sistema de ecuaciones, lo que validará la instancia concreta de las reglas semánticas involucradas.

Un algoritmo que permita alcanzar una ava se puede describir como sigue:

ALGORITMO DE EVALUACION DE ATRIBUTOS

- 1) Todas las funciones $val_a: D^{(a)} \rightarrow D_a$ están totalmente indefinidas.
- 2) Sea $q \in D$ y $X_e \xrightarrow{\alpha} X_1 \dots X_k$ la regla r de $\mathcal{G}$ usada en el nodo q . Si $(f, A) \in \mathcal{F}_r$
 $f: D_{i_1} \times \dots \times D_{i_p} \rightarrow D_{i_0}$, $A = \left((a_{i_0}, j_0), \dots, (a_{i_p}, j_p) \right)$,
 y están definidos $val_{a_{i_1}}(q_{j_1}), \dots, val_{a_{i_p}}(q_{j_p})$, entonces definir
 $val_{a_{i_0}}(q_{j_0}) = f(val_{a_{i_1}}(q_{j_1}), \dots, val_{a_{i_p}}(q_{j_p}))$
- 3) Repetir el paso 2 hasta que las etiquetas val_a no cambien.

Es claro que mediante esta algoritmo, si se comprueba que todas las etiquetas val_a son funciones totales, entonces para todos los nodos del árbol, todos sus atributos tendrán valores asignados y además satisfaciendo todas las reglas semánticas. Por tanto tenemos:

Teorema 2.- Sea $\mathcal{AG}$ una W -gramática con atributos y (D, e, gr) un W -árbol de derivación con respecto a la W -gramática $\mathcal{G}$ subyacente. Si el algoritmo de evaluación de atributos lleva a funciones totales val_a , para cada atributo a , entonces existe una única ava consistente del W -árbol (D, e, gr) .

4 APLICACIONES

En este párrafo desarrollamos dos ejemplos en los que se aplican los desarrollos realizados en los párrafos anteriores. El primero de ellos es muy simple y considera una situación en la que usamos una W -gramática con atributos difusos. El segundo, mas complejo y práctico, describe un sublenguaje (llamado L_0) de un lenguaje desarrollado por los autores. Ilustraremos la utilidad de los atributos mediante el desarrollo de la ava consistente para el árbol de derivación de un program en L_0 . En este último caso la gramática es nítida y los atributos difusos.

4.1 Expresiones aditivas

Vamos a considerar una W -gramática con atributos difusos, que nos permite evaluar expresiones numéricas con sumas y restas.

Sea la 6-tupla $\mathcal{AG} = (\mathcal{G}, Att, \mathcal{D}, \mathcal{X}, \mathcal{C}, \mathcal{F})$ donde tenemos que:

$\mathcal{G} = (N, T, \mathcal{R}, E)$ W -gramática con:

$N = \{E, ?\}$

$T = \{+, -, \text{"numero"}\}$

donde "numero" representa un número difuso en Z^+

$\mathcal{R}$ está formado por las siguientes reglas:

$E \rightarrow E+E$

$E \rightarrow E-E$

$E \rightarrow E?E$

$? \xrightarrow{\alpha} +$

$? \xrightarrow{\beta} -$

$E \rightarrow \text{"numero"}$

donde las reglas en las que no aparece grado se entiende grado 1.

$Att = \{V, sg\}$

$\mathcal{D} = \{D_v, D_{sg}\}$, donde $D_v = \{\text{numeros difusos en } Z^+\}$ y $D_{sg} = \{+1, -1\}$

Los elemetos del conjunto $\mathcal{X}$ son los siguientes :

$\text{Att}(E) = \{V\}$, $\text{Att}(?) = \{sg\}$, el resto son el conjunto vacío.

Los elementos del conjunto $\mathcal{U}$ son los siguientes :

$S(E) = \{V\}$, $S(?) = \{sg\}$, el resto son el conjunto vacío.

Los elementos del conjunto $\mathcal{F}$ son las reglas semánticas asociadas a cada una de las reglas sintácticas de la gramática. Estas reglas semánticas junto con la regla sintáctica a la que se aplica, son las siguientes:

$E \rightarrow E+E$, $A = ((V, \epsilon), (V, 1), (V, 3))$, $f: D_V \times D_V \rightarrow D_V$, $f(x, y) = x+y$

$E \rightarrow E-E$, $A = ((V, \epsilon), (V, 1), (V, 3))$, $f: D_V \times D_V \rightarrow D_V$, $f(x, y) = x-y$

$E \rightarrow E?E$, $A = ((V, \epsilon), (V, 1), (sg, 2), (V, 3))$,
 $f: D_V \times D_{sg} \times D_V \rightarrow D_V$, $f(x, a, y) = x+ay$

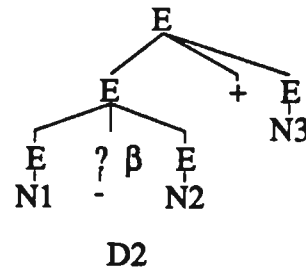
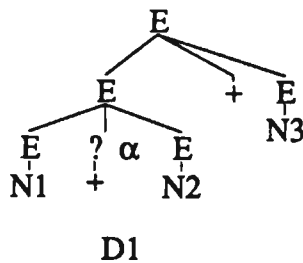
$? \xrightarrow{\alpha} +$ $A = ((sg, \epsilon))$ $f = +1$

$? \xrightarrow{\beta} -$ $A = ((sg, \epsilon))$ $f = -1$

$E \rightarrow \text{"numero"}$ $A = ((V, \epsilon))$ $f = \text{numero}$.

Es fácil ver que esta 6-tupla cumple las condiciones de una W-gramática con atributos, y además que el algoritmo de evaluación de atributos lleva a funciones totales val_a , lo que por el teorema 2 nos asegura que para cada W-arbol de derivación hay una ava consistente.

Consideremos ahora que $N1$, $N2$ y $N3$ son números difusos en Z^+ y que deseamos calcular la expresión $N1 \clubsuit N2 + N3$, donde el signo $\clubsuit$ podemos considerarlo debido a que en el original que hemos recibido no se entiende si es "+" o "-". Vamos a aplicar nuestra gramática para evaluar esta expresión. Para ello, debido a la falta de información sobre el signo que hay entre $N1$ y $N2$, consideramos los posibles W-arboles de derivación que son los siguientes:



donde el primer árbol D1 tiene grado α y el segundo árbol D2 tiene grado β .

Mediante el algoritmo de evaluación de atributos aplicado a estos W-arboles, obtenemos para el atributo V en el nodo raíz de cada árbol.

$$V1 = N1 + N2 + N3$$

$$V2 = N1 - N2 + N3$$

Teniendo en cuenta el grado de cada árbol, consideramos como valor de la expresión el conjunto difuso de nivel 2 siguiente: $V = \alpha/V1 + \beta/V2$.

Considerando, en este caso concreto, que $V1$ y $V2$ son números difusos, se podría obtener para la expresión propuesta un único valor difuso multiplicando los grados de $V1$ por α , los grados de $V2$ por β y sumando los números difusos resultantes.

4.2 El lenguaje L_0 .

Veremos en primer lugar la descripción del lenguaje mediante una gramática (nítida) con atributos difusos y posteriormente un ejemplo detallado de evaluación de atributos para un programa en este lenguaje.

Sea la 6-tupla $AG_0 = (G_0, Att, \mathbb{D}, \mathfrak{X}, \mathfrak{T}, \mathfrak{J})$ donde tenemos que:

$G_0 = (N_0, T_0, R_0, S_0)$ es la gramática libre de contexto siguiente:

$N_0 = \{ \langle letra \rangle, \langle no-cero \rangle, \langle digito \rangle, \langle numero \rangle, \langle var \rangle, \langle asig \rangle, \langle sent \rangle, \langle sents \rangle, \langle F \rangle, \langle programa \rangle \}$

$T_0 = \{ X, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :=, fuzz, (,), suc, :, inicio, fin \}$

El conjunto R_0 de reglas de la gramática, expresado en notación EBNF, es el siguiente:

$\langle letra \rangle ::= X$
 $\langle no-cero \rangle ::= 1/2/3/4/5/6/7/8/9$
 $\langle digito \rangle ::= \langle no-cero \rangle / 0$
 $\langle numero \rangle ::= \langle no-cero \rangle / \langle numero \rangle \langle digito \rangle$
 $\langle var \rangle ::= \langle letra \rangle \langle numero \rangle$
 $\langle asig \rangle ::= \langle var \rangle := 0 / \langle var \rangle := \langle F \rangle (\langle var \rangle) / \langle var \rangle := suc(\langle var \rangle)$
 $\langle F \rangle ::= fuzz \langle numero \rangle$
 $\langle sent \rangle ::= \langle asig \rangle$
 $\langle sents \rangle ::= \langle sent \rangle / \langle sents \rangle ; \langle sent \rangle$
 $\langle programa \rangle ::= inicio fin / inicio \langle sents \rangle fin$

$S_0 = \langle programa \rangle$

$Att = \{ max, v, ind, coin, cofi \}$

El significado de los elementos de Att se detallará mas adelante. Sin embargo damos ahora una breve idea: max representa el número máximo de variables usadas hasta ese momento, v el valor de un número, ind el índice de una variable, $coin$ configuración inicial y $cofi$ configuración final.

$\mathbb{D} = \{ \mathbb{Z}^+, D \}$

donde $D_{max} = D_v = D_{ind} = \mathbb{Z}^+$, el conjunto de los enteros positivos.

Sea D_k el conjunto de los subconjuntos difusos de $(\mathbb{Z}^+)^k$, entonces $D = \bigcup_{k \geq 0} D_k$

$D_{coin} = D_{cofi} = D$.

Los elementos del conjunto $\mathfrak{X}$ son los conjunto siguientes:

$Att(\langle no-cero \rangle) = Att(\langle digito \rangle) = Att(\langle numero \rangle) = \{ v \}$
 $Att(\langle var \rangle) = Att(\langle F \rangle) = \{ ind \}$
 $Att(\langle asig \rangle) = Att(\langle sent \rangle) = Att(\langle sents \rangle) = Att(\langle programa \rangle) = \{ max, coin, cofi \}$
 el resto son el conjunto vacio.

Los elementos del conjunto $\mathfrak{T}$ son los siguientes:

$S(\langle no-cero \rangle) = S(\langle digito \rangle) = S(\langle numero \rangle) = \{ v \}$
 $S(\langle var \rangle) = S(\langle F \rangle) = \{ ind \}$
 $H(\langle asig \rangle) = H(\langle sent \rangle) = H(\langle sents \rangle) = \{ coin \}$
 $S(\langle programa \rangle) = \{ max, coin, cofi \}$
 $S(\langle asig \rangle) = S(\langle sent \rangle) = S(\langle sents \rangle) = \{ max, cofi \}$
 el resto son el conjunto vacio.

Los elementos de $\mathcal{F}$ son las reglas semánticas. Estas, junto con la regla sintáctica a la que se aplica, son las que siguen:

$\langle \text{programa} \rangle \rightarrow \text{inicio fin}$

- 1) $A = ((\text{max}, \epsilon)) \quad f \in Z^+$
- 2) $A = ((\text{coin}, \epsilon), (\text{max}, \epsilon)) \quad f: Z^+ \rightarrow D, f(a) \in D_a$
- 3) $A = ((\text{cofi}, \epsilon), (\text{coin}, \epsilon)) \quad f: D \rightarrow D, f = \text{ID}$

$\langle \text{programa} \rangle \rightarrow \text{inicio} \langle \text{sents} \rangle \text{fin}$

- 1) $A = ((\text{max}, \epsilon), (\text{max}, 2)) \quad f: Z^+ \rightarrow Z^+, f = \text{ID}$
- 2) $A = ((\text{coin}, \epsilon), (\text{max}, \epsilon)) \quad f: Z^+ \rightarrow D, f(a) \in D_a$
- 3) $A = ((\text{coin}, 2), (\text{coin}, \epsilon)) \quad f: D \rightarrow D, f = \text{ID}$
- 4) $A = ((\text{cofi}, \epsilon), (\text{cofi}, 2)) \quad f: D \rightarrow D, f = \text{ID}$

$\langle \text{sents} \rangle \rightarrow \langle \text{sent} \rangle$

- 1) $A = ((\text{max}, \epsilon), (\text{max}, 1)) \quad f: Z^+ \rightarrow Z^+, f = \text{ID}$
- 2) $A = ((\text{coin}, 1), (\text{coin}, \epsilon)) \quad f: D \rightarrow D, f = \text{ID}$
- 3) $A = ((\text{cofi}, \epsilon), (\text{cofi}, 1)) \quad f: D \rightarrow D, f = \text{ID}$

$\langle \text{sents} \rangle \rightarrow \langle \text{sents} \rangle; \langle \text{sent} \rangle$

- 1) $A = ((\text{max}, \epsilon), (\text{max}, 1), (\text{max}, 3)) ; f: Z^+ \times Z^+ \rightarrow Z^+ : f(a,b) = \max(a,b)$
- 2) $A = ((\text{coin}, 3), (\text{cofi}, 1)) \quad f: D \rightarrow D, f = \text{ID}$
- 3) $A = ((\text{coin}, 1), (\text{coin}, \epsilon)) \quad f: D \rightarrow D, f = \text{ID}$
- 4) $A = ((\text{cofi}, \epsilon), (\text{cofi}, 3)) \quad f: D \rightarrow D, f = \text{ID}$

$\langle \text{sent} \rangle \rightarrow \langle \text{asig} \rangle$

- 1) $A = ((\text{max}, \epsilon), (\text{max}, 1)) \quad f: Z^+ \rightarrow Z^+, f = \text{ID}$
- 2) $A = ((\text{coin}, 1), (\text{coin}, \epsilon)) \quad f: D \rightarrow D, f = \text{ID}$
- 3) $A = ((\text{cofi}, \epsilon), (\text{cofi}, 1)) \quad f: D \rightarrow D, f = \text{ID}$

$\langle \text{asig} \rangle \rightarrow \langle \text{var} \rangle := 0$

- 1) $A = ((\text{max}, \epsilon), (\text{ind}, 1)) \quad f: Z^+ \rightarrow Z^+, f = \text{ID}$
- 2) $A = ((\text{cofi}, \epsilon), (\text{coin}, \epsilon), (\text{ind}, 1)) \quad f: D \times Z^+ \rightarrow D \quad f(A,i) = B \text{ si se verifica que: Para cada } a, \text{ con } \mu_A(a) = \alpha, \exists b, \text{ con } \mu_B(b) = \beta \text{ tal que } (\pi_i(b) = 0) \wedge (\forall j \neq i \pi_j(b) = \pi_j(a)) \wedge (\beta = \alpha)$

$\langle F \rangle \rightarrow \text{fuzz} \langle \text{numero} \rangle$

$$A = ((\text{ind}, \epsilon), (\text{v}, 2)) \quad f: Z^+ \rightarrow Z^+, f = \text{ID}$$

$\langle \text{asig} \rangle \rightarrow \langle \text{var} \rangle := \text{suc}(\langle \text{var} \rangle)$

- 1) $A = ((\text{max}, \epsilon), (\text{ind}, 1), (\text{ind}, 5)) \quad f: Z^+ \times Z^+ \rightarrow Z^+ \quad f(a,b) = \max(a,b)$
- 2) $A = ((\text{cofi}, \epsilon), (\text{coin}, \epsilon), (\text{ind}, 1), (\text{ind}, 5)) ; f: D \times Z^+ \times Z^+ \rightarrow D \quad f(A,i,j) = B \text{ ssi se verifica que: Para cada } a, \mu_A(a) = \alpha, \exists b, \text{ con } \mu_B(b) = \beta \text{ tal que } (\pi_i(b) = \pi_j(a) + 1) \wedge (\forall k \neq i \pi_k(b) = \pi_k(a)) \wedge (\beta = \alpha)$

$\langle \text{asig} \rangle \rightarrow \langle \text{var} \rangle := \langle F \rangle(\langle \text{var} \rangle)$

- 1) $A = ((\text{max}, \epsilon), (\text{ind}, 1), (\text{ind}, 5)) \quad f: Z^+ \times Z^+ \rightarrow Z^+ \quad f(a,b) = \max(a,b)$

$$2) A = ((cofi, \epsilon), (coin, \epsilon), (ind, 1), (ind, 3), (ind, 5))$$

$f: D \times Z^+ \times Z^+ \times Z^+ \rightarrow D$ $f(A, i, j, k) = B$ si se verifica que : Para cada a , con $\mu_A(a) = \alpha$

$$\exists B_a = \sum_{\lambda} \beta_{\lambda} / b_{\lambda}, \text{ tal que } (\pi_i(b_{\lambda}) = d_{\lambda}) \wedge (\forall h \neq i \pi_h(b_{\lambda}) = \pi_h(a)) \wedge (\beta_{\lambda} = \alpha \otimes f_j(\pi_k(a), d_{\lambda})) \text{ y}$$

$$B = \bigcup_a B_a, \text{ donde } f_j: Z^+ \times Z^+ \rightarrow W \text{ es una función } W\text{-calculable externa en el sentido de [CLA87]}$$

$$\langle var \rangle \rightarrow \langle letra \rangle \langle numero \rangle$$

$$A = ((ind, \epsilon), (v, 2)) \quad f: Z^+ \rightarrow Z^+, \quad f = ID$$

$$\langle numero \rangle \rightarrow \langle numero \rangle \langle digito \rangle$$

$$A = ((v, \epsilon), (v, 1), (v, 2)) \quad f: Z^+ \times Z^+ \rightarrow Z^+ \quad f(a, b) = 10a + b$$

$$\langle numero \rangle \rightarrow \langle no-cero \rangle$$

$$A = ((v, \epsilon), (v, 1)) \quad f: Z^+ \rightarrow Z^+, \quad f = ID$$

$$\langle digito \rangle \rightarrow \langle no-cero \rangle$$

$$A = ((v, \epsilon), (v, 1)) \quad f: Z^+ \rightarrow Z^+, \quad f = ID$$

$$\langle digito \rangle \rightarrow 0$$

$$A = ((v, \epsilon)) \quad f = 0$$

$$\langle no-cero \rangle \rightarrow i \quad (i = 1, \dots, 9)$$

$$A = ((v, \epsilon)) \quad f = i.$$

Teorema 3.- La 6-tupla $AG_0 = (G_0, Att, \mathbb{D}, \mathbb{X}, \mathbb{U}, \mathbb{J})$ establecida en el párrafo anterior es una W-gramática con atributos.

Demostración: La estructura definida cumple con las características de las gramáticas con atributos pues:

a) Los atributos heredados y sintetizados constituyen una partición del conjunto de atributos de cada símbolo de la gramática.

b) La existencia de una única regla semántica para cada atributo sintetizado del antecedente de una regla sintáctica y para cada atributo heredado del consecuente se cumple por inspección exhaustiva de todas las reglas semánticas.

Teorema 4.- Sea AG_0 la gramática con atributos descrita en el teorema anterior. Para cada (D, e) árbol de derivación respecto a la gramática subyacente G_0 , el "algoritmo de evaluación de atributos" lleva a funciones totales val_a , para cada atributo a .

Demostración: La haremos de manera constructiva comenzando de abajo a arriba en el árbol.

Para cada etiqueta (por el etiquetado e) $X_n, n \in Z^+$, que aparece en el árbol, mediante las reglas semánticas para el atributo v obtenemos asociado a n su valor como entero positivo y a X_n ese mismo valor como atributo ind . En este punto, las funciones val_v y val_{ind} son totales.

En los subárboles con raíz $\langle asig \rangle$, asociamos a este nodo el atributo max como el máximo de los índices de las variables y análogamente para $\langle sent \rangle$, $\langle sents \rangle$ y $\langle programa \rangle$, calculándose en este caso el máximo de los atributos max de sus hijos.

En este momento la función val_{max} es total.

Seguidamente, asociamos a la raíz del árbol ($\langle programa \rangle$) el atributo $coin$, que se hereda recorriendo el árbol en profundidad por la izquierda, hasta alcanzar un nodo $\langle asig \rangle$, lo que provoca el cálculo del atributo $cofi$ para ese nodo, y, en el caso de existir una rama derecha, este valor se transmite como valor del atributo $coin$ del nodo últimamente mencionado.

Los cálculos del atributo $cofi$ concluyen cuando llegamos de nuevo a la raíz, momento en el cual las funciones val_{coin} y val_{cofi} son totales.

Por tanto, el "algoritmo de evaluación de atributos" lleva a funciones totales val_a , para cada atributo a .

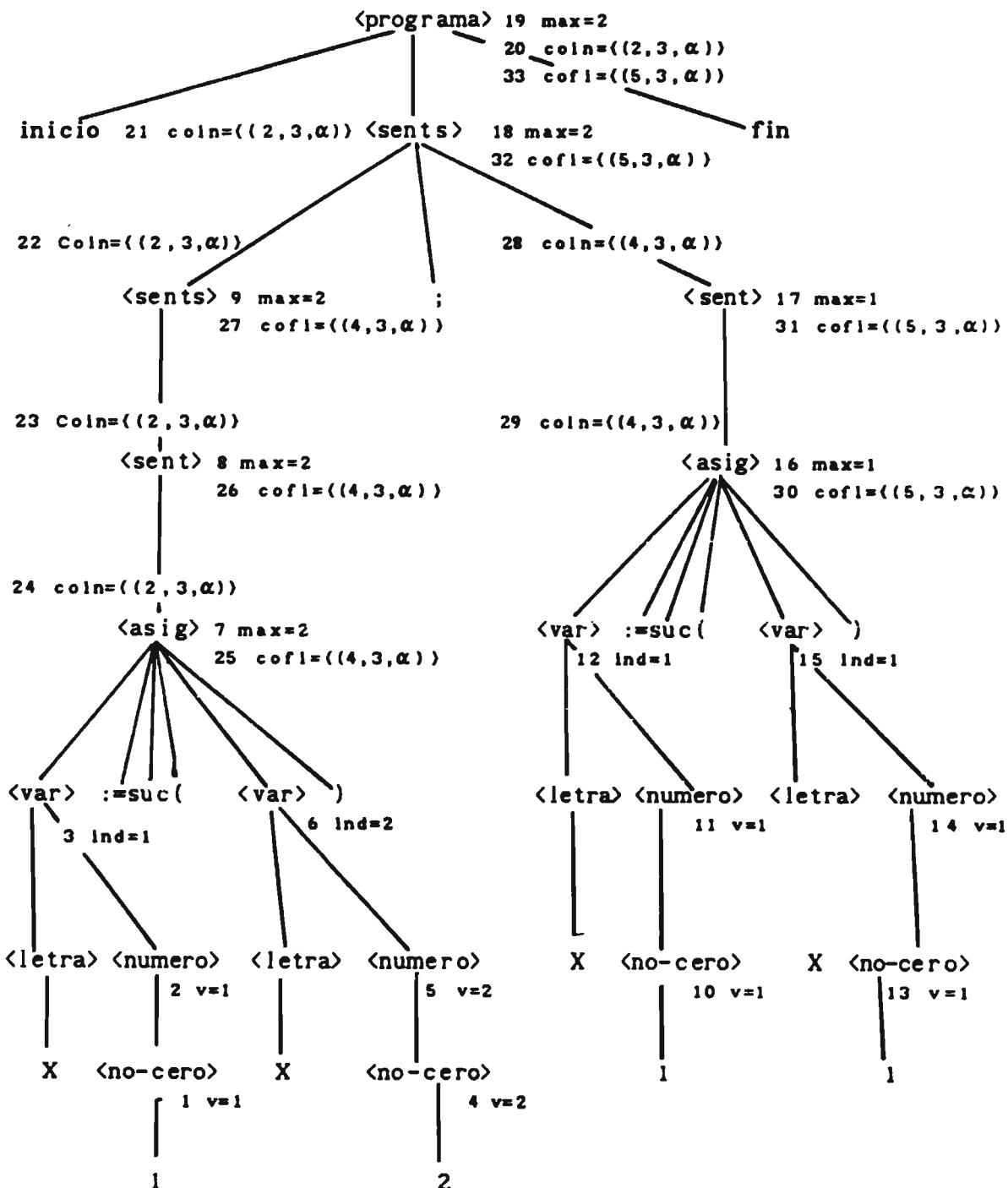
Corolario 1.- Sea AG_0 la gramática con atributos descrita en las páginas anteriores. Para cada (D,e) árbol de derivación respecto a la gramática subyacente G_0 , existe una ava consistente del mismo.

Demostración: Por los teoremas 2 y 4.

Veamos ahora un ejemplo de evaluación de atributos para un programa muy simple
Ejemplo: Sea el siguiente programa en L_0 :

```

inicio
    X1 := suc(X2);
    X1 := suc(X1)
fin
  
```



La gráfica muestra su árbol de derivación con una ava consistente.

La aplicación del algoritmo de evaluación de atributos, queda mostrada en la gráfica por los números a la izquierda de los valores de los atributos, que indican el número de orden de dicha evaluación.

En la gráfica, la notación (a,b,α) representa el elemento (a,b) con grado α (esto es: $\alpha/(a,b)$).

Notese que el lenguaje descrito no tiene sentencias de entrada ni de salida, así pues, un programa en L_0 expresa una relación funcional entre los valores de los atributos coin y cofi en el nodo raíz. En nuestro ejemplo, para $\text{coin} = \{\alpha/(a,b)\}$ se obtiene $\text{cofi} = \{\alpha/(b+2,b)\}$.

5 CONCLUSIONES Y DESARROLLOS FUTUROS

En este trabajo hemos visto como se pueden extender las gramáticas con atributos al caso difuso. Así mismo, se conserva la utilidad del caso nítido, para describir formalmente la semántica de lenguajes de programación difusa. Hemos ilustrado esta utilidad mediante la descripción completa de un lenguaje (L_0).

Una línea que puede ser interesante es considerar en las reglas semánticas (f,A) , que las f sean W-funciones; en este caso la evaluación de atributos se haría de manera difusa, dando este sentido intrínseco a las instrucciones, sin necesidad de que aparezcan explícitamente sentencias de asignación difusa.

En otra dirección, asociada al lenguaje L_0 , a partir de los atributos coin y cofi, se ha desarrollado una semántica operacional para este lenguaje; de manera que la gramática con atributos AG_0 permite describir toda la semántica estática del lenguaje.

Una ampliación del lenguaje L_0 es el lenguaje L , que está dotado de una instrucción de bucle indefinido. Este lenguaje L conserva las mismas propiedades que L_0 y además el tipo de funciones que se pueden calcular con él presentan una estrecha relación con la clase de funciones W-recursivas primitivas [CLA87].

Posterioros desarrollos llevarán a la inclusión en el lenguaje L de sentencias de bucle indefinido.

REFERENCIAS

- [CLA87]B. Clares and M.Delgado "Introduction of the concept of recursiveness of fuzzy functions", Fuzzy Sets and Systems 21 (1987), pp. 301-310
- [LEE69]E.T. Lee and L.A. Zadeh "Note on fuzzy languages", Inf. Sci. (1969), pp. 421-434
- [LOE88]J. Loeckx, K. Mehlhorn and R. Wilhelm "Foundations of programming languages", John Wiley & Sons 1988
- [MIZ70]M. Mizumoto, J. Toyoda and K. Tanaka "Fuzzy languages", Syst. Comput. Control, vol. 1 (1970), pp. 36-43
- [MOR91a]R. Morales and B. Clares "Calculabilidad mediante gramáticas difusas" Primer Congreso Español sobre Tecnologías y Lógica fuzzy, Granada 1991, pp. 167-172
- [MOR91b]R. Morales, Tesis Doctoral, Universidad de Málaga (1991)
- [SAN77]E. S. Santos "Fuzzy and probabilistic programs" In "Fuzzy automata and decision processes" M.M. Gupta, G.N. Saridis, B.R. Gaines eds. North-Holland (1977), pp.133-148

Interface entre BD Relacionales Difusas y el entorno de Prolog

O. Pons y M.A. Vila

Depto. de Ciencias de la Computación
e Inteligencia Artificial
Campus Universitario Fuentenueva
Facultad de Ciencias
-18071- GRANADA

Resumen:

Dado que los enfoques lógicos para BD clásicas ofrecen muchas ventajas y permiten una nueva visión de las mismas, seguiremos en el marco de la lógica para tratar las BD difusas.

Se trata, pues, de aprovechar los beneficios de ambas aproximaciones e integrar la lógica y las BD difusas para generar, en lo posible, nuevos sistemas. Se plantea pues la implementación de un modelo lógico de BD relacional en un sistema computacional que ofrezca las necesarias garantías de flexibilidad y potencialidad. Con esta idea, nos parece muy apropiada la elección del lenguaje Prolog y su entorno, ya que se adapta perfectamente a nuestras necesidades y nos permite el tratamiento casi directo de la representación lógica de una BD.

Palabras Clave: Base de Datos Difusa, Base de Datos Lógica, Prolog y Bases de Datos, Implementación Lógica de Base de Datos.

1. Introducción

Es un hecho sabido que el modelo relacional clásico de Bases de Datos ([Codd, 1974]), a pesar de ser el más extendido, resulta, en algunos casos, restrictivo y poco flexible dado que, en la vida real, los datos (o bien su manipulación) pueden ser imprecisos, mal conocidos, dinámicos, subjetivos,...

En un primer intento de subsanar esta deficiencia del modelo relacional clásico, la evolución de las Bases de Datos está orientándose hacia el uso de la lógica. Algunos autores ([Kowalski, 1981], [Reiter, 1984] y [Gallaire et al., 1984]) han estado estudiando la teoría de las Bases de Datos a la luz de la lógica con el objetivo de acomodar una mayor parte del mundo real en el sistema, al resultar ésta más flexible y permitir representar restricciones e imprecisión (en alguna de sus vertientes).

Las ventajas que ofrece la lógica como elemento de representación son las siguientes:

- Permite introducir información disyuntiva.
- Gracias a ella, pueden comprobarse la correctitud y la integridad de los datos.
- Ofrece mecanismos de deducción fácilmente automatizables y de gran eficacia que permiten obtener información adicional a la contenida explícitamente en la Base de Datos.
- La lógica en sí puede ser usada a su vez como mecanismo de representación y como lenguaje de consulta y manejo de los datos.
- Las restricciones de integridad de la Base de Datos pueden expresarse directamente como fórmulas bien formuladas del lenguaje subyacente.

En [Reiter, 1984] y [Gallaire et al., 1984] puede encontrarse un primer estudio sobre la formulación del concepto de Base de Datos relacional desde el punto de vista de la lógica de predicados de 1^{er} orden.

Esta representación lógica puede hacerse desde dos perspectivas distintas:

a) Como una interpretación de un especial lenguaje de primer orden, que debe ser un modelo para un conjunto de reglas de integridad. Se considera que tanto las reglas de integridad como las preguntas deben evaluarse en términos de satisfacción. Esto se conoce como *enfoque de las Interpretaciones o Modelos*.

b) Como una teoría definida sobre este mismo lenguaje. Desde este punto de vista, las reglas de integridad y las consultas se consideran teoremas a probar a partir de los axiomas que constituyen la Base de Datos. Se le denomina *enfoque de las Demostraciones*.

Hay que hacer notar, que el modelo relacional se identifica de forma muy natural en el contexto de las Interpretaciones, si bien algunos de los problemas clásicos (valores nulos, información disyuntiva, ...) no tienen una buena solución en este enfoque. Por otra parte, el enfoque de las Demostraciones nos conduce directamente al concepto de Base de Datos Lógica o Deductiva quedando una Base de Datos convencional como un caso particular de las mismas.

Otros autores han optado por abordar el problema de la imprecisión sin abandonar el entorno de los modelos relacionales clásicos de BD, para lo cual ha sido necesario extenderlos. En este sentido surgieron las llamadas **Bases de Datos Difusas**. Dada su naturaleza, la herramienta básica utilizada para abordar el problema ha sido la Teoría de Conjuntos Difusos [Zadeh, 1977].

En este trabajo tratamos el problema de esta integración y la implementación de un modelo lógico para BD Difusas. Con este propósito, daremos una ligera revisión a los conceptos de BD Difusa y de Representación Lógica de una BD.

2. Principales modelos de Bases de Datos Difusas

En esta sección, vamos a hacer un análisis de los principales modelos existentes en la literatura. En general, cualquiera de ellos cae dentro de alguna de estas tres categorías:

- Modelo Relacional Difuso
- Modelo de Unificación mediante Relaciones de Similitud.
- Modelo Relacional Posibilístico.

MODELO RELACIONAL DIFUSO

Este modelo extiende el modelo clásico mediante la adición de un grado a cada tupla. Este grado representa hasta qué punto la tupla pertenece a la relación en la que se encuentra y es un valor numérico del intervalo $[0,1]$. Más formalmente, una BD Difusa sería un conjunto de relaciones en el que cada una de ellas es una relación difusa caracterizada por una función de pertenencia como sigue:

$$BD = \{R_1, R_2, \dots, R_m\}$$

$$\mu_{R_j} : U_1 \times U_2 \times \dots \times U_n \rightarrow [0,1] \quad j=1 \dots m.$$

siendo U_i el dominio del i -ésimo atributo de la relación y $\times$ el producto cartesiano.

Aunque se han construido sistemas basados en este modelo ([Zadeh, 1978] y [Baldwin, 1987]), éste presenta el inconveniente de que cada tupla asume su carácter difuso de forma global sin que se pueda determinar la aportación difusa particular de cada uno de los atributos que la constituyen.

MODELO DE UNIFICACION MEDIANTE RELACIONES DE SIMILITUD

Este modelo, propuesto por Buckles y Petry ([Buckles-Petry, 1982], [Buckles-Petry, 1984], [Anvary-Rose, 1987]), define una Relación Difusa como un subconjunto del producto cartesiano $2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_n}$, donde 2^{D_i} es cualquier elemento del conjunto de las partes del dominio ($\mathcal{P}(D_i)$).

Los dominios permitidos para los atributos son los siguientes:

- Un conjunto finito de escalares. Ej. {negro, rojo, verde}.
- Un conjunto finito de números. Ej. {15, 16, 17}.
- Un conjunto de números difusos o etiquetas. Ej. {bajo, medio, alto}. [Buckles-Petry, 1984]

La forma de manejar y representar la incertidumbre es a través de las *Relaciones de Similitud* [Zadeh, 1971] sobre dominios escalares o numéricos y se hace manifiesta en el planteamiento de consultas, donde el usuario podrá imponer un umbral de similitud α a partir del cual todas las tuplas que sean α -similares serán *fundidas* en una sola.

En esta línea están los modelos presentados en [Shenoi-Melton, 1989], [Shenoi-Melton, 1990] y [Rundesteiner et al., 1989] donde se proponen distintas relaciones entre los valores (no de similitud).

El principal inconveniente de este modelo es que la definición de tupla no se corresponde con el concepto de ítem atómico de información, ya que una tupla en el sentido de Buckles-Petry equivale a varias unidades de conocimiento o interpretaciones (consecuencia de ello es que no se puede definir el concepto de llave primaria sobre las relaciones).

MODELOS RELACIONALES POSIBILÍSTICOS

Estos modelos representan la imprecisión en los datos por medio del uso de *Distribuciones de Posibilidad* [Zadeh, 1978]. La estructura general de una Base de Datos en este modelo es como sigue:

La Base de Datos está constituida por un conjunto de relaciones

$$BD = \{R_1, R_2, \dots, R_m\}$$

donde cada relación verifica:

$$R_j \subseteq \mathcal{P}(U_1) \times \mathcal{P}(U_2) \times \dots \times \mathcal{P}(U_n) \quad j=1 \dots m$$

$\mathcal{P}(U_i)$ representa la familia de todas las distribuciones de posibilidad en el dominio U_i y $\times$ es el Producto Cartesiano.

Bajo esta definición hay muchos modelos (que difieren entre sí en la representación de las distribuciones de posibilidad, manejo de información desconocida o nula, planteamiento de la consulta, operadores de selección, etc...) entre los que se encuentran [Fukami et al., 1979], [Umano, 1982], [Prade-Testemale, 1984], [Zemankova-Kandel, 1984], [Li-Liu, 1990] y [Medina et al., 1991].

3. Representación Lógica de una Base de Datos Difusa

En esta sección presentamos el modelo teórico que vamos a adoptar para representar una Base de Datos Difusa desde el punto de vista de la lógica [Vila, 1992]. Una de las grandes ventajas de este modelo es que utiliza como únicas herramientas la lógica de predicados de 1^{er} orden y la definición lógica de Base de Datos Relacional [Reiter, 1984] [Gallaire et al., 1984]. Este enfoque tiene además una ventaja adicional, y es que nos sitúa en el contexto más general posible y nos ofrece la posibilidad de investigar ampliaciones que nos conduzcan a las Bases de Datos Difusas Deductivas.

El modelo admite varios niveles de imprecisión en los datos:

- Atributos atómicos no difusos.
- Distribuciones de posibilidad sobre dominio discreto.
- Distribuciones de posibilidad sobre dominio continuo.

Las distribuciones de posibilidad vienen dadas por etiquetas cuya interpretación semántica debe ser incluida como información adicional. Además, se permiten consultas difusas sobre valores «crisp», lo que hace necesario almacenar también la información de las etiquetas y los modificadores asociados a las consultas.

Para ampliar el modelo lógico al caso difuso se introducen los siguientes elementos adicionales:

- Se introduce un tipo simple nuevo (MD) que servirá para representar el dominio sobre el que se moverán los grados de pertenencia.
- Se introduce la definición lógica de *tipo difuso* como sigue:

« τ es un tipo difuso sii verifica:

- 1) Toda distribución de posibilidad sobre τ tiene asociado un nombre (NOM_{τ}).
- 2) Existe un predicado POS_{τ} que servirá para representar dichas distribuciones de posibilidad.
- 3) Existe un predicado ternario (IG_{τ}) (compatibilidad) para relacionar distribuciones de posibilidad.»

Estas nuevas definiciones sirven de soporte para redefinir los conceptos de Lenguaje relacional, Base de Datos relacional e Interpretación de manera que se adapten a la representación y manejo todos los tipos de datos nuevos a considerar. Así pues se definen los conceptos de *Lenguaje Relacional Difuso*, *BD Relacional Difusa* e *Interpretación Difusa*, cuya definición formal se encuentra en [Vila et al., 1992].

Los pasos a dar para transformar una instancia de BD difusa en su correspondiente representación lógica son los dados en [Reiter, 1988] y consisten en añadir el axioma de mundo cerrado, axiomas de igualdad, axiomas de nombre único, los hechos de la BD como fórmulas atómicas, las extensiones de los predicados y, adicionalmente, los predicados relacionados con los tipos difusos existentes.

Tomando como referencia estos puntos y para aclarar todo lo expuesto veamos un ejemplo.

Supongamos una instancia de una BD en la que se guardan datos concernientes a los alumnos de un instituto y que contiene los siguientes atributos:

- NOMBRE
- EDAD (Atr. difuso dom. continuo)
- CURSO
- COMPORTAMIENTO (Atr. difuso dom. discreto)

donde el atributo EDAD es susceptible de adoptar cualquier representación difusa o no sobre dominio continuo, el atributo COMPORTAMIENTO, al estar definido sobre dominio discreto, admite cualquier representación de dominio discreto, y los demás atributos se suponen no difusos y atómicos.

TABLA ALUMNO

NOMBRE	EDAD	CURSO	COMPORTAM.
JOSE	MUY-JOVEN	1	REGULAR
MARIA	14	1	BUENO
ANTONIO	JOVEN	2	MUY-BUENO
CARLOS	16	3	BUENO
SUSANA	NORMAL	5	EJEMPLAR

Esta tabla, según el modelo lógico que hemos adoptado quedaría representada del siguiente modo:

Constantes:

JOSE, MARIA, ANTONIO, CARLOS, SUSANA, 13, ... , 30, MUY_JOVEN, JOVEN, NORMAL, MAYOR, MUY_MAYOR, 1, 2, 3, 4, 5, MUY_MALO, MALO, REGULAR, BUENO, MUY_BUENO, EJEMPLAR, 0.0, 0.1, 0.2,...,1.0.

Predicados:

NOMBRE(), EDAD(), NOM_EDAD(), POS_EDAD(_,_), EDAD(), IG_EDAD(_,_), CURSO(), COMP(), NOM_COMP(), POS_COMP(_,_), IG_COMP(_,_), =(,), ALUMNO(_,_), >(_,_).

Tipos simples: Son los predicados unarios del conjunto anterior.

Tipos difusos:

- EDAD con sus predicados asociados (NOM_EDAD, POS_EDAD, IG_EDAD)
- COMPORTAMIENTO con sus predicados asociados (NOM_COMP, POS_COMP, IG_COMP).

Axioma de nombre único:

$=(\text{JOSE}, \text{JOSE}), =(\text{ANTONIO}, \text{ANTONIO}), =(\text{MARIA}, \text{MARIA}), \dots, =(\text{SUSANA}, \text{SUSANA}),$
 $=(\text{MUY\_JOVEN}, \text{MUY\_JOVEN}), \dots, =(\text{MUY\_MAYOR}, \text{MUY\_MAYOR}), =(1,1), \dots, =(5,5),$
 $=(\text{MUY\_MALO}, \text{MUY\_MALO}), \dots, =(\text{EJEMPLAR}, \text{EJEMPLAR}), =(0,0), \dots, =(1,1).$

Ordenación del tipo MD:

$\geq(0.0,0.0), \geq(0.1,0.0), \dots, \geq(1.0,0.0), \dots, \geq(1.0,0.9), \geq(1.0,1.0).$

Extensión de los predicados:

NOMBRE(JOSE), NOMBRE(MARIA), NOMBRE(ANTONIO), NOMBRE(CARLOS), NOMBRE(SUSANA),
 EDAD(MUY_JOVEN), EDAD(14), EDAD(JOVEN), EDAD(16), EDAD(NORMAL), CURSO(1), CURSO(2),
 CURSO(3), CURSO(5), COMP(REGULAR), COMP(BUENO), COMP(MUY_BUENO), COMP(EJEMPLAR).

Hechos de la BD:

ALUMNO(JOSE, MUY_JOVEN, 1, REGULAR)
 ALUMNO(MARIA, 14, 1, BUENO)
 ALUMNO(ANTONIO, JOVEN, 2, MUY-BUENO)
 ALUMNO(CARLOS, 16, 3, BUENO)
 ALUMNO(SUSANA, NORMAL, 5, EJEMPLAR)

Predicados asociados a tipos difusos:

NOM_EDAD(13), ... , NOM_EDAD(30), NOM_EDAD(MUY_JOVEN), ... , NOM_EDAD(MUY_MAYOR),
 NOM_COMP(MUY_MALO), ... , NOM_COMP(EJEMPLAR).
 POS_EDAD(13,13,1), ... , POS_EDAD(30,30,1), POS_EDAD(13,MUY_JOVEN,1), POS_EDAD(14,MUY_JOVEN,0.9),
 ... , POS_EDAD(29,MUY_MAYOR,1), ... , POS_EDAD(30,MUY_MAYOR,1).
 POS_COMP(MUY_MALO,MALO,0.7), POS_COMP(MUY_MALO,REGULAR,0.3), ... ,
 POS_COMP(MUY_BUENO,EJEMPLAR,0.8).
 IG_EDAD(13,13,1).....

(Aquí vendrían todas las instancias del predicado POS_EDAD anteriormente descritas).

IG_EDAD(MUY_JOVEN,JOVEN,0.7), IG_EDAD(JOVEN,NORMAL,0.5), IG_EDAD(NORMAL, MAYOR, 0.5),
 IG_EDAD(MAYOR, MUY_MAYOR,0.6).

(No es necesario especificar IG_COMP porque coincide con POS_COMP).

Haria falta añadir a todos estos predicados los «simétricos» correspondientes para que la relación sea conmutativa.

Una de las grandes ventajas de esta representación es que, además, pueden especificarse reglas del problema concreto y reglas concernientes a la semántica. Un ejemplo de ello podría ser:

$\forall (X,Y,Z,W), \text{ALUMNO}(X,Y,Z,W) \rightarrow \text{NOMBRE}(X) \wedge \text{EDAD}(Y) \wedge \text{CURSO}(Z) \wedge \text{COMPORT}(W)$

que especifica el esquema relacional de la tabla ALUMNO, o bien reglas de integridad como:

$\forall (X,Y1,Y2,Z1,Z2,W1,W2), \text{ALUMNO}(X,Y1,Z1,W1) \wedge \text{ALUMNO}(X,Y2,Z2,W2) \rightarrow$
 $=(Y1,Y2) \wedge =(Z1,Z2) \wedge =(W1,W2)$

que indican que el atributo NOMBRE es clave primaria (todos los demás atributos dependen de él).

A todo lo expuesto debe sumarse también el hecho de que el *operador* de igualdad difusa (descrito por los predicados IG_EDAD e IG_COMP) no es el único operador posible sobre dominios difusos, ya que pueden definirse otros comparadores ("algo_menor", "mayor", "menor",...) siempre que el dominio subyacente permita establecer una relación de orden. Para ello no tenemos más que añadir las tablas de definición del operador correspondiente y algunos axiomas adicionales (similares a los añadidos para el operador de igualdad).

En resumen, analizando el modelo, se encuentran las siguientes ventajas:

- Hace una definición lógica de las BD difusas sin más herramienta que la lógica de 1^{er} orden.
- Permite el tratamiento de datos imprecisos de diversa naturaleza, introduciendo tan sólo unos axiomas adicionales (pueden encontrarse en [Vila et al., 1992]) a los mencionados en [Reiter, 1984]. Estos axiomas son los que permiten el manejo de datos difusos.
- El lenguaje de consulta es un cálculo relacional, que es directamente traducible a lógica de 1^{er} orden.
- Recoge como casos particulares las BD relacionales clásicas, el modelo de Relaciones de Similitud y los modelos Posibilísticos.

Por otro lado, se plantean algunos inconvenientes de cara a su implementación:

- La representación de un dato difuso sobre dominio continuo se hace especificando de forma extensiva qué valores va tomando la función de pertenencia para una serie de elementos del dominio subyacente. Esto no sólo plantea un problema de memoria sino que tenemos que asegurar que el «muestreo» que se haga sobre la función de pertenencia tenga la suficiente granularidad como para que no se produzca una pérdida de información.
- La definición propuesta obliga a que por cada tipo difuso se defina un predicado ternario asociado al mismo cuyos dos primeros parámetros se evalúan sobre los nombres de las distribuciones de posibilidad y el tercero sobre los grados de pertenencia (IG _{τ}).
- Hay mucha redundancia de información debida a la propiedad conmutativa de algunas de las relaciones.

Como puede verse, el principal problema del modelo es el de la representación de datos y operadores difusos, lo que nos ha llevado a buscar representaciones alternativas equivalentes que no presenten estos inconvenientes. En primer lugar será necesario cambiar la representación de los datos y los operadores, elegir un conjunto mínimo pero a la vez completo de tipos de datos difusos [Medina et al., 1991(b)] y buscar una representación unificada de los mismos.

4. Representación lógica unificada de datos difusos

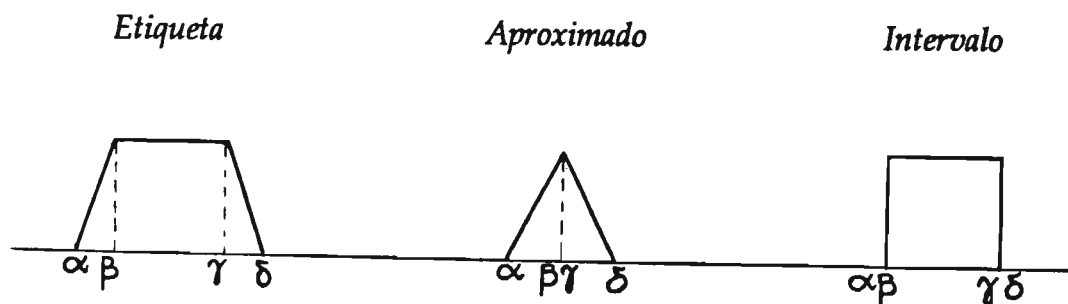
En este apartado vamos a establecer qué tipos de datos difusos son los que vamos a considerar y qué predicados serán necesarios para su representación.

4.1. Datos difusos sobre dominio contínuo:

Etiqueta lingüística: Este tipo de datos está representado por una función trapezoidal que lleva asociada un nombre. Dicha función queda perfectamente establecida por cuatro parámetros $(\alpha, \beta, \gamma, \delta)$.

Tipo aproximado: Aquí se consideran aquellos datos de los que no conocemos su valor exacto pero tenemos una idea del valor en torno al cual está. Estos datos pueden representarse por una función triangular. La representación puede ser idéntica al caso de la etiqueta lingüística sin más que hacer $\beta = \gamma$.

Tipo intervalo: Este tipo recoge información difusa de tipo intervalar, es decir, no se conoce el dato pero sí los límites de su valor. De forma similar al caso anterior, la representación puede hacerse tomando $\alpha = \beta$ y $\gamma = \delta$.



Como puede verse, es suficiente dar cuatro parámetros y un nombre para tener definido cualquier dato de uno de estos tipos, lo que nos va a facilitar enormemente su manejo. Para llevar a cabo su representación, introducimos un predicado, LABEL, que asocia el nombre de la etiqueta con sus parámetros. Por ejemplo,

```
LABEL("ALTO",1.75,1.80,1.90,1.95)
LABEL("30-35",30,30,35,35)
LABEL("APROX.7",5,7,7,9)
```

donde "ALTO" es una etiqueta, "30-35" es un intervalo y "APROX.7" es un tipo aproximado.

4.2. Datos difusos sobre dominio discreto:

Distribución de posibilidad: Este tipo se define para recoger conocimiento impreciso sobre dominios no contínuos, es decir, el dato puede tomar distintos valores de un dominio discreto con distintos valores de certeza. Un dato de este tipo podría representarse de la forma: $v_1/c_1 + v_2/c_2 + \dots + v_n/c_n$, donde v_i/c_i indica que el dato toma

el valor v_i con certeza c_i . En este caso, resulta imposible una representación análoga a la anterior, lo cual no plantea problemas puesto que el dominio de un atributo será ó continuo ó discreto pero nunca ambas cosas a la vez.

Por otro lado, tenemos el problema de que una distribución de posibilidad de este tipo no tiene un tamaño fijo, ya que depende del número de valores que intervengan en la definición del dato. Para solucionarlo basta con exigir que toda distribución de posibilidad lleve asociado un nombre. Así pues la representación puede llevarse a cabo de la siguiente manera:

Si tenemos una distribución de posibilidad $v=v_1/c_1 + v_2/c_2 + \dots + v_n/c_n$, que aparece como valor de un atributo A_i en una tupla de atributos $(A_1, A_2, \dots, A_n)$, pondremos, por un lado, el predicado correspondiente a la tupla $-TABLA(a_1, a_2, \dots, v, \dots, a_n)-$ y, para especificar la composición de v , introducimos un predicado **SIMIL**, que servirá para establecer el grado de compatibilidad de v con cada uno de sus componentes, esto es:

$$\begin{aligned} &SIMIL(v, v_1, c_1) \\ &SIMIL(v, v_2, c_2) \\ &\dots\dots\dots \\ &\dots\dots\dots \\ &SIMIL(v, v_n, c_n) \end{aligned}$$

donde añadiremos tantas instancias del predicado **SIMIL** como valores intervengan en la definición de la distribución.

Este predicado **SIMIL** nos servirá a su vez para establecer las relaciones de similitud *básicas* entre los elementos del dominio discreto, esto es

$$\begin{aligned} &SIMIL(v_1, v_2, d_1) \\ &SIMIL(v_1, v_3, d_1) \\ &\dots\dots\dots \\ &SIMIL(v_i, v_j, d_k) \end{aligned}$$

5. Implementación del modelo lógico de BD difusa

Vamos a tratar de implementar este modelo lógico de BD difusa en un sistema que ofrezca las suficientes garantías de potencialidad y generalidad, y que, a su vez, no se aparte mucho de nuestro contexto. En este sentido, nos parece muy adecuado el lenguaje Prolog y su entorno, que está establecido ya como una potente herramienta de investigación y desarrollo, y además nos permite un tratamiento casi directo de la representación lógica de nuestro problema.

Un programa en Prolog consiste en un conjunto de hechos y reglas expresados en una variante de la lógica de predicados de 1^{er} orden. Cada sentencia se representa por una cláusula que puede ser verdadera o bien verdadera pero condicionada a que se demuestren previamente una serie de sub-objetivos (regla). El Prolog utiliza el Principio de Resolución de Robinson para resolver los objetivos que se le van presentando. Dado un objetivo a ser probado, el motor del Prolog busca entre los hechos y reglas de su BD

utilizando el método de *Búsqueda en Profundidad* para lograr unificar el actual objetivo con alguna o algunas cláusulas. La BD interna del Prolog se organiza como un Heap, donde las cláusulas se alojan dinámicamente. Este almacenamiento funciona muy bien dada su extrema rapidez, pero limita mucho el número de cláusulas a considerar.

Una ventaja adicional del Prolog es que ofrece una potente herramienta, la *recursividad*, que no está presente en los lenguajes de consulta estándar.

Analizando de cerca el entorno de programación del Prolog y las BD relacionales, encontramos varios puntos en común, que difieren bastante en la forma de ser abordados:

- La existencia de una BD: Esta BD en el caso de Prolog es de tamaño relativamente pequeño, es mono-usuario y va almacenada en memoria principal. Su contenido es de reglas y hechos. En el caso de los sistemas de BD, se manejan grandes cantidades de datos almacenados en memoria masiva y que pueden ser compartidos (entorno multi-usuario).

- La consulta de información: Una consulta es el proceso por el se extrae información relevante de la BD. En el entorno del Prolog, una consulta u objetivo se construye generando deducciones en cadena que combinan hechos y reglas con el fin de confirmar o refutar la afirmación inicial. En los sistemas de BD una consulta se determina encontrando el modo más eficiente de búsqueda en memoria masiva de la información requerida. (En [Mouta et al., 1988] puede encontrarse una revisión sobre los lenguajes de consulta y su relación con el Prolog)

- La especificación de restricciones: Las restricciones definen las condiciones de correctitud para la BD. En los sistemas de BD la validación de restricciones es el proceso por el cual se preserva la correctitud de la BD evitando que en ella se almacenen datos incorrectos. En Prolog, las restricciones se expresan con reglas que se activan «automáticamente» cada vez que la BD es modificada.

En este sentido, la programación lógica ofrece un mayor poder (cuenta con la recursividad) para expresar consultas y restricciones comparado con el que ofrecen los lenguajes de manipulación de datos de los sistemas de BD. Además, la programación lógica permite que la representación de consultas y restricciones sea homogénea y con los mismos mecanismos de inferencia. Por otro lado, esta última no ofrece la tecnología necesaria para manejar grandes cantidades de datos que pueden ser compartidos.

El acoplamiento entre sistemas de BD clásicas y Prolog aparece en la literatura desde dos perspectivas muy distintas: [Ceri et al., 1990]

a) Se traslada la BD a una representación lógica equivalente y los mecanismos de manipulación se traducen también a lógica de 1^{er} orden (Prolog). Se trabaja, pues, con herramientas y representación puramente lógicas.

b) Se aprovechan las ventajas de cada una de ellas construyendo sistemas de coordinación y manipulación de ambas herramientas conjuntamente. El proceso es más complicado pero más eficiente.

Dentro de esta última perspectiva aparecen, a su vez, *dos grandes modelos de arquitectura*:

b.1) Acoplamiento débil: Sobre sistemas ya existentes se construye una nueva capa que los envuelve, pero manteniendo cada uno de ellos la estructura y comportamiento originales, es decir, los dos sistemas subyacentes se diferencian claramente. Este acoplamiento se lleva a cabo en tiempo de carga del programa en Prolog y una vez que los hechos de la BD han sido cargados en M principal, la ejecución es independiente del sistema de BD.

b.2) Acoplamiento fuerte: Se construye un único sistema desde el principio haciendo uso, tanto de la tecnología de BD como de la programación lógica, interactuando ambas entre sí.

Una vez establecido nuestro entorno de trabajo, adoptaremos inicialmente para nuestra implementación un acoplamiento débil. Aunque este tipo de acoplamiento plantea algunos inconvenientes (que analizaremos posteriormente), sí es verdad que permite obtener resultados visibles a corto plazo, lo cual es esencial para nosotros, ya que en esta primera tentativa de modelizar la BD sólo pretendemos ver que el modelo teórico es válido y que el Prolog permite un manejo y una representación sensatos de los tipos difusos introducidos.

5.1. Representación de los datos

Dada una tabla o relación de una BD, $T(A_1, A_2, \dots, A_n)$, la información en ella contenida queda representada de la siguiente forma¹:

1) *Para cada tupla de la relación, se añadirá el predicado $T(a_1, a_2, \dots, a_n)$ a la Base de Hechos del Prolog.*

2) *Para cada atributo de la tabla y para cada uno de los valores que toma en las tuplas, añadiremos el predicado $A_i(a_j)$, $i=1, \dots, n$ $j=1, \dots, m$ donde A_i es el nombre del i -ésimo atributo de la tabla y los a_j los valores que existen para dicho atributo en la tabla. Equivale a establecer el dominio actual de los atributos para una instancia concreta de la BD.*

3) *Repetir los pasos 1) y 2) para cada una de las tablas que componen la BD.*

Dado que en una BD existirá información redundante (por ejemplo, la llave primaria de una relación suele ser llave externa de otra) debemos añadir un cuarto paso que consiste en :

4) *Eliminar los predicados redundantes originados en el paso 2).*

¹ El sistema de gestión de BD utilizado para la representación inicial de los datos es el Oracle y el lenguaje de programación utilizado para implementar los operadores y reglas es el Prolog.

Con estos cuatro primeros pasos tenemos conformado lo que se llama Base de Hechos o BD del entorno de Prolog. Sin embargo, aunque contamos ya con la información real del problema, necesitamos almacenar también las definiciones de todos los posibles valores difusos de los atributos que tengan esta cualidad y que deben estar también reflejadas en la BD de partida; por tanto, para completar la representación falta un último paso, esto es:

5) Incluir en la BD del Prolog todas las definiciones de los datos difusos que aparezcan en la BD inicial mediante los predicados LABEL y SIMIL anteriormente mencionados.

Veamos cómo quedaría representada la tabla mostrada en el apartado 3 siguiendo estos pasos:

Paso 1:

ALUMNO(JOSE,MUY-JOVEN,1,REGULAR)
 ALUMNO(MARIA,14,1,BUENO)
 ALUMNO(ANTONIO,JOVEN,2,MUY-BUENO)
 ALUMNO(CARLOS,16,3,BUENO)
 ALUMNO(SUSANA,NORMAL,5,EJEMPLAR)

Paso 2:

NOMBRE(JOSE)	EDAD(MUY-JOVEN)	CURSO(1)	COMPORT(REGULAR)
NOMBRE(MARIA)	EDAD(14)	CURSO(1)	COMPORT(BUENO)
NOMBRE(ANTONIO)	EDAD(JOVEN)	CURSO(2)	COMPORT(MUY-BUENO)
NOMBRE(CARLOS)	EDAD(16)	CURSO(3)	COMPORT(BUENO)
NOBRE(SUSANA)	EDAD(NORMAL)	CURSO(5)	COMPORT(EJEMPLAR)

Paso 3:

Está incluido en los dos pasos anteriores.

Paso 4:

Eliminaríamos las instancias de predicados que estén repetidas (CURSO(1) y COMPORT(BUENO)).

Paso 5:

LABEL(MUY-JOVEN,12,17,22,27)	
LABEL(JOVEN,18,23,28,33)	
LABEL(MEDIA,30,35,40,45)	
LABEL(MAYOR,48,52,58,62)	
LABEL(MUY-MAYOR,60,65,100,100)	
	Etiquetas
SIMIL(MUY_MALO,MALO,0.7)	
SIMIL(MUY_MALO,REGULAR,0.3)	
SIMIL(MALO,REGULAR,0.5)	
SIMIL(REGULAR,BUENO,0.5)	
SIMIL(REGULAR,MUY-BUENO,0.2)	
SIMIL(BUENO,MUY-BUENO,0.8)	
SIMIL(BUENO,EJEMPLAR,0.4)	
SIMIL(MUY_BUENO,EJEMPLAR,0.8).	
	Relación de Similitud

Si además se desean hacer consultas difusas sobre atributos que no lo son (por ejemplo el atributo CURSO), deberán incluirse las definiciones de las etiquetas o relaciones de similitud correspondientes. En el caso del CURSO podríamos añadir las siguientes:

```

SIMIL(1,MUY_BAJO,1)
SIMIL(2,MUY_BAJO,0.6)
SIMIL(3,MUY_BAJO,0.3)
SIMIL(1,BAJO,1)
SIMIL(2,BAJO,1)
SIMIL(3,BAJO,0.7)
SIMIL(1,MEDIO,0.3)
SIMIL(2,MEDIO,0.6)
SIMIL(3,MEDIO,1)
SIMIL(4,MEDIO,0.6)
SIMIL(3,ALTO,0.4)
SIMIL(4,ALTO,0.8)
SIMIL(5,ALTO,1)
SIMIL(ALTO,MEDIO,0.5)
SIMIL(MEDIO,BAJO,0.5)
SIMIL(BAJO,MUY_BAJO,0.8)

```

En el caso de las Relaciones de Similitud es necesario especificar de alguna forma la *propiedad conmutativa* de las mismas. Esto, que en el modelo teórico ha de hacerse duplicando toda la información (habría que cambiar de orden los dos primeros argumentos) en nuestro caso se resuelve añadiendo simplemente la regla:

$$\text{COMP}(X,Y,G) \text{ if } \text{SIMIL}(X,Y,G) \text{ or } \text{SIMIL}(Y,X,G)$$

que indica que al comparar dos valores definidos sobre dominio discreto da lo mismo el orden en que éstos valores se presenten.

Por otro lado, la *propiedad reflexiva* que obliga a relacionar cada valor consigo mismo con valor 1 en el modelo teórico, aquí se refleja con la regla

$$\text{SIMIL}(X,X,G) \text{ if } G=1$$

que indica que si se da el caso de que los argumentos sean iguales el grado se ponga automáticamente a 1.

Nótese que se han pasado por alto algunos puntos del modelo teórico (Axioma de nombre único, Ordenación del tipo MD, Axioma de mundo cerrado) que se dan por defecto en el entorno de Prolog.

5.2. Operaciones Relacionales Difusas

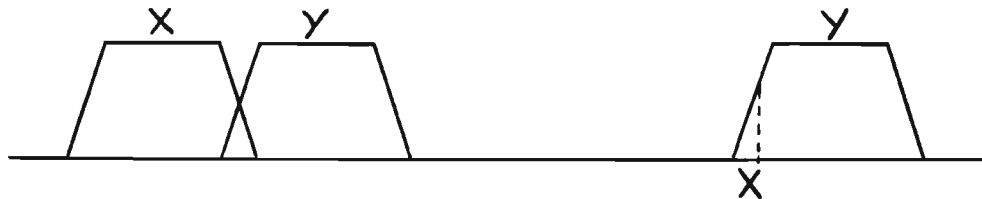
Una de las facilidades primordiales que debe ofrecer un sistema de manejo de información difusa es la posibilidad de hacer consultas bien precisas bien difusas de los datos. En este sentido, lo primero que hay que definir son los operadores relacionales que nos permitan establecer relaciones entre valores difusos y precisos entre sí con el fin de poder resolver las consultas y calcular hasta qué punto se dan dichas relaciones.

En nuestro caso, tenemos definidos los siguientes operadores relacionales difusos:

COMP(X,Y,Z): Equivale al operador de igualdad extendido al caso difuso. De los tres argumentos, los dos primeros son los valores a comparar y el tercero el grado en que se verifica la relación de igualdad entre ellos. Distingue tres casos:

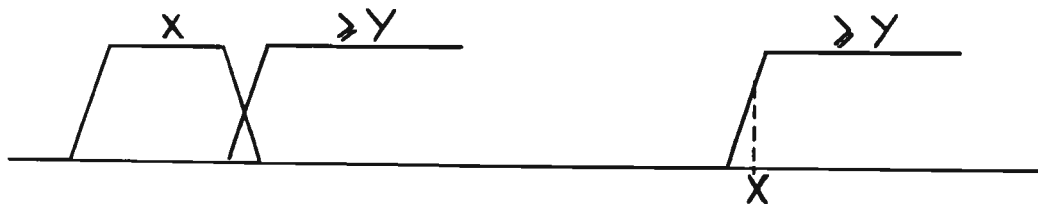
a) X e Y son valores «crisp»: el comportamiento es el clásico.

b) X e Y definidos sobre dominio continuo y al menos uno de ellos es una etiqueta: en este caso se calcula la compatibilidad haciendo uso de las definiciones de las etiquetas que hay almacenadas en la BD bajo el predicado LABEL.

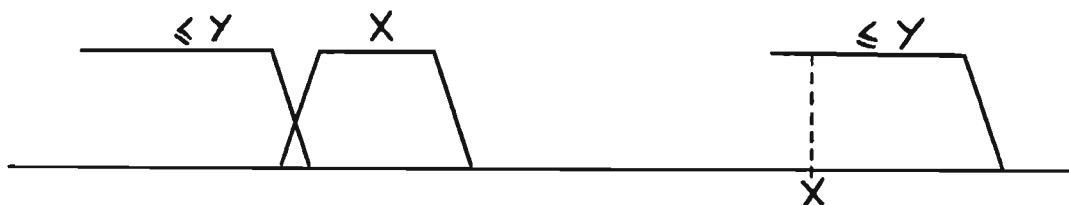


c) X e Y definidos sobre dominio discreto. El cálculo de la compatibilidad es una mera consulta a los predicados SIMIL de la BD.

MAYOR_IG(X,Y,Z): Es la extensión del operador mayor o igual ($\geq$) al caso difuso. Calcula en qué grado X es mayor o igual que Y y lo almacena en Z. Para valores «crisp» conserva el comparador clásico. Para relaciones mixtas o totalmente difusas, se construye la etiqueta *mayor o igual a Y* y se compara con X.



MENOR_IG(X,Y,Z): Se construye de forma análoga al anterior y se dan las mismas situaciones. Se construye la etiqueta *menor o igual a Y* y se compara con X.



Los operadores **MAYOR(X,Y,Z)** y **MENOR(X,Y,Z)** se construyen a partir de los anteriores como sus operadores complementarios como:

MAYOR(X,Y,Z):- **MENOR_IG(X,Y,Z1)** and $Z=1-Z1$.
(Como complemento del **MENOR_IG**)

MENOR(X,Y,Z):- **MAYOR_IG(X,Y,Z1)** and $Z=1-Z1$.
(Como complemento del **MAYOR_IG**)

que indica, en el primer caso, que X es mayor que Y con grado Z si X es menor o igual

que Y con grado $Z1$ y Z vale $1-Z1$. El segundo caso se define de igual forma con los operadores MENOR y MAYOR_IG.

5.3. Especificación de Consultas

Dado el estado actual de la implementación del sistema, para poder realizar una consulta a la Base de Hechos o para introducir nuevas reglas al sistema, hay que conocer el lenguaje Prolog. Esto, aunque pueda parecer un gran inconveniente no lo es tanto debido a que el Prolog presenta un gran parecido con el cálculo de predicados. En cualquier caso, está previsto que como trabajo futuro se realice un interface amigable al usuario.

Para especificar una consulta hay que tener en cuenta los siguientes puntos:

- Para cada atributo que intervenga en la consulta se podrá especificar un umbral de cumplimiento.
- Se seleccionarán aquellas tuplas que superen los umbrales de cumplimiento en todos y cada uno de los atributos involucrados.
- Cuando en una consulta no se especifique umbral para algún atributo, lo único que se exigirá (con respecto a dicho atributo) es que el grado de cumplimiento sea mayor que cero.

Veamos algunos ejemplos:

«Seleccionar aquellos alumnos cuya edad sea menor que media y que estén matriculados en un curso bajo»

ALUMNO(Nombre,Edad,Curso,_) and MENOR(Edad,"MEDIA",G1) and $G1 > 0$ and COMP(Curso,"BAJO",G2) and $G2 > 0$.

«Seleccionar aquellos alumnos cuyo comportamiento sea bueno con grado mínimo 0.8, que estén en un curso alto y cuya edad sea joven»

ALUMNO(Nombre,Edad,Curso,Comport) and COMP(Comport,"BUENO",G1) and $G1 \geq 0.8$ and COMP(Curso,"ALTO",G2) and $G2 > 0$ and COMP(Edad,"JOVEN",G3) and $G3 > 0$

«Seleccionar aquellos alumnos que sean inteligentes con grado mínimo 0.8 y cuyo comportamiento sea ejemplar»

ALUMNO(Nombre,Edad,Curso,Comport) and INTELIGENTE(Nombre,G) and $G \geq 0.8$ and COMP(Comport,"EJEMPLAR",G2) and $G2 > 0$.

6. Ampliaciones y Trabajos Futuros

Como ha podido observarse a lo largo del trabajo expuesto, hemos completado una primera fase de la implementación que puede decirse que es el esqueleto del sistema en el que podemos hacer uso de los operadores relacionales difusos más comunes y consultar información imponiendo determinados umbrales de creencia a la respuesta. Sin embargo, hay puntos que sería necesario mejorar, entre los que citaremos:

a) Añadir la posibilidad de definir reglas difusas. Este es el trabajo que estamos realizando actualmente y, hasta ahora, hemos podido comprobar que es relativamente fácil definir reglas que incluyan modificadores y atributos difusos así como reglas con grados de implicación distinto de 1.

b) Confeccionar un interface de usuario que no exija el conocimiento del lenguaje Prolog ni del entorno de BD de Oracle incluyendo entre sus posibilidades el que el usuario pueda consultar, introducir nuevos hechos, definir nuevas reglas y operadores relacionales y ampliar la gama de etiquetas existentes para un atributo

c) Estudiar la posibilidad de almacenar la Base de Hechos del Prolog en Memoria auxiliar ya que el almacenamiento actual se lleva a cabo en Memoria Principal y bastarían algunos miles de datos para que ésta resultara insuficiente (esto hace que la gestión adecuada de los datos sea muy dependiente de la máquina utilizada lo que hace que el sistema sea poco portable)

d) Aprovechar mejor los recursos que ofrecen los Sistemas de Gestión de BD (en nuestro caso el ORACLE) e integrarlos de forma eficiente con el entorno del Prolog

y que constituyen nuestro futuros trabajos al respecto.

7. Referencias

- [Anvary M., Rose G.F.]: *Analysis of Fuzzy Information*. Betdek Eds. CRC Press. Vol 2
- [Buckles B.P., Petry F.E., 1982]: *A Fuzzy Representation of Data for Relational Databases*. Fuzzy Sets and Systems. Vol. 7 pp. 213-226.
- [Ceri S., Gottlob G., Tanca L., 1990]: *Surveys in Computer Science. Logic Programming and Databases*. Springer-Verlag 1990.
- [Gallaire H., Minker J., Nicolas J.M., 1984]: *Logic and Databases: A Deductive Approach*. ACM Computing Surveys. Vol 16(2) pp. 153-185.
- [Kowalski R.A., 1981]: *Logic as a Database Language*. Proc. of the Advanced Seminar on Theoretical Issues in Databases (Italia).
- [Li D., Liu D., 1990]: *A Fuzzy Prolog Database System*. Ed. John Wiley & Sons. New York.

- [Medina J.M. et al., 1991(a)]: *Un modelo de Base de Datos Difusa Aplicado a Información Médica*. 1^{er} Congreso Español sobre Tecnologías y Lógica Fuzzy. Granada, 1991.
- [Medina J.M. et al., 1991(b)]: *A Knowledge Representation Model for Fuzzy Databases*. Depto. de Ciencias de la Computación e I.A. (Sometido a ACM TODS).
- [Mouta F., Howard M., Neves J.M., 1988]: *Implementing Query Languages in Prolog*. En *Prolog and Databases*. Eds. Gray & Lucas. Pp. 13-21. John Wiley & Sons.
- [Prade H. y Testemale C., 1984]: *Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries*. Information Science. Vol. 34 pp. 115-143.
- [Reiter R., 1984]: *Towards a Logical Reconstruction of Relational Database Theory*. On Conceptual Modelling. Ed. Brodie M., Mylopoulos J., Schmidt J.W. pp. 193-238.
- [Rundesteiner E. A., Hawkes L.W., Bandler W.]: *On Nearness Measures in Fuzzy Relational Data Models*. I.J.A.R. Vol 3, pp 267-298.
- [Shen Z., Ding L., Mukaidono M., 1988]: *A Theoretical Framework of Fuzzy Prolog Machine*. Fuzzy Computing. M.M. Gupta & T. Yamakawa Eds. Pp. 89-100. North Holland.
- [Umano, 1982]: *FREEDOM-0: A Fuzzy Database System*. En M.M. Gupta y E. Sanchez Eds. *Fuzzy Information and Decision Processes* Pp. 339-347. North Holland Pub. Co.
- [Vila M.A., Cubero J.C., Medina J.M., Pons O., 1992]: *A Logical Approach to Fuzzy Relational Databases*. En *Proceedings de IMPU'92*.
- [Zadeh L.A., 1977]: *Theory of Fuzzy Sets*. Encyclopedia of Computer Science and Technology. Ed. Belzer J., Holzman A., Kent A.
- [Zemankova-Leech M., Kandel A., 1984]: *Fuzzy Relational Databases: A Key for Expert Systems*. Verlag TUV Rheiland.

GEFRED. Un Modelo Generalizado de Bases de Datos Relacionales Difusas

J. M. Medina Rodríguez, A. Vila Miranda
Dpto. Ciencias de la Computación e Inteligencia Artificial
Facultad de Ciencias. Universidad de Granada
Campus de Fuentenueva. 18071 Granada (Spain)
email: medina@robinson.ugr.es
Tlf. 958-243196

Resumen

Esta comunicación presenta un modelo de Bases de Datos Relacionales Difusas cuyas características principales son: la integración en un entorno común de modelos preexistentes, la capacidad de representación de una amplia gama de información difusa y el tratamiento coherente y flexible de la misma. Todo ello con el ánimo de ofrecer un modelo con el que resolver cada problema de representación y manipulación de información difusa, atendiendo a la naturaleza del mismo, pudiendo modelar la consulta mediante la elección del operador de comparación y de la medida de compatibilidad difusa a emplear. Además, posibilita actuar de forma selectiva sobre la presición con que han de satisfacer las condiciones de una consulta los atributos envueltos en la misma.

Palabras Clave: Bases de Datos Relacionales Difusas, Conjuntos Difusos.

1 Introducción

El punto de partida de nuestro modelo es el de las Bases de Datos Relacionales [Codd, 70] el cual contempla la información almacenada como una organización de registros o tuplas agrupadas en relaciones o tablas sobre las que se realiza un reducido pero potente conjunto de operaciones tendentes a obtener, reorganizar o alterar la información presente en la base de datos.

Los dos enfoques utilizados en el modelo relacional para la captura de información (el Álgebra Relacional y Cálculo Relacional), parten de la base de que toda pregunta hecha a una BDR tiene como respuesta una nueva relación. Ninguno de ellos permite, en principio, que las preguntas se formulen de una manera imprecisa; es decir, no contestaría a preguntas tales como:

"Encontrar todos los individuos jóvenes con un salario no muy alto"

En los últimos años diversos autores han abordado el problema de flexibilizar el modelo relacional con objeto de admitir la imprecisión dentro del mismo. Esto conduce a sistemas de

bases de datos que, de alguna manera, caen dentro del campo de la Inteligencia Artificial ya que permiten tratar cuestiones e información formuladas en una terminología más cerca del lenguaje natural. La incorporación de dicha imprecisión en el Sistema de Base de Datos se puede realizar en dos niveles: el primero contempla el planteamiento de consultas imprecisas sobre Bases de Datos Clásicas y el segundo plantea el problema de la incorporación de información imprecisa en el Sistema.

En ambos casos, la teoría de subconjuntos difusos [Zadeh, 78] proporciona una potente herramienta para representar cuestiones y datos imprecisos. Así, dentro del primer planteamiento podemos encontrar los trabajos de [Bosc, Galibourg, 88] y [Kacprzyk, 86]. El tratamiento de los problemas que caen dentro del segundo planteamiento da origen a los modelos de Bases de Datos Relacionales Difusas cuyos fundamentos teóricos se mencionarán en el próximo apartado.

El modelo que presentaremos en el apartado tercero representa una síntesis entre las diferentes tendencias aparecidas para abordar el problema de representación y tratamiento de información difusa mediante Bases de Datos. Se ha realizado un esfuerzo de abstracción y de generalización tendente a integrar en un mismo desarrollo enfoques que aparecían en principio como dispares, todo ello con el único fin de facilitar un soporte coherente para la implementación de Sistemas de Bases de Datos Relacionales Difusas.

2 Principales Modelos Teóricos

Basicamente, son dos los modelos que extienden el Modelo Relacional Clásico permitiendo la incorporación de la imprecisión en la Base del Conocimiento.

En el modelo de [Buckles, Petry, 82], [Buckles, Petry, 84] (también [Anvari, Rose, 87]) las relaciones son subconjuntos del producto cartesiano $2^{D_1} \times \dots \times 2^{D_m}$, donde 2^{D_i} es cualquier elemento del conjunto de las partes del dominio D_i . La forma básica de representar la incertidumbre es a través de las Relaciones de Similitud [Zadeh, 71] sobre Dominios escalares o numéricos (también se permiten números difusos ver [Buckles, Petry, 84]). Estas relaciones establecen una partición sobre 2^{D_i} donde los elementos de cada clase son indistinguibles a un nivel de similitud α (impuesto en la consulta).

Posteriormente [Shenoi, Melton, 89] demostraron que todas las buenas propiedades que verifica este modelo seguían cumpliéndose suprimiendo la transitividad, y más en general [Shenoi, Melton, 90] demostraron que la clave estaba en particionar los dominios en clases de equivalencia. [Rundensteiner, Hawkes, Bandler, 89] generaliza el modelo utilizando relaciones que no deriven clases de equivalencia. En definitiva, se trata de construir clases en el dominio correspondiente, de forma que los elementos de una misma clase son indiferentes según el grado de similitud permitido. [Buckles, Petry, Sachar, 89] desarrollaron un Cálculo Relacional para su modelo.

En el modelo de [Prade, Testemale, 84], las relaciones correspondientes a la Base de Conocimiento (rel. básicas) tienen la misma representación en forma de tabla que en el modelo relacional clásico de Codd, diferenciando de éste en los tipos de datos que pueden aparecer en cada casilla (valor de atributo en una tupla). Mientras que en el modelo clásico la única imposición es que sean valores atómicos de un cierto dominio, en este modelo se admite que el valor de un atributo para una determinada tupla venga dado por una distribución de

posibilidad; los posibles casos particulares de estas distribuciones generan una amplia gama de alternativas para ser incluidos como tipo de datos admisibles en los modelos prácticos que se deriven de éste: conjuntos, etiquetas, números difusos,...

Esta es la misma filosofía adoptada en [Zemankova, Kandel, 84] y [Raju, Majumdar, 87]. En [Prade, Testemale, 87] también se incluye el tipo de datos correspondiente a un subconjunto del dominio (representando posibles asignaciones no excluyentes). En la aproximación de [Umano, 82], las relaciones básicas no son clásicas pues las tuplas pertenecen con cierto grado a dichas relaciones (son por tanto rel. difusas)

La imprecisión también se introduce en el proceso de consulta a la Base de Datos, al plantearla mediante un conjunto difuso sobre el dominio correspondiente. De esta forma, a partir de una relación básica (las que están en la Base de Datos) se obtiene una nueva relación que consiste en la anterior, añadiendo a cada tupla su grado de cumplimiento a la propiedad dada por el conjunto difuso que aparece en la consulta. Las relaciones así derivadas son difusas, aunque pueden ser tratadas de forma clásica interpretando el grado de cumplimiento como un campo más. Las tuplas que se seleccionan de las así obtenidas vendrán determinadas por el umbral que se imponga en dicho grado.

3 Modelo Generalizado de BD Relacionales Difusas

GEFRED se desarrolla en un ambiente posibilístico pero incorporando un tratamiento coherente para aquella información difusa cuyo origen o representación no es posibilística. Las líneas básicas que sustentan el modelo se desglosan en los apartados relativos a la estructura de datos y a la manipulación de los mismos.

3.1 Estructura de Datos

Uno de los objetivos del modelo es el de proporcionar una amplia cobertura, en cuanto a la representación y tratamiento, de información difusa de toda índole. De acuerdo con esto, el tipo de datos sobre los que opera este modelo se relaciona en la tabla 1.

Los elementos del modelo que estructuran toda esta información son:

- *DOMINIO DIFUSO GENERALIZADO, D_G .*

Definición 1 Si D es el dominio de discurso, $\tilde{\mathcal{P}}(D)$ el conjunto de todas las distribuciones de posibilidad definidas sobre D , incluidas las que definen los tipos *Unknown* y *Undefined*, (Tipos 8 y 9 resp. de la tabla 1), y *NULL* el tipo cuya definición aparece en la citada tabla, llamaremos **Dominio Difuso Generalizado, D_G** , a $D_G \subseteq \tilde{\mathcal{P}}(D) \cup \text{NULL}$.

El Dominio Difuso Generalizado constituye el elemento estructural sobre el que se articulará la representación de los datos recogidos en la tabla 1.

- 1 Un escalar simple (Ej. Aptitud=buena).
- 2 Un número simple (Ej. Edad=28).
- 3 Un conjunto de posibles asignaciones excluyentes de escalares (Ej. Aptitud={mala,buena}).
- 4 Un conjunto de posibles asignaciones excluyentes de números (Ej. Edad= {20,21}).
- 5 Una distribución de posibilidad en el dominio de los escalares (Ej. Aptitud= {0.6/mala,0.7/regular}).
- 6 Una distribución de posibilidad en el dominio de los números (Ej. Edad= {0.4/23,1.0/24,0.8/25}, números difusos, etiquetas lingüísticas)
- 7 Un número Real $\in [0,1]$ representando grados de cumplimiento (Ej. Calidad=0.9).
- 8 Un valor desconocido **Unknown** dado por la distribución de posibilidad **Unknown**= $\{1/u : u \in U\}$ sobre el dominio considerado.
- 9 Un valor indefinido **Undefined** dado por la distribución de posibilidad **Undefined**= $\{0/u : u \in U\}$ sobre el dominio considerado.
- 10 Un valor nulo **NULL** dado por **NULL**= $\{1/Unknown,1/Undefined\}$

Tabla 1: Tipos de datos

• *RELACION DIFUSA GENERALIZADA, R_{FG} .*

Definición 2 Una Relación Difusa Generalizada, R_{FG} , viene dada por un par de conjuntos "cabecera", $(\mathcal{H})$ y "cuerpo", $(\mathcal{B})$, $R_{FG} = (\mathcal{H}, \mathcal{B})$ definidos como sigue:

- La "cabecera" consiste en un conjunto fijo de ternas atributo-dominio-pertenencia,

$$\mathcal{H} = \{(A_{G1} : D_{G1} [, \mu_{A_{G1}}]), (A_{G2} : D_{G2} [, \mu_{A_{G2}}]), \dots, (A_{Gn} : D_{Gn} [, \mu_{A_{Gn}}])\} \quad (1)$$

donde a cada atributo A_{Gj} , le subyace un dominio difuso generalizado, no necesariamente distinto, D_{Gj} ($j=1,2,\dots,n$) y $\mu_{A_{Gj}}$ es un "atributo de pertenencia" que toma valores en el intervalo $[0,1]$.

- El "cuerpo" consiste en un conjunto de tuplas difusas generalizadas, donde cada tupla está compuesta por un conjunto de ternas atributo-valor-grado,

$$\mathcal{B} = \{(A_{G1} : \tilde{d}_{i1} [, \mu_{i1}]), (A_{G2} : \tilde{d}_{i2} [, \mu_{i2}]), \dots, (A_{Gn} : \tilde{d}_{in} [, \mu_{in}])\} \quad (2)$$

con ($i = 1, 2, \dots, m$, siendo m el número de tuplas de la relación), donde $\tilde{d}_{ij}$ representa el valor de dominio que toma la tupla i sobre el atributo A_{Gj} y $\mu_{ij} \neq 0$ el grado de pertenencia de este valor a la tupla i .

Observaciones:

1. Los corchetes que encierran a $\mu_{A_{G_j}}$ y a μ_{ij} denotan un carácter opcional para el *atributo de pertenencia* y por tanto para los *grados de pertenencia*, en cuanto a su aparición, de forma explícita, en la *cabecera* y *cuerpo* de la relación respectivamente.

En una relación difusa generalizada R_{FG} , en la que para un atributo dado, A_{G_j} , no tengamos definido en la *cabecera* el *atributo de pertenencia* $\mu_{A_{G_j}}$ correspondiente, entenderemos que todos los valores de dominio que adopte el *cuerpo* de la relación para dicho atributo, $\tilde{d}_{ij}$, presentarán, implícitamente, un *grado de pertenencia* a la tupla i igual a 1, osea, $\mu_{ij} = 1 \forall i = 1, 2, \dots, m$ siendo m el número de tuplas de la relación.

2. Una tupla i , que contenga un valor cero para alguno de los μ_{ij} que la componen, no pertenecerá a la relación R_{FG} .
3. La definición de *Relación Difusa Generalizada* contempla las relaciones clásicas como un caso particular, donde los dominios clásicos sobre los que se construyen los atributos que constituyen la *cabecera* de la relación, son casos particulares de la definición de *Dominio Difuso Generalizado* y donde no aparecen los *atributos de pertenencia* en la *cabecera* ni los *grados de pertenencia* en el *cuerpo*, ya que estos son considerados igual a uno.

Definición 3 Sea R_{FG} una relación difusa generalizada dada por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G_1} : D_{G_1} [\mu_{A_{G_1}}]), \dots, (A_{G_n} : D_{G_n} [\mu_{A_{G_n}}])\} \\ \mathcal{B} &= \{(A_{G_1} : \tilde{d}_{i1} [\mu_{i1}]), \dots, (A_{G_n} : \tilde{d}_{in} [\mu_{in}])\} \end{cases}$$

llamaremos **Componente de valor de una relación difusa generalizada** y lo notaremos, R_{FG}^v , a la parte de la relación dada por:

$$R_{FG}^v = \begin{cases} \mathcal{H}^v &= \{(A_{G_1} : D_{G_1}), \dots, (A_{G_n} : D_{G_n})\} \\ \mathcal{B}^v &= \{(A_{G_1} : \tilde{d}_{i1}), \dots, (A_{G_n} : \tilde{d}_{in})\} \end{cases} \quad (3)$$

$\mathcal{H}^v$ y $\mathcal{B}^v$ son las componentes de valor de la "cabecera" y el "cuerpo" respectivamente,

así mismo llamaremos **Componente de pertenencia de una relación difusa generalizada** y lo notaremos, R_{FG}^μ , a la parte de la relación dada por:

$$R_{FG}^\mu = \begin{cases} \mathcal{H}^\mu &= \{[\mu_{A_{G_1}}], \dots, [\mu_{A_{G_n}}]\} \\ \mathcal{B}^\mu &= \{[\mu_{i1}], \dots, [\mu_{in}]\} \end{cases} \quad (4)$$

$\mathcal{H}^\mu$ y $\mathcal{B}^\mu$ son las componentes de pertenencia de la "cabecera" y el "cuerpo" respectivamente.

• *CLAVE PRIMARIA GENERALIZADA, K_G .*

Definición 4 Sea R_{FG} una relación difusa generalizada dada por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{r1} [\mu_{r1}]), \dots, (A_{Gn} : \tilde{d}_{rn} [\mu_{rn}])\} \end{cases}$$

con $r = 1, 2, \dots, m$, siendo m el número de tuplas de la relación,

llamaremos **Clave Primaria Generalizada, K_G** , a un subconjunto de $\mathcal{H}$ expresado como:

$$K_G \subseteq \mathcal{H}, K_G = \{(A_{Gs} : D_{Gs}) : s \in S \subseteq \{1, \dots, n\}\} \quad (5)$$

que cumple:

1. $\forall s \in S$, D_{Gs} es un dominio numérico o escalar simple, tipos 1 y 2 de la tabla 1.
2. $\forall i, i' \in \{1, \dots, m\}$, $\exists s \in S : (A_{Gs} : d_{is}) \neq (A_{Gs} : d_{i's})$, donde el operador $\neq$ recoge el significado clásico de la desigualdad.

Con la clave primaria definida de esta forma se garantiza, de un lado, la accesibilidad de cada una de las tuplas de la relación y, de otro, la posibilidad de proporcionar un tratamiento difuso a los atributos incluidos en la clave primaria cuando esto resulte conveniente. Además, la existencia de una clave primaria posibilita que se pueda exigir a una Base de Datos Relacional Difusa realizada según el modelo propuesto que satisfaga las reglas de Identidad y Referencial.

3.2 Manipulación de los Datos

El Modelo describe un Álgebra Relacional Difusa Generalizada con la que se manipulan las $R_{FG,s}$. Las operaciones clásicas, *Unión*, *Intersección*, *Diferencia*, *Producto Cartesiano*, *Proyección*, *Join* y *Selección*, quedan extendidas a fin de que puedan operar sobre las $R_{FG,s}$ en forma coherente.

La *Selección* y el *Join* basan su funcionamiento en el empleo de comparadores difusos contruidos en torno a alguna de las medidas de compatibilidad difusas aparecidas en la literatura. Nuestro modelo adopta un marco básico, sobre el que construir los comparadores difusos partiendo del significado de las operaciones de comparación definidas sobre el dominio de discurso, y extendiendo dicho significado al tratamiento de valores del *Dominio Generalizado Difuso*.

3.2.1 Operadores de Comparación Difusa

Un aspecto crítico que ha de abordar cualquier SBRD es la definición de los operadores que se ven envueltos en las operaciones de *Selección* y *Join*. Este problema lo resuelve el Modelo mediante la definición operadores de comparación a dos niveles: a nivel del Dominio de discurso D y a nivel de D_G . Sobre el primer nivel definiremos el *Comparador Extendido*, θ_e y con su ayuda construiremos el *Comparador Difuso Generalizado*, Θ_G en el segundo.

• **COMPARADOR EXTENDIDO** (θ_e)

Definición 5 Sea D el dominio de discurso considerado, llamaremos **Comparador Extendido**, θ_e , a cualquier operador definido sobre D que pueda ser expresado en la forma:

$$\begin{aligned}\theta_e : D \times D &\longrightarrow [0, 1] \\ \theta_e(d_i, d_j) &\longmapsto [0, 1]\end{aligned}\tag{6}$$

con $d_i, d_j \in D$

El **Comparador Extendido** recoge todas las modalidades de relación existentes entre los valores del Dominio de discurso D considerado, atendiendo a la naturaleza de dichos valores y al carácter, clásico, difuso o de similitud, de la relación existente entre los mismos.

Así definido el **Comparador Extendido** (θ_e) permite modelar en forma consistente los siguientes operadores de comparación:

- Comparadores Clásicos del Álgebra Relacional tales como: $=, \neq, >, \geq, <, \leq$. Por ejemplo, el operador *igual extendido* quedaría expresado mediante $=_e(d_i, d_j) = \delta(d_i, d_j)$ donde $\delta(d_i, d_j) = 1$ para $d_i = d_j$ y $\delta(d_i, d_j) = 0$ para $d_i \neq d_j$.
- Comparadores Difusos tales como "aproximadamente igual", "mucho mayor que", etc. Este tipo de comparadores vendrán expresados mediante funciones de pertenencia. Por ejemplo, se podría modelar el operador "aproximadamente igual" mediante el correspondiente comparador extendido $\simeq_e$ con la siguiente función de pertenencia:

$$\mu_{\simeq_e}(d_i, d_j) = e^{-\beta|d_i - d_j|} \text{ con } \beta > 0$$

- Operadores de Similitud, los cuales operan sobre datos escalares en los que se ha establecido una *relación de similitud* [Zadeh, 71].
- Comparadores en Base a Propiedad. Estos comparadores operan sobre Dominios discretos y basan su actuación en la definición previa de la relación existente entre cada par de valores pertenecientes a dicho dominio en base a una propiedad observada sobre el mismo. Contienen en forma explícita el grado en que cada par de valores del producto cartesiano $D \times D$ son compatibles según la propiedad que modela este comparador.

Ejemplo 1 Supongamos $P = \{\text{malo, regular, bueno, excelente}\}$, el dominio finito de escalares asociado a un atributo que almacena el rendimiento de un empleado. Se puede considerar sobre este Dominio la propiedad "más eficiente que", la cual nos proporciona en que medida son más eficientes entre sí cada valor del dominio. Así, por ejemplo, esta propiedad pone de manifiesto que el valor de dominio "bueno" es "más eficiente que" el valor "regular" y algo más que "malo". Esto nos conduce, a la hora de modelar este tipo de operadores, a asignar a cada par de valores un grado de compatibilidad, comprendido entre 0 y 1, con la propiedad "más eficiente que". Con lo cual, por ejemplo, podríamos modelar la propiedad "más eficiente que" definida sobre el dominio P mediante la siguiente relación:

d	d'	$mas\_eficiente\_que_e(d, d')$
<i>malo</i>	<i>malo</i>	0
<i>regular</i>	<i>malo</i>	0.4
$\vdots$	$\vdots$	$\vdots$
<i>excelente</i>	<i>bueno</i>	0.5
<i>excelente</i>	<i>excelente</i>	0

En forma similar se podrían modelar otras propiedades definidas sobre el dominio P , como por ejemplo: "mucho más eficiente que", "menos eficiente que", etc.

• **COMPARADOR DIFUSO GENERALIZADO, $\Theta_G^{\theta_e}$.**

Definición 6 Sea D el dominio de discurso considerado, sea D_G el dominio difuso generalizado construido sobre él y sea θ_e un comparador extendido definido en D . Consideremos una función $\Theta_G^{\theta_e}$ definida:

$$\begin{aligned}\Theta_G^{\theta_e} : D_G \times D_G &\rightarrow [0, 1] \\ \Theta_G^{\theta_e}(\tilde{d}_1, \tilde{d}_2) &\in [0, 1]\end{aligned}\quad (7)$$

diremos que $\Theta_G^{\theta_e}$ es un Comparador Difuso Generalizado sobre D_G inducido por el comparador extendido θ_e , si cumple:

$$\Theta_G^{\theta_e}(\tilde{d}_1, \tilde{d}_2) = \theta_e(d_1, d_2) \quad \forall d_1, d_2 \in D \quad (8)$$

donde $\tilde{d}_1, \tilde{d}_2$ representan las distribuciones de posibilidad $1/d_1, 1/d_2$, inducidas, respectivamente, por los valores d_1, d_2 .

3.2.2 Álgebra Relacional Difuso Generalizado

Con los comparadores difusos anteriormente definidos y con la estructura de datos adoptada por el modelo, estamos en condiciones de extender el significado de los operadores del Álgebra Relacional, a fin de que podamos manipular en forma conveniente las "relaciones difusas generalizadas".

• **UNIÓN DIFUSA GENERALIZADA, $\cup_G$**

Definición 7 Sean R_{FG} y R'_{FG} dos relaciones difusas generalizadas dadas por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{i1} [\mu_{i1}]), \dots, (A_{Gn} : \tilde{d}_{in} [\mu_{in}])\} \end{cases}$$

$$R'_{FG} = \begin{cases} \mathcal{H}' &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B}' &= \{(A_{G1} : \tilde{d}'_{k1} [\mu'_{k1}]), \dots, (A_{Gn} : \tilde{d}'_{kn} [\mu'_{kn}])\} \end{cases}$$

con $i = 1, \dots, m$ y $k = 1, \dots, m'$, siendo m y m' las respectivas cardinalidades,

entonces la Unión Difusa Generalizada de R_{FG} y R'_{FG} vendrá definida por:

$$R_{FG} \cup_G R'_{FG} = \begin{cases} \mathcal{H}_{\cup_G} &= \{(A_{G1} : D_{G1} [, \mu_{AG1}]), \dots, (A_{Gn} : D_{Gn} [, \mu_{AGn}])\} \\ \mathcal{B}_{\cup_G} &= \begin{cases} \mathcal{B}_{\cup_G}^v &= \mathcal{B}^v \cup \mathcal{B}'^v \\ \mathcal{B}_{\cup_G}^\mu &= \{[\mu''_{i1}], \dots, [\mu''_{in}]\} \end{cases} \end{cases} \quad (9)$$

con $l = 1, \dots, m''$, siendo m'' la cardinalidad de la unión, donde

$$\mu''_{ij} = \max\{\mu_{ij}, \mu'_{ij}\} \quad (10)$$

siendo μ_{ij} el grado de pertenencia de d''_{ij} a la tupla l en la relación R_{FG} y μ'_{ij} el grado de pertenencia de d'_{ij} a la tupla l en la relación R'_{FG} .

Por tanto, la Unión Difusa Generalizada de dos relaciones Difusas Generalizadas, contiene aquellas tuplas que pertenecen a la unión de ambas relaciones, adoptando, para el grado de pertenencia a cada tupla de la unión, μ''_{ij} , el máximo de los grados de pertenencia adoptados por dichos valores en las relaciones de origen, $\mu''_{ij} = \max\{\mu_{ij}, \mu'_{ij}\}$.

• INTERSECCIÓN DIFUSA GENERALIZADA, $\cap_G$

Definición 8 Sean R_{FG} y R'_{FG} dos relaciones difusas generalizadas dadas por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [, \mu_{AG1}]), \dots, (A_{Gn} : D_{Gn} [, \mu_{AGn}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{i1} [, \mu_{i1}]), \dots, (A_{Gn} : \tilde{d}_{in} [, \mu_{in}])\} \end{cases}$$

$$R'_{FG} = \begin{cases} \mathcal{H}' &= \{(A_{G1} : D_{G1} [, \mu_{AG1}]), \dots, (A_{Gn} : D_{Gn} [, \mu_{AGn}])\} \\ \mathcal{B}' &= \{(A_{G1} : \tilde{d}'_{k1} [, \mu'_{k1}]), \dots, (A_{Gn} : \tilde{d}'_{kn} [, \mu'_{kn}])\} \end{cases}$$

con $i = 1, \dots, m$ y $k = 1, \dots, m'$, siendo m y m' las respectivas cardinalidades,

entonces la Intersección Difusa Generalizada de R_{FG} y R'_{FG} vendrá definida por:

$$R_{FG} \cap_G R'_{FG} = \begin{cases} \mathcal{H}_{\cap_G} &= \{(A_{G1} : D_{G1} [, \mu_{AG1}]), \dots, (A_{Gn} : D_{Gn} [, \mu_{AGn}])\} \\ \mathcal{B}_{\cap_G} &= \begin{cases} \mathcal{B}_{\cap_G}^v &= \mathcal{B}^v \cap \mathcal{B}'^v \\ \mathcal{B}_{\cap_G}^\mu &= \{[\mu''_{i1}], \dots, [\mu''_{in}]\} \end{cases} \end{cases} \quad (11)$$

con $l = 1, \dots, m''$, siendo m'' la cardinalidad de la intersección, donde

$$\mu''_{ij} = \min\{\mu_{ij}, \mu'_{ij}\} \quad (12)$$

siendo μ_{ij} el grado de pertenencia de d''_{ij} a la tupla l en la relación R_{FG} y μ'_{ij} el grado de pertenencia de d'_{ij} a la tupla l en la relación R'_{FG} .

Así pues, la Intersección Difusa Generalizada de dos relaciones Difusas Generalizadas, contiene aquellas tuplas que pertenecen a la intersección de ambas relaciones, adoptando, para el grado de pertenencia a cada tupla de la intersección, μ''_{ij} , el mínimo de los grados de pertenencia adoptados por dichos valores en las relaciones de origen, $\mu''_{ij} = \min\{\mu_{ij}, \mu'_{ij}\}$.

• **DIFERENCIA DIFUSA GENERALIZADA, $-_G$**

Definición 9 Sean R_{FG} y R'_{FG} dos relaciones difusas generalizadas dadas por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{i1} [\mu_{i1}]), \dots, (A_{Gn} : \tilde{d}_{in} [\mu_{in}])\} \end{cases}$$

$$R'_{FG} = \begin{cases} \mathcal{H}' &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B}' &= \{(A_{G1} : \tilde{d}'_{k1} [\mu'_{k1}]), \dots, (A_{Gn} : \tilde{d}'_{kn} [\mu'_{kn}])\} \end{cases}$$

con $i = 1, \dots, m$ y $k = 1, \dots, m'$, siendo m y m' las respectivas cardinalidades, entonces la **Diferencia Difusa Generalizada** de R_{FG} y R'_{FG} vendrá definida por:

$$R_{FG} -_G R'_{FG} = \begin{cases} \mathcal{H}_{-G} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B}_{-G} &= \begin{cases} \mathcal{B}_{-G}^v &= \mathcal{B}^v - \mathcal{B}'^v \\ \mathcal{B}_{-G}^\mu &= \{[\mu''_{l1}], \dots, [\mu''_{ln}]\} \end{cases} \end{cases} \quad (13)$$

con $l = 1, \dots, m''$, siendo m'' la cardinalidad de la diferencia, donde

$$\mu''_{lj} = \min\{\mu_{lj}, (1 - \mu'_{lj})\} \quad (14)$$

siendo μ_{lj} el grado de pertenencia de d''_{lj} a la tupla l en la relación R_{FG} y μ'_{lj} el grado de pertenencia de d'_{lj} a la tupla l en la relación R'_{FG} .

Como puede observarse, la Diferencia Difusa Generalizada de dos relaciones Difusas Generalizadas, contiene aquellas tuplas que pertenecen a la diferencia de ambas relaciones, adoptando, para el grado de pertenencia a cada tupla de la diferencia, μ''_{lj} , el mínimo entre el grado de pertenencia a la tupla en la relación R_{FG} y el complemento del grado de pertenencia a la tupla en la relación R'_{FG} , $\mu''_{lj} = \min\{\mu_{lj}, (1 - \mu'_{lj})\}$.

• **PRODUCTO CARTESIANO DIFUSO GENERALIZADO, $\times_G$**

Definición 10 Sean R_{FG} y R'_{FG} dos relaciones difusas generalizadas dadas por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{i1} [\mu_{i1}]), \dots, (A_{Gn} : \tilde{d}_{in} [\mu_{in}])\} \end{cases}$$

$$R'_{FG} = \begin{cases} \mathcal{H}' &= \{(A'_{G1} : D'_{G1} [\mu_{A'_{G1}}]), \dots, (A'_{Gn'} : D'_{Gn'} [\mu_{A'_{Gn'}}])\} \\ \mathcal{B}' &= \{(A'_{G1} : \tilde{d}'_{k1} [\mu'_{k1}]), \dots, (A'_{Gn'} : \tilde{d}'_{kn'} [\mu'_{kn'}])\} \end{cases}$$

con $i = 1, \dots, m$ y $k = 1, \dots, m'$, siendo m y m' las respectivas cardinalidades y n, n' los respectivos grados, entonces el **Producto Cartesiano Difuso Generalizado** se define como:

$$R_{FG} \times_G R'_{FG} = \begin{cases} \mathcal{H}_{\times_G} &= \mathcal{H} \times \mathcal{H}' \\ \mathcal{B}_{\times_G} &= \mathcal{B} \times \mathcal{B}' \end{cases} \quad (15)$$

Como se puede observar el **Producto Cartesiano Difuso Generalizado** opera igual que el producto cartesiano usual, solo que considera a los atributos de pertenencia y a sus valores como partes integrantes de las relaciones.

• **PROYECCIÓN DIFUSA GENERALIZADA, $\mathcal{P}_G$**

Definición 11 Sea R_{FG} una relación difusa generalizada dada por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{r1} [\mu_{r1}]), \dots, (A_{Gn} : \tilde{d}_{rn} [\mu_{rn}])\} \end{cases}$$

con $r = 1, 2, \dots, m$, siendo m el número de tuplas de la relación,

Sea X un subconjunto de $\mathcal{H}$ expresado como:

$$X \subseteq \mathcal{H}, X = \{(A_{Gs} : D_{Gs} [\mu_{A_{Gs}}]) : s \in S, s' \in S'; S, S' \subseteq \{1, \dots, n\}\}$$

entonces diremos que la **Proyección Difusa Generalizada de R_{FG} sobre X** , $\mathcal{P}_G(R_{FG}; X)$ es una relación difusa generalizada dada por:

$$\mathcal{P}_G(R_{FG}; X) = \begin{cases} \mathcal{H}_P &= X \\ \mathcal{B}_P &= \{(A_{Gs} : \tilde{d}_{rs} [\mu_{rs'}])\} \end{cases} \quad (16)$$

con $s \in S, s' \in S', S, S' \subseteq \{1, \dots, n\}$.

La proyección se puede realizar sobre cualquier subconjunto de la *cabecera*, pudiendo incluir éste "atributos de valor" así como "atributos de pertenencia". El resultado de la proyección es una "selección vertical" sobre el *cuerpo* de la relación.

• **SELECCIÓN DIFUSA GENERALIZADA, S_G**

Definición 12 Sea R_{FG} una relación difusa generalizada dada por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{r1} [\mu_{r1}]), \dots, (A_{Gn} : \tilde{d}_{rn} [\mu_{rn}])\} \end{cases}$$

con $r = 1, 2, \dots, m$, siendo m el número de tuplas de la relación,

sea $\tilde{a} \in D_G$ una constante, sea $\Theta_G^{\theta_g}$ un "comparador difuso generalizado" y sea $\gamma \in [0, 1]$ un "umbral de cumplimiento", entonces la **Selección Difusa Generalizada realizada sobre R_{FG} por la condición inducida por $\Theta_G^{\theta_g}$ compuesto con $\tilde{a}$ sobre el atributo A_{Gi} y cualificada por γ** , $S_G(R_{FG}; \Theta_G^{\theta_g}(A_i, \tilde{a}) \geq \gamma)$, es una relación R_{FG}^S dada por:

$$R_{FG}^S = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{r'1} [\mu_{r'1}]), \dots, (A_{Gi} : \tilde{d}_{r'i} [\mu_{r'i}']), \dots, (A_{Gn} : \tilde{d}_{r'n} [\mu_{r'n}'])\} \end{cases} \quad (17)$$

con

$$\mu_{r'i} = \Theta_G^{\theta_g}(\tilde{d}_{r'i}, \tilde{a}) \geq \gamma, \quad (18)$$

$r' = 1, \dots, m'$, siendo m' la cardinalidad de la selección.

- Reglas de Composición sobre la Selección Generalizada, S_G

Definición 13 Sea R_{FG} una relación difusa generalizada dada por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{r1} [\mu_{r1}]), \dots, (A_{Gn} : \tilde{d}_{rn} [\mu_{rn}])\} \end{cases}$$

con $r = 1, 2, \dots, m$, siendo m el número de tuplas de la relación,

- sea $S \subseteq \{1, \dots, n\}$ siendo n el "grado" de R_{FG} ,
- sean $\tilde{a}_i \in D_{Gi}$, $i \in S$ un conjunto de constantes no necesariamente distintas y A_{Gi} , $i \in S$ un conjunto de atributos de la cabecera de R_{FG} no necesariamente distintos,
- sean $\Theta_{Gi}^{\theta_i}$, $i \in S$ un conjunto de comparadores difusos generalizados, no necesariamente distintos,
- sean $\gamma_i \in [0, 1]$, un conjunto de "umbrales de cumplimiento",
- y sean $S_G(R_{FG}; \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i)$, $i \in S$ el conjunto de selecciones difusas generalizadas inducidas por las constantes $\tilde{a}_i$ y los comparadores difusos generalizados $\Theta_{Gi}^{\theta_i}$ sobre los atributos A_{Gi} de R_{FG} y cualificadas por γ_i ,

entonces definimos las siguientes composiciones sobre la Selección Generalizada:

1. Llamaremos **Complemento de la Selección Generalizada**,

$\neg S_G(R_{FG}; \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i) = S_G(R_{FG}; \neg \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i)$, a la relación difusa generalizada, R_{FG}^S dada por:

$$R_{FG}^S = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{r'1} [\mu_{r'1}]), \dots, (A_{Gi} : \tilde{d}_{r'i} [\mu_{r'i}]), \dots, (A_{Gn} : \tilde{d}_{r'n} [\mu_{r'n}])\} \end{cases} \quad (19)$$

con

$$\mu_{r'i} = 1 - \Theta_{Gi}^{\theta_i}(\tilde{d}_{r'i}, \tilde{a}_i) \geq \gamma_i \quad (20)$$

$r' = 1, \dots, m'$, siendo m' la cardinalidad de la selección.

2. Se define la **Conjunción sobre la Selección Generalizada** mediante la expresión:

$$\bigwedge_{i \in S} S_G(R_{FG}; \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i) = \bigcap_{i \in S} S_G(R_{FG}; \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i) \quad (21)$$

3. La **Disyunción sobre la Selección Generalizada** viene dada por la expresión:

$$\bigvee_{i \in S} S_G(R_{FG}; \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i) = \bigcup_{i \in S} S_G(R_{FG}; \Theta_{Gi}^{\theta_i}(A_i, \tilde{a}_i) \geq \gamma_i) \quad (22)$$

La **Selección Difusa Generalizada** así definida, proporciona un mecanismo por el cual se ajustan, los "grados de pertenencia" de los "valores de atributo" según las condiciones impuestas en la consulta y se eliminan aquellas tuplas que no satisfacen un "umbral de cumplimiento" establecido. Para ello, se hace uso de los "comparadores difusos generalizados" empleados para modelar las condiciones. También se han introducido reglas mediante las cuales componer consultas complejas a partir de unas más simples.

• *JOIN DIFUSO GENERALIZADO, $\bowtie_G$*

Definición 14 Sean R_{FG} y R'_{FG} dos relaciones difusas generalizadas, no necesariamente distintas, dadas por:

$$R_{FG} = \begin{cases} \mathcal{H} &= \{(A_{G1} : D_{G1} [\mu_{A_{G1}}]), \dots, (A_{Gn} : D_{Gn} [\mu_{A_{Gn}}])\} \\ \mathcal{B} &= \{(A_{G1} : \tilde{d}_{j1} [\mu_{j1}]), \dots, (A_{Gn} : \tilde{d}_{jn} [\mu_{jn}])\} \end{cases}$$

$$R'_{FG} = \begin{cases} \mathcal{H}' &= \{(A'_{G1} : D'_{G1} [\mu_{A'_{G1}}]), \dots, (A'_{Gn'} : D'_{Gn'} [\mu_{A'_{Gn'}}])\} \\ \mathcal{B}' &= \{(A'_{G1} : \tilde{d}'_{k1} [\mu'_{k1}]), \dots, (A'_{Gn'} : \tilde{d}'_{kn'} [\mu'_{kn'}])\} \end{cases}$$

con $j = 1, \dots, m$ y $k = 1, \dots, m'$, siendo m y m' las respectivas cardinalidades y n, n' los respectivos grados,

sea $\Theta_G^{\theta_\epsilon}$ un "comparador difuso generalizado" y sea $\gamma \in [0, 1]$ un "umbral de cumplimiento", entonces la **Join Difusa Generalizada** realizada sobre $R_{FG} \times_G R'_{FG}$ mediante la condición $\Theta_G^{\theta_\epsilon}$ inducida sobre los atributos A_{Gi} de R_{FG} y $A'_{Gi'}$ de R'_{FG} , cualificada por γ , $\bowtie_G (R_{FG} \times_G R'_{FG}; \Theta_G^{\theta_\epsilon}(A_i, A'_{i'}) \geq \gamma)$, viene dada por:

$$R_{FG}^{\bowtie} = \begin{cases} \mathcal{H}^{\bowtie} &= \mathcal{H} \times \mathcal{H}' \\ \mathcal{B}^{\bowtie} &= \{(A_{G1} : \tilde{d}_{r1} [\mu_{r1}]), \dots, (A_{Gi} : \tilde{d}_{ri} [\mu''_{ri}]), \dots, \\ &\quad (A'_{Gi'} : \tilde{d}'_{ri'} [\mu''_{ri'}]), \dots, (A'_{Gn''} : \tilde{d}'_{rn''} [\mu'_{rn''}])\} \end{cases} \quad (23)$$

con $n'' = n + n'$, donde r es la cardinalidad del Join y

$$\mu''_{ri} = \mu''_{ri'} = \Theta_G^{\theta_\epsilon}(\tilde{d}_{ri}, \tilde{d}'_{ri'}) \geq \gamma \quad (24)$$

La *Join Difusa Generalizada* se puede entender como una especie de *Selección Difusa Generalizada* realizada sobre el *Producto Cartesiano Difuso Generalizado* de las dos relaciones implicadas, y cuyo resultado es la variación de los grados de pertenencia de los atributos sobre los que se ejecuta la Join y la supresión de aquellas tuplas que no presenten un "grado de Join" superior o igual a un umbral establecido.

4 Ejemplo de Representación y Manipulación de una BD Relacional Difusa Según GEFRED

Para ilustrar el funcionamiento del modelo vamos a emplear el ejemplo de una relación que recoge datos, algunos de ellos de naturaleza difusa, sobre un conjunto de empleados. Este ejemplo no agota todas las posibilidades del modelo GEFRED, tanto en la representación, como en la manipulación de información difusa.

Suponemos que la información que almacenan los atributos NOMBRE y CALLE son de naturaleza precisa. Los dominios que subyacen a EDAD y SALARIO, son numéricos y tienen definidos sobre ellos las distribuciones de posibilidad, (expresadas en forma trapezoidal), etiquetadas y definidas como muestran las figuras 1, 3 y 2. Se contempla la presencia de valores "Unknown" en ambos atributos pero, dada la naturaleza de la relación, no existen

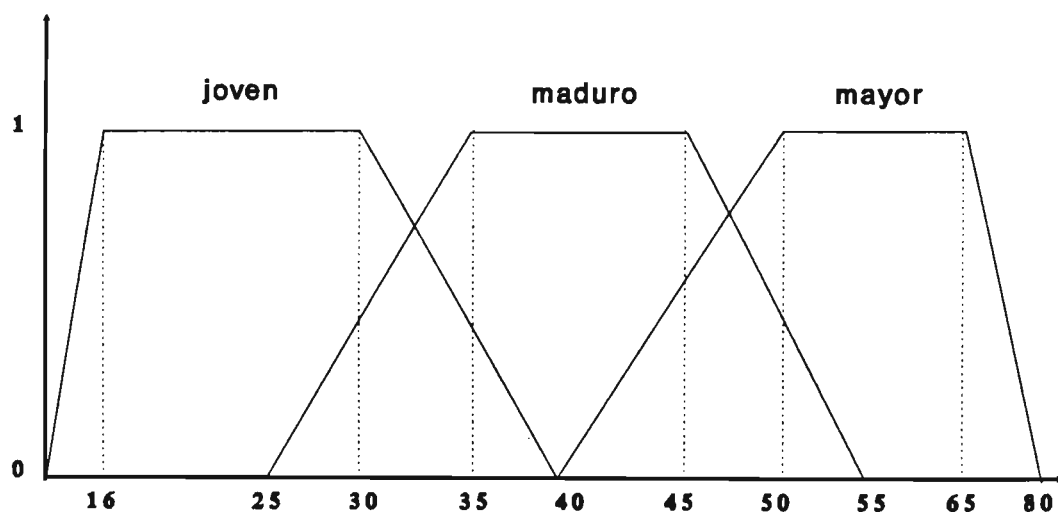


Figura 1: Etiquetas lingüísticas definidas sobre el dominio EDAD.

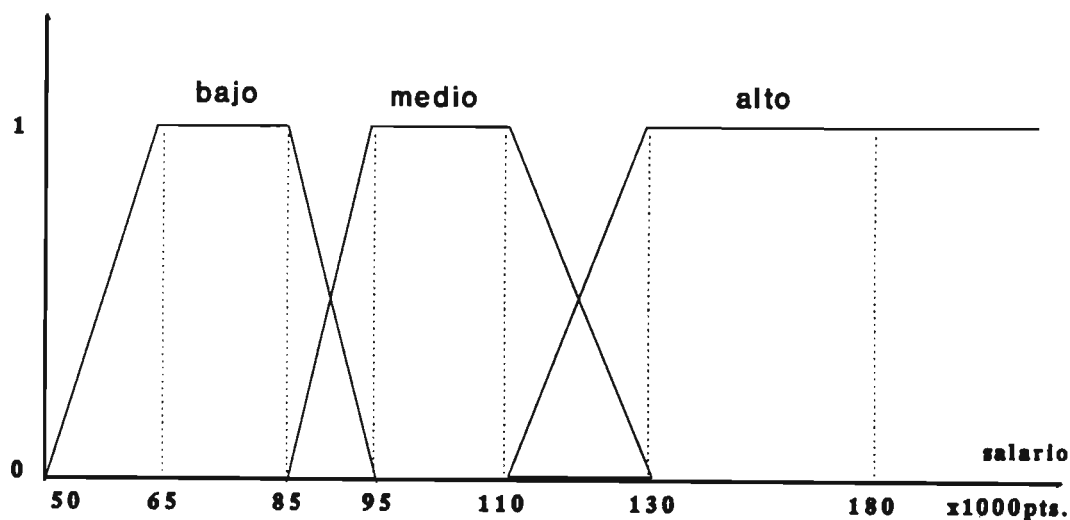


Figura 2: Etiquetas lingüísticas definidas sobre el dominio SALARIO.

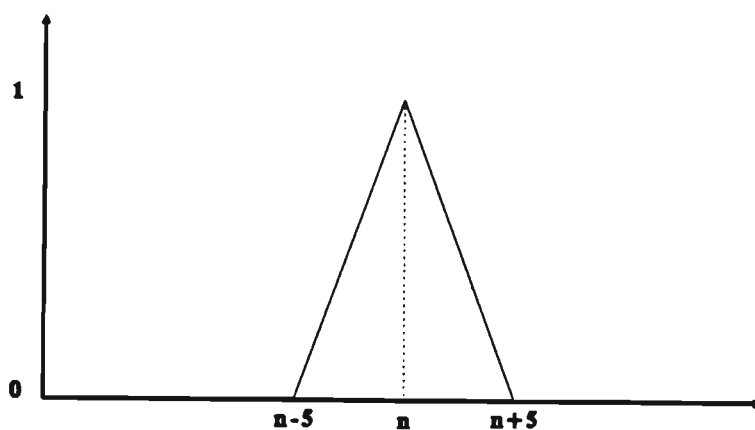


Figura 3: Etiqueta "Aproximadamente" sobre el dominio EDAD.

$\mathcal{H}$	NOMBRE	CALLE	EDAD	RENDIMIENTO	SALARIO
$\mathcal{B}$	Luis	Recogidas	&.8/30,1/31	Bueno	110000
	Antonio	Reyes Católicos	Maduro	Regular	100000
	Juan Carlos	Camino Ronda	Joven	Malo	90000
	Francisco	P. A. Alarcón	Mayor	Excelente	150000
	Julia	Puerta Real	Joven	Bueno	130000
	Inés	Manuel de Falla	#28	Bueno	125000
	Javier	Gran Vía	*30,35	Regular	105000

El símbolo # significa "aproximadamente" y su definición aparece en la figura 3, & indica distribución de posibilidad y * intervalo.

Tabla 2: Tabla de Empleados, $\mathbf{Emp}$

valores "undefined" y por tanto tampoco "NULL". El dominio que subyace al atributo RENDIMIENTO, es del tipo 1 de la tabla 1, en la tabla 3 se especifican los valores del dominio así como la relación de similitud que existe entre ellos. También permite en principio valores "Unknown".

De acuerdo con el modelo GEFRED, la "cabecera" de la relación difusa generalizada que estructura la información vendría dada por:

$$\mathcal{H} = \{ \text{NOMBRE} : D_{\text{NOMBRE}}, D_{\text{CALLE}} : \text{CALLE}, \text{EDAD} : D_{\text{EDAD}}, \text{RENDIMIENTO} : D_{\text{RENDIMIENTO}}, \text{SALARIO} : D_{\text{SALARIO}} \} \quad (25)$$

Donde, no aparecen los atributos de pertenencia porque el grado de pertenencia de cada valor de atributo a la tupla es igual a 1. El "cuerpo" de la relación está constituido por todas las tuplas mostradas en la relación de la tabla 2.

Para poner de manifiesto el funcionamiento del *Álgebra Relacional Difuso Generalizado*, vamos a estudiar el proceso de resolución de una consulta ejemplo.

En la primera consulta estamos interesados en encontrar empleados que con un rendimiento "bueno" perciban un salario "alto", o que con una edad "mayor" tengan un salario "medio". En términos más precisos, podríamos formular la consulta:

Consulta: Dame el NOMBRE, la EDAD, el RENDIMIENTO y el SALARIO, así como el grado en que cada atributo satisface la condición, de aquellos empleados que presenten RENDIMIENTO "bueno" (con un grado ≥ 0.9) y un SALARIO "alto" (con un grado ≥ 0.7) ó que tengan una EDAD "mayor" (con un grado ≥ 0.6) y un SALARIO "medio" (con un grado ≥ 0.7)

Los "comparadores difusos generalizados" implícitos en la consulta deberán, ó estar definidos en la implementación, o bien estar almacenados como componentes del meta-conocimiento de la Base de Datos Difusa. GEFRED, en este aspecto, no precisa como se ha de modelar el comparador de forma específica. Esto es así, porque la naturaleza de la información, condiciona, además de la representación de los datos, la forma de operar con ellos. En nuestro ejemplo vamos a emplear los siguientes modelos para los comparadores

$s_e(d, d')$	Malo	Regular	Bueno	Excelente
Malo	1	0.8	0.5	0.1
Regular	0.8	1	0.7	0.5
Bueno	0.5	0.7	1	0.8
Excelente	0.1	0.5	0.8	1

Tabla 3: Relación de analogía sobre el dominio RENDIMIENTO.

implicados en la consulta:

- Para la compatibilidad en los atributos EDAD y SALARIO empleamos el "comparador extendido" habitual $=_e(d, d') = \delta(d, d')$ y el "comparador difuso generalizado" inducido:

$$\Theta^{=e}(\tilde{d}, \tilde{d}') = \sup_{(d, d') \in D \times D} \min(=_e(d, d'), \pi_{\tilde{d}}(d), \pi_{\tilde{d}'}(d')) \quad (26)$$

donde $\pi_{\tilde{d}}(d)$ y $\pi_{\tilde{d}'}(d')$ son las distribuciones de posibilidad asociadas a $\tilde{d}$ y a $\tilde{d}'$, respectivamente.

- Para RENDIMIENTO emplearemos el "comparador extendido" $s_e(d, d')$ dado en la Tabla 3. El "comparador difuso generalizado" usado será:

$$\Theta^{s_e}(\tilde{d}, \tilde{d}') = \sup_{(d, d') \in D \times D} \min(s_e(d, d'), \pi_{\tilde{d}}(d), \pi_{\tilde{d}'}(d')) \quad (27)$$

donde nuevamente, $\pi_{\tilde{d}}(d)$ y $\pi_{\tilde{d}'}(d')$ son las distribuciones de posibilidad asociadas a $\tilde{d}$ y a $\tilde{d}'$, respectivamente.

Una vez precisados los comparadores, la consulta, en los términos del Álgebra de GEFRED, queda expresada como sigue:

$$\mathcal{P}_G((\mathcal{S}_1 \wedge_G \mathcal{S}_2) \vee_G (\mathcal{S}_3 \wedge_G \mathcal{S}_4); X) \quad (28)$$

$\mathcal{H}$	NOMBRE	CALLE	EDAD	REND	μ_{REND}	SALARIO
$\mathcal{B}$	Luis	Recogidas	&.8/30,1/31	Bueno	1	110000
	Julia	Puerta Real	Joven	Bueno	1	130000
	Inés	Manuel de Falla	#28	Bueno	1	125000

Los valores de μ_{REND} se obtienen de la relación de similitud mostrada en la Tabla 3. Las tuplas que presentan un valor $< \gamma = 0.9$ para el grado de pertenencia del atributo RENDIMIENTO no pertenecen a la relación.

Tabla 4: Cálculo de $\mathcal{S}_1$

$\mathcal{H}$	NOMBRE	CALLE	EDAD	RENDIMIENTO	SALARIO	μ_{SAL}
$\mathcal{B}$	Francisco	P. A. Alarcón	Mayor	Excelente	150000	1
	Julia	Puerta Real	Joven	Bueno	130000	1
	Inés	Manuel de Falla	#28	Bueno	125000	0.75

Los valores de μ_{SAL} , que se calculan según la ecuación 26, para etiquetas trapezoidales pueden ser calculados de forma trigonométrica. Las tuplas que presentan un valor $< \gamma = 0.7$ para el grado de pertenencia del atributo SAL no pertenecen a la relación.

Tabla 5: Cálculo de $\mathcal{S}_2$

donde:

- $\mathcal{S}_1 = \mathcal{S}_G(\text{Emp}; \Theta^*(\text{RENDIMIENTO}, \text{Bueno}) \geq 0.9)$ con "Bueno" $\in D_{\text{REND}}$.
- $\mathcal{S}_2 = \mathcal{S}_G(\text{Emp}; \Theta^*(\text{SALARIO}, \text{Alto}) \geq 0.7)$ siendo "Alto" la etiqueta que referencia la distribución de posibilidad mostrada en la figura 2.
- $\mathcal{S}_3 = \mathcal{S}_G(\text{Emp}; \Theta^*(\text{EDAD}, \text{Mayor}) \geq 0.6)$ siendo "Mayor" la etiqueta que referencia la distribución de posibilidad mostrada en la figura 1.
- $\mathcal{S}_4 = \mathcal{S}_G(\text{Emp}; \Theta^*(\text{SALARIO}, \text{Medio}) \geq 0.7)$ siendo "Medio" la etiqueta que referencia la distribución de posibilidad mostrada en la figura 2.
- $X = \{NOMB, EDAD, \mu_{EDAD}, REND, \mu_{REND}, SAL, \mu_{SAL}\}$

En las tablas 4, 5, 6, 7, 8, 9, 10 y 11, se detallan las diferentes operaciones llevadas a cabo para resolver la consulta.

$\mathcal{H}$	NOMBRE	CALLE	EDAD	μ_{EDAD}	RENDIMIENTO	SALARIO
$\mathcal{B}$	Antonio	Reyes Católicos	Maduro	0.75	Regular	100000
	Francisco	P. A. Alarcón	Mayor	1	Excelente	150000
	Julia	Puerta Real	Joven	1	Bueno	130000

Los valores de μ_{EDAD} , que se calculan según la ecuación 26, para etiquetas trapezoidales pueden ser calculados de forma trigonométrica. Las tuplas que presentan un valor $< \gamma = 0.6$ para el grado de pertenencia del atributo EDAD no pertenecen a la relación.

Tabla 6: Cálculo de $\mathcal{S}_3$

$\mathcal{H}$	NOMBRE	CALLE	EDAD	RENDIMIENTO	SALARIO	μ_{SAL}
$\mathcal{B}$	Luis	Recogidas	&.8/30,1/31	Bueno	110000	1
	Antonio	Reyes Católicos	Maduro	Regular	100000	1
	Inés	Manuel de Falla	#28	Bueno	125000	0.75
	Javier	Gran Vía	*30,35	Regular	105000	1

Los valores de μ_{SAL} , que se calculan según la ecuación 26, para etiquetas trapezoidales pueden ser calculados de forma trigonométrica. Las tuplas que presentan un valor $< \gamma = 0.7$ para el grado de pertenencia del atributo SAL no pertenecen a la relación.

Tabla 7: Cálculo de $\mathcal{S}_4$

$\mathcal{H}$	NOMBRE	CALLE	EDAD	REND	μ_{REND}	SALARIO	μ_{SAL}
$\mathcal{B}$	Julia	Puerta Real	Joven	Bueno	1	130000	1
	Inés	Manuel de Falla	#28	Bueno	1	125000	0.75

Para el cálculo de la "Conjunción Generalizada Difusa" se halla la "Intersección Generalizada Difusa" de las relaciones calculadas por las consultas $\mathcal{S}_1$ y $\mathcal{S}_2$.

Tabla 8: Cálculo de $\mathcal{S}_1 \wedge_G \mathcal{S}_2$

$\mathcal{H}$	NOMBRE	CALLE	EDAD	μ_{EDAD}	REND	SALARIO	μ_{SAL}
$\mathcal{B}$	Antonio	Reyes Católicos	Maduro	0.75	Regular	100000	1

Para el cálculo de la "Conjunción Generalizada Difusa" se halla la "Intersección Generalizada Difusa" de las relaciones calculadas por las consultas $\mathcal{S}_3$ y $\mathcal{S}_4$.

Tabla 9: Cálculo de $\mathcal{S}_3 \wedge_G \mathcal{S}_4$

$\mathcal{H}$	...	EDAD	μ_{EDAD}	RENDIMIENTO	μ_{REND}	SALARIO	μ_{SAL}
$\mathcal{B}$	...	Maduro	0.75	Regular	1	100000	1
	...	Mayor	1	Excelente	0.8	150000	1
	...	Joven	1	Bueno	1	130000	1
	...	#28	1	Bueno	1	125000	0.75

Para el cálculo de la "Disyunción Generalizada Difusa" se halla la "Unión Generalizada Difusa" de $(\mathcal{S}_1 \wedge_G \mathcal{S}_2)$ y $(\mathcal{S}_3 \wedge_G \mathcal{S}_4)$.

Tabla 10: Cálculo de $((\mathcal{S}_1 \wedge_G \mathcal{S}_2) \vee_G (\mathcal{S}_3 \wedge_G \mathcal{S}_4))$

$\mathcal{H}$	NOMBRE	EDAD	μ_{EDAD}	RENDIMIENTO	μ_{REND}	SALARIO	μ_{SAL}
$\mathcal{B}$	Antonio	Maduro	0.75	Regular	1	100000	1
	Francisco	Mayor	1	Excelente	0.8	150000	1
	Julia	Joven	1	Bueno	1	130000	1
	Inés	#28	1	Bueno	1	125000	0.75

Se proyecta sobre los atributos, ya sean de valor ó de pertenencia.

Tabla 11: $\mathcal{P}_G((\mathcal{S}_1 \wedge_G \mathcal{S}_2) \vee_G (\mathcal{S}_3 \wedge_G \mathcal{S}_4); X)$

5 Conclusiones

El modelo que presentamos en este trabajo aporta las siguientes características frente a otras propuestas recogidas en la literatura:

- Proporciona tratamiento difuso a la información: tanto a los datos difusos, como a las relaciones que existen entre ellos.
- Los tipos de datos con los que opera es bastante extenso en comparación con otras propuestas. Su naturaleza difusa es mas dispersa que en modelos precedentes.
- Organiza en forma consistente la información difusa. Las "relaciones difusas generalizadas" estructuran, tanto la información de partida, como la que resulta de las operaciones realizadas sobre la misma. Una tupla difusa generalizada, se entiende como una serie de valores de atributo, difusos o no, que presentan un grado de pertenencia, mayor o menor, a la relación que dicha tupla expresa. En el ejemplo anterior una de las tuplas que aparece como resultado de la consulta, p.e., es {Antonio, (Maduro,0.75), (Regular,1), (100000,1)}, esto significa que, en la base de datos hay una tupla que responde a las condiciones de la consulta con un grado 0.75 para los requerimientos de edad, y con un grado 1, para los de rendimiento y salario.
- Dentro del conjunto de metainformación que el experto o usuario debe proporcionar a la Base de Datos se encuentra, además de la definición que adopta para las diferentes etiquetas, la del modelo de comparador difuso que va asociar a cada una de las comparaciones a establecer para cada dominio definido.
- El modelo presenta gran flexibilidad para el tratamiento y evaluación de la información difusa, basado en la libertad a la hora de seleccionar un comparador difuso generalizado u otro y en que se puede controlar el grado en que se satisfacen las condiciones individuales de una consulta.
- Por último una gran parte de los modelos precedentes se pueden considerar, bajo ciertas condiciones, como casos particulares de GEFRED. Por tanto, este modelo se puede emplear para representarlos.

Bibliografía

- [Anvari, Rose, 87] Anvari M., Rose G.F. "Fuzzy Relational Databases". Analysis of Fuzzy Information. Bezdek ed. Vol II CRC Press. (1987)

- [Bosc, Galibourg, 88] Bosc P., Galibourg M., Hamon G. "Fuzzy Querying with SQL: Extensions and Implementation Aspects". *Fuzzy Sets and Systems*. v.28 pp. 333-349. (1988)
- [Buckles, Petry, 82] Buckles B.P., Petry F.E. "A Fuzzy Representation of Data for Relational Databases". *Fuzzy Sets and Systems*, 7. 213-226. (1982)
- [Buckles, Petry, 84] Buckles B.P., Petry F.E. "Extending the Fuzzy Database with Fuzzy Numbers". *Information Sciences*, 34. 145-155. (1984)
- [Buckles, Petry, Sachar, 89] Buckles B.P., Petry F.E., Sachar H.S. "A Domain Calculus for Fuzzy Relational Databases". *Fuzzy Sets and Systems*, 29. 327-340. (1989)
- [Codd, 70] Codd E.F. "A Relational Model of Data for Large Shared Data Banks". *Commun. ACM*, 13. (6). pp. 377-387 (1970)
- [Kacprzyk, 86] Kacprzyk J., Ziolkowski A. "Database Queries with Fuzzy Linguistic Quantifiers". *IEEE Transactions on Syst., Man, and Cyb.* Vol SMC-16 n°3.(1986)
- [Medina et al, 91(a)] Medina J. M., Vila M.A., "Un Modelo de Bases de Datos Difusa Aplicado a Información Médica". 1^{er} Congreso Español sobre Tecnologías y Lógica Fuzzy. Granada. (1991)
- [Medina et al, 91(b)] Medina J. M., Vila M.A., "A Knowledge Representation Model for Fuzzy Databases". *Sometido a ACM TODS*.(1991)
- [Prade, Testemale, 84] Prade H., Testemale C. "Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries". *Information Sciences*, 34. 115-143. (1984)
- [Prade, Testemale, 87] Prade H., Testemale C. "Representation of Soft Constraints and Fuzzy Attribute Values by means of Possibility Distributions in Databases". Bezdek ed. *Analysis of Fuzzy Information*. Vol II CRC Press. (1987)
- [Raju, Majumdar, 87] Raju K., Majumdar A. "The Study of Joins in Fuzzy Relational Databases". *Fuzzy Sets and Systems* 21.pp19 – 34.(1987)
- [Rundensteiner, Hawkes, Bandler, 89] Rundensteiner E.A., Hawkes L. W., Bandler W. "On Nearness Measures in Fuzzy Relational Data Models". *International Journal of Approximate Reasoning*. 3.pp267 – 298.(1989)
- [Shenoi, Melton, 89] Shenoi S., Melton A. "Proximity Relations in the Fuzzy Relational Database Model". *Fuzzy Sets and System*, 31. 285-296. (1989)
- [Shenoi, Melton, 90] Shenoi S., Melton A. "An Extended Version of the Fuzzy Relational Database Model". *Information Sciences*, 52. 35-52. (1990)
- [Umano, 82] Umano M. "Freedom-0 : A Fuzzy Database System". *Fuzzy Information and Decision Processes*. Gupta-Sanchez edit. North-Holland Pub. Comp. (1982)
- [Vila et al. 1992] Vila M.A. et al. "A Logic Approach to Fuzzy Relational Databases", *IPMU'92* (1992).
- [Vila Cátedra, 92] Vila M.A., "Las Bases de Datos Relacionales y su Tratamiento Lógico". *Proyecto de Cátedra*. (1992).
- [Zadeh, 71] Zadeh L.A. "Similarity Relations and Fuzzy Orderings". *Information Sciences*. vol. 3 177-200. (1971)
- [Zadeh, 78] Zadeh L.A. "Fuzzy Sets as a Basis for a Theory of Possibility". *Fuzzy Sets and Systems*. 1, 3-28. (1978)
- [Zemankova, Kandel, 84] Zemankova M., Kandel A. "Fuzzy Relational Data Bases - A Key to Expert Systems". *Verlag TUV Rheinland*. (1984)

MEDIDAS DE BONDAD PARA SISTEMAS DE CONTROL BORROSO

Autores F. Matía, A. Jiménez
R. Sanz, R. Galán.
Departamento de Automática (DISAM)
Universidad Politécnica de Madrid
Madrid (España)

Resumen: Este trabajo muestra una serie de medidas útiles en el análisis de la configuración de un sistema de control borroso, que sirven al mismo tiempo para la detección de errores de configuración. Se ha pretendido que las medidas sean lo más elementales posibles, de forma que puedan ser fácilmente utilizables y redefinidas según las propias necesidades. En muchos casos se han obtenido por extensión de medidas ya existentes en la teoría de conjuntos borrosos.

Palabras clave: Lógica Borrosa, Sistemas de Control Borroso, Análisis de Sistemas no Lineales, Diseño de Sistemas de Control, Medidas de Bondad.

1. Introducción

Los sistemas de control borroso (SCB) se han venido diseñando tradicionalmente en base a reglas de control expresadas en términos lingüísticos. Las operaciones entre los conjuntos borrosos que lo definen, utilizan gran parte del aparato matemático de la lógica borrosa.

Sin embargo, es conocido el problema de la configuración y ajuste de este tipo de sistemas, pese a la facilidad con que teóricamente debería quedar modelado el conocimiento del experto en base a reglas. Ello se debe, principalmente, a que la obtención independiente de cada una de ellas, puede provocar la aparición de incoherencias, falta de completitud o inconsistencia entre las acciones de control propuestas.

Por este motivo, se presenta una filosofía diferente de enfocar la configuración de un SCB. El sistema de control es contemplado como una función de control, determinada unívocamente por su mapa de inferencia. La elección de las etiquetas lingüísticas, las funciones de pertenencia, junto con las reglas de control, condicionan las características de dicho mapa. Por ello es posible predecir su apariencia en base a estos factores y utilizar su información provechosamente.

Además, contrastar la no-linealidad de un SCB con el regulador lineal (PID) equivalente, permite al ingeniero de control analizar las irregularidades derivadas de una incorrecta configuración. Entre las técnicas utilizadas para valorar este contraste, se emplean medidas de bondad para la detección de singularidades y para la cuantificación de las asimetrías, de la dispersión de las reglas y de la entropía o *borrosidad* del regulador.

2. Análisis Descriptivo de un SCB

2.1. Conceptos Básicos

Consideremos un SCB definido en base a reglas del tipo:

SI (E es ..) Y (DE es ..) Y (SE es ..) ENTONCES (U es ..)

siendo E el error, DE la derivada del error, SE la suma o integral del error y U la acción de control. A este tipo de controlador lo llamaremos Controlador Fuzzy PID (FPID), por analogía con las entradas/salidas de un controlador clásico PID. En este trabajo analizaremos sólo, por sencillez de cálculo y representación gráfica, controladores de tipo FP con reglas de la forma

SI (DE es ..) ENTONCES (DU es ..)

siendo DU la derivada de la acción de control. La definición de un SCB pasa por establecer los conjuntos borrosos asociados a DE (DE_i) y a DU (DU_i), y las N reglas de control. Además, por simplicidad, y dado que no existe encadenamiento de reglas, los conjuntos borrosos asociados a la salida pueden definirse por su centro de gravedad c_i , puntos discretos sobre el universo DU (Figura 1). Las reglas quedan entonces de la forma

SI (DE es DE_i) ENTONCES ($du = c_i$)

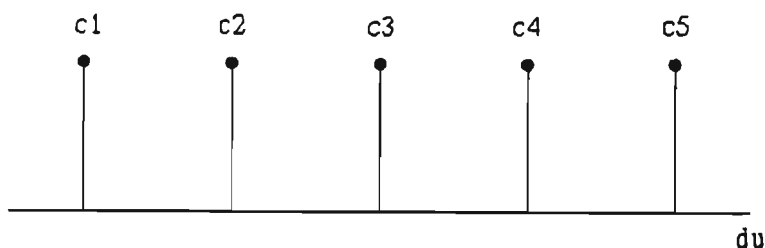


Figura 1. Acciones de Control

Llamaremos *función de información* contenida en los conjuntos borrosos DE_i a la función suma de todas las funciones de pertenencia para cada valor de de :

$$I(de) = \sum_{i=1}^N \mu_i(de)$$

Diremos entonces que el sistema de control está normalizado, o que es *normal*, si los conjuntos borrosos μ_i definidos sobre el universo DE son normales y la cantidad de información es constante e igual a la unidad, como se muestra en la Figura 2 para funciones de pertenencia triangulares. Esto también es posible con funciones de pertenencia polinómicas.

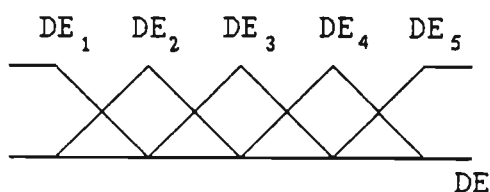


Figura 2. Conjuntos Borrosos

La salida del controlador se obtiene calculando el centro de gravedad:

$$du(de) = \frac{\sum_{i=1}^N c_i \cdot \mu_i(de)}{\sum_{i=1}^N \mu_i(de)}$$

2.2 Puntos Maestros

A partir de aquí es posible llevar a cabo una configuración inicial del controlador mediante el establecimiento de sus puntos maestros y el ajuste paramétrico de los conjuntos borrosos. Estos puntos maestros vienen determinados por la posición relativa de las funciones de pertenencia asociadas a las variables de entrada y las conclusiones de las reglas de control. Su determinación es unívoca si los conjuntos borrosos son *normales* y cumplen los requisitos adicionales especificados anteriormente. Por otra parte, la forma de las funciones de pertenencia es fácilmente ajustable en función de un parámetro de diseño que determina su grado de linealidad.

En primer lugar consideraremos los puntos β_i del universo DE que coinciden con el valor máximo de cada conjunto borroso (en el caso de funciones triangulares). Para estos puntos se verifica que la única regla que se dispara es la regla i , por lo que la salida tomará el valor c_i y la función de control resultante $du(de)$ pasará exactamente por el punto (β_i, c_i) . A estos puntos los llamaremos *puntos maestros* o puntos guía.

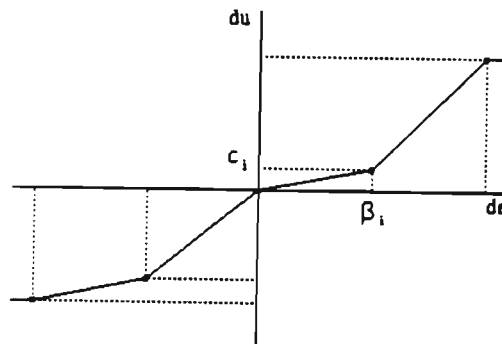


Figura 3. Puntos Maestros

2.3 Contraste

Entre cada dos puntos maestros, la forma de la función de control depende de la forma de los conjuntos borrosos DE_i . Si definimos las funciones de pertenencia mediante la función reflejada en la Tabla I, con λ positiva, entonces se verifican las siguientes correspondencias:

- para $\lambda = 0$, las funciones de pertenencia son triangulares, y la función de control resultante es lineal por tramos
- para $\lambda = 1$, las funciones de pertenencia son polinómicas de segundo grado, y la función de control resultante es polinómica por tramos, con derivada nula en su paso por cada punto maestro

- para $0 < \lambda < 1$, las funciones de pertenencia tienen una forma intermedia a las anteriores y lo mismo se refleja en la función de control
- para $\lambda > 1$, las funciones de pertenencia tienden a ser rectangulares (conjuntos clásicos) a medida que $\lambda \rightarrow \infty$ son triangulares, mientras que la función de control tiende a comportarse como una función relé

	$\gamma - \beta < de < \gamma - \beta/2$	$\gamma - \beta/2 < de < \gamma$	$\gamma < de < \gamma + \beta/2$	$\gamma + \beta/2 < de < \gamma + \beta$
$\lambda > 0$ General	$\frac{1}{2} \left[2 \left(1 + \frac{de - \gamma}{\beta} \right) \right]^{\lambda+1} \frac{1}{2} \left[\frac{2(de - \gamma)}{\beta} \right]^{\lambda+1} 1 - \frac{1}{2} \left[2 \frac{ de - \gamma }{\beta} \right]^{\lambda+1} \frac{1}{2} \left[2 \left(1 - \frac{de - \gamma}{\beta} \right) \right]^{\lambda+1}$			
$\lambda = 0$ Triangular	$1 + \frac{de - \gamma}{\beta}$	$1 + \frac{de - \gamma}{\beta}$	$1 - \frac{de - \gamma}{\beta}$	$1 - \frac{de - \gamma}{\beta}$
$\lambda = 1$ Polinómica	$2 \left(1 + \frac{de - \gamma}{\beta} \right)^2$	$1 + 2 \left(\frac{de - \gamma}{\beta} \right)^2$	$1 - 2 \left(\frac{de - \gamma}{\beta} \right)^2$	$2 \left(1 - \frac{de - \gamma}{\beta} \right)^2$

Tabla I. Función de Contraste

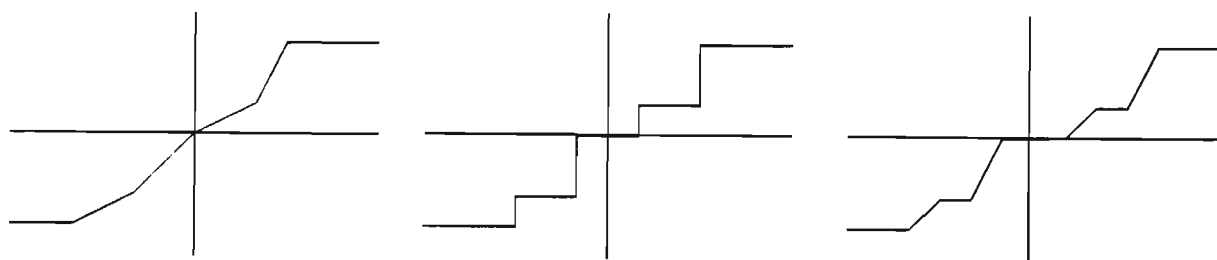


Figura 4. El Contraste en SCB

A la función $\lambda(de)$ la llamaremos *función de contraste*.

Sin embargo, en un SCB no normal, en que la suma de funciones de pertenencia sea distinta de la unidad provoca que, tras calcular el centro de gravedad, la función de control no pase por los puntos maestros (ver Figura 5).

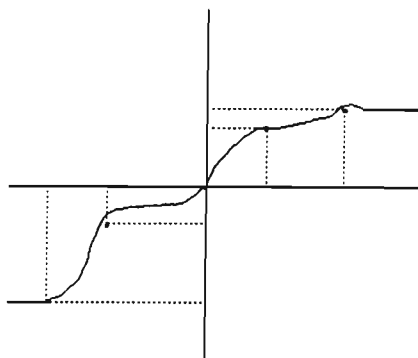


Figura 5. Curva de un SCB no Normal

2.4 Obtención del Conocimiento

Dependiendo de la fuente de información de que disponemos para definir el sistema de control (básicamente los conjuntos borrosos y las reglas), podemos clasificar los SCB en estructurados y no estructurados.

En los primeros, las funciones de pertenencia corresponden a uno de los siguientes tipos: triangulares, polinómicas, trapezoidales o rectangulares, con lo que se puede utilizar el contraste definido anteriormente. En el caso, bastante habitual, de que el experto no nos suministre directamente los conjuntos borrosos, sino que lo haga indirectamente definiendo las reglas de control sobre puntos o sobre regiones de funcionamiento, las funciones de pertenencia asociadas se pueden obtener fácilmente a partir de los mismos:

- a partir de puntos de funcionamiento se obtienen funciones triangulares,
- a partir de regiones, si éstas no son contiguas, se obtienen funciones trapezoidales, mientras que si lo son, lo que se obtienen son funciones rectangulares.

El caso de SCB no estructurados se presenta cuando el experto nos suministra todas las reglas de control, incluyendo la definición de los conjuntos borrosos, pero esta última es de forma caótica (no estructurada), como se muestra en la Figura 6. En este caso, la mejor solución para analizar la definición del SCB pasa por transformar, en la medida de lo posible, estas funciones de pertenencia en otras más estructuradas (combinación de triangulares y trapezoidales) mediante aproximación. Con esto ya hemos conseguido un SCB estructurado que podemos analizar mediante las técnicas que se describirán en el apartado 3.

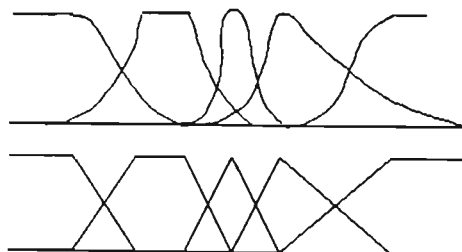


Figura 6. Aproximación de Funciones de Pertenencia Caóticas

2.5 Problemas de Definición

Resulta sencillo pensar en numerosos casos en los que la función de control resultante no es tan estructurada como las vistas hasta el momento. Por ejemplo, si la cantidad de información $I(\text{de})$ es nula en algún tramo entre dos puntos maestros, aparece una zona en la que la función de control no está definida (singularidad). O si existe una zona en la que sólo hay definida una regla (o conjunto borroso) la función de control es constante en esa zona.

Siguiendo esta idea, como derivación de un controlador normal podemos definir *controlador débil* como aquel para el que la cantidad de información es inferior a la unidad en alguna zona. Y, análogamente, *controlador fuerte* como aquel en el que la cantidad de información supera en alguna zona a la unidad. En ambos casos es bastante frecuente que la función de control no pase por los puntos maestros.

2.6 Soluciones

Como solución a los problemas planteados, parece interesante la elaboración de índices o medidas que nos valoren aspectos como los ya vistos de la cantidad de información, y otros como la no-linealidad, asimetrías, dispersión, consistencia, etc, de cara a detectar irregularidades en la configuración o incoherencias entre las reglas y que consideraremos en los siguientes apartados.

Por ejemplo, si tenemos un controlador débil, podemos determinar en qué zonas tenemos una menor información, con el fin de pedir al experto que vuelque su conocimiento en esa zona. En el caso de un controlador fuerte, el exceso de información es siempre bueno, pues en cualquier caso ayudará a camuflar entre todas las reglas colindantes posibles errores en las mismas.

3. Medidas para SCB

3.1 Enfoque lingüístico

Atendiendo al enfoque lingüístico del controlador, es posible analizar aspectos como la interacción entre reglas, la aparición de reglas inconsistentes, situaciones para las que no hay propuesta ninguna acción de control y que puedan dar lugar a singularidades en la función de control, etc.

Definimos la *completitud* de un SCB como $C(\text{SCB}) = \min I(\text{de}) \quad \forall \text{de}$. Por tanto, la completitud será mayor que cero si existe solapamiento entre los conjuntos borrosos que lo definen, es decir, se verifica según que

$$\forall \text{de}, \exists i \in \{1, \dots, N\} \mid \mu_i(\text{ce}) > 0$$

Por el contrario, un SCB será incompleto cuando tenga completitud nula, lo que se puede producir si hay insuficientes reglas.

Decimos que un SCB es *coherente* si sólo existe una regla para cada conjunto borroso asociado a la variable de entrada, es decir

$$\forall \text{DE}_i \exists! \text{ regla } R_i: \text{DE} \rightarrow \text{DU} \mid c_i = R_i(\text{DE}_i)$$

Las incoherencias aparecerán en el momento en que exista más de una acción de control propuesta para una misma situación.

Se define la *consistencia* entre dos reglas i y j como

$$C_{ij} = \sup \min(\mu_i(de), \mu_j(de)), \text{ para todo } de, \text{ con } C_{ij} = C_{ji}$$

Decimos que no existe interacción entre la regla i y las restantes, si la función de control pasa por el punto maestro (β_i, c_i) . En caso contrario, se puede obtener una medida de la interacción como

$$\text{Interacción(SCB)} = \frac{1}{N} \cdot \sum_{i=1}^N |du(\beta_i) - c_i| = \frac{1}{N} \cdot \sum_{i=1}^N \left| \frac{\sum_{j=1}^N c_j \cdot \mu_j(\beta_i)}{\sum_{j=1}^N \mu_j(\beta_i)} - c_i \right|$$

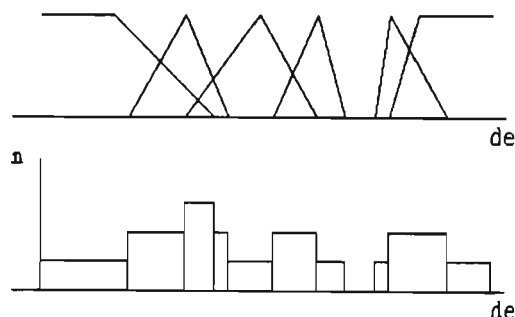


Figura 7. Densidad de Reglas

Definimos la densidad de reglas como

$$d = \frac{\int_{DE} n(de) dde}{\text{card}(DE)}$$

donde $n(de)$ es el número de reglas para cada valor de de , y $\text{card}(DE)$ es la amplitud del universo de discurso.

Definimos la dispersión de reglas como

$$D = \frac{1 - d}{1 + d}$$

y la concentración de reglas de manera análoga

$$C = -D = \frac{d - 1}{d + 1}$$

En el caso extremo de mínima densidad, la dispersión es máxima y la concentración mínima.

Como vimos anteriormente, la cantidad de información de un SCB puede interpretarse como una medida de la fortaleza del sistema de control: un controlador con poca información es un controlador débil. Definimos entonces la fiabilidad de un SCB como

$$I_{\min}^* = \max \{ 1, I_{\min} \} = \max \{ 1, \inf I(de) \quad \forall de \in DE \}$$

Partiendo de estas medidas, podemos construir un diagrama que, para cada ce , represente la evolución de I en función de D . La Figura 8 muestra un ejemplo de la evolución de $I(D)$ a medida vamos quitando reglas al sistema de control. Para ello, elegimos el criterio optimista, consistente en eliminar en cada paso aquella regla que consigue mantener una mayor $I_{\min}$. Llamaremos al diagrama así construido *diagramas inverso*. Definimos entonces la *robustez* del controlador como el área bajo la función de fiabilidad

$$R(SCB) = \frac{\int_{D_0}^1 I_{\min}^*(D) dD}{1 - D_0}$$

siendo D_0 la dispersión de las reglas del controlador inicial. Podemos también establecer una medida de la *velocidad de pérdida de información* como

$$V = - \frac{dI_{\min}}{dD}$$

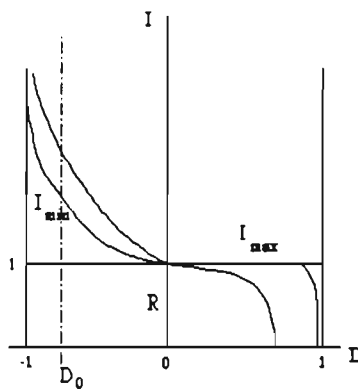


Figura 8. Información frente a Dispersión (Diagrama Inverso)

El *diagrama directo* puede obtenerse iniciando el proceso sin ninguna regla y añadiendo una a una sucesivamente.

Llamamos *función de traslación* $TRAS(DE_i; T \cdot \delta)$ a aquella que traslada un conjunto borroso DE_i una fracción de T (número real) sobre su universo de discurso. Esta fracción a desplazar viene definida en cada punto por el valor que toma una función de pertenencia δ (ver Figura 9):

$$\text{dado } DE_i = \sum \mu_{DE_i}(\beta_i) / \beta_i$$

$$\text{entonces } TRAS(DE_i; T \cdot \delta) = \sum \mu_{DE_i}(\beta_i) / (\beta_i + T \cdot \delta(\beta_i))$$

El interés de estas funciones consiste en aplicar la misma traslación a todos y cada uno de los conjuntos borrosos que definen el SCB, con lo cual lo que se obtiene indirectamente es un desplazamiento de la curva de control. Si definimos un controlador borroso como

$$SCB = \sum du_i / \beta_i$$

entonces podemos definir la *traslación inferior* como:

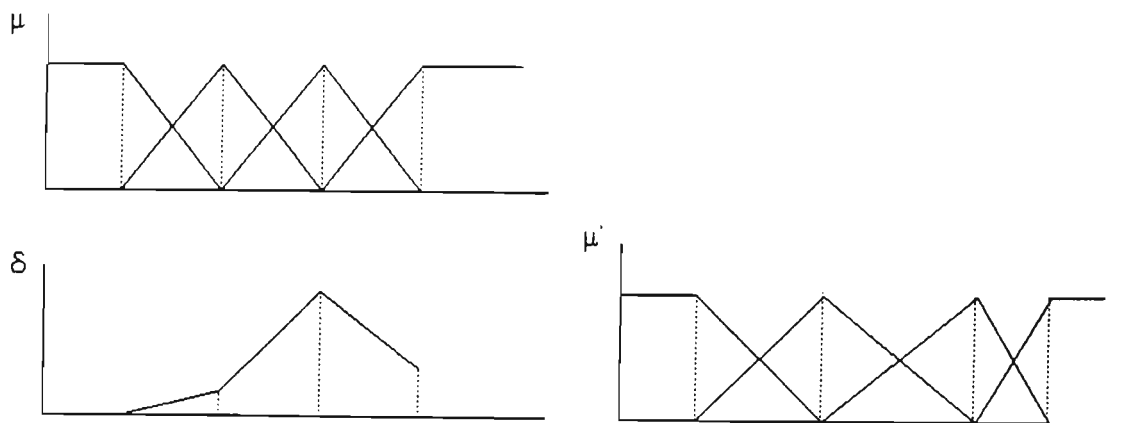


Figura 9. Función de Traslación

$$TI(SCB; T \cdot \delta) = \sum du_i / (\beta_i + T \cdot \delta(\beta_i))$$

y análogamente la *traslación superior* pero aplicada a los conjuntos borrosos de salida:

$$TS(SCB; T \cdot \delta) = \sum (du_i + T \cdot \delta(du_i)) / \beta_i$$

El efecto sobre la función de control de aplicar una traslación inferior o una superior aparece reflejado en la Figura 10. En la primera ésta se desplaza horizontalmente a DU constante, y en la segunda se desplaza verticalmente a DE constante.

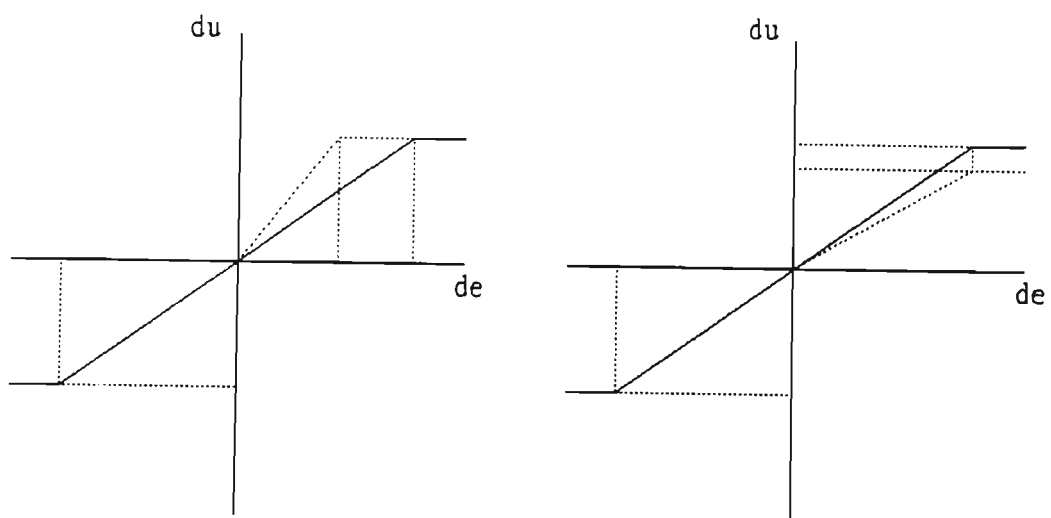


Figura 10. Traslación Inferior y Traslación Superior

Un análisis interesante consiste en comprobar cómo es posible expresar un SCB

cuya función de control sea lineal por tramos, como combinación de traslaciones aplicadas a un SCL. En el caso más genérico de que el SCB no esté definido mediante funciones de pertenencia triangulares, es decir, que tenga un contraste distinto de la unidad, es posible emplear una función que intensifique o reduzca el contraste en distintas zonas para eliminar la falta de linealidad por tramos.

Si definimos el *contraste* de un SCB como

$$\text{CONT}(\text{SCB}) = \sum \lambda(b_i) / b_i$$

donde b_i representa cada uno de los tramos del universo de discurso con contraste constante, para un sistema lineal se verifica que

$$\text{CONT}(\text{SCL}) = \sum 0 / b_i$$

Definimos entonces la *intensificación de contraste* (Figura 11) como

$$\text{IC}(\text{CONT}(\text{SCB}); \epsilon) = \sum \text{INT}(\lambda(b_i); \epsilon) / b_i$$

$$\text{con } \text{INT}(\lambda; \epsilon) = \begin{cases} (1+\epsilon) \cdot \lambda, & \text{si } \epsilon \leq 0 \\ \frac{1}{1-\epsilon} \cdot \lambda, & \text{si } \epsilon \geq 0 \end{cases}$$

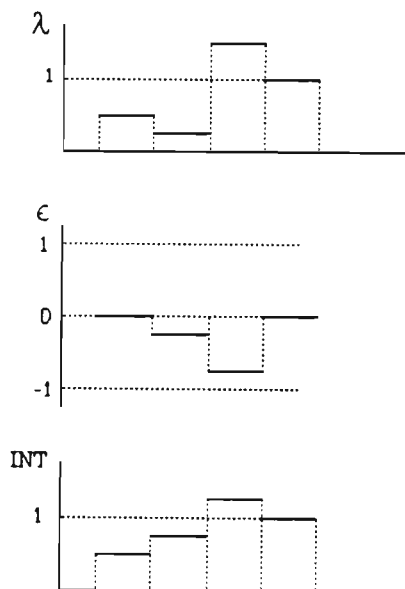


Figura 11. Intensificación de Contraste

Puede observarse que:

- para obtener λ infinito, hay que intensificar el contraste con $\epsilon=1$
- para obtener λ nulo, hay que reducir el contraste con $\epsilon=-1$

Por tanto, es factible descomponer un SCB en combinaciones sucesivas de traslaciones e intensificaciones de contraste.

3.2. El Enfoque Funcional

Considerando el SCB como una función $du(de)$, podemos muestrear el universo de entrada DE cada $(\beta_i - \beta_{i-1})/k$ unidades, obteniendo $k-1$ puntos (de_i, du_i) entre cada dos puntos maestros. El número total de puntos, por tanto, será $k \cdot (N-1) + 1$.

El primer análisis rápido que se puede llevar a cabo es el estadístico, obteniendo medidas como por ejemplo:

- valor medio
- mediana
- varianza (medida de la dispersión)
- MEDA
- coeficiente señal-ruido
- rango
- coeficiente de asimetría
- coeficiente de apuntamiento
- covarianza
- índice de correlación (medida de la linealidad)
- matriz de varianzas y covarianzas
- varianza generalizada

A continuación se puede analizar la no-linealidad de un SCB. Para ello necesitamos obtener el sistema de control lineal (SCL) más próximo, calculando por mínimos cuadrados la recta de regresión $(de_i, d\hat{u}_i)$ que pasa por el punto de equilibrio. De la comparación de ambos controladores (SCB y SCL) podemos llevar a cabo un análisis de los residuos: mediante un contraste de linealidad podemos obtener una medida de la linealidad entre 0 y 1. Llamando d_i al valor absoluto de los residuos, podemos examinar el valor de d_{media} , d_{min} (habitualmente 0) y d_{max} (con este valor podemos detectar puntos singulares cuando el valor en cuestión se desvía mucho del rango previsto de du). Otras medidas que pueden calcularse asociadas a los residuos son:

- valor medio
- valor absoluto medio
- desviación cuadrática media
- residuos negativos
- residuos positivos
- desviación normalizada

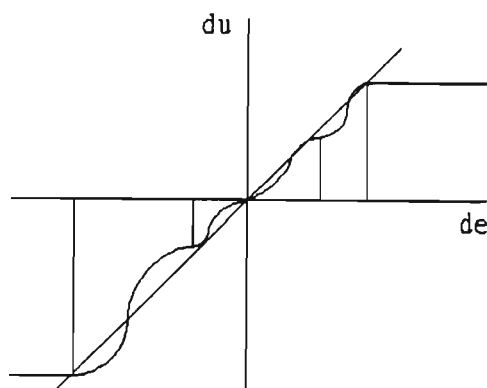


Figura 12. Recta de Regresión

Definimos la *nitidez* de un SCB como

$$Nit(SCB) = \frac{1}{k(N-1)+1} \cdot \sum_{i=1}^{k(N-1)+1} \frac{1}{|du_i - d\hat{u}_i|}$$

Una función de *similitud* entre dos sistemas de control (SC) puede definirse como una función $X_s: CS \times CS \rightarrow [0, 1]$ con las siguientes propiedades

$$\begin{aligned} X_s(CS_1, CS_1) &= 1 && \text{(reflexiva)} \\ X_s(CS_1, CS_2) &= X_s(CS_2, CS_1) && \text{(simétrica)} \\ X_s(CS_1, CS_3) &\geq \max_{\forall CS_2} \min (X_s(CS_1, CS_2), X_s(CS_2, CS_3)) && \text{(transitiva)} \end{aligned}$$

La *borrosidad* surge de la falta de similitud entre un SCB y el SCL más próximo, por lo que podemos definir una medida de la vaguedad de forma que el SCB sea menos vago cuando ésta esté más próxima a la unidad. Esta medida puede ser representada como una función de disimilitud entre un SCB y el SCL, que verifica $X_d(SCB, SCL) = 1 - X_s(SCB, SCL)$. Por ejemplo

$$X_d(SCB, SCL) = \sum_{i=1}^{k(N-1)+1} |du_i - c_i|$$

donde c_i representa ahora no sólo los puntos maestros, sino también la función lineal por tramos que pasa por ellos. desde otro punto de vista, la desviación total entre la función de control du y la función lineal más próxima puede ser descompuesta en dos partes: $(du_i - d\hat{u}_i) = (du_i - c_i) + (c_i - d\hat{u}_i)$ o bien $DT = DB + DL$

- $DB = (du_i - c_i)$ representa la borrosidad debida a la interacción entre reglas
- $DL = (c_i - d\hat{u}_i)$ representa la borrosidad debida a la no linealidad de las reglas de control

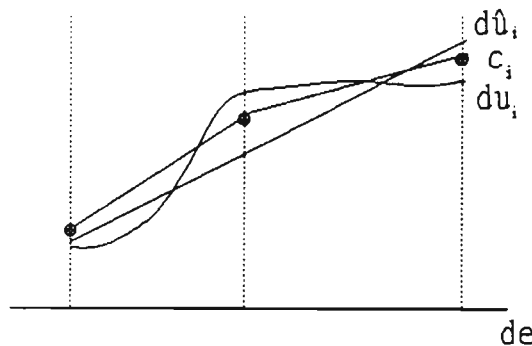


Figura 13. Descomposición de la desviación respecto al PID equivalente

Por tanto, la cantidad total de borrosidad entre el SCB y el SCL más próximo es la suma de ambas. Definimos la *energía normalizada* de un SCB como

$$E(SCB) = \sum_{i=1}^{k(N-1)+1} E_i, \quad \text{con } E_i = \frac{\int_{-\infty}^{du_i} \mu_{dii}(du) ddu}{\int_{-\infty}^{\infty} \mu_{dii}(du) ddu}$$

y la energía complementaria como

$$E(SCB) = \sum_{i=1}^{k(N-1)+1} E_i, \quad \text{con } E_i = \frac{\int_{-\infty}^{\overline{du}_i} \mu_{d\bar{u}_i}(du) ddu}{\int_{-\infty}^{\infty} \mu_{d\bar{u}_i}(du) ddu}$$

donde $\overline{du}_i = 2d\hat{u}_i - du_i$ es la simétrica de du_i y $\mu_{d\bar{u}_i}$ se define como aparece en la Figura 14 y de manera simétrica, por ejemplo

$$\mu_{d\bar{u}_i}(du) = e^{-\frac{|du_i - d\bar{u}_i|}{\gamma}}$$

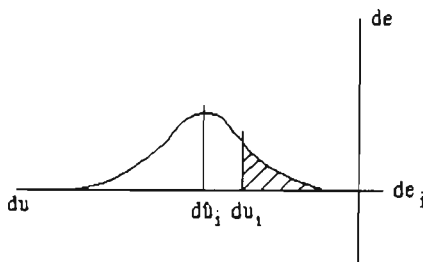


Figura 14. Estimación de la Energía de un SCB

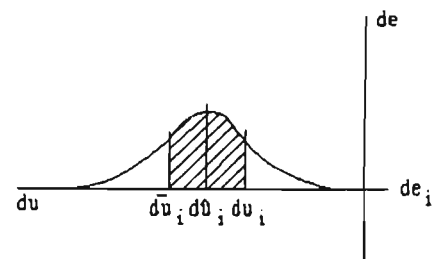


Figura 15. Estimación de la Entropía de un SCB

Adviértase que cuanto más próximo esté el SCB al SCL, E está más próxima de 0.5. En otro caso, E se aproxima a 0 o a 1. A partir de estas ideas definimos la entropía de un SCB como

$$S(SCB) = \sum_{i=1}^{k(N-1)+1} S_i, \quad \text{with } S_i = |E_i - \overline{E}_i| = \frac{\left| \int_{du_i}^{\overline{du}_i} \mu_{d\bar{u}_i}(cu) ddu \right|}{\int_{-\infty}^{\infty} \mu_{d\bar{u}_i}(du) ddu}$$

Cuanto más cerca está el SCB del SCL, más próxima está S de cero. En otro caso, S está próximo a 1. Con estas medidas podemos comparar la borrosidad y la entropía de dos o más controladores borrosos.

Por último comentar que, aunque no se ha hecho en todos los casos vistos, la mayoría de las medidas presentadas se prestan a ser normalizadas.

4. Conclusión

El análisis de diversas medidas aplicadas a controladores borrosos puede ser interpretado como una herramienta para certificar el correcto diseño de un SCB. Pero la borrosidad o entropía no debe ser considerada como una medida de calidad. Su necesidad surge de que en ocasiones es necesario pensar sobre si un controlador borroso es realmente borroso, pues en determinadas situaciones un controlador clásico puede llegar a ser mejor que uno borroso, y podemos vernos forzados a emplear un SCB lineal. En cualquier caso, existen dos formas diferentes de obtener provecho de estas medidas:

- en la detección de errores de configuración
- en la mejora de las reglas de control

También se han apuntado otras ideas. Por ejemplo, no siempre es necesario eliminar las reglas superfluas, como reglas que son combinación lineal de otras, excepto en los casos en que se pueden detectar importantes incoherencias, pues suele resultar interesante integrar información procedente de diferentes fuentes. También pueden darse situaciones en las que la regresión lineal no es la mejor aproximación de cara a analizar desviaciones del sistema, y se requiere una función lineal por tramos. Estos ejemplos sugieren que pueden desarrollarse mejoras interesantes futuros trabajos.

Referencias

- [1] I.Z.Batyrshin, *On Fuzziness Measures of Entropy on Kleene Algebras*, Fuzzy Sets and Systems 34 (1) (1990) 47-60.
- [2] R.E.Bellman, L.A.Zadeh, *Decision-Making in a Fuzzy Environment*, Management Sci. 17, (1970) B-141-B-164.
- [3] J.J.Buckley, H.Ying, *Fuzzy Controller Theory: Limit Theorems for Linear Fuzzy Control Rules*, Automática 25 (3) (1989) 469-472.
- [4] A.De Luca and S.Termini, *Entropy and Energy Measures of a Fuzzy Set*. In *Advances in Fuzzy Sets Theory and their Applications*, M.M.Gupta, R.K.Ragade and R.R.Yager, North-Holland, Amsterdam (1979).
- [5] M.C.García Alegre, *Inteligencia Artificial en el Control de Procesos: Controladores Borrosos*, Mundo Electrónico 214 (1991).
- [6] A.Jiménez, F.Matía, *Aplicaciones de Fuzzy Logic en Tiempo Real*, en el curso *La Inteligencia Artificial y el Control en Tiempo Real*, Universidad Internacional Menéndez Pelayo, Santander (1991).
- [7] A.Kandel, *Fuzzy Techniques in Pattern Recognition*, Wiley-Interscience. John Wiley & Sons (1982).
- [8] P.J.King, E.H.Mamdani, *The Application of Fuzzy Control Systems to Industrial Processes*, Automática 13 (1977) 235-242.
- [9] Y.F.Li, C.C.Lau, *Development of Fuzzy Algorithms for Servo Systems*, IEEE International Conference on Robotics and Automation, (Pennsylvania, 1989).
- [10] F.Matía, A.Jiménez, R.Galán, R.Sanz, *Fuzzy Controllers: Lifting the Linear - Nonlinear Frontier*, Fuzzy Sets and Systems 53 (1) (1993). En imprenta.
- [11] W.Pedrycz, *Fuzzy Control and Fuzzy Systems*, Research studies press ltd. (1989).
- [12] D.Peña, *Estadística, Modelos y Métodos* (Alianza Universidad Textos, Madrid, 1987).
- [13] T.J.Procyk, E.H.Mamdani, *A Linguistic Self-Organizing Process Controller*, Automática 15 (1978).
- [14] A.Sgarro, *Fuzziness Measures for Fuzzy Rectangles*, Fuzzy Sets and Systems 34 (1) (1990) 39-45.
- [15] W.Siler, H.Ying, *Fuzzy Control Theory: the Linear Case*, Fuzzy Sets and Systems 33 (1988) 275-290.
- [16] M.Sugeno, *An Introductory Survey of Fuzzy Control*, Information Sciences 36 (1985) 59-83.
- [17] E.Trillas and T.Riera, *Entropies in Finite Fuzzy Set*, Inf. Sci. 15 (1978) 159-168.
- [18] W.X.Xie and D.Bedrosian, *An Information Measure for Fuzzy Sets*. IEEE Trans. Syst. Man, Cybern. SMC-14 (1) (1984) 151-156.
- [19] H.Ying, W.Siler and J.J.Buckley, *Fuzzy Control Theory: A Nonlinear Case*, Automática 26 (3) (1990) 513-520.
- [20] L.A.Zadeh, *Similarity Relations and Fuzzy Orderings*, Information Sciences 3 (1971) 177-200.

REALIZACIÓN DE UN CONTROLADOR FUZZY SOBRE UN PROCESADOR DIGITAL DE SEÑAL

I. del Campo, J.R. González de Mendivil, J.R. Garitagoitia y J.M. Tarela
Dpto. de Electricidad y Electrónica, Fac. de Ciencias
Universidad del País Vasco UPV/EHU
Lejona (Vizcaya)

Resumen: En este trabajo se presenta la realización de un Controlador Fuzzy sobre un Procesador Digital de Señal (DSP). La arquitectura interna de los DSPs, orientada al tratamiento intensivo de datos a alta velocidad, y un conjunto de instrucciones adecuado, los hacen más eficientes en la realización de los algoritmos fuzzy que un microprocesador de propósito general. Los aspectos más destacables de esta realización son a) flexibilidad, b) velocidad de respuesta (85.000 FIPs para inferencias máx-prod), y c) bajo costo.

Palabras clave: control fuzzy, hardware, DSPs, herramientas CASE.

1. Introducción.

El gran número y diversidad de aplicaciones industriales de la lógica fuzzy ha propiciado la realización tanto de entornos de tipo CASE para el desarrollo de sistemas fuzzy, como de realizaciones hardware específicas para el procesamiento de inferencias fuzzy [Sel90] [Sch92]. En las aplicaciones de control, la selección adecuada del hardware es fundamental para el éxito de la aplicación.

Los aspectos más importantes a tener en cuenta en la selección del hardware son: a) velocidad de respuesta, b) dimensiones, c) costo de diseño y fabricación, y d) flexibilidad con respecto a la selección de los parámetros de diseño del controlador.

Existen distintas alternativas para la realización de controladores fuzzy que abarcan desde una realización software en lenguaje de alto nivel para un ordenador de propósito general, hasta un hardware de alta velocidad de tipo ASIC (Circuito Integrado de Aplicación Específica), o una solución intermedia basada en un microprocesador. La primera opción es la más flexible, ya que puede ser utilizada en distintas aplicaciones cambiando únicamente el programa, sin embargo, para la mayoría de las aplicaciones al control su relación costo/velocidad es demasiado alta. En el extremo opuesto se encuentran los ASICs. Actualmente existen en el mercado distintas familias de ASICs de inferencias fuzzy, cuyas características son a) baja relación costo/velocidad, b) pequeñas dimensiones y c) arquitecturas fijas limitadas a los algoritmos fuzzy estándar [TOG91] [OMR92]. En gran número de aplicaciones industriales, una solución intermedia basada en un microprocesador, es la mejor alternativa. La realización que proponemos, basada en un Procesador Digital de Señal (DSP), está dentro de esta última línea.

El trabajo que presentamos está organizado como sigue. En la sección 2 se realiza una descripción de las características más destacables de los DSPs, en particular se presenta el DSP56001 de Motorola, utilizado en esta aplicación. La sección 3 describe la realización del controlador fuzzy sobre un DSP, se describe la organización de memorias y la realización de los algoritmos fuzzy, poniendo énfasis en la adecuación de los recursos del DSP para la realización de dichos algoritmos. En la sección 4 se describen distintas configuraciones de sistemas fuzzy basadas en DSPs. En la sección 5 se sitúa la realización dentro de un entorno integrado de tipo CASE para el desarrollo de sistemas fuzzy. Finalmente, se incluye una sección dedicada a las conclusiones del trabajo.

2. El Procesador Digital de Señal (DSP).

Un Procesador Digital de Señal (DSP) es un microprocesador de propósito especial diseñado originalmente para el procesamiento digital de señales en tiempo real. En la actualidad existe ya un gran número de aplicaciones basadas en DSPs, principalmente en áreas como telecomunicaciones, procesamiento de imagen y sonido, control de alta velocidad e instrumentación [MOT86]. Recientemente se han utilizado DSPs en otras áreas, como la realización de redes neuronales [Alo91].

La arquitectura interna del DSP está especialmente orientada al tratamiento intensivo de datos a alta velocidad, esta característica junto con un conjunto de instrucciones adecuado, hacen que los DSPs sean también eficientes en el procesamiento de los algoritmos fuzzy [Cam92]. En esta aplicación se ha utilizado un DSP de propósito general, el DSP56001 de Motorola.

2.1. Arquitectura interna de los DSPs.

En el mercado existen DSPs que realizan una función específica como un filtro FIR o una transformada de Fourier, en los que son programables únicamente ciertos coeficientes, y también DSPs de propósito general que permiten la programación de una amplia gama de algoritmos típicos en el procesamiento de señales.

No existe una arquitectura que responda a todos los DSPs de propósito general, sin embargo, en la figura 1 se presenta un esquema que recoge sus características más comunes. En la figura se distinguen tres grandes bloques: a) **unidades de ejecución**, b) **memorias** y c) **periféricos**. Las tres unidades de ejecución son la Unidad de Control de Programa (CPU), la Unidad Aritmética Lógica de datos (ALU) y la Unidad de Generación de Direcciones (AGU). Estas tres unidades operan independientemente sobre sus propios registros, de forma que los buses de datos quedan libres para realizar movimientos de datos en paralelo.

Los DSPs disponen de espacios separados de memoria. Incorporan un espacio de **memoria de programa** y uno o varios espacios de **memoria de datos** de acceso independiente. En general estos espacios son internos con posibilidad de expansión externa. Disponen además de múltiples **buses de datos y direcciones** que permiten transferencias simultáneas de datos e instrucciones entre las distintas unidades del sistema [Men89].

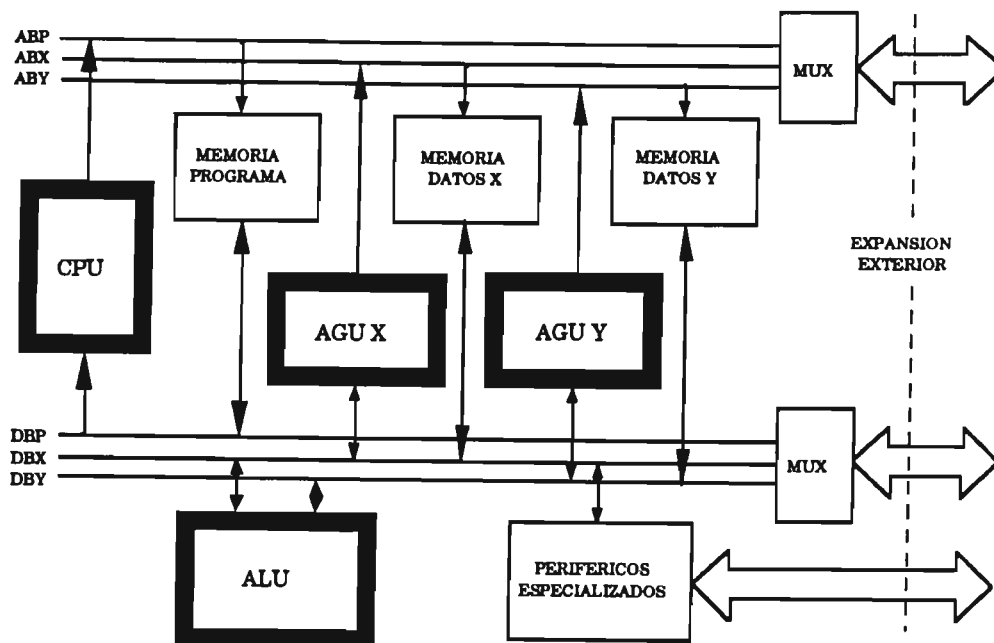


Figura 1: Arquitectura interna de un DSP de propósito general. ABP, ABX y ABY representan los Buses de Direcciones de Programa, de Datos X y de Datos Y respectivamente. DBP, DBX y DBY corresponden a los Buses de Datos para Programa, para Datos X y para Datos Y respectivamente.

2.2. Características del DSP56001 de Motorola.

El DSP56001 de Motorola es un DSP de propósito general realizado mediante tecnología VLSI CMOS de bajo consumo de potencia. Puede ejecutar hasta 13.5 MIPS con un reloj de 27 MHz, esta velocidad le permite ejecutar una instrucción compleja en tan solo 74.07 ns. Para conseguir este rendimiento se ha diseñado una ALU de un único ciclo de instrucción, una arquitectura interna de múltiples buses y un conjunto de instrucciones adecuado.

Los diseñadores han elegido un formato para los datos en coma fija fraccional en complemento a dos con 24 bits de precisión. La ALU dispone de acumuladores de 56 bits que aseguran la precisión en los cálculos intermedios.

Dispone de tres espacios de memoria independientes, una memoria de programa, la memoria de datos X y la memoria de datos Y. Cada memoria de datos interna está compuesta por una memoria RAM de 256 x 24 bits y una memoria ROM de 256 x 24 bits. La memoria de programa interna consiste en 256 x 24 bits de memoria RAM. Estos tres espacios de memoria pueden expandirse externamente hasta 64K x 24 bits.

En cuanto al conjunto de instrucciones, está formado por las típicas de un microprocesador convencional, más un conjunto de instrucciones dedicadas al procesamiento de señal. Hay que destacar que la mayoría de las instrucciones lógicas y aritméticas permiten especificar hasta dos transferencias de datos en paralelo con la operación. Así, es posible combinar dos o tres instrucciones convencionales en una única instrucción. Una instrucción característica es la de multiplicar-acumular (MAC) que multiplica dos registros y suma el resultado al contenido de un tercero:

MAC X0,Y0,A X:(R0)+,X0 Y:(R4)+,Y0

Esta instrucción multiplica el contenido del registro X0 con el contenido del registro Y0 y suma el resultado al contenido del acumulador A. En paralelo lleva un nuevo dato desde memoria al registro X0 a través del bus de datos X, utilizando direccionamiento indirecto mediante el registro R0 que se incrementa después de la operación. De forma similar actualiza el contenido de los registros Y0 y R4. Como puede verse, la clave para la velocidad del DSP56001 está en que las transferencias de datos entre las unidades y memoria se realiza en paralelo con la ejecución de operaciones internas en estas mismas unidades [MOT86] [Klo86] [Men90].

3. Realización de un controlador fuzzy sobre un DSP.

La adecuación del DSP para la realización de un controlador fuzzy se centra básicamente en tres aspectos:

- i) La **arquitectura interna** del DSP, con múltiples buses de datos y direcciones, unidades que operan sobre sus propios registros, y espacios separados de memoria. En particular la **naturaleza dual de la memoria de datos**, que permite disponer reglas y hechos en memoria de forma natural.
- ii) El **conjunto de instrucciones**, dedicado al procesamiento de señal, se adecúa también a la realización de los algoritmos fuzzy que mayor aceptación han tenido en aplicaciones al control. Cabe destacar el mecanismo de inferencias max-prod y la defuzzificación mediante el cálculo del centro de gravedad, como se verá más adelante.
- iii) Los **periféricos de Entrada/Salida**. Dispone de dos puertas E/S de propósito general denominadas puertas B y C. La puerta B puede utilizarse como puerta de E/S de propósito general o como puerta de servicio al periférico (host). La puerta C puede utilizarse como puerta E/S de propósito general o dedicada a los periféricos serie SCI y SSI. Gracias a estos periféricos el DSP puede funcionar en una amplia gama de configuraciones (maestro-esclavo, multiprocesador, etc.).

A continuación se describe con más detalle la realización del Controlador Fuzzy que hemos desarrollado para el DSP56001.

3.1. Organización de las memorias de datos.

El formato de regla utilizado en esta aplicación es de uno o dos antecedentes y un consecuente "if (X es A_i and Y es B_j) then Z es C_i ". Las funciones de pertenencia, que caracterizan tanto a los antecedentes y consecuentes de las reglas como a los hechos, se definen sobre universos discretos, con una discretización máxima de 32 niveles. Cada función de pertenencia se define en memoria de datos como un bloque de 33 posiciones consecutivas de memoria. La primera posición indica los niveles de discretización (número de elementos del universo), y las 32 restantes se reservan para los valores de la función de pertenencia (figura 2).

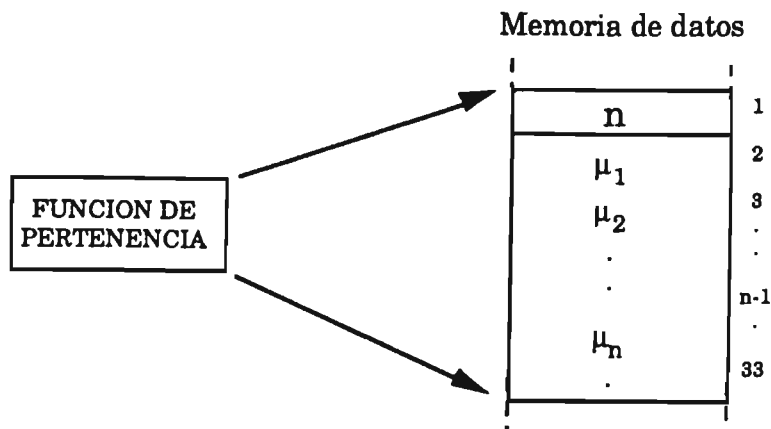


Figura 2: Formato utilizado en la definición de funciones de pertenencia. Cada función de pertenencia se define en memoria de datos como un bloque de 33 posiciones consecutivas de memoria. La primera posición se reserva para indicar el número de niveles de discretización (n) y las restantes para la definición de la función.

Con objeto de sacar el máximo partido al **paralelismo en la captura de datos**, se ha dispuesto la Base de Reglas en memoria de datos X y la Base de Hechos en memoria interna de datos Y. Así, en un único ciclo de instrucción, es posible acceder a reglas y hechos, en paralelo con la ejecución de una operación aritmética o lógica. Es importante notar que únicamente se ha ocupado una memoria externa (la de datos X). Con este mapeado de memorias se evita la penalización, mediante ciclos de espera, por acceso simultáneo a dos o más memorias externas. La necesidad de introducir ciclos de espera se debe a que el DSP56001 dispone de un único bus de datos y direcciones externo.

En la figura 3 se ha representado el mapeado de memorias de datos X e Y. En los 64 K de memoria de datos X se pueden acomodar **hasta 630 reglas**. Como puede verse en la figura, las últimas 64 posiciones de memoria de datos X se reservan para los periféricos internos y en las 32 posiciones anteriores se ha dispuesto una Tabla de Escalado que se utiliza en las rutinas de conversión de información.

En la memoria de datos Y, se ha reservado la memoria RAM interna para Hechos y Tablas de Entrada/Salida. En las 256 posiciones de RAM interna es posible definir **hasta 7 variables de entrada/salida**.

3.2. Realización de los algoritmos fuzzy.

El núcleo del programa consta de tres procesos que se ejecutan de forma secuencial a) Conversión de información concreta en fuzzy C/F, b) Inferencias y c) Conversión de información fuzzy en concreta F/C. Estos procesos involucran distintos lazos de programa.

El proceso a) incluye un lazo externo sobre el número de entradas, y un lazo interno sobre los niveles de discretización de la variable de entrada. El proceso b) incluye lazos anidados de dos niveles, el externo se realiza sobre el número de reglas. El nivel interno consta de tres lazos secuenciales que se realizan sobre los niveles de discretización correspondientes al primer antecedente, al segundo antecedente y al consecuente de la regla. Finalmente, el proceso c) consta de un lazo externo sobre el número de salidas y otro interno sobre los niveles de discretización de la variable de salida.

Los algoritmos fuzzy, así como los algoritmos típicos del procesamiento de señal, se caracterizan porque sus programas implican lazos anidados. El DSP56001 incluye una instrucción, DO, que permite realizar lazos anidados controlados por hardware. Esta instrucción sustituye a las instrucciones "decremento, comparación y salto" que se utilizan para construir los lazos por software. La instrucción DO simplifica notablemente el código y ejecuta el núcleo del lazo sin añadir ciclos de instrucción adicionales, únicamente los de ejecución de la instrucción al comienzo del lazo.

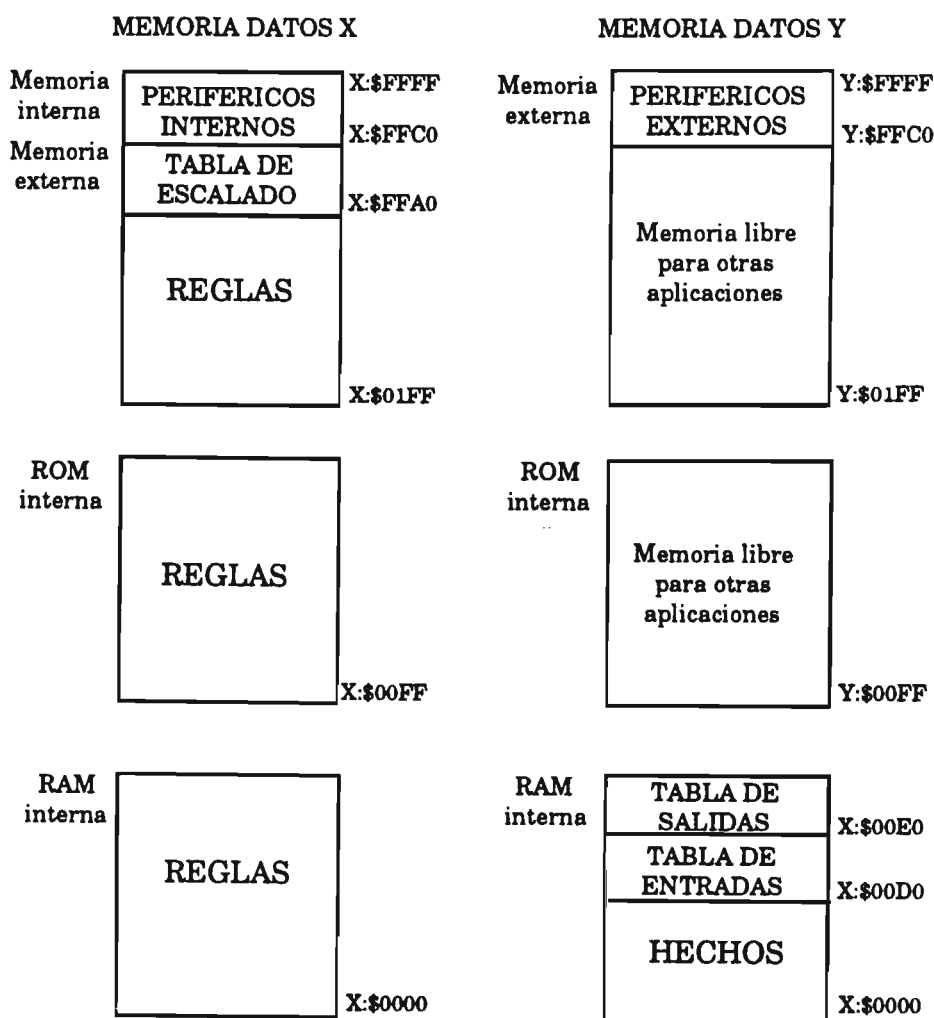


Figura 3: Organización de las memorias de datos. Mapeado de la Base de Reglas en memoria de datos X y de la Base de Hechos en memoria de datos Y.

Para dotar de flexibilidad a la realización, hemos incluido distintos algoritmos fuzzy, tanto de conversión de información como de inferencias. Hemos optado por aquellos algoritmos que mayor aceptación han tenido en aplicaciones al control y que mejor se adecúan a la arquitectura interna y al conjunto de instrucciones del DSP.

a) Conversión de información concreta en fuzzy C/F.

El proceso de conversión de información C/F consta de dos fases, escalado y fuzzificación. La fase de **escalado** es común a todos los algoritmos de conversión C/F, la de **fuzzificación**, sin embargo, puede abordarse mediante distintas estrategias dependiendo de la naturaleza de los datos medidos. Los dos operadores

de fuzzyficación que hemos incluido en la realización, **fuzS** y **fuzG**, suponen que las medidas no están afectadas por errores.

La primera estrategia de fuzzyficación es la más utilizada en aplicaciones al control. Esta genera un conjunto fuzzy cuya función de pertenencia vale 1 en el valor medido, x_0 , y 0 en el resto de los puntos del universo (Fuzzy Singleton):

$$\text{fuzS}(x_0) = \mu_A(x) = \begin{cases} 1 & \text{si } x=x_0 \\ 0 & \text{en otro caso} \end{cases} \quad (1)$$

Este operador representa una manera natural de definir una función de pertenencia sin introducir difuminación en la medida. Sin embargo, si se trata con universos discretos, se está introduciendo un error de aproximación del valor medido, x_0 , al punto del universo más próximo a él. Dicho error puede tener efectos notables si la discretización elegida no es suficientemente fina. Con el objeto de tener en cuenta la discretización, hemos definido además un nuevo operador de fuzzyficación, **fuzG**, que conserva el Centro de Gravedad de la medida:

$$\begin{aligned} \text{si } x_i \leq x_0 \leq (x_i+x_{i+1})/2 \quad \text{entonces} \quad \mu_A(x) &= \begin{cases} 1 & \text{si } x=x_i \\ (x_i-x_0)/(x_0-x_{i+1}) & \text{si } x=x_{i+1} \\ 0 & \text{en otro caso.} \end{cases} \\ \text{si } (x_i+x_{i+1})/2 \leq x_0 \leq x_{i+1} \quad \text{entonces} \quad \mu_A(x) &= \begin{cases} (x_{i+1}-x_0)/(x_0-x_i) & \text{si } x=x_i \\ 1 & \text{si } x=x_{i+1} \\ 0 & \text{en otro caso.} \end{cases} \end{aligned} \quad (2)$$

en otras palabras, si el valor medido, x_0 , pertenece al intervalo $[x_i, x_{i+1}]$, entonces se define una función de pertenencia tal que vale 0 en todos los puntos del universo excepto en los extremos del intervalo. En el extremo del intervalo más próximo a la medida se define como 1 y en el otro extremo del intervalo toma un valor tal que *el centro de gravedad de la función coincide con el valor medido*. Con este operador se consigue una variación continua de los centros de gravedad de las funciones de pertenencia aunque los universos sean discretos.

La función de **escalado** es una función real que lleva el rango de variación de una variable de entrada al sistema fuzzy [mín,máx], al rango de variación del universo fuzzy correspondiente, $[1,n]$, siendo n el nivel de discretización de dicho universo fuzzy:

$$\text{esc}: [\text{mín}, \text{máx}] \rightarrow [1, n], \quad 1 \leq n \leq 32, \quad n \in \mathbb{N}.$$

$$\text{esc}(x) = (x - \text{mín}) (n - 1) / (\text{máx} - \text{mín}) + 1 \quad (3)$$

Seguidamente, y con objeto de destacar algunos aspectos interesantes de la aplicación, incluimos un extracto de código ensamblador que corresponde al escalado de un dato de entrada. Para comprender mejor el conjunto de instrucciones, hay que recordar que el formato de los datos del DSP56001 es de coma fija fraccional en complemento a dos. Los datos en los registros aritméticos y en memoria deben ser vistos como fracciones tomando valores entre -1.0 y 1.0 (estrictamente en el intervalo $[-1.0, 1.0)$). Así, el intervalo $[\text{mín}, \text{máx}]$ se reduce al intervalo $[0.0, 1.0)$ y la expresión (3) se convierte en:

$$\text{esc}(x) = x(n-1) + 1 \quad (4)$$

Notación:

x_0 : dato en Puerta de Entrada, $x_0 \in [0.0, 1.0)$

n : niveles de discretización del universo fuzzy, $1 \leq n \leq 32$, $n \in \mathbb{N}$.

x_e : dato escalado al universo fuzzy, $x_e \in [1, n]$, $x_e \in \mathbb{R}$.

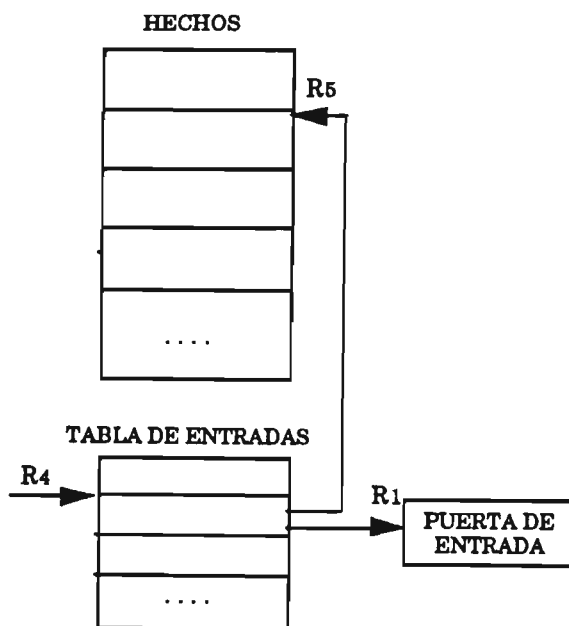
Código ensamblador:

<u>Etiqueta</u>	<u>código de operación</u>	<u>operandos</u>	<u>movimientos paralelo</u>
MOVE	X:(R1),X0	Y:(R4)+,Y0	; $X0=x_0$ (dato), $Y0=(n-1)/32$
MPY	X0,Y0,A	#(1/32.0),B	; $A=x_0(n-1)/32$
ADD	B,A		; $A=[x_0(n-1) + 1]/32$
REP	#5		
ASL	A		; $A=x_e=x_0(n-1) + 1$

La gestión de datos y apuntadores se ha representado en la figura 4.a. El formato fraccional para los datos se adecúa especialmente a la realización de inferencias fuzzy, ya que las funciones de pertenencia únicamente toman valores entre 0 y 1. Sin embargo, en los algoritmos de conversión C/F y F/C, se manejan datos fuera del rango fraccional, por lo que es necesario introducir factores de escala, cuyo manejo se muestra en este ejemplo. Los apuntadores R1 y R4 apuntan respectivamente al dato x_0 en Puerta de Entrada y al factor de escala $(n-1)/32$ en la Tabla de Entradas.

- I1. En la primera instrucción se cargan en paralelo el dato y el factor de escala en los registros X0 e Y0 respectivamente.
- I2. En la segunda instrucción se multiplican estos operandos depositándose el resultado en el acumulador A, en paralelo con esta operación se carga el acumulador B con la fracción $1/32$.
- I3. La siguiente instrucción suma el contenido de los acumuladores A y B cuyo resultado es el dato escalado al universo fuzzy, x_e según (3.2), y escalado a su vez al rango fraccional, $x_e/32$. Este resultado queda almacenado en el acumulador A.
- I4,5. Finalmente, en las dos últimas instrucciones, se multiplica el contenido de A por 32 para obtener x_e , ya que aunque el DSP56001 maneja datos fraccionales, los datos en los acumuladores pueden tomar valores en el intervalo $[-256.0, 256.0)$.

a) Fuzzificación



b) Defuzzificación

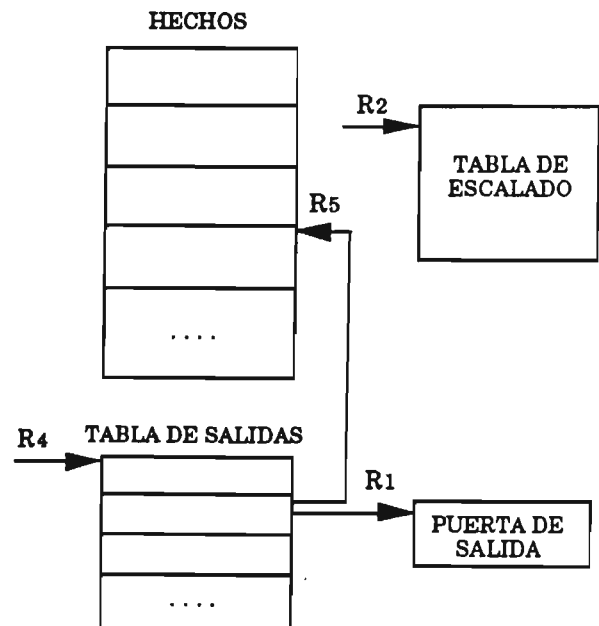


Figura 4: Gestión de datos y apuntes en las rutinas de conversión de información. a) Escalado y fuzzificación y b) Defuzzificación y escalado.

b) Mecanismos de inferencias.

Los mecanismos de inferencias que hemos incluido en el repertorio de algoritmos fuzzy del DSP son los métodos **máx-mín** y **máx-prod**. Dado un conjunto de reglas R_i : "if x es A_i and y es B_i then C es C_i ", y un par de observaciones (A', B') , la conclusión C' se obtiene como sigue:

$$\mu_{C'}(z) = \max \mu_{C'_i}(z), \text{ siendo } \mu_{C'_i}(z) = \alpha_i * \mu_{C_i}(z), \alpha_i = \alpha_i^A * \alpha_i^B, \text{ donde}$$

$$\alpha_i^A = \max_x [\mu_{A'}(x) * \mu_{A_i}(x)] \text{ y } \alpha_i^B = \max_x [\mu_{B'}(x) * \mu_{B_i}(x)], \quad (5)$$

"*" representa el operador mínimo o producto.

Hemos seleccionado una parte del algoritmo de inferencias máx-prod, el cálculo de α para una regla con un único antecedente, para destacar el paralelismo en el movimiento de datos desde memoria de datos X y memoria de datos Y , con la ejecución de la operación de multiplicación. Hay que destacar que estas operaciones se realizan en un único ciclo de instrucción (74.07 nsg). Para facilitar la comprensión del código, vamos a particularizar (5) para una regla con un único antecedente. Dada una reglas R : If x es A then z es C , y un hecho A' , la conclusión C' para una única regla se obtiene como sigue:

$$\mu_{C'}(z) = \alpha * \mu_C(z), \text{ donde} \quad (6)$$

$$\alpha = \max_x [\mu_{A'}(x_i) * \mu_A(x_i)], 1 \leq i \leq n_A, \quad (7)$$

"." representa el operador producto algebraico.

Notación:

n_A : niveles de discretización del universo fuzzy X , $1 \leq n_A \leq 32$, $n_A \in \mathbb{N}$.

$\mu_A(x_i)$: función de pertenencia del antecedente de la regla, $x_i \in X$.

$\mu_{A'}(x_i)$: función de pertenencia del hecho de entrada (dato fuzzy), $x_i \in X$.

Algoritmo:

ALGORITMO cálculo de alfa;

```

Begin
  i:=0;
  Cobegin
    alfa:=0;
    leer( $\mu_A(x_{i+1})$ );
    leer( $\mu_{A'}(x_{i+1})$ );
    i:=i+1;
  Coend
  While i≤nA do
    Begin
      Cobegin
        prod:= $\mu_A(x_i) \cdot \mu_{A'}(x_i)$ ;
        leer( $\mu_A(x_{i+1})$ );
        leer( $\mu_{A'}(x_{i+1})$ );
        i:=i+1
      Coend;
      If prod>alfa then alfa:=prod
    End
  End; {cálculo de alfa}

```

Código ensamblador:

<u>etiqueta</u>	<u>código de operación</u>	<u>operandos</u>	<u>movimientos paralelo</u>		
	CLR	B	X:(R1)+,X0	Y:(R4)+,Y0	; X0= $\mu_A(x_1)$, Y0= $\mu_{A'}(x_1)$
	DO	nA,end			
	MPY	X0,Y0,A	X:(R1)+,X0	Y:(R4)+,Y0	; A= $\mu_A(x_1) \cdot \mu_{A'}(x_1)$, ; X0= $\mu_A(x_{i+1})$, Y0= $\mu_{A'}(x_{i+1})$
	CMP	B,A			; compara B con A
	TGE	A,B			; deja el mayor en B
end					

Los apuntadores R1 y R4 apuntan al primer par de datos ($\mu_A(x_1), \mu_{A'}(x_1)$).

- I1. En la primera instrucción se limpia el acumulador B y en paralelo con esta operación se carga el primer par de datos en los registros X0 e Y0 respectivamente, además, se actualizan los apuntadores.
- I3. La primera instrucción en el lazo DO realiza el producto del primer par de datos dejando el resultado en el acumulador A, carga un nuevo par de datos y actualiza los apuntadores.

I3.4. Las dos instrucciones restantes calculan el máximo según (7). La gestión de datos y apuntadores puede verse en la figura 5.

El cálculo de α se ejecuta en $8+6n$ ciclos de reloj, un DSP56001 con un reloj de 27 MHz puede procesar este código en $3.8 \mu s$, para un valor típico de $n_A=16$. Una inferencia completa por el método máx-prod requiere solo $11.7 \mu s$ para cada regla.

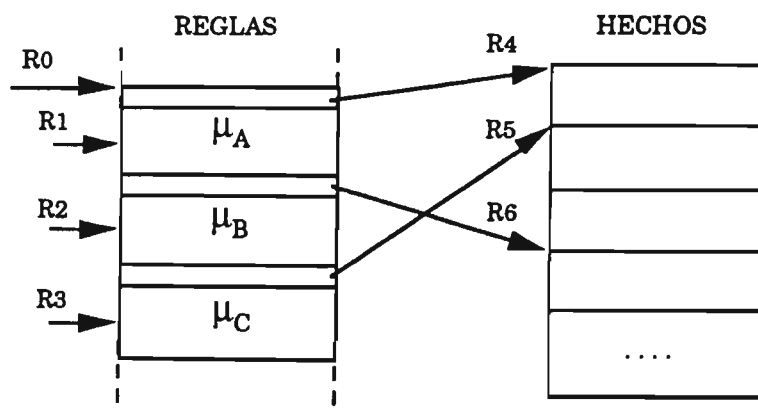


Figura 5: Gestión de datos y apuntadores en las rutinas de inferencias.

c) Conversión de información fuzzy en concreta F/C.

La conversión de información fuzzy en concreta implica una fase de **defuzzificación** y otra fase de **escalado**. Como estrategias de defuzzificación se han incluido las funciones **defuz_G** basada en el cálculo del centro de gravedad, y **defuz_M** basada en el cálculo del máximo o promedio de máximos de la función de pertenencia.

Centro de Gravedad:

$$z_0 = \text{defuz}_G(z) = \frac{\sum_{i=1}^{n_C} \mu_C(z_i) z_i}{\sum_{i=1}^{n_C} \mu_C(z_i)} \quad (8)$$

Promedio de Máximos:

$$z_0 = \text{defuz}_G(z) = \sum_{i=1}^I z_i / I \quad (9)$$

z_i : máximos del mismo valor.

I : número de máximos con el mismo valor.

Como ejemplo, proponemos la defuzzificación por el método defuz_G.

Notación:

n_C : niveles de discretización del universo fuzzy Z , $1 \leq n_C \leq 32$, $n_C \in \mathbb{N}$.

$\mu_C(z_i)$: función de pertenencia de la conclusión, $z_i \in Z$

z_0 : conclusión

Algoritmo:

ALGORITMO cálculo del centro de gravedad;

```

Begin
  i:=0;
  Cobegin
    numerador:=0;
    leer(nC)
  Coend;
  Cobegin
    denominador:=0;
    leer(zi+1);
    leer(  $\mu_C(z_{i+1})$ );
    i:=i+1
  Coend;
  While i≤nC do
    Begin
      numerador:=numerador+  $\mu_C(z_i)*z_i$ ;
      Cobegin
        denominador:=denominador +  $\mu_C(z_i)$ ;
        leer(zi+1);
        leer( $\mu_C(z_{i+1})$ );
        i:=i+1
      Coend;
    End,
    cdg:=numerador/denominador
  End; (cálculo del centro de gravedad)

```

Código ensamblador:

<u>etiqueta</u>	<u>código de operación</u>	<u>operandos</u>	<u>movimientos paralelo</u>	
	CLR	A	Y:(R5)+,N5	; N5=nC
	CLR	B	X:(R2)+,X0	; X0=1/32, Y0= $\mu_C(z_1)$
	DO	nC,end		;
	MAC	X0,Y0,A		; A=A+(i/32) $\mu_C(z_i)$
	ADD	Y0,B	X:(R2)+,X0	Y:(R5)+,Y0 ; B=B+ $\mu_C(z_i)$,
end				; X0=(i+1)/32, Y0= $\mu_C(z_{i+1})$
	DO	#5,shift		
	ASR	A		; divide numerador/32 (A)
	ASR	B		; y denominador (B) en 3.4
shift				; por 32
	MOVE	B,X0		; X0=B
	AND	#\$FE,CCR		; limpia bit de carry
	REP	#\$18		; divide A/X0 (24 bits)
	DIV	X0,A		; A=z ₀ /32

La gestión de datos y apuntadores se ha representado en la figura 4.b. Se observa la definición de una Tabla de Escalado para defuzzificación. Esta tabla ocupa 32 posiciones consecutivas en memoria de datos X, los coeficientes de la tabla corresponden a la fracción $i/32$ para valores enteros de i entre 1 y 32 (ver mapeado de memoria en figura 3).

El primer lazo del algoritmo realiza el cálculo del numerador, que se ha escalado al rango fraccional haciendo uso de la Tabla de Escalado, y del denominador de (9). Obsérvese el paralelismo de la operación suma (ADD) con la captura de los factores de escala en memoria de datos X y los valores de la función de pertenencia en memoria de datos Y.

El segundo lazo realiza un nuevo escalado dividiendo numerador y denominador por 32, con ello no se alterará el resultado de la división y se asegura que tanto dividendo y divisor como cociente estarán dentro del rango fraccional.

4. Configuraciones de sistemas fuzzy basadas en DSPs.

El DSP56001 permite distintos modos de operación, puede operar en modo "stand-alone", o en modo maestro-esclavo, como coprocesador de un microprocesador de propósito general:

i) En **modo stand-alone** el DSP ejecuta su propio programa residente en memoria (figura 6).

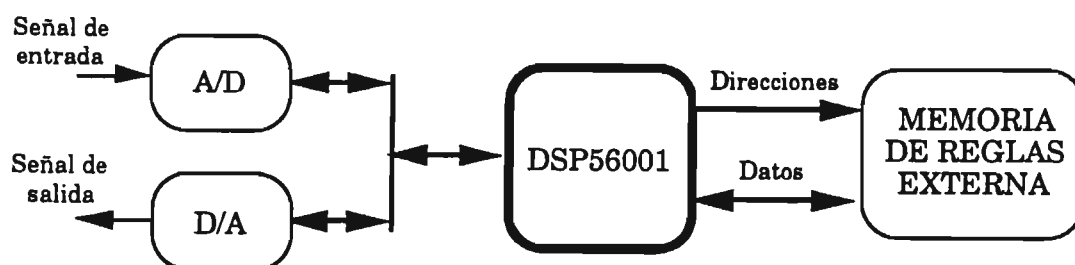


Figura 6: Configuración de Sistema Fuzzy basada en DSP en modo "stand-alone".

ii) En **modo maestro-esclavo**, el procesador principal (host) actúa como controlador del Sistema Fuzzy y realiza la interfaz con el usuario, mientras el DSP realiza el procesamiento de inferencias. Para ello el DSP dispone de una puerta paralelo de 8 bits, full-duplex, que puede conectarse directamente al bus de datos de un microprocesador, otro DSP o un hardware DMA. Una peculiaridad de este periférico es que el procesador principal puede solicitar la ejecución de rutinas especiales, definidas por el usuario, por medio de mecanismos de interrupciones. Este periférico permite ver al DSP desde otro procesador como un coprocesador fuzzy.

Esta característica de los DSPs hace posible la modificación on-line de los parámetros de un Controlador Fuzzy. Así, sería posible conmutar on-line entre distintos algoritmos fuzzy o modificar la definición de reglas y funciones de pertenencia. Esta opción es de gran importancia en la realización de pruebas y ajustes en tiempo real. Si la evolución del Sistema Fuzzy es satisfactoria, puede desconectarse el DSP del procesador principal y operar en modo stand-alone.

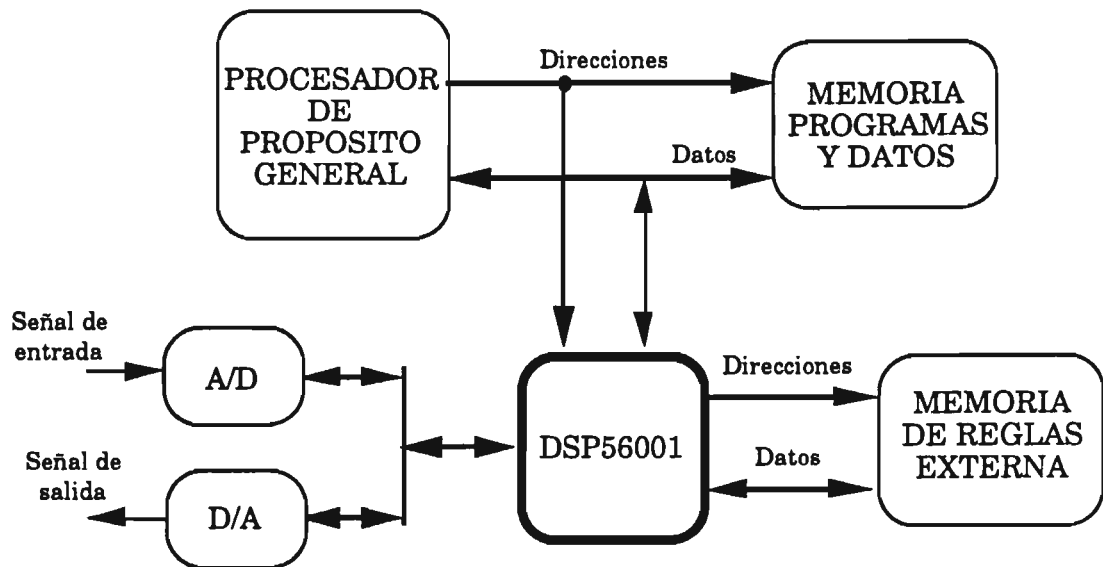


Figura 7: Configuración de Sistema Fuzzy basado en DSP en modo maestro-esclavo.

El DSP56001 admite así mismo configuraciones multiprocesador en las que varios DSPs realizan tareas diferentes comunicándose entre sí a través de un árbitro, o configuraciones multiprocesador en cascada en las que cada DSP realiza parte de una tarea común comunicándose entre sí [MOT86].

5. Entorno CASE para el desarrollo de sistemas fuzzy.

Como herramienta de ayuda al diseñador, hemos realizado un entorno de tipo CASE para el desarrollo de Sistemas Fuzzy [Cam91]. Este entorno brinda soporte al ciclo completo de desarrollo del sistema. Las utilidades que hemos incorporado están orientadas a agilizar tanto la definición, ajuste y verificación del sistema, como la traducción del prototipo sobre un hardware específico. Con el fin de abarcar una gama amplia de problemas y dar opción al diseñador para que seleccione un hardware adecuado a las características de la aplicación, hemos incluido distintas alternativas que consisten en un software en lenguaje de alto nivel, una arquitectura fuzzy para un CI de tipo EPLD y la realización optimizada para un Procesador Digital de Señal, como caso particular de microprocesador.

Conclusiones.

En este trabajo presentamos la realización de un controlador fuzzy sobre un Procesador Digital de Señal (DSP). La naturaleza dual de la memoria de datos, que permite el acceso en paralelo a la Base de Reglas y a la Base de Hechos, y un conjunto de instrucciones adecuado, son la clave para la velocidad de procesamiento de inferencias fuzzy del DSP. El DSP56001 utilizado en esta aplicación, con un reloj de 27 MHz, puede procesar aproximadamente 62.000 reglas/seg con un mecanismo de inferencias máx-mín y 85.000 reglas/seg con un mecanismo de inferencias máx-prod. Esta velocidad se consigue para funciones de pertenencia definidas sobre universos discretos de 16 niveles.

Las características más destacables de esta realización son a) flexibilidad con respecto a la selección de algoritmos fuzzy, b) velocidad de respuesta suficientemente alta para gran número de aplicaciones en tiempo real y c) bajo

costo. Es importante así mismo la posibilidad de ampliar el repertorio actual de algoritmos, por ejemplo, definiendo algoritmos basados en redes neuronales, que junto con los algoritmos fuzzy constituyen una herramienta muy potente en aplicaciones al control.

Finalmente, hay que destacar la utilidad de disponer de un entorno de tipo CASE, como el que hemos desarrollado, como soporte en el diseño de un prototipo y su posterior traducción sobre un hardware de alta velocidad.

Referencias.

- [Alo91] Alonso, J., Caresana, M., Lanza, A., Panetsos, F. and Ríos, J., "A DSP based board for neural network simulation", *Microprocessing and Microprogramming*, vol. 29, nº 5, pp.263-271, 1991.
- [Cam91] del Campo, I., Garitagoitia, J.R. y Tarela, J.M., "Un Sistema de Desarrollo de Aplicaciones Basadas en la Lógica Borrosa orientado al Control de Procesos", *proc. del Primer Congreso Español sobre Tecnologías y Lógica Fuzzy*, pp. 91-95, Granada, 17-20 Junio, 1991.
- [Cam92] del Campo, I., González de Mendivil, J.R., Garitagoitia, J.R. and Tarela, J.M., "A DSP Based Fuzzy Logic Controller", *proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, July 17-22, 1992.
- [Klo86] Kloker, K., "The Motorola DSP56000 Digital Signal Processor", *IEEE Micro Magazine*, pp. 29-49, Dec. 1986.
- [Men89] González de Mendivil, J.R. y Garitagoitia, J.R., "Introducción a los DSPs", *Revista Española de Electrónica*, pp. 36-41, Julio/Agosto 1989.
- [Men90] González de Mendivil, J.R. y Garitagoitia, J.R., "El Procesador Digital de Señal DSP56000/1", *Revista Española de Electrónica*, pp. 86-92, Junio 1990.
- [MOT86] Motorola, "DSP56000/DSP56001 Digital Signal Processor User's Manual", Motorola DSP56000UM/AD, 1986.
- [Sch92] Schwartz, D.G. and Klir, G.J., "Fuzzy logic flowers in Japan", *IEEE Spectrum*, pp. 32-35, July 1992.
- [Sel90] Self, K., "Designing with fuzzy logic", *IEEE Spectrum*, pp. 42-44, 105, November 1990.
- [OMR92] OMRON, "Los Productos de OMRON", *Curso de Ingeniería Fuzzy organizado por el Instituto Nacional de Tecnología Aeroespacial (INTA)*, Madrid, Julio 1992.
- [TOG91] Togai Infralogic, "Product Summary and Suggestions", *información técnica*, Octubre 1991.

Controlador Difuso Basado En Contextos

J.L. Castro y J.M. Zurita

*Departamento de Ciencias de la Computación e Inteligencia Artificial
Facultad de Ciencias. Universidad de Granada
Campus de Fuentenueva
18071 Granada (Spain)*

Resumen

Uno de los inconvenientes con que nos enfrentamos al hacer uso de la lógica difusa es el esfuerzo computacional en tiempo requerido [2]. De hecho, muchos de los procesos que trabajan con este tipo de lógica se ven seriamente afectados. El problema es aún más acuciante cuando se necesita una respuesta rápida a unos determinados requerimientos. Este es el caso por ejemplo de los sistemas de control, en los cuales un retardo excesivo puede ocasionar la invalidez del sistema para controlar un determinado proceso.

El objetivo de este artículo es mostrar un mecanismo, basado en *contextos*, que permita un ahorro considerable de tiempo en los procesos de cálculo.

Palabras Clave: Controlador difuso, Restricción Difusa, Contexto.

1 Introducción

Los sistemas de control difuso están adquiriendo progresivamente un mayor interés en la industria como unidad implementable en muchos de sus productos con notables resultados.

Un sistema de control difuso se puede definir como un conjunto de reglas (suministradas por un experto) que indican cómo deben de intervenir las variables de control del sistema en función de las variables de salida proporcionadas por éste, de forma que se obtenga un estado de funcionamiento predeterminado descrito por el experto para ese sistema [4].

Siendo $\mathcal{X}_1, \dots, \mathcal{X}_n$ observaciones efectuadas a la salida del sistema que denominaremos variables de salida del sistema, con referenciales $\mathcal{R}_{\mathcal{X}_1}, \dots, \mathcal{R}_{\mathcal{X}_n}$ respectivamente, e $\mathcal{Y}$ variable de control con referencial $\mathcal{R}_{\mathcal{Y}}$, un esquema usual de sistema de control difuso es el que se observa en la figura 1.

El sistema de reglas de control [1] será de la forma

Si $\mathcal{X}_1$ es $\mathcal{A}_1$, $\mathcal{X}_2$ es $\mathcal{A}_2$, ..., $\mathcal{X}_n$ es $\mathcal{A}_n$ entonces $\mathcal{Y}$ es $\mathcal{B}$,

$$R(\mathcal{X}_1) = \mathcal{A}_1, R(\mathcal{X}_2) = \mathcal{A}_2, \dots, R(\mathcal{X}_n) = \mathcal{A}_n, R(\mathcal{Y}) = \mathcal{B}$$

tal que,

$$\forall x_1 \in \mathcal{R}_{\mathcal{X}_1}, x_2 \in \mathcal{R}_{\mathcal{X}_2}, \dots, x_n \in \mathcal{R}_{\mathcal{X}_n}, y \in \mathcal{R}_{\mathcal{Y}},$$

se obtiene

$$\mu_{\mathcal{A}_i}(x_i) = \alpha_i, \mu_{\mathcal{B}}(y) = \beta; \forall i \in 1, \dots, n; \forall \alpha_i, \beta \in [0, 1]$$

siendo α_i y β el grado de creencia en que x_i e y son considerados $\mathcal{A}_i$ y $\mathcal{B}$ respectivamente.

El conjunto de restricciones diferentes asociadas a una variable suele fluctuar en un número comprendido entre 2 y 7.

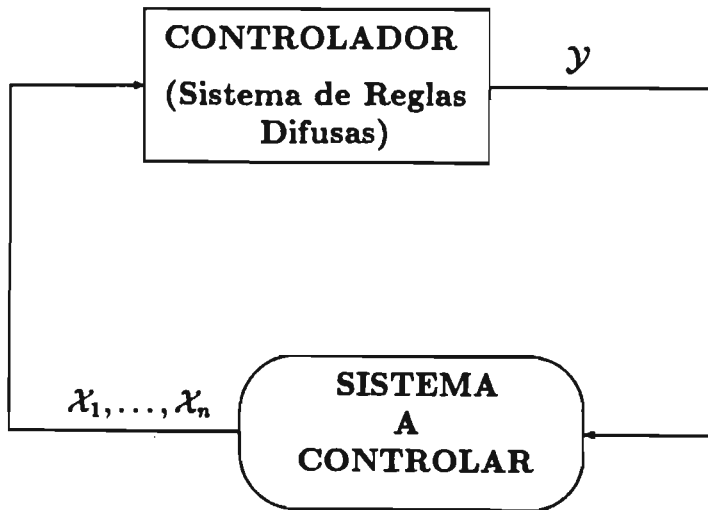


Figura 1: Esquema de interacción de un controlador difuso

2 Modelo Propuesto

Sea $\mathcal{E} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m\}$ el conjunto de todas las variables de salida del sistema a controlar. Notaremos por $\mathcal{C} = \{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n\}$ el conjunto completo de variables que controlan el correcto equilibrio de su funcionamiento.

Introduciremos el operador de relación $\propto$ que asociará los resultados testeados a la salida del sistema con los efectos que deberán producir en el control del mismo.

Expresaremos

$$\mathcal{X}_{i_1} \wedge \mathcal{X}_{i_2} \wedge \dots \wedge \mathcal{X}_{i_k} \propto \mathcal{Y}_i$$

$$\forall i \in 1, \dots, n; k \in 1, \dots, m$$

para indicar que la variable de control $\mathcal{Y}_i$ es factible de modificar su valor de entrada siempre que cualquiera de los valores del subconjunto de variables de salida $\{\mathcal{X}_{i_1}, \mathcal{X}_{i_2}, \dots, \mathcal{X}_{i_k}\}$ sea modificado.

Dado $CONT(\mathcal{Y}_i)$, notaremos por $M_{\mathcal{X}_{i_1}}, M_{\mathcal{X}_{i_2}}, \dots, M_{\mathcal{X}_{i_k}}$, el número de estados diferentes que interesan en las variables $\mathcal{X}_{i_1}, \mathcal{X}_{i_2}, \dots, \mathcal{X}_{i_k}$ respectivamente para sus valores de salida.

Así, para almacenar todas las posibles tuplas $(y_i, x_{i_1}, x_{i_2}, \dots, x_{i_k})$ verificándose $(x_{i_1}, x_{i_2}, \dots, x_{i_k}) \in CONT(y_i), y_i \in \mathcal{R}_{\mathcal{Y}_i}$, necesitaremos

$$(M_{\mathcal{X}_{i_1}} * M_{\mathcal{X}_{i_2}} * \dots * M_{\mathcal{X}_{i_k}})$$

unidades de almacenamiento. Siendo por tanto,

$$\sum_{i=1}^n (M_{\mathcal{X}_{i_1}} * M_{\mathcal{X}_{i_2}} * \dots * M_{\mathcal{X}_{i_k}})$$

la memoria global requerida.

En lo referente al tiempo empleado en el espacio de búsqueda,

$$O(\lg(M_{\mathcal{X}_{i_1}} * M_{\mathcal{X}_{i_2}} * \dots * M_{\mathcal{X}_{i_k}}))$$

será el orden de complejidad necesitado para localizar el valor $y_i \in \mathcal{R}_{\mathcal{Y}_i}$, tal que $(x_{i_1}, x_{i_2}, \dots, x_{i_k}) \in CONT(y_i)$, ó en su caso, insertar dicha tupla en la estructura de almacenamiento.

Un serio inconveniente es la posibilidad de que los valores $M_{\mathcal{X}_{i_1}}, M_{\mathcal{X}_{i_2}}, \dots, M_{\mathcal{X}_{i_k}}$ pueden ser excesivamente grandes, hecho que conllevaría que la cantidad de memoria requerida puede ser demasiado elevada.

Es importante hacer notar que los órdenes se han calculado anteriormente en el peor de los casos. Esto significa que todas las combinaciones posibles de los valores $M_{\mathcal{X}_{i_1}}, M_{\mathcal{X}_{i_2}}, \dots, M_{\mathcal{X}_{i_k}}$ asociados a la variable de control $\mathcal{Y}_i$ se darán durante el funcionamiento del sistema; hecho que afortunadamente no suele suceder.

Tres factores influyen notablemente en la eficiencia del proceso de control examinado anteriormente:

- Precisión
- Estabilidad
- Independencia

Si el sistema es muy preciso en las observaciones $\mathcal{X}_1, \dots, \mathcal{X}_m$, mayor será a su vez el conjunto de elementos a considerar dentro de los respectivos referenciales $\mathcal{R}_{\mathcal{X}_1}, \dots, \mathcal{R}_{\mathcal{X}_m}$ y mayor serán por tanto los valores $M_{\mathcal{X}_{i_1}}, M_{\mathcal{X}_{i_2}}, \dots, M_{\mathcal{X}_{i_k}}$. Esto redundaría en un incremento de la capacidad de almacenamiento y de búsqueda.

La estabilidad del sistema supondrá que éste se mantiene entre unos determinados valores de salida más o menos estables y poco cambiantes de una etapa a otra, que de suceder

Extendiendo la definición,

$$\mathcal{X}_{i_1} \wedge \mathcal{X}_{i_2} \wedge, \dots, \wedge \mathcal{X}_{i_k} \propto \mathcal{Y}_i \iff x_{i_1} \wedge x_{i_2} \wedge, \dots, \wedge x_{i_k} \propto y_i$$

$$x_{i_k} \in \mathcal{R}_{\mathcal{X}_j}; y_i \in \mathcal{R}_{\mathcal{Y}_i}, i, j \in 1, \dots, n; k \in 1, \dots, m$$

Llamaremos *Contexto asociado a una variable de control* $\mathcal{Y}_i$ que notaremos por $CONT(\mathcal{Y}_i)$ [3],

$$CONT(\mathcal{Y}_i) = \{\mathcal{X}_{i_1} \mathcal{X}_{i_2} \dots, \mathcal{X}_{i_k} / \mathcal{X}_{i_1} \wedge \mathcal{X}_{i_2} \wedge, \dots, \wedge \mathcal{X}_{i_k} \propto \mathcal{Y}_i\}$$

y adicionalmente,

$$CONT(y_i) = \{(x_{i_1} x_{i_2} \dots, x_{i_k}) / x_{i_1} \wedge x_{i_2} \wedge, \dots, \wedge x_{i_k} \propto y_i\}$$

El controlador actuará de la siguiente forma:

Dada la m-tupla $(x_1, \dots, x_m) \in (\mathcal{R}_{\mathcal{X}_1} \times \dots \times \mathcal{R}_{\mathcal{X}_m})$ generada a la salida del sistema que se controla, deberemos obtener la n-upla $(y_1, \dots, y_n) \in (\mathcal{R}_{\mathcal{Y}_1} \times \dots \times \mathcal{R}_{\mathcal{Y}_n})$ para que se consiga el equilibrio requerido (mediate el conjunto de reglas difusas suministradas por el experto) del sistema.

Algoritmo

Si existe una tupla $(x_{i_1}, x_{i_2}, \dots, x_{i_k})$ que no ha modificado su valor respecto a la anterior etapa, y es además componente de un determinado contexto $CONT(y_i)$, el valor y_i a la entrada del sistema será el de la etapa anterior y no necesitará ser calculado.

Si no, para cada variable $\mathcal{Y}_i$ cuyo contexto haya quedado afectado por algún cambio en la nueva m-upla, el valor de salida y_i se calculará de la siguiente forma:

Se Buscará la tupla $(y'_i, x_{i_1}, x_{i_2}, \dots, x_{i_k})$,

Si se encuentra: el valor de salida y_i será y'_i .

Si no, se calculará dicho valor operando mediante el sistema de reglas de inferencia difuso para posteriormente almacenar la nueva tupla $(y_i, x_{i_1}, x_{i_2}, \dots, x_{i_k})$ en la estructura.

3 Análisis

En este apartado haremos un estudio de eficiencia en tiempo y en espacio para establecer los ámbitos de viabilidad en que puede ser aplicado.

así disminuiría notablemente tanto la cantidad de memoria necesaria como la cantidad de tiempo empleado.

Diremos que dos variables de control $\mathcal{Y}_i$ e $\mathcal{Y}_j$ son independientes si verifican,

$$CONT(\mathcal{Y}_i) \cap CONT(\mathcal{Y}_j) = \phi$$

El tiempo y el espacio se verían de nuevo favorecidos si esta propiedad es verificada por el sistema que pretendemos controlar.

Resumiendo, podemos acertar a decir que el modelo de control presentado será, tanto más efectivo, cuanto más baja sea la precisión, más alta la estabilidad y mayor independencia generen las variables de entrada del sistema que controla.

4 Observaciones

Bien es cierto que los órdenes de complejidad calculados dependen fuertemente de la estructura de datos seleccionada para la representación de la información manejada.

Una posible estructura (que es la que se ha tenido en mente en el desarrollo del apartado anterior), mantendría un árbol binario de búsqueda para cada variable de control $\mathcal{Y}_i, i \in 1, \dots, n$. Los nodos de cada árbol estarán formados por tuplas de la forma $(y_i, x_{i_1} \ x_{i_2} \ , \dots, \ x_{i_k})$ tal que $(x_{i_1}, x_{i_2}, \dots, x_{i_k}) \in CONT(y_i)$. Es decir, almacenaría los contextos generadores de y_i durante el desarrollo del proceso de control.

Sean $\Theta_1, \Theta_2, \dots, \Theta_m$, relaciones de ordenes definidas sobre las variables el conjunto de variables $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m$. Sea $<$, la relación de orden definida sobre los nodos de un árbol binario de búsqueda. La expresión

$$(y_i, x_{i_1} \ x_{i_2} \ , \dots, \ x_{i_k}) < (y_j, x_{j_1} \ x_{j_2} \ , \dots, \ x_{j_k})$$

vendrá obviamente definida a partir de las relaciones individuales anteriores.

Una implementación por punteros de dicha estructura supondrá del orden del $\lg$ del número de elementos del árbol generado para cada una de las variables $\mathcal{Y}_i$.

5 Conclusiones

En este artículo hemos pretendido ofrecer un apoyo a la Teoría de Sistemas de Control Difuso, precisamente en uno de los aspectos que nos ha parecido de especial relevancia.

El hecho de requerir una lógica difusa subyacente, así como el fin último práctico a que ha sido dedicado en aplicaciones industriales, el tiempo es un factor a valorar y sobre el que hemos pretendido incidir en el diseño de este tipo de sistemas.

Bibliografía

- [1] Bellman R.E. and Zadeh L.A. (1977). *Local and Fuzzy Logics*. J.M. Dunn and G. Epstein (eds.). Modern Uses of Multiple-Valued Logic 103-165. Dordrecht Holland
- [2] Castro J.L. and Zurita J.M. (1992). *A Fuzzy Logic ATMS*. Non-Classical Logics and their Applications. 14th Linz Seminar on Fuzzy Set Theory. Austria.
- [3] De Kleer (1986). *An Assumption-based TMS*. Artificial Intelligence, 28.
- [4] Sugeno M. (ed.) (1985). *Industrial Applications of Fuzzy Control*. North-Holland

SEGUIMIENTO FRECUENCIAL MEDIANTE LÓGICA DIFUSA

Autores : M. Reyes de los Mozos,
E. Valderrama, J. Carrabina,
N. Avellana, F. Lisa

Departamento de Diseño
Centre Nacional de Microelectrònica
Universitat Autònoma de Barcelona
Bellaterra (ESPAÑA)

Resumen : En este artículo se presenta un F.L.C. (*Fuzzy Logic Controller*) que realiza el control y seguimiento de la frecuencia de un VCXO, de forma que la mantiene estable dentro del margen de error de 1ppm (partes por millón), en el rango de temperaturas de -55 a +75 °C. Se determinan las funciones modelo para los conjuntos fuzzy, así como la función de implicación para el proceso de inferencia, con el fin de optimizar el proceso de defuzzificación. El FLC tiene una estructura de red, lo que permite realizar un número elevado de cálculos en paralelo, reduciendo el tiempo de respuesta del controlador.

Palabras clave : T.C.X.O, F.L.C., Cuantificación, Bloque de inferencia, Defuzzificación.

1. Introducción

Muchas aplicaciones con osciladores de cristal requieren una estabilidad mayor, por encima del rango de temperaturas de operación, de la que se puede llegar a alcanzar con un cristal no compensado a temperatura ambiente. En todas estas aplicaciones es deseable vencer todos los problemas derivados de la dependencia del cristal con respecto a la temperatura. Esto puede llegar a conseguirse siguiendo una de las técnicas de compensación que existen hoy en día. Actualmente estas técnicas utilizan sistemas tanto analógicos como digitales, llegándose a conseguir un error próximo a 1ppm en el rango de temperaturas desde -55 a 75 °C. El problema básico de todas estas técnicas es la necesidad de conocer las curvas de comportamiento del cristal de cuarzo en función de la temperatura, obligando al fabricante de cristales a realizar una serie de medidas en diferentes condiciones, incrementando el coste final del producto, haciéndolo menos competitivo en el mercado.

En este artículo se presenta un controlador basado en lógica difusa que realiza el seguimiento y control de la frecuencia proporcionada por un V.C.X.O. (*Voltage Controlled Crystal Oscillator*), de forma que el sistema completo se comporte como un T.C.X.O. (*Temperature Compensated Crystal Oscillators*). La principal mejora aportada por esta nueva

familia de TCXO es la no necesidad de las curvas de comportamiento en función de la temperatura, lo cual reduce el coste final del producto.

2. Controlador de lógica difusa

Durante los últimos años el control fuzzy ha aparecido como una de las más activas y productivas áreas para los investigadores en aplicaciones de la teoría de conjuntos fuzzy. Varias son las razones de este crecimiento en aplicaciones con FLC (*Fuzzy Logic Controller*), siendo efectiva su utilización en el contexto de procesos complejos mal definidos, que pueden ser controlados por un operador humano sin conocimiento de su dinámica fundamental. La lógica fuzzy - que es la lógica sobre la que un FLC está basado - está mucho más próximo al pensamiento humano y lenguaje natural que los tradicionales sistemas lógicos, lo cual hace a los FLC's más atractivos que los controladores clásicos en problemas de control del tipo como el antes comentado.

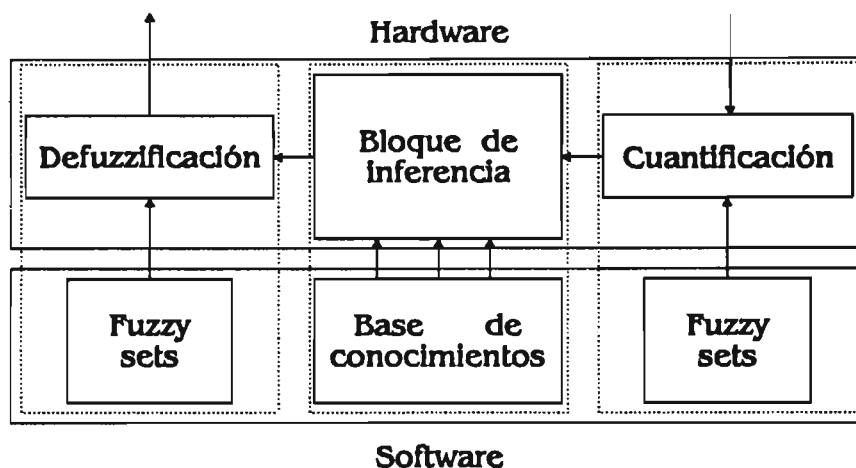


Fig. 1. : Esquema de bloques de un F.L.C.

La parte esencial de un FLC es un conjunto de reglas de control lingüísticas relacionadas entre sí mediante el concepto de implicación fuzzy y una serie de reglas de composición de inferencia. Si se construye un FLC, tal que este evalúe las reglas en paralelo, reducimos el tiempo de respuesta del FLC, obteniendo una señal de control en tiempo real. En la figura 1 se muestra el esquema de bloques del FLC utilizado en esta aplicación. El FLC se compone de tres módulos totalmente independientes : cuantificación, bloque de inferencia y defuzzificación. Cada uno de estos módulos se divide en dos bloques : uno corresponde al hardware del sistema, y el otro corresponde al software, donde tendremos las reglas de control y la definición de los conjuntos fuzzy.

2.1. Cuantificación

A un FLC le llega como entrada los valores medidos sobre la planta, en este caso la entrada es la frecuencia a la que oscila el VCXO, y el diferencial de esta frecuencia. Estas medidas se han de proyectar sobre el espacio de operación del FLC, se ha de realizar la función de cuantificación o fuzzificación.

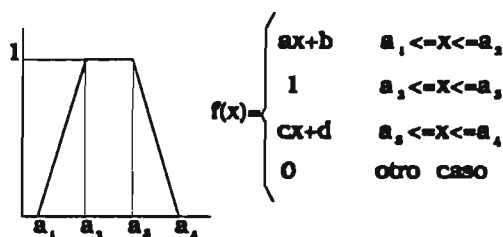


Fig. 2. : Modelo de función para los conjuntos fuzzy de los universos de condición

Definimos en un principio dos universos de condición : frecuencia, variación de frecuencia. En cada uno de estos universos se han definido siete conjuntos fuzzy : NG, NM, NP, IC, PP, PM, PG (NG : Negativo Grande, NM : Negativo Medio, ...). Varios son los modelos de función para definir un conjunto fuzzy, en la figura dos se muestra el adoptado en este caso. La elección de este tipo de función se debe a la sencillez en que se puede realizar el cálculo de cada uno de los μ_i .

2.2. Bloque de inferencia

El bloque de inferencia de un FLC es un típico sistema MIMO (*multiinput and multioutput*) donde, dado un número de entradas μ_i , el sistema proporciona un número de salidas μ_o , siendo el grado en que se produce cada uno de los conjuntos fuzzy definidos en cada uno de los universos consecuentes. Este complejo sistema MIMO puede dividirse en un número finito de sistemas MISO (*multiinput and single-output*) independientes entre ellos. Cada uno de estos sistemas MISO se compone de una regla de control fuzzy, caracterizando, el conjunto de todas, el comportamiento dinámico del sistema fuzzy.

Una regla de control fuzzy es una declaración condicional fuzzy en que el antecedente es una condición en su dominio de aplicación, y el consecuente es una acción de control para la planta. En este caso las reglas fuzzy de control son definidas como

$$\underline{\text{SI}} \ f_k \text{ es } XX \ \underline{\text{Y}} \ \Delta f_k \text{ es } YY, \underline{\text{ENTONCES}} \ y = f(x)$$

donde f_k es la frecuencia a la que oscila el VCXO, Δf_k indica la variación de la frecuencia de oscilación, y $f(x)$ es una función modelo como la mostrada en la figura 3. Este modelo de

función viene determinado por el proceso de defuzzificación, aspecto que se tratará en el próximo apartado.

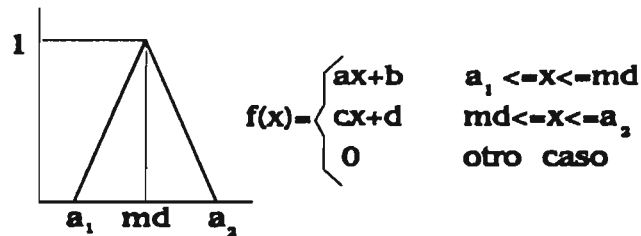


Fig. 3. : Modelo de función para los conjuntos fuzzy del universo consecuente

Una regla de control fuzzy como la mostrada anteriormente es implementada mediante la implicación fuzzy (*relación fuzzy*) R_i definida :

$$\mu_{R_i} \Delta \mu((f_i \Delta \Delta f_i) \rightarrow y_i)(u, v, w) = [\mu_{f_i}(u) \wedge \mu_{\Delta f_i}(v)] \rightarrow \mu_{y_i}(w) \quad (1)$$

donde la función de implicación se realiza siguiendo la generalización del modus ponens

$$A \rightarrow B = \sup \{ c \in [0, 1], A * c \leq B \} \quad (2)$$

donde * denota el producto algebraico. En la figura 4 se muestra el proceso de evaluación de una regla tal y como se ha desarrollado en este trabajo.

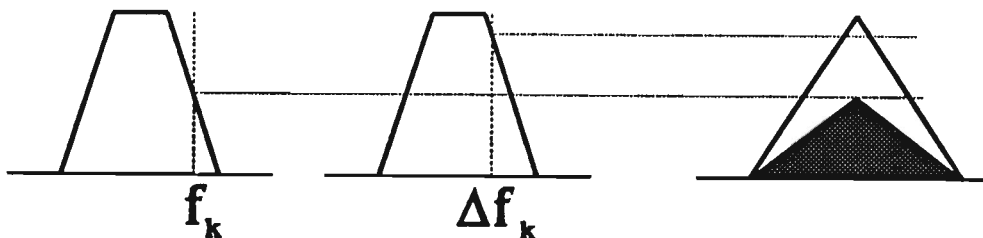


Fig. 4. : Proceso de inferencia : correlación-producto

2.3. Defuzzificación

En el bloque de inferencia se evalúan las reglas, obteniendo de cada una de ellas una función, resultado del proceso de implicación fuzzy. Defuzzificación es el proceso de proyección desde este espacio de acciones de control fuzzy, definido sobre el universo de salida, al espacio de acciones de control no-fuzzy (*crisp*), en este caso la tensión a aplicar sobre el VCXO. Este proceso es el cuello de botella de todo FLC que quiera trabajar en tiempo real siendo eficiente. Varias son las posibles estrategias a seguir, siendo la más utilizada la del cálculo del "centro de

masas" de la combinación de los resultados obtenidos en la evaluación de las reglas de control fuzzy. La expresión que describe esta técnica es,

$$v_k = \frac{\int y m_o(y) dy}{\int m_o(y) dy}, \quad (3)$$

donde discretizando la expresión queda

$$v_k = \frac{\sum_{j=1}^p y_j m_o(y_j)}{\sum_{j=1}^p m_o(y_j)}, \quad (4)$$

siendo p el número de posibles tensiones que puede proporcionar el FLC.

Esta solución no es factible si se desea obtener una respuesta en tiempo real sin tener un excesivo hardware. El tiempo de cálculo de la expresión es elevado, debido básicamente al número de multiplicaciones que se deben realizar. Para reducir este tiempo de cálculo se obtendrá el centro de masas mediante la expresión

$$v_k = \frac{\sum_{j=1}^7 \mu_o(y_j) y_j J_j}{\sum_{j=1}^7 \mu_o(y_j) J_j}, \quad (5)$$

donde J_j es el área del j -ésimo conjunto fuzzy consecuente, y_j es la moda de la función que define el conjunto j -ésimo de universo de acciones, y $\mu_o(y_j)$ se deduce de la expresión

$$\mu_o(y_j) = \sum_{i=1}^N \mu_{o_i}(y_j). \quad (6)$$

El cálculo del centro de masas según la ecuación (5) es válido si se definen siete conjuntos fuzzy en el universo de acciones de control, siendo todos ellos simétricos y unimodales, tal como se mostraba en la figura 3. Además se ha de utilizar para el proceso de inferencia la operación correlación-producto, tal como se indicó en el apartado anterior.

Es posible reducir aún más el número de multiplicaciones si las funciones con que se definen los conjuntos tienen las mismas pendientes, de esta forma tendríamos $J_1 = \dots = J_7$, reduciendo la expresión del proceso de defuzzificación.

3. Implementación del F.L.C.

Para la implementación del FLC se ha pensado en un sistema distribuido, en un sistema donde el grado de paralelismo sea lo más alto posible, para ello se introduce una estructura en red, análoga a una red neural. Esta red está constituida por un número finito de unidades de proceso, clasificadas en cuatro capas, donde los nodos de una capa están conectados con los nodos de las capas inmediatamente anterior y posterior, tal y como se muestra en la figura 5.

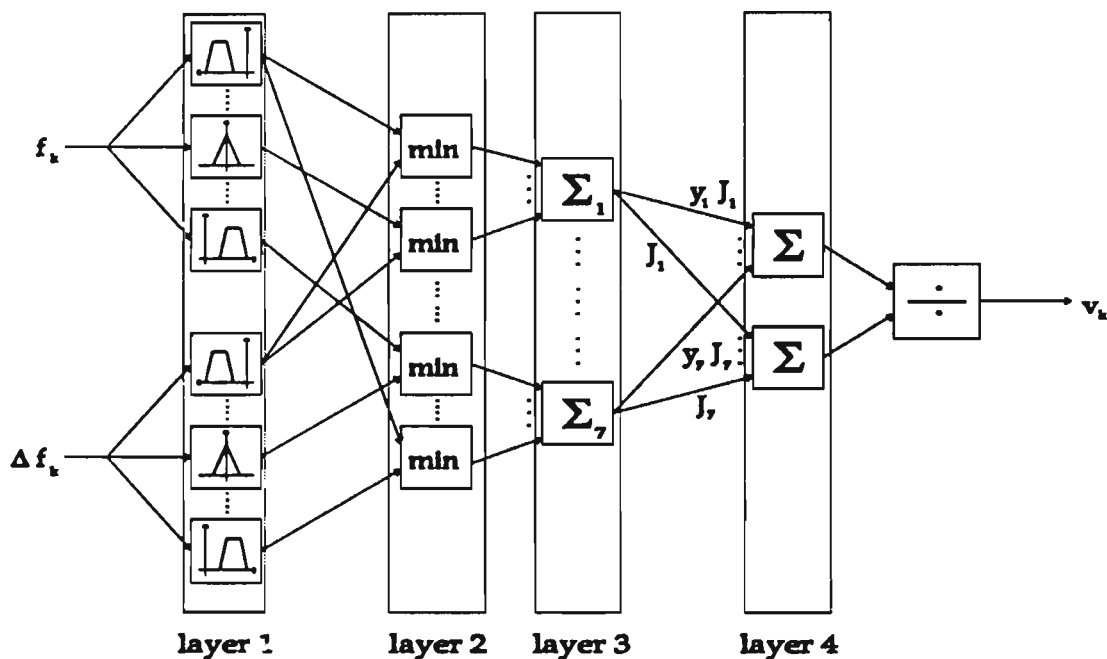


Fig. 5. : Estructura del F.L.C.

Los nodos de la primera capa llevan a cabo el proceso de cuantificación para cada uno de los conjuntos fuzzy definidos en cada uno de los universos de condición,

$$O_{ij} = f_{ij}(I_i) \quad i=1,2 ; \quad j=1,\dots,7 \quad (7)$$

donde f_{ij} es la función que caracteriza el conjunto fuzzy j -ésimo definido en el universo de condición i -ésimo.

En los nodos de la segunda capa se evalúa el grado en que se cumple cada una de las reglas que componen la base de conocimientos del FLC,

$$O_i = \min(i_1, i_2) . \quad (8)$$

La tercera etapa se compone exactamente de siete nodos, donde cada uno de ellos evalúa el grado de actividad de un determinado conjunto fuzzy del universo consecuente,

$$O_i = \sum_{j=1}^N (w_{ij} * h_j) , \quad (9)$$

donde w_{ij} es el peso asignado a cada una de las conexiones entre los nodos del layer 2 y 3. El peso corresponde con la importancia otorgada a cada una de las reglas, en el presente caso los pesos son igual a 0 o 1.

La última capa se compone únicamente de dos nodos, ambos se caracterizan por la expresión,

$$O_{ij} = \sum_{j=1}^7 (w_{ij} * h_j) , \quad (10)$$

donde los pesos de las conexiones son ,

$$w_{1j} = y_j J_j , \quad w_{2j} = J_j ,$$

siendo y_j la moda del conjunto fuzzy j-ésimo del universo consecuente, y J_j es el resultado de la integral,

$$J_j = \int_{a_{j1}}^{a_{j2}} f_j(x) dx , \quad (11)$$

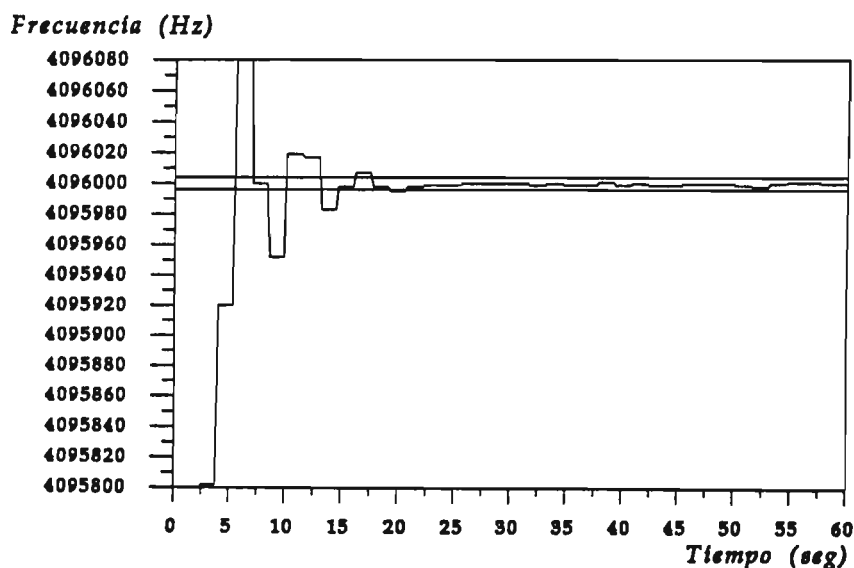
siendo f_j la función característica del conjunto fuzzy j-ésimo del universo consecuente.

4. Resultados experimentales

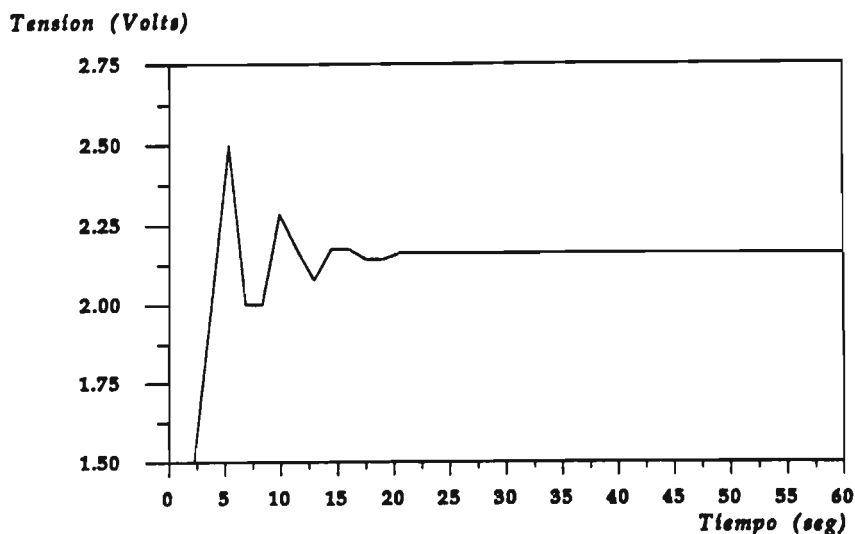
En una primera versión se implementa el FLC mediante un PC-AT, siguiendo la estructura en red antes comentada. En esta primera fase se pretenden obtener unos primeros resultados del comportamiento del sistema, variando por una lado la definición de los conjuntos fuzzy, tanto

para la condición como para el consecuente, y por otro lado variando el conjunto de reglas de control que forman la base de conocimientos. Para todas las pruebas se ha utilizado un VCXO del tipo CT-1612085 de 4.096 MHz de T.F.E..

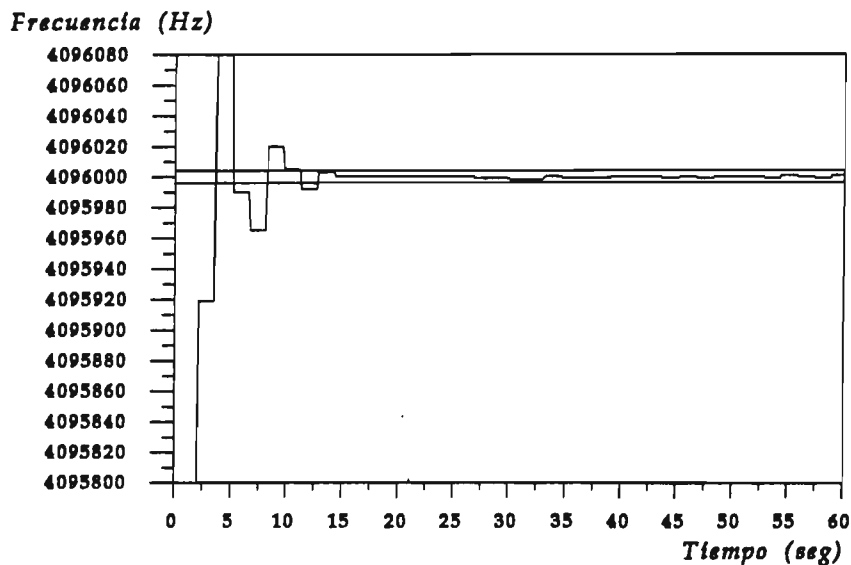
En las primeras pruebas se llega a la conclusión de que el universo de condición Δf_k se puede despreciar, ya que no influye para nada en el comportamiento del VCXO.



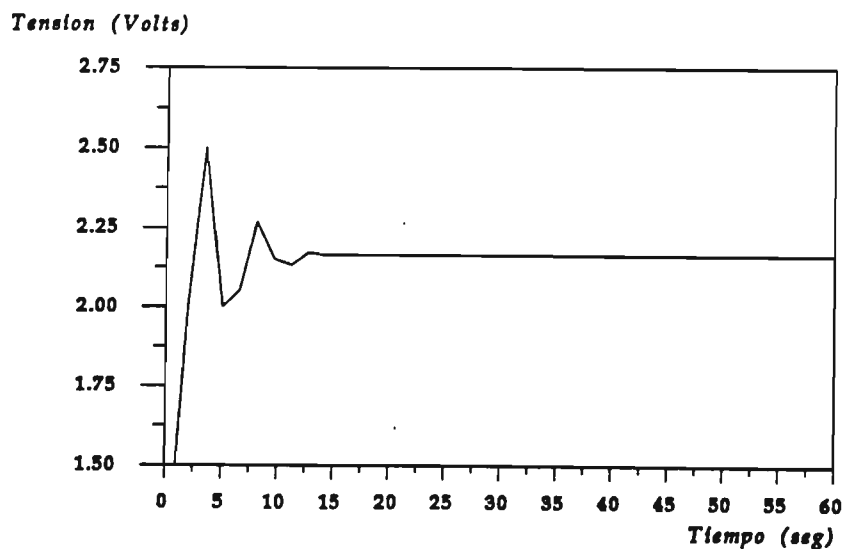
Grafica 1.a : Frecuencia de oscilación del VCXO a una temperatura ambiente de 27°C con el primer grupo de reglas fuzzy de control.



Grafica 1.b. : Tensión suministrada por el FLC a una temperatura ambiente de 27°C con el primer grupo de reglas fuzzy de control.

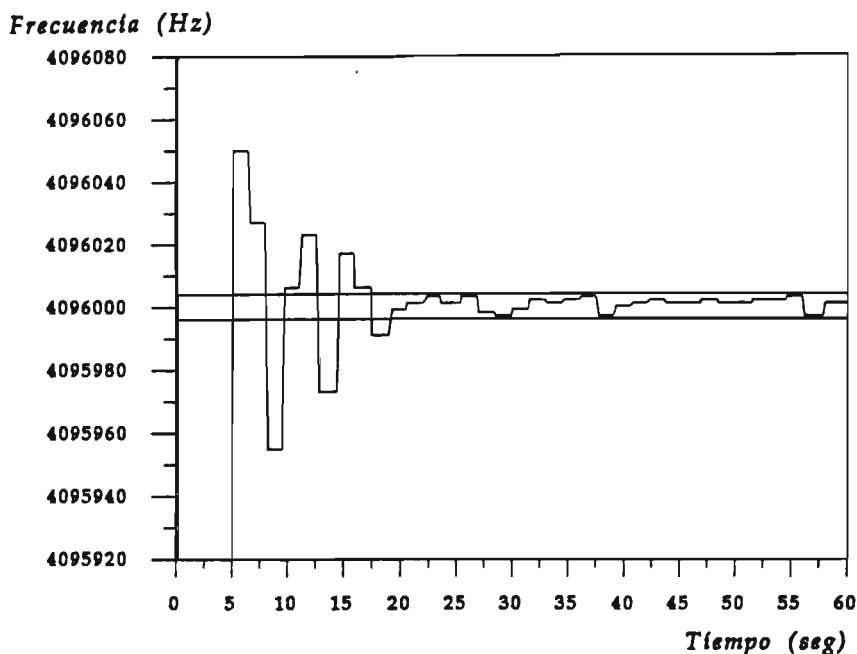


Grafica 2.a. : Frecuencia de oscilación del VCXO a una temperatura ambiente de 27°C con el segundo grupo de reglas fuzzy de control.

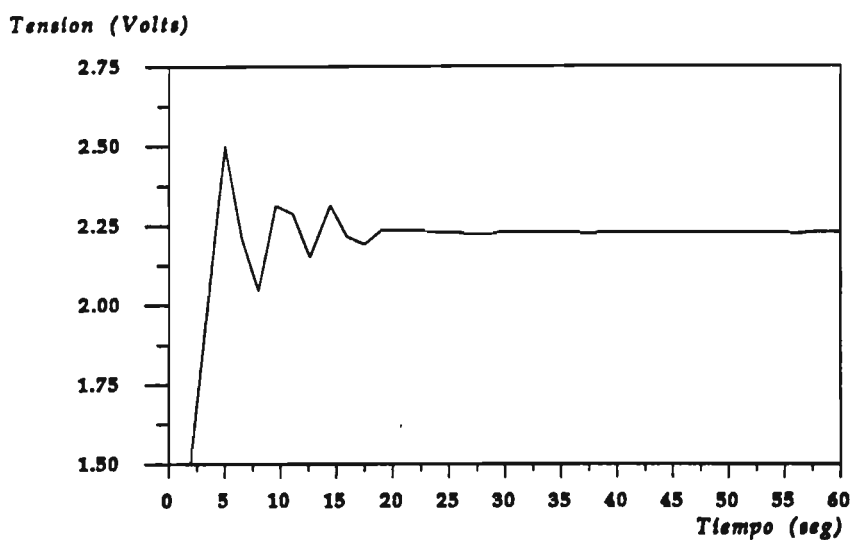


Grafica 2.b. : Tensión suministrada por el FLC a una temperatura ambiente de 27°C con el segundo grupo de reglas fuzzy de control.

En las siguientes pruebas se ha sometido al VCXO a diferentes temperaturas, obteniendo los resultados que se muestran en las gráficas 1,2,3 y 4. Las gráficas 1 y 2 muestra la frecuencia de oscilación a 27 °C bajo el control del FLC, comprobando que se cumple el objetivo marcado, tener un error inferior a 1ppm. En la gráfica 2 se alcanza la frecuencia objetivo más rápido que en el caso de la gráfica 1, esto se debe al cambio de reglas de control entre un caso y otro. La



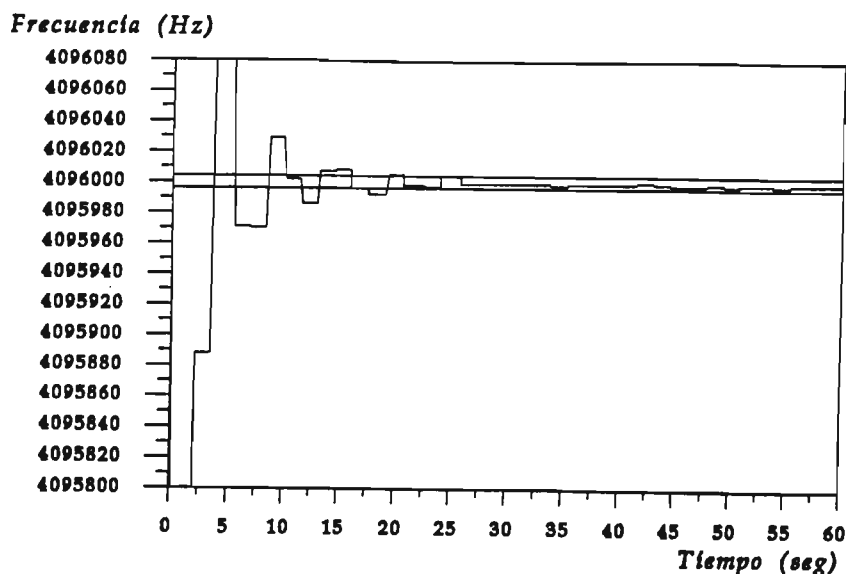
Grafica 3.a. : Frecuencia de oscilación del VCXO a una temperatura ambiente de 70°C con el primer grupo de reglas fuzzy de control.



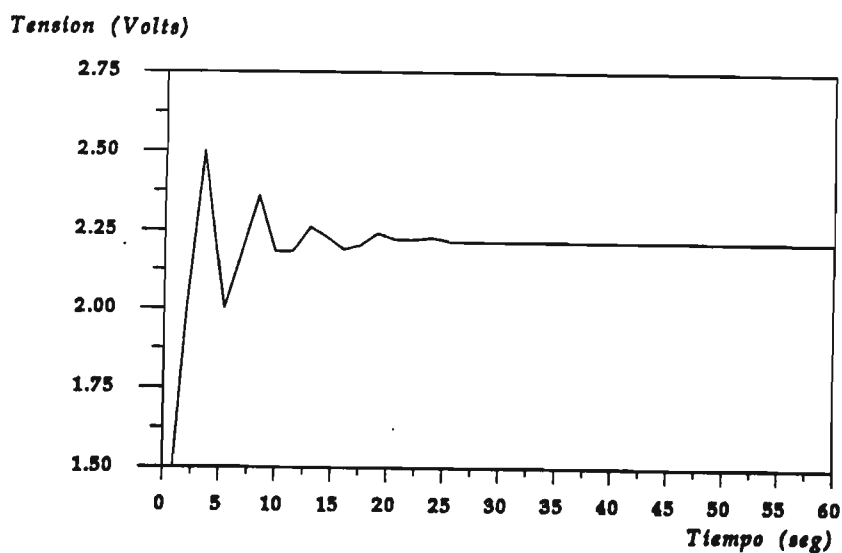
Grafica 3.b. : Tensión suministrada por el FLC a una temperatura ambiente de 70°C con el primer grupo de reglas fuzzy de control.

elección de las reglas de control es un punto crucial para que el FLC sea eficiente y óptimo. Otro punto crucial en la estabilidad y eficacia del sistema es cómo se definen los conjuntos fuzzy.

En las gráficas 3 y 4 se muestran los resultados obtenidos cuando la placa está a una temperatura de 70 °C, llegando a las mismas conclusiones.



Grafica 4.a. : Frecuencia de oscilación del VCXO a una temperatura ambiente de 70°C con el segundo grupo de reglas fuzzy de control.



Grafica 4.b. : Tensión suministrada por el FLC a una temperatura ambiente de 70°C con el segundo grupo de reglas fuzzy de control.

En el conjunto de las gráficas se observa que el FLC es lento a la hora de proporcionar una respuesta, esto se debe por una parte al proceso secuencial (inherente al PC) del cálculo de las funciones de pertenencia, del proceso de inferencia y del proceso de defuzzificación. Y por otra parte al frecuencímetro utilizado como interface VCXO-PC.

5. Conclusiones

Con la introducción de los controladores basados en lógica difusa se podrán tratar todos aquellos problemas en que el proceso de modelización matemática es muy costoso, llegando incluso a ser imposible obtener una expresión válida que describa el comportamiento dinámico de la planta. El modelo resultado del proceso anterior puede ser de difícil manejo, lo cual deriva en un elevado coste en el diseño del controlador. Para disminuir estos costes, sin necesidad de conocer la dinámica de la planta, se introducen los FLC. Otro tipo de problemas abordables con FLC's son aquellos donde el modelo que describe el comportamiento de la planta varía con el tiempo. La aplicación presentada en este artículo reúne ambas características, por un lado el laborioso y costoso proceso de obtención del modelo del VCXO, y por otro lado la dependencia del cristal respecto a la temperatura.

Actualmente la modelización fuzzy de un proceso no es todavía muy conocida, debido principalmente a las dificultades en el modelaje de una proceso en una estructura lingüística. De todas formas un FLC es atractivo para tratar con seguridad y confianza problemas como los comentados en el párrafo anterior.

Integrando el FLC mediante técnicas VLSI podemos reducir el tiempo de respuesta del sistema, por un lado por las dimensiones del circuito y su correspondiente velocidad de cálculo, así como por la inferencia en paralelo de las reglas fuzzy de control.

Podemos mejorar todavía más el FLC presentado, para ello podemos añadir características propias de las redes neurales, como sería cierta capacidad de aprendizaje, modificando tanto las reglas como los conjuntos fuzzy definidos previamente a lo largo de su funcionamiento, aprendiendo de la experiencia. De esta forma el FLC aprende y se adapta a los cambios del entorno que le rodea, mejorando con ello su comportamiento.

6. Referencias

- [1] B. Parzen, A. Ballato : Design of crystal and other harmonic oscillators, Wiley Interscience, 1983.
- [2] S. Schodowski : "Design of a High Stability Voltage Controlled Crystal Oscillator", U.S. Army Electronics Command Fort Monmouth, New Jersey, Febrero 1972.
- [3] B. Kosko : Neural networks and fuzzy systems : a dynamical systems approach to machine intelligence, New Jersey, Prentice Hall, 1992.
- [4] C. C. Lee : "Fuzzy logic in control systems : Fuzzy logic controller - Parts I & II ", IEEE Trans. Syst. Man Cybern., vol. 20, no. 2, pp. 404-435, Mar./Apr. 1990.

ESTUDIO COMPARATIVO DE CIRCUITOS BÁSICOS DE LÓGICA DIFUSA

S. Sánchez Solano, I. Baturone, A. Barriga, J.L. Huertas

Dpto. de Diseño de Circuitos Analógicos
Centro Nacional de Microelectrónica
Sevilla (España)

Resumen: Se presenta un estudio comparativo de distintos circuitos generadores de funciones de pertenencia. Junto a la estructura clásica de Yamakawa, se analizan dos nuevas estructuras propuestas por los autores. El estudio está enfocado hacia la búsqueda de implementaciones óptimas desde el punto de vista de su aplicabilidad a casos prácticos. Se demuestra asimismo que una de las técnicas de implementación propuestas permite la realización eficiente de operadores máximo y mínimo. Los diferentes circuitos han sido implementados en tecnología CMOS de 1.5 μ . Los resultados de simulación ilustran su comportamiento estático y dinámico.

Palabras Clave: Lógica difusa, circuitos generadores de funciones de pertenencia, operadores máximo y mínimo, implementaciones CMOS.

1. Introducción.

La aplicación de técnicas de control difuso a problemas que requieren soluciones en tiempo real ha motivado un creciente interés en las implementaciones hardware de estos sistemas. Como alternativa a las realizaciones digitales reportadas en la literatura [1-2], durante los últimos años, algunos autores han considerado la aplicación de procedimientos menos convencionales, dando lugar a los denominados "circuitos difusos analógicos" [3-4]. Los sistemas de control basados en esta última aproximación proporcionan un elevado número de inferencias por segundo con un consumo de área reducido. Por otra parte, las técnicas analógicas se corresponden mejor que las digitales con el carácter multivaluado de la lógica difusa.

Un sistema de control difuso consta de un conjunto de reglas expresadas por relaciones lingüísticas del tipo "IF - THEN" y un mecanismo de inferencia capaz de procesar dicha información. Junto a estos elementos básicos, es preciso disponer de los bloques de interfase que transforman los valores deterministas que constituyen las entradas y salidas del sistema bajo control en los valores difusos utilizados por el controlador (Figura 1). La regla composicional de inferencia más ampliamente utilizada es la propuesta por Mamdani [5], que utiliza como operadores básicos el máximo y el mínimo. Desde el punto de vista de su implementación electrónica, tanto los operadores anteriores como el fuzzifier pueden realizarse mediante funciones lineales a tramos.

2. Espacio de Diseño.

Recientemente se han reportado diferentes circuitos para la síntesis de funciones lineales a tramos, operando tanto en modo de corriente [6] como en modo de transconductancia [7]. La utilización de intensidades para procesar la información presenta una serie de ventajas frente al uso de tensiones. Los circuitos resultantes operan a mayor velocidad y las operaciones de suma y resta son directamente implementadas mediante conexiones.

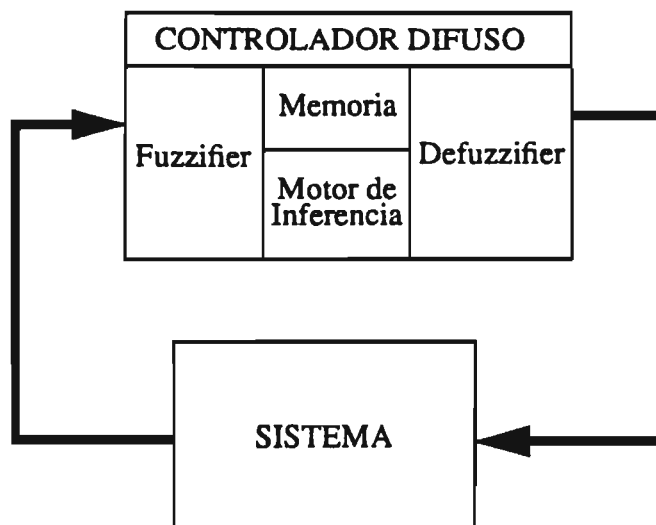


Figura 1: Estructura de un sistema de control difuso.

Los circuitos que utilizan como elementos básicos espejos de corriente realizan las operaciones de suma, escalado o rectificación con un reducido número de transistores. No presentan problemas de estabilidad porque son configuraciones en lazo abierto (salvo los transistores operando como diodos). Además, al presentar nudos de baja impedancia en el camino de la señal, permiten un comportamiento adecuado a altas frecuencias.

Los OTAs constituyen las celdas básicas de los circuitos que operan en modo de transconductancia. La principal característica de estos elementos es que admiten valores de tensión en las entradas, con lo que se evita la necesidad de interfases tensión/intensidad entre las salidas del sistema bajo control y las entradas del controlador. Como contrapartida, las realizaciones con OTAs requieren mayor área de silicio y presentan una serie de problemas relacionados con el ajuste de las tensiones de *offset* y el rango lineal de operación.

El trabajo descrito en esta comunicación se centra en el estudio de circuitos que operan en modo de corriente. El espacio de diseño viene delimitado por dos coordenadas. Por un lado, se analizan diferentes técnicas de generación de funciones de pertenencia lineales a tramos (la técnica de Yamakawa y dos nuevas técnicas presentadas por los autores). Por otro lado, dado que las distintas técnicas se basan en la utilización de espejos de corriente, se determinan cuáles son los tipos de espejos que mejor cumplen los objetivos de diseño. En la literatura pueden encontrarse estudios comparativos de diferentes espejos de corriente CMOS. Dichos estudios direccionan tanto los aspectos de comportamiento estático [8], como dinámico [9] de los distintos espejos. Los resultados reportados en esta comunicación se limitan a la utilización de espejos simples y de las dos estructuras cascode mostradas en la Figura 2.

3. Circuitos Generadores de Funciones de Pertenencia (MFCs).

Los conjuntos difusos que representan a las etiquetas lingüísticas empleadas en las técnicas de razonamiento aproximado se describen matemáticamente mediante funciones de pertenencia. Usualmente estas funciones corresponden a triángulos o trapecios simétricos. En concreto, nuestro estudio se ha centrado en la generación de funciones de pertenencia del tipo anterior, con una pendiente de valor 2 y que cubren un universo de discurso en la variable de entrada I_{in} de unos $90\mu A$. El diseño de los circuitos se ajustó a las geometrías mínimas que verificasen los requerimientos estáticos en este rango de operación.

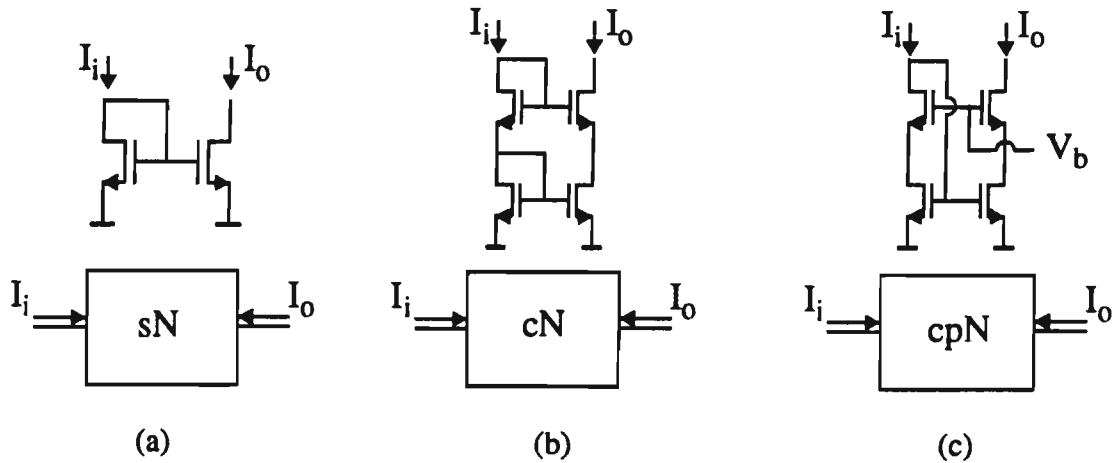


Figura 2: (a) Espejo de corriente simple. (b) Espejo cascode autopolarizado. (c) Espejo cascode con polarización externa.

En el siguiente subapartado se analizan tres estructuras de circuitos generadores de funciones de pertenencia (MFCs). Todas ellas utilizan como bloques básicos espejos cascode autopolarizados. A continuación, en el subapartado 3.2, se estudian las implementaciones de la mejor de estas estructuras con cada uno de los tres espejos de la Figura 2.

3.1. Estructuras de MFCs.

En la Figura 3-a se presenta la celda básica propuesta por Yamakawa [10] para un circuito capaz de proporcionar la característica de transferencia mostrada en la Figura 3-b. Las intensidades I_1 , I_2 e I_{ref} actúan como parámetros que definen la forma de la función lineal a tramos (función de pertenencia). Las pendientes ascendente y descendente vienen definidas por la relación geométrica entre la etapa de entrada y cada una de las etapas de salida del espejo N1.

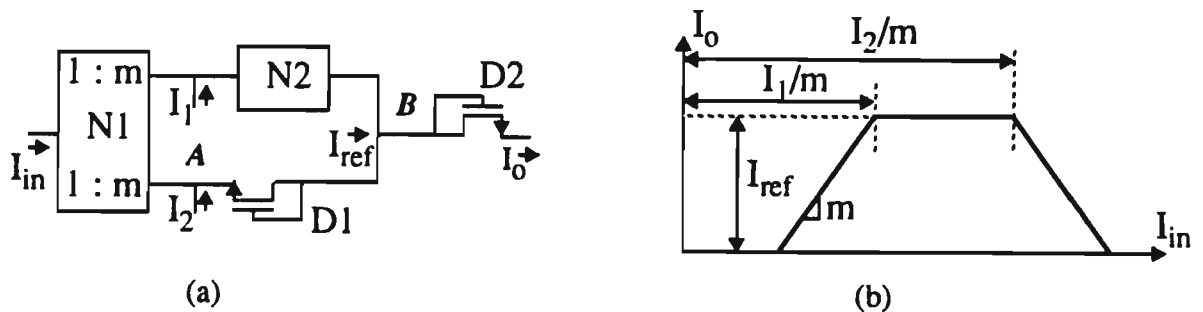


Figura 3: MFC propuesto por Yamakawa. (a) Esquema de circuito. (b) Característica de transferencia.

La realización con espejos cascode autopolarizados del circuito de la Figura 3-a presenta una serie de problemas al conectar esta celda al motor de inferencia difuso, ya que la etapa de entrada del operador MIN es otro espejo cascode NMOS (que sustituye al diodo D2 de la figura). Bajo estas circunstancias, considerando la ley cuadrática para un transistor en saturación y despreciando la conductancia de salida y el efecto sustrato, la tensión en el nudo B viene expresada por la siguiente relación:

$$V_B = 2 \left(V_{Tn} + \sqrt{\frac{2I_o}{\beta_n}} \right)$$

donde $\beta = \mu C_{ox} W/L$, μ es la movilidad de los portadores, C_{ox} es la capacidad de puerta por unidad de área, V_T es la tensión umbral y W y L son, respectivamente, la anchura y la longitud de canal del transistor, .

Si nos centramos en una situación en la que $(mI_{in}) > I_2 > I_1$, sólo conduce la rama del diodo D1, por lo que la tensión en el nudo A viene dada por:

$$V_A = V_B - V_{GS_{D1}} = 2V_{Tn} - V_{T_{D1}} + 2\sqrt{\frac{2I_o}{\beta_n}} - \sqrt{\frac{2(I_{ref} - I_o)}{\beta_{D1}}}$$

En la zona donde I_o es prácticamente cero, si consideramos que el diodo se realiza mediante un transistor NMOS:

$$V_A = V_{Tn} - \sqrt{\frac{2I_{ref}}{\beta_n}}$$

Por otro lado, para que los transistores de la etapa de salida del espejo N1 estén trabajando en saturación debe cumplirse:

$$V_{A_{min}} = 2V_{GS} - V_{Tn} = 2\sqrt{\frac{2I_{in}}{\beta_n}} + V_{Tn}$$

De acuerdo con las dos expresiones anteriores, para que el circuito opere correctamente tiene que verificarse la siguiente relación:

$$2\sqrt{\frac{2I_{in}}{\beta_n}} + V_{Tn} \leq V_{Tn} - \sqrt{\frac{2I_{ref}}{\beta_n}}$$

Esta relación sólo se cumple para $I_{in} = I_{ref} = 0$, poniendo de manifiesto que no es posible implementar esta estructura con espejos de corriente cascode autopolarizados. Por este motivo, el circuito que hemos integrado responde a la configuración indicada en la Figura 4, donde se ha sustituido el diodo D1 por dos espejos en cascada. La celda diseñada incluye asimismo los espejos necesarios para proporcionar las intensidades que actúan como parámetros de la función de pertenencia.

Al ajustar las geometrías del circuito de la Figura 4 hay que tener en cuenta que el nudo C no debe tener una tensión muy alta, con objeto de conseguir que el espejo cP3 trabaje en zona de saturación. Por el contrario, el nudo D no puede alcanzar una tensión muy baja para que cN1 no presente problemas. Cuando se utiliza esta estructura para generar funciones triangulares, conforme el rango de I_{in} aumenta, una desviación muy pequeña en el comportamiento ideal de cN1, cP2 ó cP3, puede conducir a desviaciones importantes en la altura de los triángulos y la posición de sus vértices. Este error no es constante en todas las funciones de pertenencia (si así fuera se podría evitar cambiando el valor del uno lógico) sino que aumenta de forma no lineal con I_{in} .

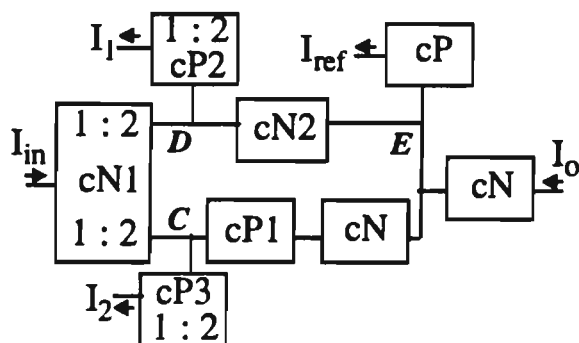


Figura 4: MFC basado en la estructura de Yamakawa.

Para el caso particular en que las funciones de pertenencia deseadas sean triangulares, la estructura que se presenta en la Figura 5-a tiene algunas ventajas, ya que minimiza el número de espejos necesarios para su realización y simplifica la elección de los parámetros que definen la forma de la función (como se muestra en la Figura 5-b). La característica de transferencia triangular se genera mediante la suma de dos tramos lineales (uno de ellos de pendiente doble) definidos por las intensidades I_1 e I_2 .

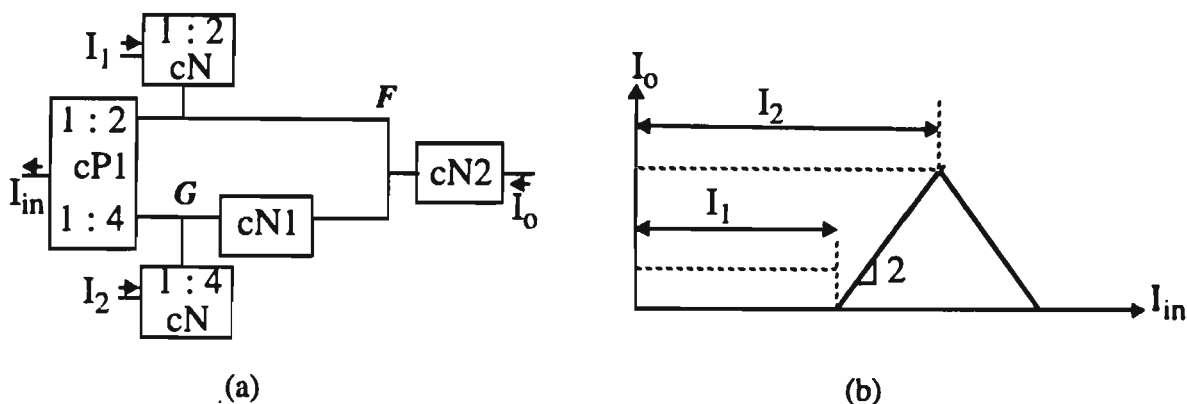


Figura 5: MFC para funciones triangulares. (a) Esquema de circuito. (b) Característica de transferencia.

Con objeto de que el espejo cP1 trabaje en zona de saturación, las tensiones en el nudo F y, sobre todo, en el nudo G deben ser ajustadas con precisión. El nudo G es tan crítico como lo era el C en la estructura anterior (Figura 4) por lo que surgen problemas similares. Tanto en esta configuración, como en la de Yamakawa, hay tres espejos (los que replican las intensidades I_{in} , I_1 e I_2) por los que circulan intensidades 2 ó 4 veces superiores al valor de I_{in} .

La tercera estructura analizada se ilustra en la Figura 6-a. En esta ocasión, los parámetros que determinan la forma de la función de pertenencia son las intensidades denominadas I_{aux} , I_{sat} e I_{ref} en la Figura 6-b. Las pendientes ascendente y descendente vienen definidas por la relación geométrica del espejo cN1. Esta estructura saca partido de la naturaleza simétrica de la función de pertenencia, utilizando en la etapa de entrada un rectificador que genera una función triangular, acotada posteriormente por el espejo cN1 [11].

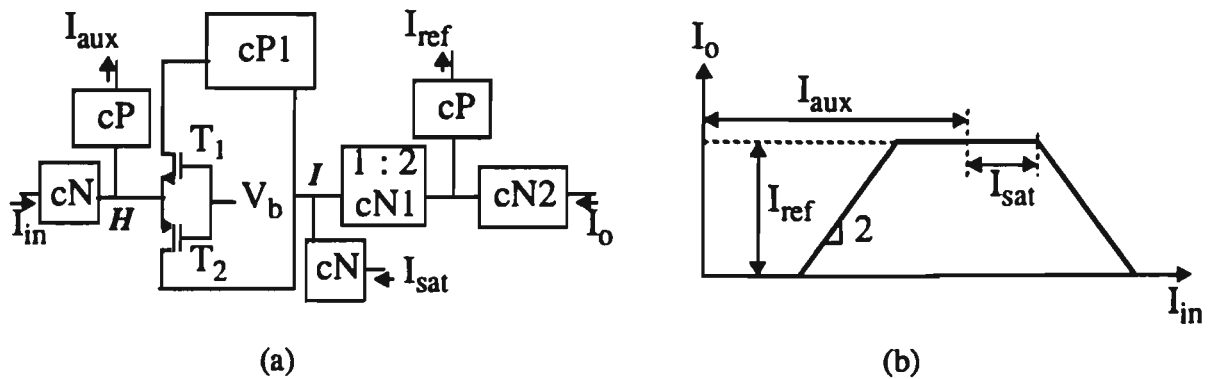


Figura 6: MFC con rectificador. (a) Esquema de circuito. (b) Característica de transferencia.

Los transistores T_1 y T_2 actúan como una llave de intensidades que sólo deja pasar a las siguientes etapas I_{in} - I_{aux} . El comportamiento lineal de estas etapas sólo debe ser asegurado en un rango de intensidades inferior a I_{ref} . Como consecuencia, a diferencia de las dos estructuras descritas anteriormente, se consigue una mayor precisión en las alturas de las funciones triangulares independientemente del valor de I_{in} . Por otra parte, sólo hay dos espejos que soportan intensidades altas (aquellos que replican I_{in} e I_{aux}) y que pueden presentar problemas de no linealidad que se traduzcan en un error en las posiciones de los vértices de los triángulos. No obstante, dado que ambos espejos comparten el nudo H , este error puede minimizarse consiguiendo una baja impedancia en dicho nudo.

Los valores de tensión que se fijan en H dependen de la forma en que se implemente la llave de intensidades. Una primera aproximación es tomar un transistor NMOS y otro PMOS con la misma polarización V_b en sus puertas, tal como se indica en la Figura 6-a. En tal caso, cuando se produce la conmutación entre T_1 y T_2 , la tensión en el nudo de entrada presenta un salto de, al menos, $V_{Tn} + |V_{Tp}|$.

Una solución para disminuir la impedancia en el nudo H consiste en polarizar por separado los transistores NMOS y PMOS. En este caso, si llamamos V_{bn} y V_{bp} a las tensiones de polarización, para asegurar que sólo uno de los transistores esté conduciendo, debe verificarse la siguiente desigualdad:

$$V_{bn} - V_{bp} < V_{Tn} + |V_{Tp}|$$

Otra posibilidad para disminuir el salto en la tensión del nudo H es introducir un inversor que proporcione una realimentación no lineal desde este nudo hacia los terminales de puerta de T_1 y T_2 [12]. Cuando el transistor PMOS conduce, la tensión en H alcanza un valor alto, por lo que interesa que V_b sea bajo. Por el contrario, cuando es el transistor NMOS el que conduce, la tensión en H desciende e interesa un valor alto de V_b . Estos son precisamente los cambios que proporciona el inversor.

Los resultados de simulación mostrados en la Figura 7 ilustran que las diferentes estructuras verifican los requerimientos estáticos que se consideraron en el diseño de los circuitos. Los tres casos que aparecen en la figura corresponden a la generación de una familia de nueve funciones de pertenencia triangulares. Las áreas ocupadas por cada uno de los circuitos y sus correspondientes consumos de potencia se comparan en los diagramas de barras de las Figuras 8-a, 8-b. (Los resultados de la tercera estructura se refieren al circuito de la Figura 6-a con T_1 y T_2 polarizados independientemente).

Al analizar el comportamiento dinámico de las tres estructuras (Figura 8-c) hay que tener en cuenta que los mayores retrasos se producen en aquellos nudos donde hay transistores que

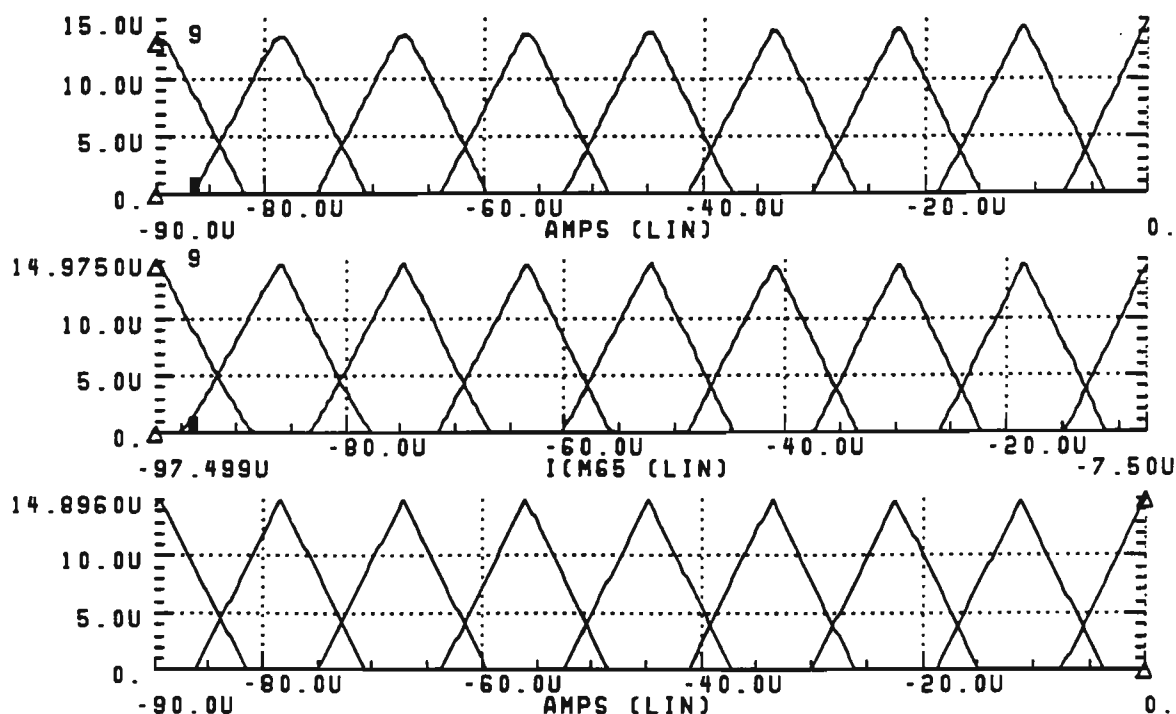


Figura 7: Resultados de simulación de los tres MFCs.

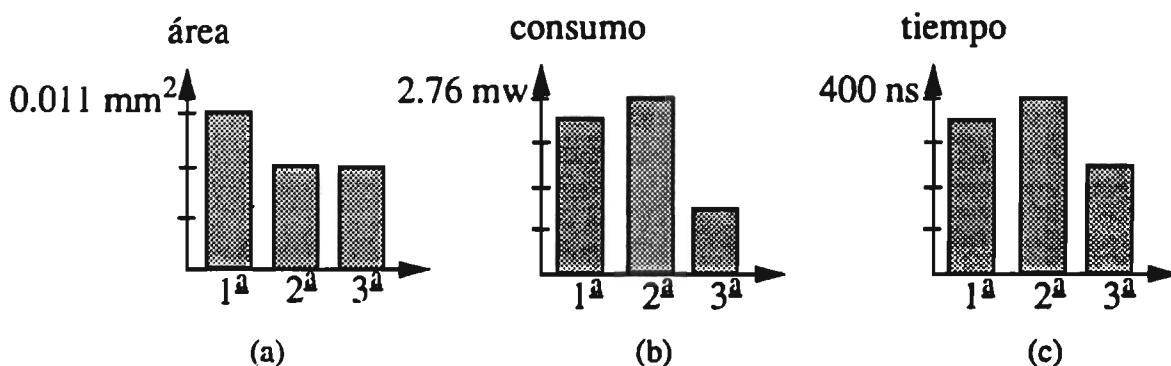


Figura 8: Comparación de los tres MFCs. (a) Área ocupada. (b) Consumo de potencia. (c) Comportamiento dinámico.

conmutan entre los estados de corte y conducción. Los nudos más críticos son aquellos en que una etapa PMOS de salida tiene que cargar a una NMOS de entrada o al revés. Esta situación se da en el nudo *C* de la estructura de Yamakawa, en los nudos *F* y *G* de la segunda estructura y en el nudo *I* de la tercera. Sin embargo, en esta última estructura es la conmutación entre T_1 y T_2 la que limita en mayor medida el rango dinámico del circuito.

Los resultados anteriores ponen de manifiesto que, para la tecnología de fabricación utilizada y para el rango de aplicación propuesto, la tercera estructura constituye la mejor elección. Esta estructura presenta una ventaja adicional. La pendiente de las funciones de pertenencia viene determinada por la relación geométrica entre los transistores de las etapas de entrada y salida de un único espejo de corriente (CN1 en la Figura 6-a), lo que facilita en gran medida la programabilidad del MFC resultante [13].

3.2. Espejos de Corriente.

El segundo aspecto considerado en nuestro estudio es la elección del tipo de espejo de corriente a utilizar en la implementación de un MFC basado en la tercera de las estructuras descritas en el apartado anterior. En la Figura 9 se muestran los esquemas de circuito de los tres espejos de corriente utilizados: espejo simple (s), espejo cascode autopolarizado (c) y espejo cascode con polarización externa (cp).

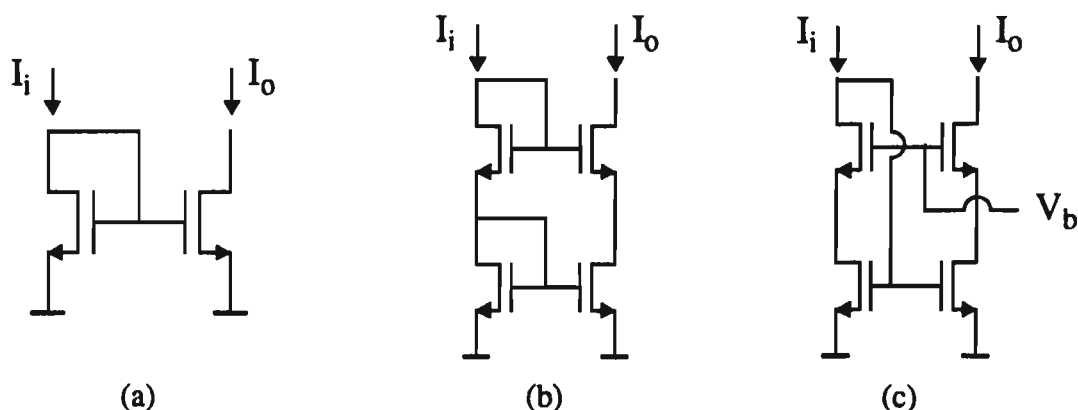


Figura 9: Diferentes espejos de corriente. (a) Simple. (b) Cascode autopolarizado. (c) Cascode con polarización externa.

El espejo de corriente simple utiliza menor número de transistores que los otros dos espejos. No obstante, el efecto de modulación de canal obliga a usar transistores con grandes longitudes para conseguir precisión en la réplica de intensidades. La utilización de estructuras cascode, por el contrario, minimiza este efecto aumentando la resistencia de salida de los circuitos resultantes.

Motivado por el uso de geometrías grandes y por la propia naturaleza de los espejos simples, el MFC implementado con este tipo de espejos es el más lento. Por el contrario, los espejos cascode, además de permitir el uso de geometrías muy pequeñas, proporcionan una mejora considerable en la velocidad de operación de los circuitos.

En los diagramas de barras de la Figura 10 se comparan las áreas de silicio y los tiempos de respuesta característicos de las diferentes opciones de diseño. Como puede apreciarse en la figura, la elección del tipo de espejo vendrá determinada por un compromiso área/velocidad entre las dos estructuras cascode.

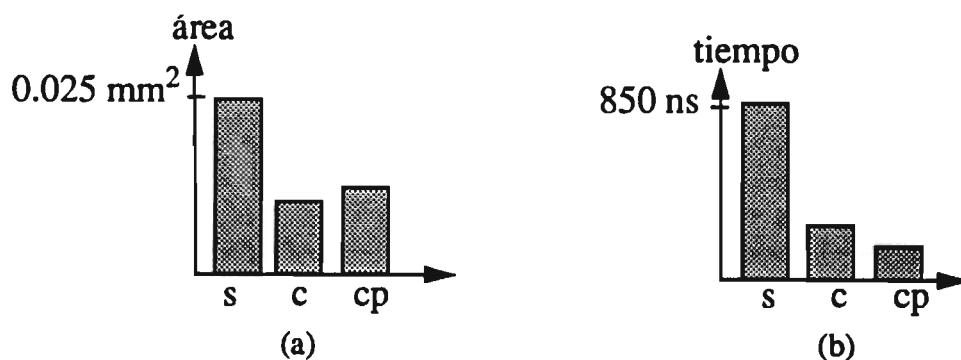


Figura 10: Utilización de diferentes espejos. (a) Área ocupada. (b) Comportamiento dinámico.

4. Operadores MIN y MAX.

La mayor parte de los sistemas de control difusos descritos en la literatura utilizan operadores mínimo y máximo para implementar el mecanismo de inferencia. Desde el punto de vista electrónico estos operadores son fácilmente realizables trabajando en modo de corriente, ya que sólo requieren operaciones de suma y diferencia acotada:

$$I_{\min}(I_1, I_2) = I_1 \ominus (I_1 \ominus I_2)$$

$$(I_2 \ominus I_1) = \begin{cases} I_2 - I_1 & \text{si } I_2 > I_1 \\ 0 & \text{en otro caso} \end{cases}$$

$$I_{\max}(I_1, I_2) = (I_2 \ominus I_1) + I_1$$

En este apartado comparamos los operadores MIN y MAX implementados de acuerdo con la estructura de Yamakawa frente a otros basados en un circuito similar a la tercera de las estructuras del apartado anterior. La diferencia entre ambas opciones es que, mientras que el circuito de Yamakawa utiliza diodos para realizar la operación de diferencia acotada, en la nueva estructura dicha operación se realiza mediante una pareja de transistores que proporciona dos caminos excluyentes al paso de la intensidad. Como en el caso de los MFCs, todos los circuitos han sido implementado en tecnología CMOS de 1.5 μ m. El objetivo de diseño fue la obtención de geometrías mínimas y buenos tiempos de respuesta para un rango de aplicación de 15 μ A. Para comparar las características de ambas estructuras se emplean espejos cascode autopolarizados. Posteriormente se analizan las consecuencias de utilizar diferentes tipos de espejos para implementar la nueva estructura.

El circuito correspondiente al operador MIN basado en la estructura de Yamakawa se muestra en la Figura 11-a. La Figura 11-b ilustra el circuito que responde a la nueva estructura. Como comentamos en el apartado 3.1, las tensiones en las puertas de los transistores T1 y T2 pueden fijarse mediante un esquema de polarización común o de polarización independiente (como aparece en la figura), o bien introduciendo una realimentación a través de un inversor.

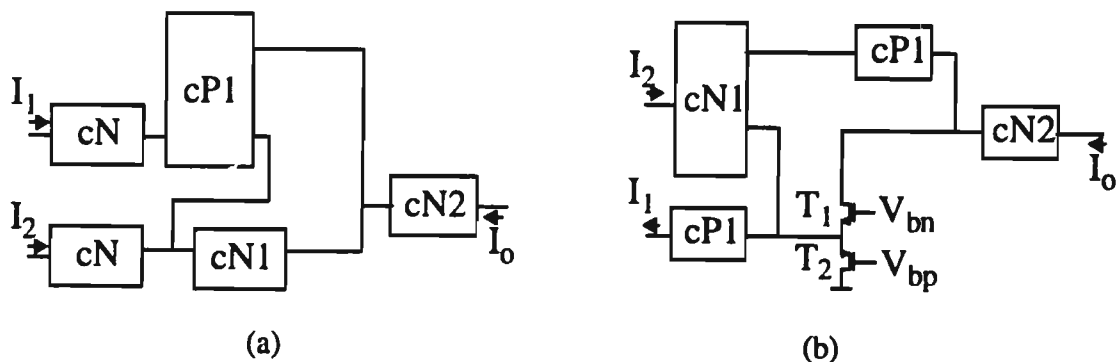


Figura 11: Operador MIN. (a) Estructura de Yamakawa. (b) Estructura propuesta.

En la Figura 12 se comparan los resultados de área y de tiempo de respuesta obtenidos de simulaciones con *layout* extraídos. La respuesta transitoria de la segunda estructura es muy sensible a la impedancia del nudo común a T1 y T2, pero con las técnicas anteriormente referidas conseguimos buenos resultados.

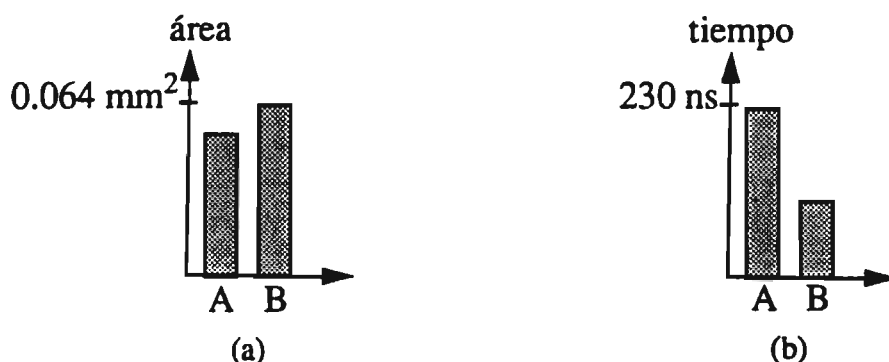


Figura 12: Comparación de operadores MIN. (a) Área ocupada. (b) Comportamiento dinámico.

Los esquemas de circuito de los operadores MAX se muestran en la Figura 13. El circuito de la Figura 13-a se basa en la estructura de Yamakawa, mientras que el de la Figura 13-b corresponde a la nueva estructura presentada en esta comunicación. Los resultados de simulación se comparan en los diagramas de la Figura 14.

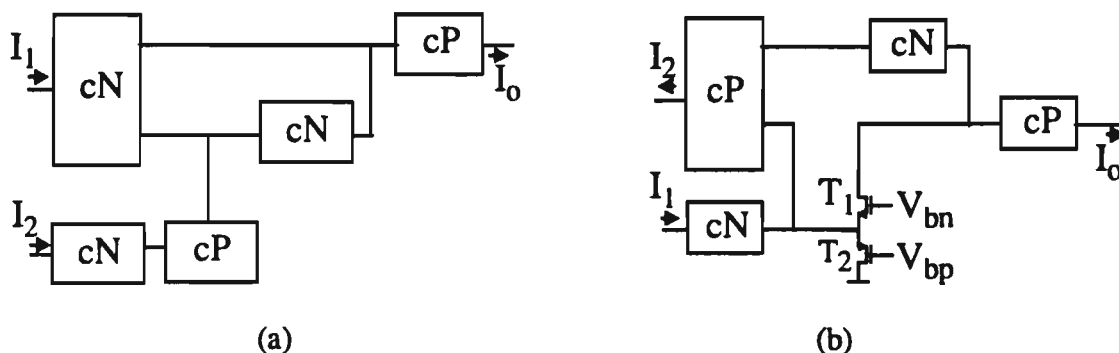


Figura 13: Operador MAX. (a) Estructura de Yamakawa. (b) Estructura propuesta.

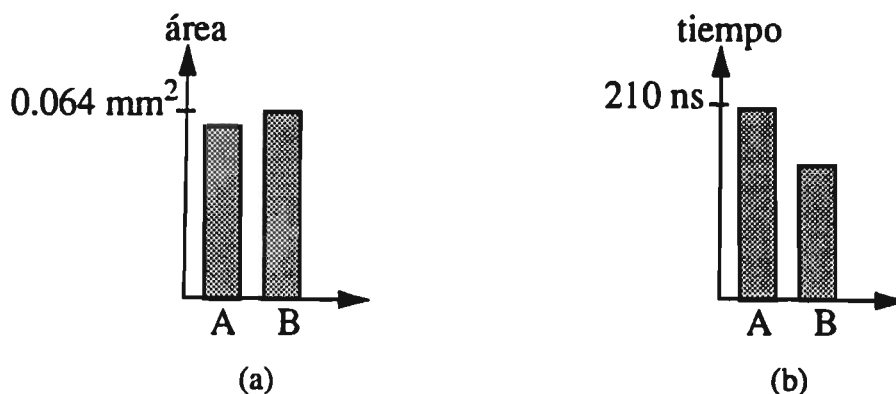


Figura 14: Comparación de operadores MAX. (a) Área ocupada. (b) Comportamiento dinámico.

Por último, teniendo en cuenta que los operadores MIN y MAX basados en nuestra estructura presentan mejor comportamiento dinámico que los de Yamakawa, hemos analizado la utilización de diferentes tipos de espejos en los circuitos que proponemos. Los resultados referentes a los espejos cascode se muestran en la Figura 15. No se incluyen los obtenidos con espejos normales porque presentan un área superior al doble de la ocupada por las implementaciones cascode y unos tiempos de respuesta un orden de magnitud por encima de los conseguidos con estos espejos.

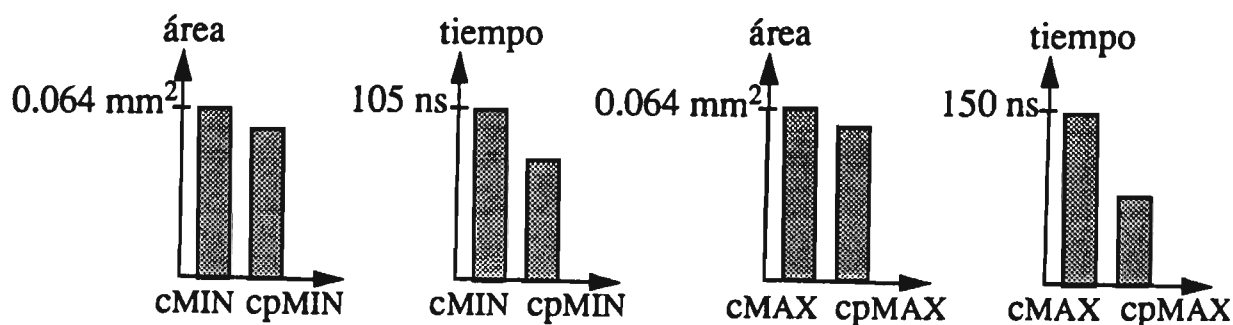


Figura 15: Comparación de operadores MIN y MAX con diferentes tipos de espejos.

5. Conclusiones

El estudio comparativo realizado en esta comunicación pone de manifiesto las ventajas de una nueva estructura de circuito para la generación de funciones de pertenencia. Estas ventajas se traducen en un menor consumo de área y potencia, así como en la reducción de los tiempos de respuesta. Cuando se considera la utilización de diferentes tipos de espejos de corriente, los espejos cascode ofrecen mejores prestaciones que los simples, tanto en tamaño (cascode autopolarizado) como en velocidad de operación de los circuitos (cascode con polarización externa). La misma filosofía de circuito utilizada en la realización de los MFCs puede aplicarse al diseño de operadores MIN y MAX con mejor respuesta temporal. La utilización de espejos cascode con polarización externa optimiza los objetivos de diseño propuestos para estos operadores.

6. Referencias

- [1] M. Togai and S. Chiu, "A Fuzzy Chip and a Fuzzy Inference Accelerator for Real-Time Approximate Reasoning", Proc. 17th IEEE Int. Symp. Multiple-Valued Logic, pp. 25-29, May 1987.
- [2] Wayne Dettloff, Kathy Yount and H. Watanabe, "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture", IEEE Solid-State Circuits, April 1990.
- [3] T. Yamakawa, T. Miki and F. Ueno, "The design and fabrication of the current mode fuzzy logic semi-custom IC in the standard CMOS IC technology", Proc. 15th IEEE Int. Symp. Multiple-Valued Logic, pp. 76-82, May 1985.
- [4] T. Yamakawa, "High-Speed Fuzzy Controller Hardware System: The Mega FIPS Machine", Elsevier Science Pub. Comp. Inc., 1988.

- [5] E. H. Mandani and S. Assilan, "An experiment in linguistic synthesis with a fuzzy logic controller", *Int. J. Man-Mach. Stud.*, Vol, 7, pp. 1-3, 1975.
- [6] J. Ramirez-Angulo, E. Sanchez-Sinencio and A. Rodríguez-Vázquez, "A piecewise-linear function approximation using current mode circuits", *ISCAS*, 1992.
- [7] T. Inoue, F. Ueno and T. Motomura, "Analysis and Design of Analog CMOS Building Blocks for Integrated Fuzzy Inference Circuits", *ISCAS*, 1991.
- [8] Z. Wang, "Analytical determination of output resistance and DC matching errors in Mos current mirrors", *IEE Proc.* Vol 137, Pt. G, No 5, Oct. 1990.
- [9] F. V. Fernández, R. Domínguez-Castro, A. Rodríguez-Vázquez y J. L. Huertas, "Modelado Dinámico Comparativo de Espejos de Corriente CMOS", *VI Congreso de Diseño de Circuitos Integrados*, Santander, 1991.
- [10] T. Yamakawa and H. Kabuo, "A Programmable Fuzzifier Integrated Circuit: Synthesis, Design and Fabrication", *Elsevier Science Pub. Comp. Inc.*, 1988.
- [11] J. L. Huertas, S. Sánchez-Solano, A. Barriga, I. Baturone, "Serial Architecture For Fuzzy Controllers: Hardware Implementation Using Analog/ Digital VLSI Techniques", 2nd. International Conference on Fuzzy Logic and Neural Networks, Iizuka, 1992.
- [12] A. Rodríguez-Vázquez, M. Delgado-Restituto, S. Espejo and J. L. Huertas, "Generation of chaotic signals using current mode techniques", 2nd. International Conference on Fuzzy Logic and Neural Networks, Iizuka, 1992.
- [13] I. Baturone, S. Sánchez-Solano, A. Barriga, J. L. Huertas, "Una implementación hardware de circuitos básicos de lógica difusa", *VII Congreso de Diseño de Circuitos Integrados*, Toledo, 1992.

REALIZACIÓN CMOS DE UN CIRCUITO DE FUNCIÓN DE PERTENENCIA PROGRAMABLE: EL BLOQUE BÁSICO DE PROCESAMIENTO BORROSO

Autores: A.Ruiz, J.Gutiérrez
Depto. Tecnología Fotónica U.P.M.
Campus de Montegancedo s/n.
28660 Boadilla del Monte (España)

Resumen: Se presenta el circuito de función de pertenencia DPFC_7. Sus características son: función de pertenencia programable, interfaz digital CMOS, procesamiento interno analógico en modo corriente e integrable sin componentes externos. Ha sido diseñado con objeto de obtener un bloque básico eficiente para procesamiento fuzzy, e incluirlo en una futura arquitectura.

Palabras Clave: Hardware fuzzy.

1. INTRODUCCIÓN

La primera cuestión a tener en cuenta cuando intentamos procesar información borrosa es su representación. En [Ruiz92] presentamos las primeras ideas sobre un método de procesamiento fuzzy diferente de los anteriores conocidos. El punto de partida de las mismas es la gran cantidad de información que se encierra en una función de pertenencia. El modo eficiente de procesarla es codificar las funciones trapezoidales en 12 bits, llamándolos CMF (Función de Pertenencia Codificada). Pero necesitamos un decodificador para interpretarlos. En [Ruiz91] se presentó una primera versión de éste circuito, el PFC_6. Ahora lo hemos mejorado, obteniendo el DPFC_7 (Circuito Borrosificador Programable Digital). Se presentan a continuación los criterios de diseño, la descripción del circuito, los resultados de simulación y la implementación de un prototipo en VLSI.

2. OBJETIVOS

- Disponer un módulo hardware que realice el método de procesamiento de [Ruiz92], decodificando una CMF y procesándola mediante la operación de borrosificación (dado un conjunto borroso y un elemento de dominio, obtener su grado de pertenencia).
- Este módulo debe ser más rápido que los del método puramente digital [Togai87], donde la función de pertenencia se almacena en un mapa de bits en memoria RAM.

- El diseño debe ser modular y tener una buena interfaz con objeto de ser la base de una arquitectura borrosa.

3. CRITERIOS DE DISEÑO

Analógico vs. Digital: Las operaciones internas se realizan fácilmente en modo corriente analógico. Sin embargo, la interfaz externa es digital para evitar la degradación de corriente y facilitar la conexión con bloques CMOS estándar.

Velocidad vs. Precisión: En esta versión la velocidad no es el principal objetivo, aunque sí se miden los tiempos de retardo. Se garantizan primero la precisión a la salida y la definición de las escalas de los rangos de muestreo.

4. INTRODUCCIÓN AL CIRCUITO

El DPFC_7 (fig.5.1) realiza un sencillo cálculo de grado de pertenencia. En la figura 4.1 se muestra una función de pertenencia trapezoidal. El eje X es el dominio de elementos y se representa en modo corriente como el intervalo $[0\mu A, 60\mu A]$. La

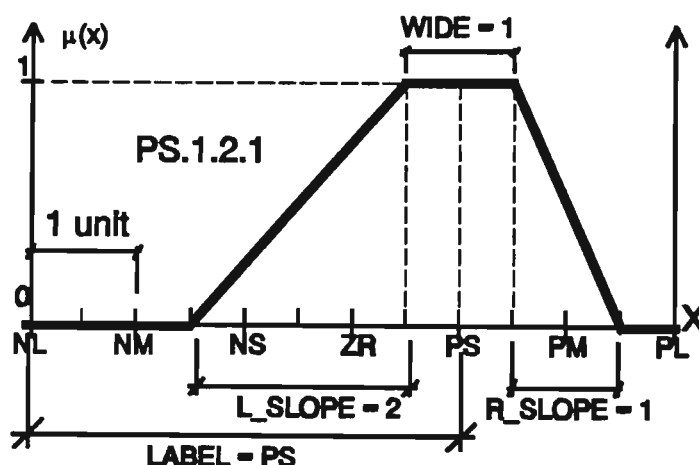


Fig. 4.1. Función de pertenencia codificada.

entrada x se muestrea cada $2\mu A$ obteniéndose 31 muestras codificadas en un bus de 5 bits ($x[0:4]$).

El intervalo borroso $[0, 1]$ en el eje Y (grado de pertenencia) se representa en modo corriente como el intervalo $[0\mu A, 49\mu A]$ (salida I_o). Se divide cada $7\mu A$, obteniéndose 8 muestras que representan grados multivaluados (bus $D[0:6]$).

La función de pertenencia se codifica mediante 4 parámetros (B, W, L y R)

cuyos valores se expresan en unidades del eje X. Cada unidad es igual a $10\mu A$ ó 5 muestras. Estos parámetros se codifican en buses de 3 bits.

5. DESCRIPCIÓN DEL CIRCUITO

El DPFC_7 se muestra en la figura 5.1. Las entradas y salidas son digitales y

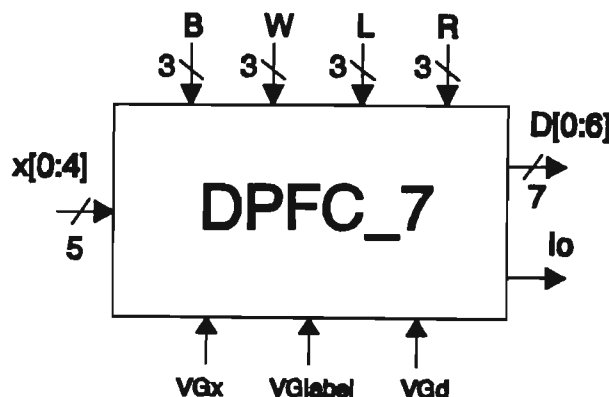


Fig. 5.1. Bloque externo de DPFC_7.

compatibles CMOS, con la excepción de I_o (analógica en modo corriente), V_{Gx} , V_{Glabel} y V_{Gd} , siendo estas 3 últimas (analógicas en modo corriente) entradas que ajustan voltajes de puerta. La característica programable se debe a la posibilidad de cambiar dinámicamente los parámetros de los buses de la CMF.

El proceso interno es analógico (figura 5.2), llevado a cabo en los bloques

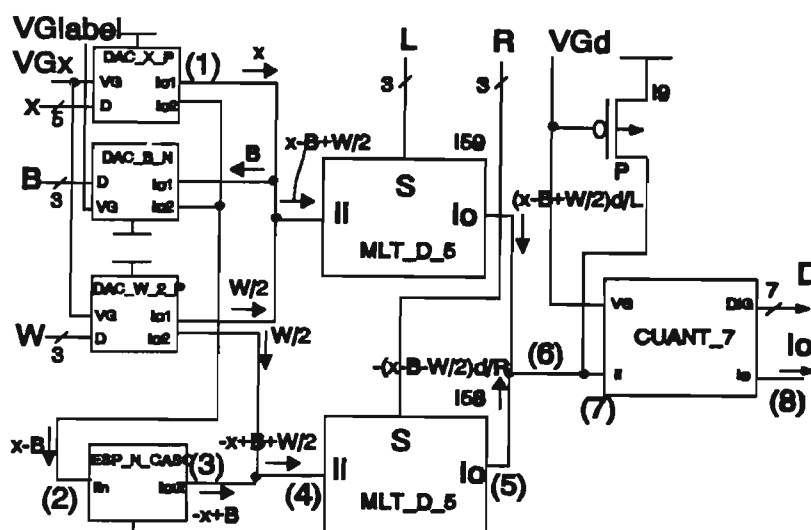


Fig. 5.2. DPFC_7

MLT_D_5. Antes y después de estos bloques están los bloques de entrada (DACs y **ESP_N_CASC**) y salida (**CUANT_7**) que realizan las conversiones analógica y digital, además de sumar o restar las corrientes correspondientes. Por lo tanto, estos bloques realizan el muestreo de los ejes X e Y.

Las etapas de procesamiento del lado derecho de la función de pertenencia se

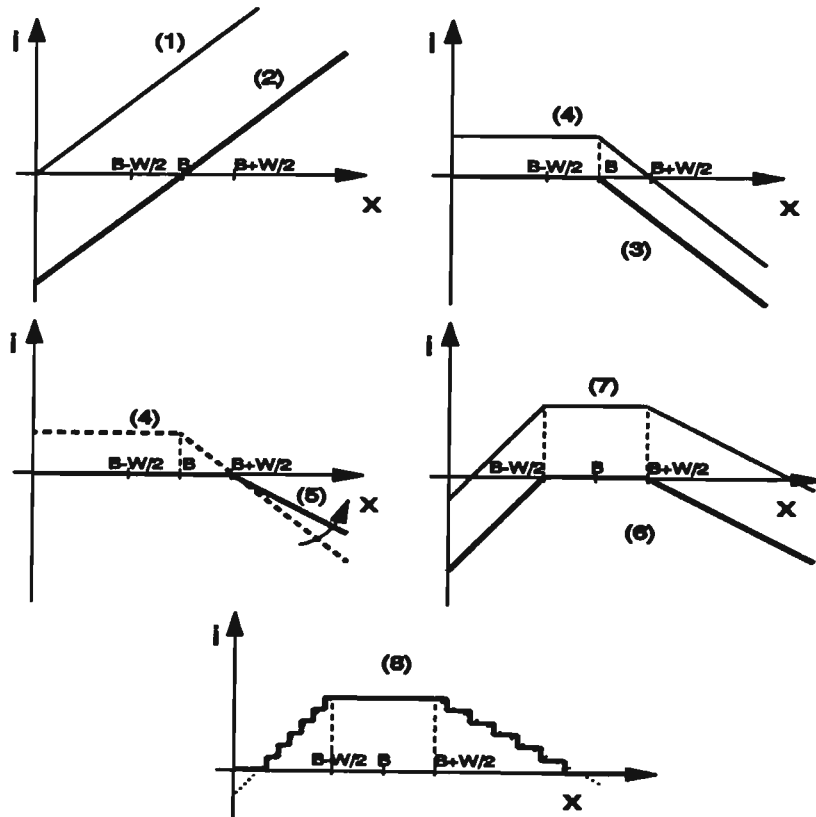


Fig. 5.3. Etapas de proceso interno de la función de pertenencia.

aprecian en la figura 5.3, donde se ha representado la corriente a través de cada rama numerada del circuito de la figura 5.2:

PASO (1): Una sencilla función $y=x$ es la salida de **DAC_X_P** al nodo siguiente, que conecta con la entrada de **ESP_N_CASC** y la salida de **DAC_B_N**.

PASO (2): **DAC_B_N** resta el valor de corriente B , resultando en un desplazamiento vertical.

PASO (3): El bloque **ESP_N_CASC** es un espejo de corriente que realiza dos operaciones: el "corte" de las corrientes negativas y el cambio del sentido de la corriente. El segmento negativo resultante tiene pendiente -1 .

PASO (4): Se añade el parámetro $W/2$ (módulo **DAC_W_2_P**).

- PASO (5):** (módulo **MLT_D_5**) Primero el corte de las corrientes positivas y después amplificación y cambio de pendiente, dependiendo del parámetro **R**.
- PASO (6):** Se añaden los dos lados (**L** y **R**) de la función. El segmento nulo de un lado no afecta al otro.
- PASO (7):** Se suma el parámetro **d**, valor en corriente del grado de pertenencia "1" (pmos **I9** como fuente de corriente).
- PASO (8):** Se cortan las corrientes negativas (**CUANT_7**) y se discretiza la corriente, pasando a un dominio multivaluado (8-valuado). Esta es la corriente que sale de **Io**. Además, se realiza una conversión analógico-digital, obteniéndose el bus digital decodificado **D**.

La corriente que realmente atraviesa las diferentes ramas es ligeramente diferente a las de la figura 5.3 porque los efectos de corte se producen en la entrada de los bloques, afectando así al paso previo.

Los tres conversores de entrada (DACs) constan de fuentes de corriente

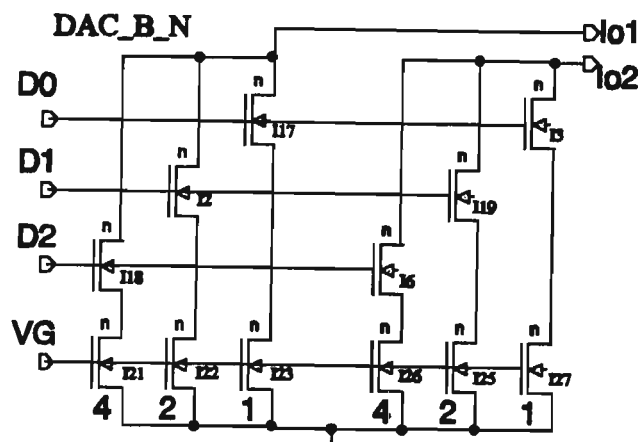


Fig. 5.4. DAC_B_N.

ponderadas, cada una de ellas controlada con un transistor de paso. En las figuras 5.4 y 5.5 se muestran DAC_B_N y DAC_W_2_P respectivamente. Son conversores DA de tipo **n** y **p**, es decir, con corrientes de salida negativa y positiva respectivamente, y mostrando una estructura simétrica de fuentes nmos y pmos. El voltaje **VG** se debe ajustar externamente para obtener la unidad de corriente que atraviesa la fuente de menor peso (numerada 1 en DAC_B_N y 1/2 en DAC_W_2_P). Las unidades de corriente son $10\mu\text{A}$ para DAC_B_N, $2\mu\text{A}$ para DAC_X_P y $5\mu\text{A}$ para DAC_W_2_P. Los pmos de estos dos últimos se han diseñado para tener la misma longitud de canal, y por lo tanto el mismo voltaje de ajuste de puerta **VG**. Las anchuras de canal son proporcionales a la corriente suministrada, dentro de un margen razonable de anchura desde $10\mu\text{m}$ a $160\mu\text{m}$. La ventaja es una entrada común **VGx** (fig. 5.2), reduciéndose el número de pines y señales de ajuste externas.

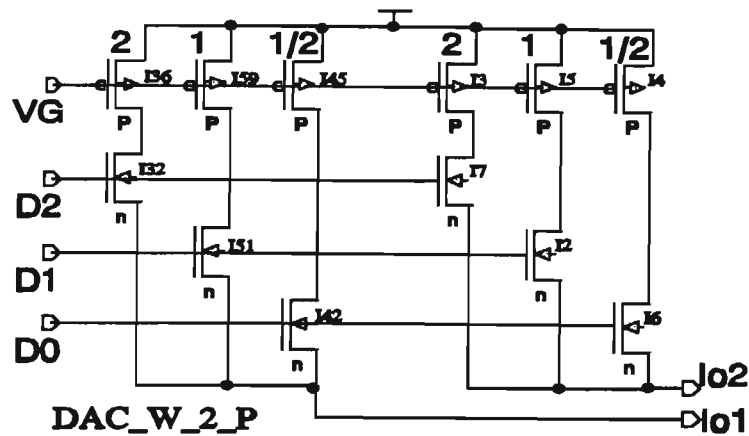


Fig. 5.5. DAC_W_2_P.

ESP_N_CASC es un espejo de corriente tipo n diseñado con 4 nmos, conectados en 2 niveles de cascada. Este diseño reduce el error de la corriente de salida.

El núcleo del DPFC_7 son los multiplicadores MLT_D_5 (fig. 5.6). Su función es cambiar las pendientes de la función de pertenencia al valor que indiquen los

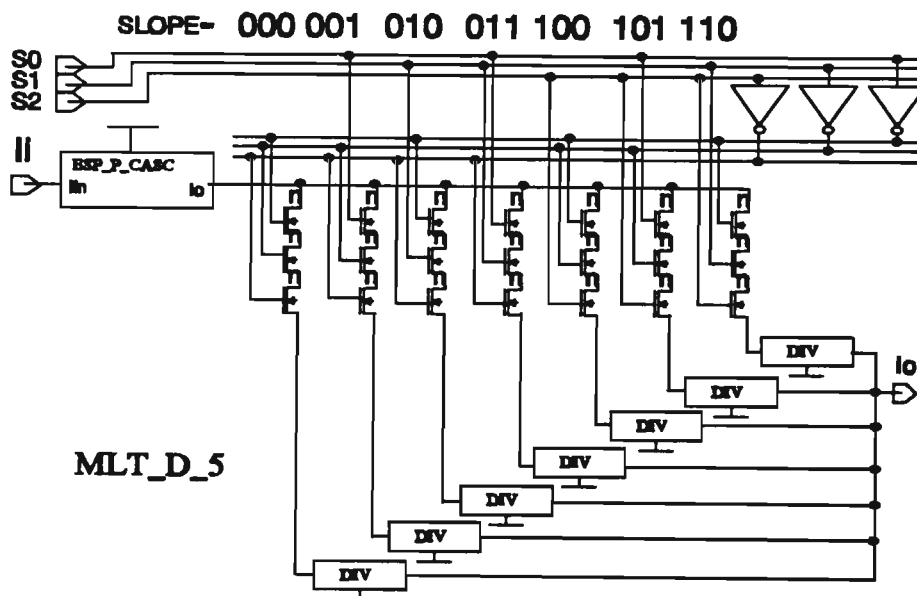


Fig. 5.6. MLT_D_5.

parámetros L y R. Esto se hace amplificando la corriente en los módulos DIV. Son espejos de corriente con ganancia, debida a la proporción entre los anchos de canal de los nmos de salida y los de entrada. Cada DIV representa una ganancia diferente, y en consecuencia tenemos una multiplicación discreta. Hay 7 módulos DIV, relativos a los 7 posibles valores de los parámetros L y R. El módulo DIV correcto se selecciona mediante un multiplexor hecho con una matriz de nmos de paso. Cada columna se compone de 3 nmos de paso en serie que representan un canal del multiplexor, pues sus puertas se conectan a la correspondiente combinación binaria

del bus S, conectado externamente al parámetro de pendiente L o R. Ambos multiplicadores son idénticos, simplificando el diseño del layout de la celda. El espejo de corriente tipo p ESP_P_CASC a la entrada se necesita para el corte de corriente del paso (5) de la figura 5.3.

La última etapa es CUANT_7 (figura 5.7), cuantificador y conversor AD. La cuantificación se lleva a cabo haciendo 7 copias de la corriente de entrada, cuyo

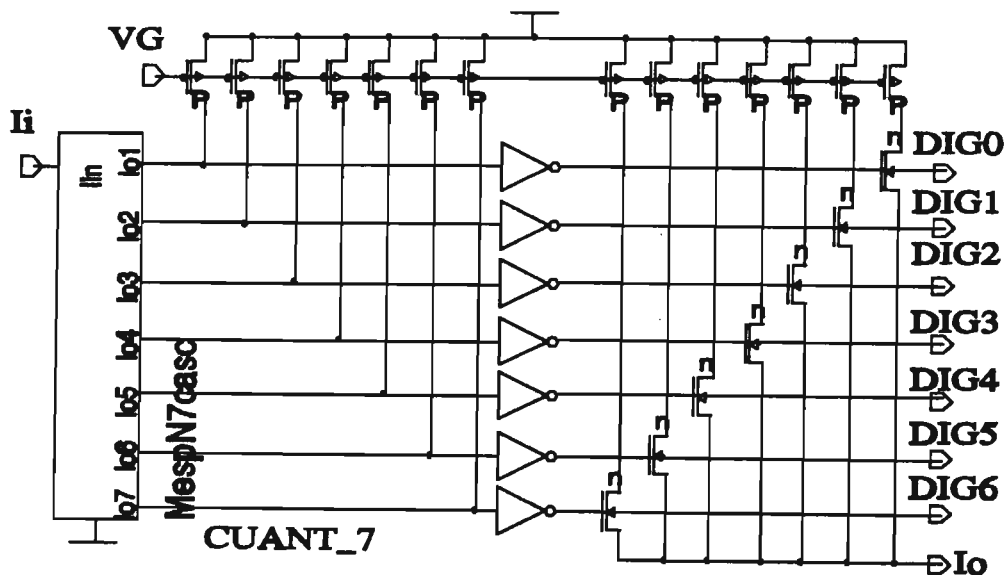


Fig. 5.7. CUANT_7.

aspecto es el del paso (7) de la figura 5.3. Esto lo realiza *MespN7casc*, un espejo de corriente tipo n en cascada de 7 salidas, que además corta las corrientes de entrada negativas, obteniendo de este modo los segmentos laterales de grado de pertenencia 0.

Cada una de estas 7 copias se "compara" con una corriente umbral entre los dos niveles de corriente de referencia adyacentes del código de 7 valores. Si los valores de referencia son I_u , $2I_u$, $3I_u$,..., $7I_u$ las corrientes umbrales son $0.5I_u$, $1.5I_u$,..., $6.5I_u$, siendo I_u igual a $7\mu A$. Esta comparación se realiza conectando cada salida del espejo a una fuente de corriente (pmos) que suministra el valor umbral correspondiente. El nodo se conecta también a la entrada de un inversor lógico. Veamos como se comporta uno de estos nodos: el punto de trabajo se determina por la menor de las corrientes entre la de entrada y la umbral, pues ambas se generan con fuentes de corriente MOS. Si la corriente de entrada es menor que la umbral, el nmos de salida de *MespN7casc* obliga a la fuente pmos a estar en región resistiva, el único punto donde la curva característica del pmos puede suministrar la misma corriente (la corriente de entrada al inversor es despreciable). Esto hace que el voltaje del nodo sea alto, cercano a v_{dd} , donde tiene lugar la región resistiva del pmos. El valor alto del voltaje a la entrada del inversor provoca un "0" lógico en su salida. Un proceso similar ocurre cuando la corriente de entrada es mayor que la umbral, pero en este caso el nmos es el que está en región resistiva. El voltaje del nodo es bajo, y la salida

del inversor un "1" lógico.

Las salidas de los inversores son el bus de salida DIG[6:0]. Por ello es un bus digital decodificado, con 8 niveles de grado de pertenencia. Por ejemplo, el grado 4 de pertenencia se representa con los valores DIG[6:0]=0011111.

La etapa final de CUANT_7 es la conversión a modo corriente. Esto se realiza sumando 7 corrientes de referencia idénticas ($I_u=7\mu A$), cada una activada con un transistor de paso controlado por el correspondiente bit del bus. El grado de pertenencia máximo se representa con una corriente de $49\mu A$.

6 RESULTADOS DE SIMULACIÓN

Como la salida de DPFC_7 es un bus digital, los resultados se miden como el porcentaje de grados de salida correctos relativos a los valores teóricos. Estos se calculan redondeando el valor analógico al valor de referencia más cercano. El error del circuito se debe a la desigualdad geométrica entre mosfets de la oblea, idénticos en el diseño (esto no se mide en la simulación) y la pendiente no nula de la corriente de drenador en zona de saturación, minimizada al alargar el canal del mosfet.

Hay 7 flancos diferentes de función de pertenencia (parámetros L ó R de 0 a 6). Cada unidad del eje X se divide en 5 muestras, por lo que tenemos 226 muestras a simular. Se han simulado 88 muestras (39% del total), incluyendo los flancos más representativos. El número de grados incorrectos es 10 (11'4%), siendo 9 de ellas errores de ± 1 nivel de pertenencia y 1 un voltaje CMOS intermedio. Este último no es un error, pues se origina cuando un grado analógico correcto es igual al nivel umbral. El error de los 9 grados incorrectos se ha medido con el error relativo respecto a la escala total (FS), que compara el valor de salida incorrecto con el grado continuo teórico. El error medio es 9'4%FS y el error máximo es 12'9%FS.

El retardo de propagación en conmutación (desde cambio en x, B, W, L o R a la salida) está alrededor de las decenas de nanosegundos, pero en unas pocas muestras el retardo aumenta notablemente, llegando a 500ns, 800ns y 1450ns en casos específicos. Este efecto se debe a corrientes pequeñas que atraviesan determinados nodos, que no pueden evacuar rápidamente las cargas acumuladas en los mosfets.

El DPFC_7 se ha diseñado a nivel de layout (fig 6.1) para su integración. El proceso tecnológico es CMOS $1'5\mu m$ con Doble Metal. El área ocupada es $6mm^2$ incluyendo anillo de pads y $1'1mm^2$ sin él. Esta última, como se puede apreciar en la figura, se puede reducir en aproximadamente un 50%.

Hay 218 mosfets en el diseño, la tensión de alimentación es 5V y el anillo de pads se compone de pads tanto analógicos como digitales, con líneas de alimentación y masa independientes. No se dispone de resultados de test físico ya que en el momento de escribir este documento el chip se estaba fabricando.

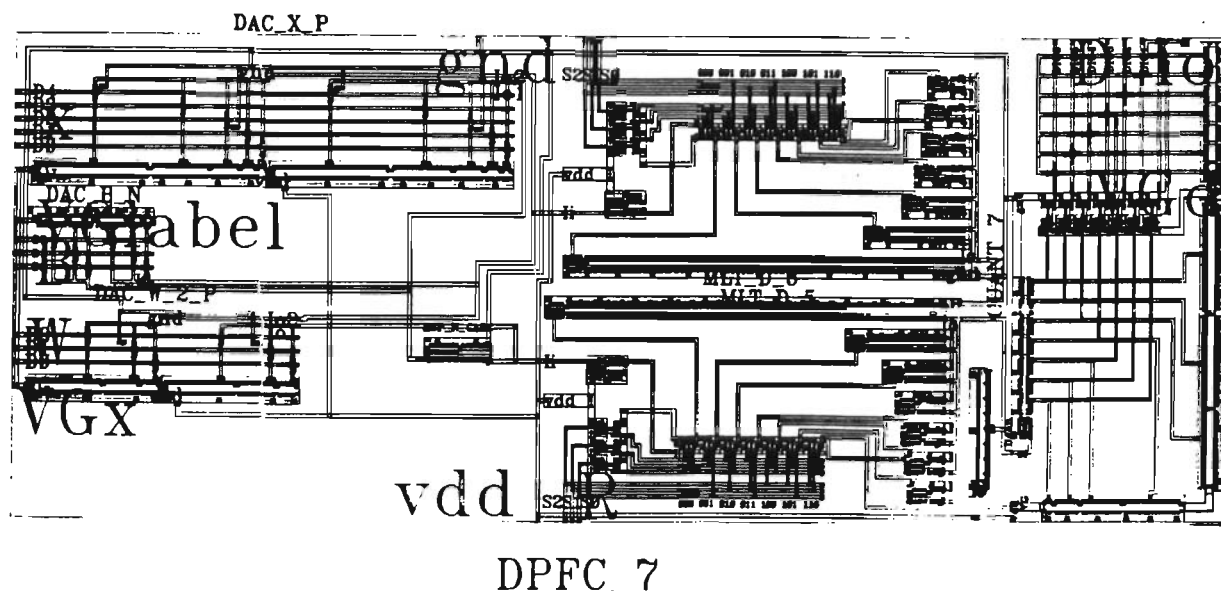


Fig 6.1. Layout de DPFC_7.

7 CONCLUSIONES

Debido a estar a la espera de la fabricación del chip, la conclusión en este momento es haber diseñado un módulo que encaja con la descripción funcional. Los resultados de simulación indican que la etapa de salida se debe mejorar para obtener niveles CMOS correctos. La velocidad se estudiará en la próxima versión.

REFERENCIAS

[Ruiz91] A.Ruiz, J.Gutiérrez, "Sistema de codificación de datos fuzzy para su eficaz procesamiento en hardware", I Congreso Español sobre Tecnologías y Lógica Fuzzy, Granada, junio 1991.

[Ruiz92] A.Ruiz, J.Gutiérrez, "An efficient fuzzy data codified set for general purpose and its implementation" report FIM 64.1/TF/92. Facultad de Informática, Universidad Politécnica de Madrid, Spain, 1992.

[Togai87] M.Togai, S.Chui, "A Fuzzy Logic Accelerator and a Programming Environment for Real-Time Fuzzy Control", Preprints of 2nd. IFSA Congress, Tokyo, July, 1987.

[Yamakawa88] T.Yamakawa, H.Kabuo, "A programmable fuzzyfier integrated circuit - synthesis, design, and fabrication-", Information Sciences 45, 75-112, Elsevier Sci. Publ. Co. Inc., NY, 1988.

SOLUCIONES SISTOLICAS EN CONTROL BORROSO

A. Bugarín, S. Barro y R. Ruiz*

Departamento de Electrónica e Computación. Facultade de Física. Universidade de Santiago de Compostela. 15706 Santiago de Compostela.

* Departamento de Ingeniería Electromecánica. E.T.S.I.I. de Cartagena. Universidad de Murcia. 30203 Cartagena (Murcia).

Resumen: En el presente trabajo se describe una arquitectura sistólica para la evaluación de Sistemas de Control Borroso. Para ello se hace uso del homomorfismo existente entre ciertas estrategias de evaluación de estos sistemas y el producto clásico Matriz*Vector. Esto permite la utilización de técnicas de particionamiento y proyección sobre arquitecturas sistólicas descritas para este tipo de producto (en nuestro caso la DBT: Dense to Band matrix transformation by Triangular-block partitioning).

Palabras Clave: Control Borroso, Arquitectura sistólica, Particionamiento DBT.

1. Introducción.

Los sistemas de control basados en lógica borrosa [Zadeh, 1965; Zadeh, 1971] intentan modelar el conocimiento y experiencia acumulados por el experto humano sobre un proceso concreto, y no el proceso en sí. Por tanto, este modo de abordar el diseño de sistemas de control resulta particularmente adecuado en aplicaciones complejas, para las que no se dispone de modelos matemáticos bien establecidos. El diseño de la base de reglas que describe a un controlador borroso sigue fundamentalmente dos pautas: la proyección de la experiencia humana a través de la traducción del lenguaje natural o bien un proceso más metódico de análisis del modo de operación del experto (especialmente interesante cuando el operador no tiene en mente modelos bien definidos sobre el proceso de control). En ambos casos, el conocimiento se estructura en una base de conocimiento (BC) formada por reglas del tipo IF-THEN, conocidas como Sentencias Condicionales Borrosas (SCBs), las cuales establecen una serie de aseveraciones fundamentadas en valores de determinadas variables lingüísticas. La estructura de dichas reglas queda caracterizada por la clara división existente entre la parte antecedente (en la que únicamente intervienen variables de estado) y la parte consecuente (en la que únicamente intervienen variables de control), a diferencia de lo que sucede en los que podríamos denominar sistemas con reglas encadenadas, donde una misma variable puede encontrarse en la parte antecedente de una(s) regla(s) y en la parte consecuente de otra(s).

La elección de las variables (de estado y de control) implicadas en el diseño de la BC condiciona fuertemente su estructura (número de reglas, número de antecedentes, ...), siendo imprescindible, además, realizar una elección adecuada que permita un elevado grado de mimetismo entre la BC propuesta y el conocimiento del controlador humano.

2. Metodología.

Las reglas que conforman la BC se encuentran relacionadas entre sí por la necesidad de ser consideradas globalmente todas y cada una de ellas y, por tanto, de agrupar en una única distribución de posibilidad final todas las distribuciones de posibilidad inferidas para cada una de las variables de control. Este vínculo puede representarse simbólicamente por medio del conectivo "*also*", que opera sobre el conjunto de reglas de la BC de la siguiente manera:

$$BC = \text{ALSO}(R^1, R^2, \dots, R^r, \dots, R^R) \quad R^r: \text{Regla } r\text{-ésima} \quad (1)$$

Una formulación genérica de la BC debe consentir que estas reglas sean del tipo Múltiple-Antecedente Múltiple-Consecuente, donde las variables asociadas a la parte antecedente de las reglas son idénticas para cada regla, y lo mismo ocurre con aquellas asociadas a la parte consecuente. La formulación más general de una regla permitiría el encadenamiento de antecedentes y consecuentes mediante conectivos conjunción y disyunción. Sin embargo, es fácil eliminar las referencias explícitas a los operadores disyunción mediante una simple reescritura de las reglas de la BC. De este modo, el único operador de encadenamiento, tanto de los distintos antecedentes como de los distintos consecuentes, es el operador conjunción, el cual suele representarse simbólicamente con la expresión AND. Por tanto, la formulación de la regla r -ésima de una BC resulta como sigue

$$R^r: \text{ IF } X_1 \text{ IS } A_1^r \text{ AND } X_M \text{ IS } A_M^r \text{ THEN } Y_1 \text{ IS } B_1^r \text{ AND } \dots \text{ AND } Y_N \text{ IS } B_N^r \quad (2) \\ \text{ALSO } \dots$$

Una simple reordenación de los consecuentes (composición de los operadores AND y ALSO) permite reestructurar la BC como:

$$R_1^r: \text{ IF } X_1 \text{ IS } A_1^r \text{ AND } \dots \text{ AND } X_M \text{ IS } A_M^r \text{ THEN } Y_1 \text{ IS } B_1^r \\ \text{ALSO } \dots \text{ ALSO} \quad (3) \\ R_N^r: \text{ IF } X_1 \text{ IS } A_1^r \text{ AND } \dots \text{ AND } X_M \text{ IS } A_M^r \text{ THEN } Y_N \text{ IS } B_N^r$$

de modo que siempre es posible reformular una BC Múltiple-Antecedente Múltiple-Consecuente con reglas del tipo Múltiple-Antecedente único-consecuente. Esto nos permite caracterizar genéricamente la regla r -ésima de una BC de un Sistema de Producción Borroso orientado al control de procesos como

$$R_n^r: \text{ IF } X_1 \text{ IS } A_1^r \text{ AND } \dots \text{ AND } X_M \text{ IS } A_M^r \text{ THEN } Y_n \text{ IS } B_n^r \\ \text{ALSO } \dots \quad (4)$$

$$n=1, \dots, N \quad r=1, \dots, R$$

donde M representa el número de antecedentes de cada regla,
 N el número de consecuentes,

A_m^r , $m=1,\dots,M$, son valores lingüísticos de las variables lingüísticas de estado X_m , y B_n^r es un valor lingüístico de la variable lingüística de control Y_n .

Denotaremos por

$$\tilde{B}_n^r = \left\{ \tilde{b}_{n,j_n}^r \right\}, \quad j_n=1,\dots,J_n, \quad n=1,\dots,N, \quad (5)$$

la distribución de posibilidad inferida para la variable Y_n , tras la aplicación de la regla R_n^r sobre las distribuciones de posibilidad observadas para cada una de las variables de control, representadas por:

$$\tilde{A}_m = \left\{ \tilde{a}_{m,i_m} \right\}, \quad i_m=1,\dots,I_m, \quad m=1,\dots,M. \quad (6)$$

La Regla Composicional de Inferencia [Zadeh, 1973] interpreta el proceso de evaluación de una regla como la composición de dos relaciones: una relación de conjunción definida sobre el espacio de antecedentes inducida por la observación simultánea de un conjunto de valores para las variables de estado -que denotaremos por $S_{(\tilde{A}_1,\dots,\tilde{A}_M)}$ - y la relación de implicación, inducida por la propia existencia de la regla, que liga antecedentes y consecuentes y que, por comodidad, denotaremos, igual que la regla correspondiente, por R_n^r . La composición de ambas relaciones puede expresarse a través del operador SUP-*, de modo que

$$\tilde{b}_{n,j_n}^r \triangleq \bigvee_{U_1 \times \dots \times U_M} S_{(\tilde{A}_1,\dots,\tilde{A}_M)}(i_1,\dots,i_M) * R_n^r(i_1,\dots,i_M,j_n) \quad (7)$$

donde $a \vee b = \text{SUP}(a,b)$.

Consideramos que la relación asociada a los "valores observados" para las variables de estado se define como:

$$\begin{aligned} S_{(\tilde{A}_1,\dots,\tilde{A}_M)}: U_1 \times \dots \times U_M &\rightarrow [0,1] \\ i_1,\dots,i_M &\rightarrow S_{(\tilde{A}_1,\dots,\tilde{A}_M)}(i_1,\dots,i_M) \triangleq \tilde{a}_{1,i_1} \wedge \dots \wedge \tilde{a}_{M,i_M} \end{aligned} \quad (8)$$

considerando como operador conjunción que define la relación de conjunción la t-norma mínimo ($\wedge$).

Por su parte, la definición de la relación de implicación será

$$\begin{aligned} R_n^r: U_1 \times \dots \times U_M \times J_n &\rightarrow [0,1] \\ i_1,\dots,i_M,j_n &\rightarrow R_n^r(i_1,\dots,i_M,j_n) \end{aligned} \quad (9)$$

Puesto que la t-norma escogida en la mayoría de las arquitecturas descritas en la literatura es el operador mínimo ($\wedge$) -Regla Composicional de Inferencia SUP-mín-, tenemos que la expresión (7) puede transformarse en

$$\tilde{b}_{n,j_n}^r = \bigvee_{U_1 \times \dots \times U_M} S_{(\tilde{A}_1,\dots,\tilde{A}_M)}(i_1,\dots,i_M) \wedge R_n^r(i_1,\dots,i_M,j_n) \quad (10)$$

y, utilizando las expresiones (8) y (9), obtenemos la ecuación que describe la distribución de posibilidad inferida punto a punto:

$$\tilde{b}_{n,j_n}^r = \bigvee_{i_1=1}^{I_1} \dots \bigvee_{i_M=1}^{I_M} \left[\tilde{a}_{1,i_1} \wedge \dots \wedge \tilde{a}_{M,i_M} \right] \wedge R_n^r(i_1,\dots,i_M,j_n) \quad (11)$$

Por otra parte, la implicación borrosa empleada en la práctica totalidad de las soluciones es la de Mamdani [1977], definida como

$$R_n^r(i_1, \dots, i_M, j_n) = a_{1,i_1}^r \wedge \dots \wedge a_{M,i_M}^r \wedge b_{n,j_n}^r \quad (12)$$

Por tanto, tendremos,

$$\tilde{b}_{n,j_n}^r = \bigvee_{i_1=1}^{I_1} \dots \bigvee_{i_M=1}^{I_M} [\tilde{a}_{1,i_1} \wedge \dots \wedge \tilde{a}_{M,i_M}] \wedge [a_{1,i_1}^r \wedge \dots \wedge a_{M,i_M}^r \wedge b_{n,j_n}^r] \quad (13)$$

Dado que una BC consta de R reglas, y que la distribución de posibilidad inferida para cada variable de control se obtiene como la unión de las distribuciones parciales obtenidas tras la aplicación de cada una de las reglas, tendremos,

$$\tilde{b}_{n,j_n} = \bigvee_{r=1}^R \bigvee_{i_1=1}^{I_1} \dots \bigvee_{i_M=1}^{I_M} [\tilde{a}_{1,i_1} \wedge \dots \wedge \tilde{a}_{M,i_M} \wedge a_{1,i_1}^r \wedge \dots \wedge a_{M,i_M}^r \wedge b_{n,j_n}^r] \quad (14)$$

donde

$$\tilde{B}_n = \{\tilde{b}_{n,j_n}\} \quad j_n = 1, \dots, J_n, \quad n = 1, \dots, N \quad (15)$$

representa la distribución obtenida finalmente para la variable Y_n tras la aplicación de la base de reglas.

A partir de la expresión (14) se derivan dos metodologías de evaluación claramente distintas, cuyo ámbito de aplicación discutiremos brevemente. Una de ellas, que denominaremos "clásica", ya que es asumida mayoritariamente [Manzoul, 1987; Togai y Watanabe, 1986; Watanabe y col., 1990], consiste en la obtención de un grado de compatibilidad para cada variable de estado entre su distribución de posibilidad observada y la distribución asociada a los valores lingüísticos que se describen para dicha variable en cada regla. De esta forma, se reduce a un único valor numérico el proceso de evaluación de cada proposición antecedente. Mediante una posterior integración de todos los grados de compatibilidad correspondientes a las distintas premisas que componen la parte antecedente de una misma regla, se obtiene un grado de compatibilidad global, que se asume como grado de cumplimiento o verificación de la parte antecedente de dicha regla. Dicho grado de compatibilidad global modulará la distribución asociada a la variable lingüística reflejada en la parte consecuente de la regla, obteniéndose la distribución inferida por dicha regla para la variable de control correspondiente. El proceso de evaluación finaliza tras la evaluación de todas las reglas, lo cual implica la necesidad de componer las distribuciones obtenidas para la misma variable en una única distribución final (proceso asociado al conectivo *also* que liga todas las reglas que conforman la base).

La segunda metodología de evaluación que hemos apuntado [Manzoul y Serrate, 1988; Ruiz y col., 1991; Manzoul y Jayabharathi, 1992], consiste, básicamente, en la "construcción" de una **Matriz de Implicación Global (MIG)** que liga la parte antecedente con la parte consecuente de las reglas de acuerdo con la relación de implicación asumida. Para una aplicación de control concreta, una vez obtenida esta matriz, cuyo cálculo se realiza en tiempo diferido, el proceso de evaluación de la BC presenta, como veremos, una gran similitud con la operación clásica **Matriz*Vector**.

Retomando las expresiones (12) y (14), tenemos

$$\tilde{b}_{n,j_n} = \bigvee_{r=1}^R \bigvee_{i_1=1}^{I_1} \dots \bigvee_{i_M=1}^{I_M} [\tilde{a}_{1,i_1} \wedge \dots \wedge \tilde{a}_{M,i_M} \wedge R_n^r(i_1, \dots, i_M, j_n)] \quad (16)$$

Intercambiando operadores,

$$\tilde{b}_{a,j_n} = \bigvee_{i_1=1}^{I_1} \dots \bigvee_{i_M=1}^{I_M} \bigvee_{r=1}^R [\tilde{a}_{i_1,i_1} \wedge \dots \wedge \tilde{a}_{i_M,i_M} \wedge R_n^r(i_1, \dots, i_M, j_n)] \quad (17)$$

y, aplicando la propiedad distributiva entre los operadores $\wedge$ y $\vee^1$, tenemos, finalmente,

$$\tilde{b}_{a,j_n} = \bigvee_{i_1=1}^{I_1} \dots \bigvee_{i_M=1}^{I_M} [\tilde{a}_{i_1,i_1} \wedge \dots \wedge \tilde{a}_{i_M,i_M} \wedge R_n(i_1, \dots, i_M, j_n)] \quad (18)$$

donde la expresión

$$R_n(i_1, \dots, i_M, j_n) \triangleq \bigvee_{r=1}^R R_n^r(i_1, \dots, i_M, j_n) \quad (19)$$

define analíticamente la Matriz de Implicación Global R_n .

En una notación más compacta, podemos poner

$$\tilde{b}_{a,j_n} = \bigvee_{\substack{i_1=1 \\ \vdots \\ i_M=1}}^{I_1 \dots I_M} [\tilde{a}_{i_1,i_1} \wedge \dots \wedge \tilde{a}_{i_M,i_M} \wedge R_n(i_1, \dots, i_M, j_n)] \quad (20)$$

Del análisis de la expresión (20) se deduce una diferencia significativa frente a la otra metodología de evaluación como es la total independencia del tamaño de la MIG respecto del número total de reglas de la BC. Esta característica, junto a la posibilidad de obtención de la MIG en tiempo diferido, dotan a esta estrategia de evaluación de unas características que hacen que, en determinadas situaciones, resulte computacionalmente más atractiva.

Dado que la dimensión de la MIG es $M+1$ (expresión (19)), donde M es el número de antecedentes de cada regla, una implementación directa de la expresión (18) requeriría arquitecturas multidimensionales (hipercubo, sistólicos multidimensionales, ...). Sin embargo, pensando en una posible integración VLSI de la solución, debe forzarse una reducción de la dimensionalidad de dicha expresión que permita su proyección sobre redes de procesadores bidimensionales. La necesaria reducción de la dimensionalidad de la expresión (18) para conseguir una total analogía con la expresión clásica del producto Matriz*Vector, debe realizarse a través de la transformación de la MIG $M+1$ -dimensional en una matriz bidimensional equivalente, y la transformación del conjunto de valores observados en un vector unidimensional equivalente. La realización de ambas transformaciones implicará establecer una aplicación entre el espacio de los subíndices de la MIG y otro espacio de menor dimensión (2-D en el caso de la matriz y 1-D en el caso del vector).

Definimos la MIG 2-D equivalente a R_n como

$$Dn_{j,s} = R_n(i_1, \dots, i_M, j_n) \quad (21)$$

donde a cada coeficiente s le corresponde un vector de índices establecido mediante una aplicación F definida como:

$$\begin{aligned} F: \{1, \dots, S\} &\rightarrow \{1, \dots, I_1\} \times \dots \times \{1, \dots, I_M\} \\ s &\rightarrow F(s) = (i_1, \dots, i_M) \end{aligned} \quad (22)$$

con

¹ Esta propiedad puede formularse como $\bigvee_{k=1}^K a_k \wedge b = b \wedge \bigvee_{k=1}^K a_k$.

$$S = \prod_{m=1}^M I_m \quad (23)$$

La aplicación F reordena los $S \cdot J_n$ elementos de la matriz $(M+1)$ -dimensional en elementos de una matriz 2-D formada por S filas y J_n columnas.

La correspondencia entre cada subíndice s de la matriz D_{j_n} y cada vector de índices $(i_1, \dots, i_M)$ de la MIG puede expresarse analíticamente como solución única de la siguiente ecuación diofántica

$$s = i_M + \sum_{k=1}^{M-1} (i_k - 1) \times \prod_{m=k+1}^M I_m \quad (24)$$

sujeta a las condiciones

$$\begin{aligned} 1 \leq i_m \leq I_m & \quad \forall m=1, \dots, M \\ i_m \in \mathbb{N} & \quad \forall m=1, \dots, M \end{aligned} \quad (25)$$

que, efectivamente se verifican por la propia definición de los índices.

La Figura 1 muestra una representación esquemática del proceso de transformación descrito.

La transformación del término relacionado con los valores observados en un vector se realiza mediante la misma aplicación F , que reordena los índices de manera análoga a como opera con la MIG. La definición de dicho vector puede establecerse como:

$$\tilde{a}_s = \tilde{a}_{1,i_1} \wedge \dots \wedge \tilde{a}_{M,i_M} \quad s=1, \dots, S \quad (26)$$

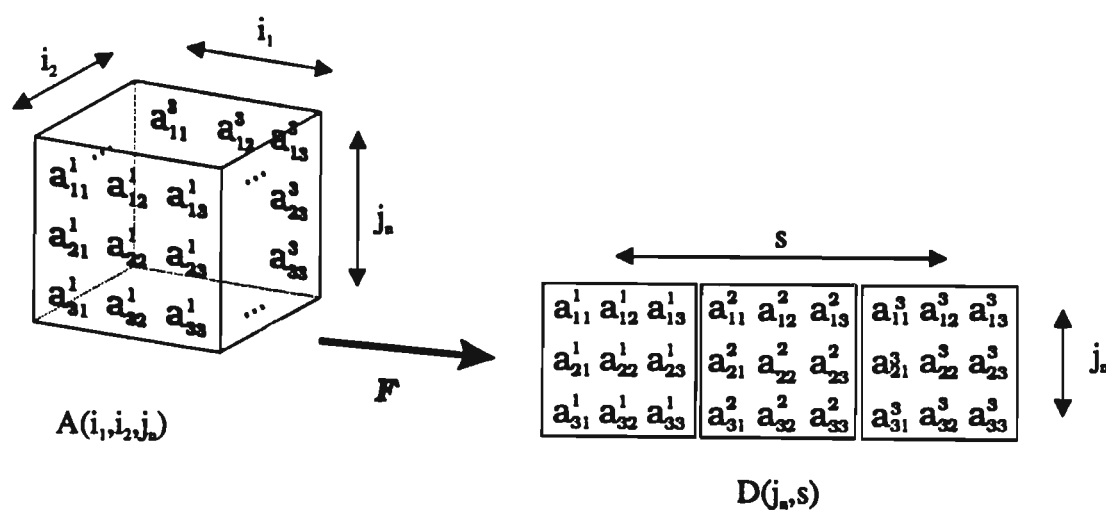


Figura 1: Obtención del problema transformado a partir del problema denso original.

Finalmente, teniendo en cuenta las definiciones (20), (21) y (26), podemos reescribir (18) como

$$\tilde{b}_{n,j_n} = \bigvee_{s=1}^s Dn_{j_n,s} \wedge \tilde{a}_s, \quad j_n=1,\dots,J_n \quad (27)$$

expresión que coincide, formalmente, con la expresión clásica del producto Matriz*Vector, sin más que sustituir los operadores " $\wedge$ " y " $\vee$ " por " $\times$ " y " Σ ", respectivamente. De esta forma, mediante un simple intercambio de bloques de operación, puede realizarse la evaluación de Sistemas de Control Borrosos a través de arquitecturas diseñadas para el producto de matrices. Sin embargo, dado el tamaño de la matriz, una implementación directa implicaría un número de PEs intratable. Pensando en una solución realizable resulta inevitable aplicar técnicas de particionamiento y proyección de algoritmos adecuadas para este tipo de operación. Nuestra solución resulta de adaptar la técnica DBT (*Dense to Band matrix transformation by Triangular-block partitioning*) propuesta por Navarro y col. [1987]. El objetivo no es otro que conseguir una estructura *hardware* de tamaño no dependiente del tamaño del problema (en nuestro caso, del tamaño de la BC a ejecutar).

3. Arquitectura sistólica.

Un array sistólico está constituido por una red de procesadores elementales (PEs) que de manera rítmica y regular transfieren datos entre sí, realizando en cada etapa una serie de operaciones breves y manteniendo un flujo regular en toda la red. Sus características de modularidad, regularidad e interconexión local los hacen enormemente atractivos para su implementación según técnicas VLSI.

Se han presentado en la literatura diversos planteamientos de arquitecturas sistólicas orientadas a la evaluación de SPDs desde las dos perspectivas esbozadas. En esta línea, Manzoul [1987] propone una solución basada en la estrategia de cálculo de grados de compatibilidad, consistente en considerar una BC constituida por una única variable en la parte antecedente o condición y otra en la parte consecuente o acción. Propone una proyección del proceso de ejecución de la BC sobre dos redes sistólicas lineales conectadas en serie, una de ellas dedicada al cómputo de los grados de compatibilidad y la otra a la modulación de la parte consecuente de las reglas. Ambas redes se basan en el mismo Procesador Elemental, radicando la diferencia básica entre ambas redes en el interconexionado existente entre los PEs que las componen.

Manzoul y Serrate [1988] plantean una solución para BC del tipo Unico-Antecedente Unico-Consecuente basada en la MIG, a través de dos redes sistólicas: la primera dedicada a la obtención de la MIG, poco interesante, ya que esta tarea puede realizarse en tiempo diferido, y la segunda al proceso de ejecución, a partir de dicha matriz, de la base de SCBs. Esta última es una red sistólica lineal constituida por un total de PEs igual al grado de discretización fijado para el universo de discurso donde se definen las variables de control (I), que consume $2 \cdot I$ ciclos de reloj hasta la obtención del valor final.

La generalización del modo de evaluación basado en MIGs a reglas con múltiples variables, tanto en la parte acción como en la parte condición, nos aproxima un poco más a la realidad de los sistemas de control borrosos, mayoritariamente caracterizados por presentar más de un antecedente en las partes *IF* y *THEN* de las reglas que componen su base de

conocimiento. El propio Manzoul y Tayal [1990] abordan una arquitectura pseudo-sistólica para la evaluación de Sistemas de Control Borroso de este tipo. Su solución, no obstante, no es más que una aproximación a la evaluación estricta de la BC, basada en obtener la MIG para una BC tipo Múltiple-Antecedente Múltiple-Consecuente y considerar las N submatrices que resultan de proyectar la MIG sobre los universos de discurso asociados a las N variables de control (por N denotamos el número de variables de control). Un planteamiento estricto de la evaluación a través de MIGs de Sistemas de Control Borrosos Múltiple-Antecedente Múltiple-Consecuente consiste, como ya hemos comentado, en la reformulación del Sistema de Control Borroso en reglas del tipo Múltiple-Antecedente Único-Consecuente (resultando $R \cdot N$ reglas). Estas reglas pueden agruparse en subsistemas con igual variable de acción en su parte consecuente, dando lugar a N MIGs, que permitirán realizar el proceso de inferencia para cada nuevo valor observado de las variables de estado [Ruiz y col., 1991]. Un trabajo posterior de Manzoul y Jayabharathi [1992], confirma esta conclusión, al abandonar los autores su método de evaluación inicial y utilizar el desdoblamiento y posterior reordenación de la base de reglas que hemos expuesto. No obstante, esta nueva solución propone una implementación que sigue una metodología radicalmente distinta a las anteriores, sustituyendo la arquitectura sistólica por un dispositivo FPGA que implementa, en forma de tabla (*look-up table*), todas las posibles respuestas del Sistema de Control Borroso. La imposibilidad de operar cuando los valores de entrada a la base de reglas son distribuciones de posibilidad, y no valores puntuales, restringe el ámbito de aplicación de esta solución.

Sin embargo, en todas las soluciones sistólicas comentadas, el número de PEs requeridos depende de las dimensiones del problema a tratar (número de elementos en los universos de discurso donde se definen las variables lingüísticas y número de éstas). Así pues, resulta interesante desde el punto de vista práctico, el desarrollo de una arquitectura sistólica en la que se pueda abordar la evaluación de Sistemas de Control Borrosos independientemente de sus dimensiones, dando lugar a una estructura hardware altamente versátil en cuanto a sus posibilidades de aplicación.

Haciendo uso del paralelismo existente entre la evaluación de Sistemas de Control Borrosos y el producto clásico Matriz*Vector (tal y como se desprende de la expresión (27)), tomamos como base una técnica de proyección sobre arquitecturas sistólicas empleada en operaciones de cálculo matricial [Navarro y col., 1987], adaptándola a las necesidades específicas de la evaluación de este tipo de sistemas.

3.1. Técnica de particionamiento DBT.

En la Figura 2 se muestra un ejemplo del fundamento teórico de la técnica DBT. La Figura 2a presenta el cálculo de un producto Matriz*Vector siguiendo la técnica tradicional, donde puede apreciarse cómo el número de operandos y el número de operaciones a realizar para la obtención de cada componente del vector resultado depende directamente del tamaño de la matriz. En la Figura 2b se presenta el planteamiento y la resolución de un problema transformado a partir del anterior. Lógicamente, la resolución del problema transformado no constituye directamente la solución del problema inicial, haciéndose necesaria una etapa de composición de las soluciones parciales (Figura 2c) obtenidas. La resolución del problema inicial mediante ambos procedimientos implica la realización del mismo número de operaciones, y la intervención del mismo número de operandos. Sin embargo, la segunda

metodología presenta una ventaja de tipo "estructural", consistente en que el número de operaciones a realizar -o, equivalentemente, el número de operandos que intervienen- depende exclusivamente del número de elementos no nulos que hay en cada columna o fila de la matriz transformada² (su ancho de banda) y es escogido arbitrariamente. De esta forma, se divide el cómputo global en varias etapas, de modo que en cada una de las cuales interviene un número de operandos que se fija arbitrariamente. El cómputo podrá realizarse, pues, en una estructura cuyo tamaño sea independiente de las dimensiones del problema original. Este hecho, añadido a la elevada reiteración y sencillez de las operaciones implicadas, hacen que el procedimiento descrito resulte altamente adecuado para su proyección sobre una arquitectura sistólica.

3.2. Aplicación del método DBT a SPBs.

Lógicamente, un punto clave en la adaptación de este método a la evaluación de SPBs lo constituye la obtención del problema transformado a partir del problema original, cuya formulación puede establecerse como

$$\tilde{B}_n = D_n \circ \tilde{A} \quad (28)$$

donde el símbolo $\circ$ se asocia al operador matricial SUP-mín y $\tilde{A} = \{\tilde{a}_s\} \quad s=1, \dots, S$ representa el vector de valores observados.

El problema transformado presenta una estructura idéntica (consiste igualmente en una composición $\circ$ de matriz y vector), que puede formularse como

$$\tilde{B}T_n = DT_n \circ \tilde{A}_T \quad (29)$$

donde DT_n representa la matriz de banda transformada a partir de la matriz densa D_n , $\tilde{A}_T$ representa el vector de valores observados transformado, y $\tilde{B}T_n$ el vector transformado de valores inferidos.

La obtención del problema transformado expresado en la ecuación (29) se realiza en dos fases: transformación de la matriz densa D_n y transformación del vector $\tilde{A}$.

El cálculo de la matriz transformada DT_n se realiza mediante la división de la matriz original D_n en submatrices cuadradas de dimensión ω . El valor de ω , que se establece arbitrariamente, fija el número de operaciones que se realizarán en cada una de las etapas en que se va a dividir el cálculo de los valores inferidos, y habrá de coincidir con el número de PEs que constituyan la arquitectura sobre la cual se va a realizar la proyección del problema. Una vez establecido ω , se obtienen dos valores $S', T' \in \mathbb{N}$, tales que el número de filas y el de columnas de la matriz original D_n (respectivamente, J_n y S) verifiquen

$$\begin{aligned} S &= \omega \cdot S' \\ T \Delta J_n &= \omega \cdot T' \end{aligned} \quad (30)$$

para lo cual puede ser necesario añadir columnas y/o filas de ceros a la matriz original.

² La matriz transformada es de tipo matriz de banda (los únicos elementos que pueden ser no nulos se agrupan en una banda diagonal), a diferencia de la matriz original, que es una matriz densa (sin restricción de ningún tipo).

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} \begin{matrix} a_1^1 & a_1^2 \\ a_2^1 & a_2^2 \end{matrix} & \begin{matrix} a_1^3 & a_1^4 \\ a_2^3 & a_2^4 \end{matrix} & \begin{matrix} a_1^5 & a_1^6 \\ a_2^5 & a_2^6 \end{matrix} \\ \begin{matrix} a_3^1 & a_3^2 \\ a_4^1 & a_4^2 \end{matrix} & \begin{matrix} a_3^3 & a_3^4 \\ a_4^3 & a_4^4 \end{matrix} & \begin{matrix} a_3^5 & a_3^6 \\ a_4^5 & a_4^6 \end{matrix} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

$$\begin{aligned} y_1 &= a_1^1 x_1 + a_1^2 x_2 + a_1^3 x_3 + a_1^4 x_4 + a_1^5 x_5 + a_1^6 x_6 \\ y_2 &= a_2^1 x_1 + a_2^2 x_2 + a_2^3 x_3 + a_2^4 x_4 + a_2^5 x_5 + a_2^6 x_6 \\ y_3 &= a_3^1 x_1 + a_3^2 x_2 + a_3^3 x_3 + a_3^4 x_4 + a_3^5 x_5 + a_3^6 x_6 \\ y_4 &= a_4^1 x_1 + a_4^2 x_2 + a_4^3 x_3 + a_4^4 x_4 + a_4^5 x_5 + a_4^6 x_6 \end{aligned}$$

a)

$$\begin{pmatrix} y_1^1 \\ y_2^1 \\ y_1^2 \\ y_2^2 \\ y_1^3 \\ y_2^3 \\ y_1^4 \\ y_2^4 \\ y_3^1 \\ y_4^1 \\ y_3^2 \\ y_4^2 \\ y_3^3 \\ y_4^3 \end{pmatrix} = \begin{pmatrix} \begin{matrix} a_1^1 & a_1^2 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ a_2^3 & a_2^4 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} a_3^5 & a_3^6 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ a_4^1 & a_4^2 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} a_5^1 & a_5^2 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} a_6^3 & a_6^4 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} a_1^5 & a_1^6 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

$$\begin{aligned} y_1^1 &= a_1^1 x_1 + a_1^2 x_2 \\ y_2^1 &= a_2^3 x_3 + a_2^4 x_4 \\ y_1^2 &= a_1^5 x_5 + a_1^6 x_6 \\ y_2^2 &= a_2^1 x_1 + a_2^2 x_2 \\ y_1^3 &= a_3^3 x_3 + a_3^4 x_4 \\ y_2^3 &= a_4^1 x_1 + a_4^2 x_2 \\ y_1^4 &= a_1^5 x_5 + a_1^6 x_6 \\ y_2^4 &= a_2^3 x_3 + a_2^4 x_4 \\ y_3^1 &= a_3^1 x_1 + a_3^2 x_2 \\ y_4^1 &= a_4^3 x_3 + a_4^4 x_4 \\ y_3^2 &= a_3^5 x_5 + a_3^6 x_6 \\ y_4^2 &= a_4^1 x_1 + a_4^2 x_2 \end{aligned}$$

$$\begin{aligned} y_1 &= y_1^1 + y_1^2 + y_1^3 \\ y_2 &= y_2^1 + y_2^2 + y_2^3 \\ y_3 &= y_3^1 + y_3^2 + y_3^3 \\ y_4 &= y_4^1 + y_4^2 + y_4^3 \end{aligned}$$

c)

b)

Figura 2: Producto Matriz*Vector. a) Expresión clásica. b) Problema transformado y obtención de resultados parciales. c) Composición de resultados parciales.

De esta forma, la matriz densa original D_n queda dividida en un total de $S' \cdot T'$ submatrices cuadradas (que denotaremos por $C_{s',t'}(i,j)$), cada una de las cuales se divide en sus partes triangular inferior (C^L), diagonal (C^D) y triangular superior (C^U), de modo que

$$C_{s',t'}(i,j) = C_{s',t'}^L(i,j) + C_{s',t'}^D(i,j) + C_{s',t'}^U(i,j) \triangleq C_{s',t'}^L(i,j) + C_{s',t'}^{DU}(i,j) \quad (31)$$

A partir de estas componentes triangulares, se construye la matriz de banda DT_n , siguiendo el proceso de reordenación de dichas componentes que se muestra en la Figura 1, y cuya realización algorítmica se describe a continuación

```

for t'=1 to T'
{
  for s'=(t'-1)S'+1 to S'+t'-1
  {
    DTns',s' = Ct',s'-(t'-1)S'DU
    DTns',s'+1 = Ct',s'+1-(t'-1)S'L
  }
  DTnt'S',t'S' = Ct',S'DU
  DTnt'S',t'S'+1 = Ct',1L
}

```

de donde se deduce que el número de filas de la matriz transformada será de $S'^T \omega$, y el de columnas $(S'^T + 1)\omega - 1$.

La transformación del vector original de valores observados se realiza, tal y como se puede deducir de la Figura 1, mediante una simple repetición del vector inicial hasta igualar el número de columnas de DT_n . La expresión que formaliza esta transformación es la siguiente:

$$\tilde{a}_T = \tilde{a}_{s \bmod S'} \quad s=1, \dots, (S'^T + 1)\omega - 1 \quad (32)$$

La característica principal que se desprende de la resolución del problema transformado es la división del proceso global de inferencia en S' etapas. Como en cada una de estas etapas se realiza un total de ω operaciones, parece adecuado el uso de una estructura *hardware* cíclica acumulativa de tamaño ω PEs (fijémonos en que ω es elegido arbitrariamente). En el apartado siguiente, presentamos una arquitectura con estas características, que implementa los bloques de resolución del problema transformado y agregación de los resultados parciales, resolviendo, pues, el problema global de evaluación de un Sistema de Control Borroso. No olvidemos que el bloque correspondiente al planteamiento del problema transformado se realiza en tiempo diferido, previamente al proceso de evaluación.

3.3. Descripción de la arquitectura.

La Figura 3 muestra el esquema de la celda básica (o PE) a partir de la cual se construye la propuesta de arquitectura de evaluación de Sistemas de Control Borrosos de control que presentamos en este apartado. Las interconexiones entre PEs, así como el flujo global de datos se muestran en la Figura 4, para un ejemplo concreto con 3 PEs y una MIG formada por $S=6$ columnas y $T=3$ filas.

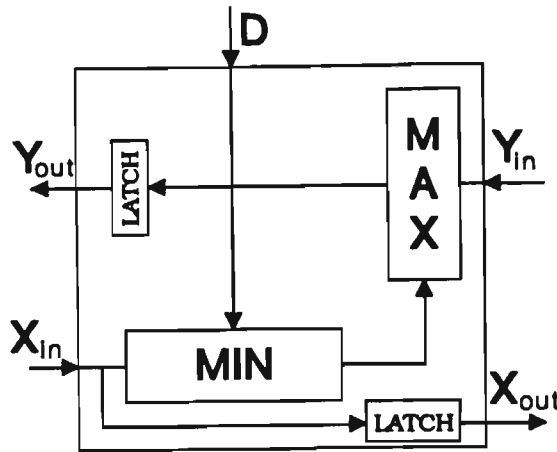


Figura 3: PE básico de la solución propuesta.

Sobre este ejemplo puede apreciarse la relación existente entre el ancho de banda de la matriz transformada y el número de procesadores de la arquitectura, que, como ya se comentó anteriormente, debe coincidir. Puesto que el ancho de banda es ω , cada una de las S' etapas en que se divide el cálculo del proceso global de inferencia implica la realización de ω operaciones. De esta forma, cada PE realizará una operación en cada una de dichas etapas y transferirá al siguiente PE el resultado parcial, de manera que se concluya dicha etapa cuando se dispone del resultado a la salida del primer PE. Esta dependencia entre el número de PEs y el ancho de banda de la matriz transformada condicionará, como veremos, algunos de los parámetros indicativos del rendimiento de la solución, como son la frecuencia de salida de datos y el tiempo total consumido en la evaluación del Sistema de Control Borroso.

El volcado de los coeficientes de la matriz en la red de PEs sigue una secuencia extremadamente simple, introduciéndose en cada PE una de las diagonales de la matriz de banda, intercalando ciclos de espera a intervalos regulares (cada dos ciclos de reloj) tras los ciclos iniciales de espera y sincronización. Igualmente simple resulta la introducción de las componentes del vector transformado de valores observados, que se realiza en la entrada X_{in} del primer PE, alternando igualmente ciclos de espera. El cómputo de estas componentes (expresión (26)) se supone realizado en una etapa previa, que, por claridad de la figura, no

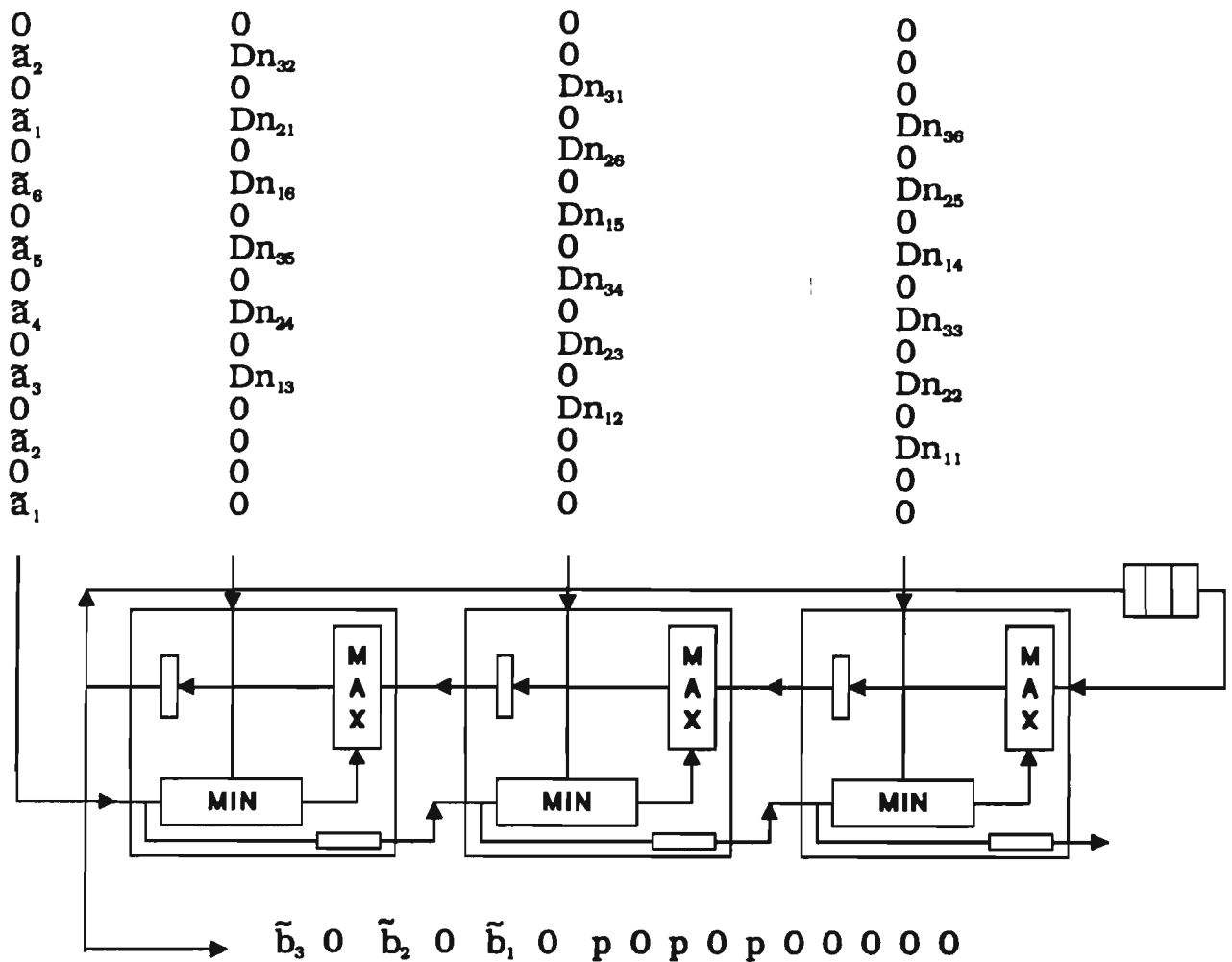


Figura 4: Red de PEs y flujo de datos ($\omega=3$; $S=6$; $T=3$) (El símbolo p a la salida del primer PE denota un resultado parcial).

se ha representado. Este bloque consistirá, básicamente, en una unidad de almacenamiento de las distribuciones de posibilidad observadas (constituida por M bloques -uno por cada uno de los antecedentes- de I palabras -una por cada punto del universo de discurso-) y una unidad de cálculo de mínimos de M entradas. El proceso de generación de las componentes del vector transformado puede realizarse simultáneamente con la evaluación de la matriz, con lo cual se consigue que el tiempo total de computación no se vea alterado significativamente.

Una atención mayor requiere, por el contrario, el control de flujo de realimentación de los resultados parciales. El lazo de realimentación realiza la función de agregación de los resultados parciales (Figura 2c), proporcionando el sincronismo adecuado para su correcto funcionamiento.

Al realizarse el cómputo global de cada componente del vector de valores inferidos en S' etapas de ω operaciones, los resultados parciales obtenidos durante el cómputo de una etapa deben realimentarse a la red de PEs para su acumulación con los resultados que se obtengan en la siguiente etapa. Realizando consideraciones de carácter temporal, el análisis de la solución indica que el primer resultado parcial se obtiene en el ciclo $2\omega-1$, instante a partir del cual se van generando, cada dos ciclos de reloj, los siguientes resultados parciales de esta primera etapa, hasta completar el total de ω componentes (que son las que se calculan en cada etapa). Dado que la siguiente etapa no se inicia hasta que han transcurrido ω ciclos (instante $3\omega-1$), estos resultados parciales deben ser retenidos hasta este instante, en el cual deberán volcarse en la entrada Y_m del ω -ésimo PE (que es el que realiza el primer cómputo de cada etapa). El sincronismo necesario se consigue de una manera simple, introduciendo una memoria FIFO de ω palabras en el lazo de realimentación, como se muestra en la Figura 4. La primera realimentación tendrá lugar cuando el elemento $\tilde{a}_{\omega+1}$ del vector haya alcanzado la entrada X_m del ω -ésimo PE, y, a partir de ese instante, cada dos ciclos de reloj, hasta que se hayan calculado los resultados finales (lo cual sucede en el ciclo $2S-1$ y se completa en el $2(S+\omega)-3$). A partir de aquí, se repite la secuencia de introducción de valores, hasta completar las S' etapas que componen el cálculo global.

4. Análisis y comparación.

Un parámetro importante, como indicativo del rendimiento de la red de PEs, es el ciclo de reloj en que se obtiene la última componente del vector de inferencias (tiempo de computación global), puesto que mide la separación temporal mínima necesaria entre dos observaciones sucesivas de las variables de estado. Esto sucederá $2S(T'-1)$ ciclos después de la obtención de la ω -ésima componente de dicho vector. Es decir, en el ciclo

$$\Delta = 2(S+\omega)-3 + 2S(T'-1) = 2(ST'+\omega)-3 \quad (33)$$

Otras soluciones sistólicas orientadas a la evaluación de Sistemas de Control Borrosos, que no presentan particionamiento del problema [Manzoul y Serrate, 1988; Kung, 1988³], arrojan un resultado idéntico si se verifica, además,

$$S'=1 \quad \delta \quad T'=1 \quad (34)$$

o, equivalentemente,

$$S=\omega \quad \delta \quad T=\omega \quad (35)$$

(recordemos que la solución de Manzoul se propone para sistemas del tipo Unico-Antecedente Unico-Consecuente, con lo cual la MIG es cuadrada, y, por tanto, $S=T$).

Dado que, en general, $S \gg T$, la opción $T=\omega$ implicará, manteniendo un mismo tiempo de computación global, la necesidad de un número menor de PEs, por lo cual será preferible frente a las soluciones que no presentan partición, y que necesitan implementar $S+T-1$ PEs.

Para que ambos parámetros temporales sean efectivamente idénticos, es necesario garantizar que las MIG de partida, en uno y otro método, sean exactamente iguales. Esto no

³ La solución de Kung se plantea para el producto Matriz*Vector, aunque sus resultados son totalmente comparables, al ser ambos procesos estructuralmente idénticos.

está garantizado, puesto que el método de particionamiento propuesto puede exigir la adición de columnas de ceros para que sea posible realizar la división de la MIG en submatrices cuadradas. No será necesario añadir columnas de ceros si

$$\exists \alpha \in \mathbb{N} \mid S = \alpha T \quad (36)$$

lo cual se verifica en la práctica totalidad de las soluciones implementadas, pues al ser usual considerar grados de discretización uniformes para todos los universos de discurso, tendremos

$$\begin{aligned} S \triangleq \prod_{m=1}^M I_m &= I^M \\ T \triangleq J_n &= I \end{aligned} \quad (37)$$

con lo que, trivialmente, la condición exigida se satisface.

Pese a todo, en el caso en que (36) no se verificase, el número de columnas a añadir sería, como máximo, de $T-1$. En una primera aproximación, para la mayoría de los casos en que la BC se pueda estructurar en reglas del tipo Múltiple-Antecedente Único-Consecuente, tendremos $S \gg T$, con lo cual

$$S + T - 1 \approx S \quad (38)$$

y la diferencia en cuanto al tiempo de computación global se puede considerar casi nula.

Un posible inconveniente en la solución propuesta lo constituye la eficiencia de valor $\frac{1}{2}$, asociada a los PEs que constituyen la red, y que viene impuesta por el propio método de particionamiento. Otra circunstancia negativa, ésta relacionada con el análisis presentado en el trabajo anterior, es que, en determinadas circunstancias (alto grado de discretización, elevado número de antecedentes), el tamaño de la MIG sea excesivamente grande, y la solución poco práctica. En estos casos, se hace necesario un planteamiento que se ajuste a la metodología alternativa (regla a regla), incorporando, en la medida de lo posible, técnicas de particionamiento y proyección similares a las descritas aquí. A este trabajo estamos dedicando nuestra atención actualmente.

Referencias

- Kung, SY (1988). En T Kailath (Ed.), *VLSI array processors*. Prentice Hall, Englewood Cliffs, N. Jersey.
- Mamdani, EH (1977). Application of Fuzzy Logic to Approximate Reasoning using Linguistic Systems. *IEEE Trans. on Computers* 26, 1182-1191.
- Manzoul, MA (1987). Fuzzy Inference on a Systolic Array. *Proc. 18th Modelling and Simulation Conference* 1103-1108.
- Manzoul, MA y HA Serrate (1988). Fuzzy Systolic Arrays. *Proc. 18th International Symposium on Multiple-Valued Logic*.

Manzoul, MA y S Tayal (1990). Systolic VLSI Array for Multi-Variable Fuzzy Control Systems. *Cybernetics and Systems* 21, 27-42.

Manzoul, MA y D Jayabharathi (1992). Fuzzy Controller on FPGA Chip. *Proc. 1st IEEE Conf. on Fuzzy Systems*, 1309-1316.

Navarro, JJ, JM Llabería y M Valero (1987). Partitioning: An Essential Step in Mapping Algorithms into Systolic Array Processors. *IEEE Computer*, July, 77-89.

Ruiz, R, S Barro, A Bugarín y J Presedo (1991). Fuzzy production systems on a VLSI architecture. *Proc. 1991 IFSA Congress*, Bruselas, 175-178.

Togai, M y H Watanabe (1986). Expert System on a Chip: An Engine for Real-Time Approximate Reasoning. *IEEE Expert Fall*, 55-62.

Watanabe, H, WD Dettloff y KE Yount (1990). A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture. *IEEE Journal of Solid-State Circuits* 25-2, 376-382.

Zadeh, LA (1965). Fuzzy Sets. *Information and Control* 8, 338-353.

Zadeh, LA (1971). Towards a theory of fuzzy systems. En RE Kalman y N DeClaris (Eds.), *Aspects of network and systems theory*. Holt, Rinehart, Winston.

Zadeh, LA (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. on Syst., Man and Cyber.* 3-1, 28-44.

DISEÑO SISTEMÁTICO DE UN PROCESADOR SISTÓLICO DE INFERENCIAS DIFUSAS.

F. Fernández, A. Ruiz, J. Gutiérrez
Departamento de Tecnología Fotónica
Facultad de Informática
Madrid (España)

Resumen: En este artículo se analiza la complejidad computacional de diversos algoritmos de inferencia borrosos descritos utilizando una notación matricial pseudo-APL, y se derivan, de forma sistemática, algoritmos equivalentes mediante la aplicación de las correspondientes propiedades de las relaciones binarias borrosas. Basándose en el análisis previo, se muestra el diseño de un procesador sistólico borroso, para resolver problemas de un tamaño arbitrario, con un número de procesadores fijo, independiente del tamaño del problema.

Palabras Clave: Controlador borroso, Procesador sistólico, Particionado, Método de proyección de dependencias.

1. Introducción

Desde que en 1965 el profesor iraní Lotfi A. Zadeh de la universidad de Berkeley dio a conocer su innovadora teoría de conjuntos borrosos, el número de sus aplicaciones en diversos campos de la ciencia no ha dejado de crecer. Los conjuntos borrosos constituyen una herramienta de gran potencia expresiva para describir situaciones o reglas que son intrínsecamente ambiguas o que no están exactamente definidas.

El concepto de continuidad, subyacente en dicha teoría, juega un papel esencial en la descripción de sistemas de muy diverso tipo. Dicha continuidad implica la no existencia de zonas nítidas de separación entre la certeza o falsedad de una determinada característica. La función de pertenencia de un conjunto borroso, caracterizado por una determinada propiedad vaga, permite capturar de forma simple la naturaleza continua de una determinada situación, proceso o regla.

El diseño de un sistema automático de control, requiere generalmente el conocimiento matemático preciso del sistema a controlar. Sin embargo para muchos procesos complejos, la construcción de un modelo que caracterice su comportamiento es difícil o prácticamente imposible. En estos casos un controlador borroso puede ser la solución más efectiva.

En un controlador borroso se considera generalmente una estructura de relación aproximada entre condiciones y acciones en forma de reglas de tipo "If-Then". Estas reglas de inferencia permiten expresar, de forma estructurada, relaciones entre las variables del proceso a controlar; las funciones de pertenencia asociadas permiten evaluar de forma numérica el grado de aproximación de una situación concreta a un conjunto de condiciones vagas previamente establecido.

En los últimos años se han desarrollado diversas aplicaciones basadas en lógica borrosa. Los resultados obtenidos han demostrado las ventajas substanciales de estos sistemas, frente al procesamiento lógico clásico, en ciertos campos tales como el control automático o el procesamiento de señal.

Un problema crucial en muchas de las aplicaciones de control borroso, especialmente en las de tiempo real, es el tiempo de proceso. Dicho tiempo de computación, en una máquina de tipo von Neumann, puede llegar a ser muy grande debido al gran número de operaciones requerido para evaluar las reglas correspondientes. Esto es especialmente cierto cuando el número de reglas y entradas es grande, como es el caso de muchos sistemas prácticos. Afortunadamente, el rápido avance de las técnicas VLSI han permitido actualmente la integración, en un simple chip, de una impresionante potencia de cómputo, así como el diseño de circuitos específicos orientados hacia la realización de un determinado algoritmo o grupos de algoritmos del tipo cómputo intensivo.

El problema de diseño de chips de inferencias borrosas ha recibido en los últimos años diversas soluciones de tipo analógico [YAMAKAWA 88] [PATYRA 91] y digital [TOGAI 86][WATANABE 88][ISIK 88] [CHIUEH 92]. El mecanismo de inferencia utilizado generalmente en éstos, para caracterizar reglas del tipo "If A Then B", es el operador de composición máximo-mínimo sobre las funciones de pertenencia correspondientes [LEE 90]. La composición de antecedentes (And) y reglas (Or) se realiza usualmente mediante operaciones de mínimo y máximo. Las técnicas de desborrosificación más comúnmente utilizadas están basadas en el cálculo del máximo, media de máximos y centro de área del conjunto borroso correspondiente [LEE 90][KOSKO 92].

La tecnología microelectrónica favorece especialmente las arquitecturas regulares síncronas que tengan un gran número de elementos de proceso idénticos y conectados sólo localmente. Estas arquitecturas se conocen con el nombre de estructuras sistólicas, habiéndose diseñado una gran cantidad de procesadores de este tipo [KUNG 88], para la resolución de problemas de cómputo intensivo, en procesamiento de señal, procesamiento de imágenes, teoría de grafos, álgebra lineal, etc.

Con objeto de garantizar un diseño correcto de un procesador sistólico en un corto período de tiempo, es conveniente el uso de técnicas formales de síntesis y de herramientas CAD que relacionen la especificación algorítmica con la realización última del sistema; existiendo por tanto, dentro de este campo, un gran interés en el desarrollo de metodologías de diseño. Actualmente se dispone de numerosos resultados sobre la relación entre la especificación de algoritmos a alto nivel, mediante ecuaciones de recurrencia, y las correspondientes arquitecturas paralelas regulares [COSNARD 92] [MEGSON 92] [QUINTON 89] [JAGADISH 87] [RAO 85]. También se cuenta con herramientas formales para realizar el particionado [NELIS 87] [HORIIE 87] y el diseño de las señales de control de una estructura sistólica [RAJOPADHYE 89] [TEICH 91]. Las técnicas de resincronización [LEISERSON 83] [KUNG 88] resultan así mismo convenientes para simplificar y optimizar la función temporal de las células de proceso.

A continuación, en la sección 2 se establece primeramente la notación matricial pseudo-APL utilizada para la descripción de los elementos básicos de una inferencia borrosa: conjuntos y relaciones; se describen diversos operadores matriciales y las principales propiedades de los mismos. Esta representación matricial resulta especialmente indicada para la descripción inicial del algoritmo borroso, para la transformación sistemática en diversos algoritmos equivalentes, y como lenguaje de especificación inicial de la correspondiente arquitectura paralela regular.

En la sección 3 se analiza la complejidad del cómputo matricial de diversos algoritmos de control borroso, y los problemas asociados a la superposición de reglas y a la multiplicidad de entradas (antecedentes) y salidas (consecuentes).

En la sección 4 se considera la estructura general de un procesador borroso, se analizan diversas opciones de diseño y las principales formas de realización en VLSI. Finalmente se describe un algoritmo eficiente para el cómputo del centro de área, mediante un sistema de ecuaciones de recurrencia.

En la sección 5 se describe un procesador sistólico para realizar el producto interno máximo-mínimo de dos vectores, válido igualmente para el producto de matriz por vector, que resuelve problemas de un tamaño arbitrario utilizando un número de procesadores fijo, independiente del tamaño del problema.

Finalmente se describen brevemente las conclusiones en la sección 6.

2. Representación matricial de conjuntos y relaciones borrosas

Si se desea introducir un cierto grado de formalismo en el diseño de una arquitectura orientada a la realización de un determinado algoritmo, es conveniente la utilización de estructuras y propiedades que expresen las principales características del sistema a realizar. Y aunque no a todos los factores que influyen en el diseño puede dárseles una especificación formal simple, las ventajas que reporta la utilización de herramientas formales, supera en muchos casos la intención inicial de la mismas.

A continuación, se concreta la notación matricial utilizada para especificar, en lenguaje de alto nivel, los diversos algoritmos de inferencia borrosos [KOSKO 92] [LEE 90] [MANZOUL 88]. Dicha notación permite analizar cómodamente la complejidad computacional de un algoritmo. La consideración de algunas propiedades facilita también la transformación sistemática de la especificación inicial del algoritmo, así como la simplificación del diseño de la arquitectura correspondiente.

2.1 Definición vectorial de un conjunto borroso discreto

Dado el conjunto universal índice (linealmente ordenado) U de n elementos:

$$U = \{1, 2, \dots, n\} \quad n \in \mathbb{N}$$

La imagen de la función de pertenencia o función característica de dicho conjunto universal se puede describir mediante el vector columna característico U de n dimensiones:

$$U = U[1:n] = [1(i)] \quad i=1, \dots, n = [1, 1, \dots, 1]^T$$

Un conjunto borroso A contenido en U se puede también describir mediante el vector columna característico A de n dimensiones, imagen de su función característica:

$$A = A[1:n] = [a(i)] \quad i=1, \dots, n \quad a(i) \in L=[0, 1]$$

Utilizándose la misma designación para el conjunto borroso que para su vector característico (función característica) correspondiente [NOVAK 89].

Las operaciones de unión, intersección y complementación de conjuntos borrosos se definirán análogamente como operaciones vectoriales entre sus vectores característicos:

$$A \vee B = [a(i) \vee b(i)] \quad i=1, \dots, n$$

$$\begin{aligned} A \wedge B &= [a(i) \wedge b(i)] \quad i=1, \dots, n \\ \neg A &= [1-a(i)] \quad i=1, \dots, n \end{aligned}$$

donde los símbolos $\vee$, $\wedge$ y $\neg$ se utilizan, en este artículo, para designar respectivamente las operaciones (sobre escalares o vectores), de máximo, mínimo y complementación.

Un conjunto unitario, correspondiente al valor exacto i dentro del universo del discurso U , se puede representar mediante un vector columna unitario característico I_i de n dimensiones:

$$I_i = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$$

cuya única componente no nula (igual a 1) se encuentra en la posición i .

2.2 Definición matricial de una relación borrosa binaria

Si U y V son dos conjuntos universales índice, con n y m elementos respectivamente, y si R es una relación borrosa de V en U , es posible representar a la relación R mediante su matriz característica R de orden $n \times m$:

$$R = R[1:n, 1:m] = [r(i, j)] \quad i=1, \dots, n \quad j=1, \dots, m$$

donde $r(i, j) \in L$ indica el grado de pertenencia, del par ordenado (i, j) en la relación R . Obsérvese que de acuerdo a esta notación, el conjunto inicial V de la relación se corresponde con las columnas de la matriz y el conjunto final U con las filas. Habiéndose utilizado también la misma designación para la relación binaria borrosa que para su matriz característica correspondiente.

Con objeto de unificar, se puede también considerar un conjunto borroso A como una relación binaria borrosa (relación de pertenencia) del conjunto universal unitario $\text{Nombre\_de\_}A$ en el conjunto universal U de n elementos. Dicha relación binaria estará determinada por una matriz columna característica coincidente con su vector característico:

$$A = A[1:n, 1] = [a(i, 1)] \quad i=1, \dots, n \quad a(i, 1) = a(i)$$

Inversamente, la anterior unificación permite también considerar a toda matriz columna característica de una relación como el vector columna característico de un conjunto asociado.

Por tanto, las propiedades y operaciones de las relaciones binarias borrosas, serán también reinterpretables desde el punto de vista de relaciones de pertenencia.

2.3 Descripción matricial de operaciones sobre relaciones binarias borrosas

Al existir una biyección entre relaciones binarias borrosas y matrices características, es posible definir las operaciones y propiedades de las relaciones binarias borrosas sobre sus matrices características correspondientes.

Si R y S son dos relaciones binarias borrosas del conjunto universal V en el conjunto universal U , las operaciones de unión, intersección y complementación de relaciones binarias borrosas se definen

como operaciones entre sus matrices características:

$$\begin{aligned} R \vee S &= [r(i,j) \vee s(i,j)] & i=1,\dots,n \quad j=1,\dots,m \\ R \wedge S &= [r(i,j) \wedge s(i,j)] & i=1,\dots,n \quad j=1,\dots,m \\ \neg R &= [1-a(i,j)] & i=1,\dots,n \quad j=1,\dots,m \end{aligned}$$

donde los símbolos $\vee$, $\wedge$ y $\neg$ denotan también las operaciones matriciales de máximo, mínimo y complemento a la unidad.

Al ser las operaciones de máximo y mínimo, asociativas y conmutativas, se puede definir también, la unión e intersección de un número finito relaciones binarias borrosas S_k ($k=1..e$), mediante:

$$\vee / (k=1..e) R_k = [\vee / (k=1..e) r_k(i,j)] \quad i=1,\dots,n \quad j=1,\dots,m$$

$$\wedge / (k=1..e) R_k = [\wedge / (k=1..e) r_k(i,j)] \quad i=1,\dots,n \quad j=1,\dots,m$$

donde los símbolos $\vee / (k=1..e)$ y $\wedge / (k=1..e)$ denotan la aplicación de la operación $\vee$ o $\wedge$ (escalar o matricial) en el correspondiente conjunto índice ($k=1..e$) (notación de operador de reducción o sumatorio, pseudo-APL o FP [BACKUS 78]). Si la operación se extiende a todo el rango de variación (defecto: el conjunto índice universal) se puede omitir el índice en el operador:

$$\vee / (k=1..e) R_k = \vee / R_k$$

$$\wedge / (k=1..e) R_k = \wedge / R_k$$

Análogamente se definen las operaciones de unión e intersección entre una relación binaria borrosa o matriz característica R de orden $n \times m$ y una α de orden 1×1 (escalar):

$$\begin{aligned} R \vee \alpha &= [r(i,j) \vee \alpha] & i=1,\dots,n \quad j=1,\dots,m \\ R \wedge \alpha &= [r(i,j) \wedge \alpha] & i=1,\dots,n \quad j=1,\dots,m \end{aligned}$$

La composición máximo-mínimo de dos relaciones binarias borrosas T y R definidas a través de sus matrices características: T de orden $p \times n$ y R de orden $n \times m$, viene definida por el producto matricial correspondiente [KLIR 88] [NOVAK 89], utilizando las operaciones de máximo y mínimo en lugar de la adición y multiplicación:

$$T \circ R = [\vee / (k=1..n) t(i,k) \wedge r(k,j)] \quad i=1,\dots,p \quad j=1,\dots,m$$

donde el símbolo $\circ$ denota el producto matricial, el operador $\vee$ denota la operación de máximo ($\vee$) sobre los elementos indicados ($k=1..n$), y el símbolo $\wedge$ la operación de mínimo.

Un algoritmo borroso se puede generalmente expresar como una composición de relaciones binarias borrosas. El número de operaciones mínimo ($\wedge$) y máximo ($\vee$) que es necesario realizar en la composición de dos relaciones binarias borrosas, T de orden $p \times n$ y R de orden $n \times m$ es

Número de operaciones mínimo: $p(n)m$

Número de operaciones máximo: $p(n-1)m$.

Número total de operaciones : $p(m)(2n-1)$

Las operaciones de un algoritmo borroso: producto exterior e interno de los vectores columna característicos A y B de orden n, se pueden expresar también en forma de producto matricial de sus correspondientes relaciones binarias de pertenencia:

$$\text{Producto interno. } A^T \circ B = [\vee / (k=1..n) a(1,k) \wedge b(k,1)]$$

$$\text{Producto exterior. } A \circ B^T = [a(i,1) \wedge b(1,j)] \quad i,j=1,\dots,n$$

Como caso particular, se tiene que el producto interno de un vector $A = [a(1), \dots, a(n)]^T$ por un vector unitario I_i de la misma dimensión n (que corresponde a un valor exacto), tiene la siguiente expresión simplificada:

$$A^T \circ I_i = a(i)$$

La operación de truncamiento del vector columna característico A por un escalar α (o matriz 1x1) se puede expresar matricialmente mediante la siguiente composición de relaciones:

$$A \circ \alpha = [a(i,1) \wedge \alpha] \quad i=1,\dots,n$$

A continuación se indican algunas propiedades [NOVAK 89] [LEE 90] [KOSKO 92] de las operaciones con relaciones binarias borrosas (R, S, T, Si ($i=1..e$) y Sij ($(i=1..e) (j=1..s)$)), que son de interés para la transformación sistemática y estudio de la complejidad de un algoritmo borroso.

■ Conmutativas

$$R \wedge S = S \wedge R \quad R \vee S = S \vee R$$

■ Asociativas

$$\begin{aligned} (R \circ S) \circ T &= R \circ (S \circ T) \\ (R \wedge S) \wedge T &= R \wedge (S \wedge T) \\ (R \vee S) \vee T &= R \vee (S \vee T) \end{aligned}$$

■ Distributivas

$$\begin{aligned} R \vee (S \wedge T) &= (R \vee S) \wedge (R \vee T) & (S \vee T) \wedge R &= (S \vee R) \wedge (T \vee R) \\ R \wedge (S \vee T) &= (R \wedge S) \vee (R \wedge T) & (S \vee T) \wedge R &= (S \wedge R) \vee (T \wedge R) \end{aligned}$$

$$\begin{aligned} R \circ (S \vee T) &= (R \circ S) \vee (R \circ T) & (S \vee T) \circ R &= (S \circ R) \vee (T \circ R) \\ R \circ (S \wedge T) &= (R \circ S) \wedge (R \circ T) & (S \wedge T) \circ R &= (S \circ R) \wedge (T \circ R) \end{aligned}$$

■ Distributivas generalizadas

$$\begin{aligned} R \vee (\wedge / Si) &= \wedge / (R \vee Si) & (\wedge / Si) \vee R &= \wedge / (Si \vee R) \\ R \wedge (\vee / Si) &= \vee / (R \wedge Si) & (\vee / Si) \wedge R &= \vee / (Si \wedge R) \end{aligned}$$

$$\begin{aligned} R \circ (\vee / Si) &= \vee / (R \circ Si) & (\vee / Si) \circ R &= \vee / (Si \circ R) \\ R \circ (\wedge / Si) &= \wedge / (R \circ Si) & (\wedge / Si) \circ R &= \wedge / (Si \circ R) \end{aligned}$$

donde se supone que los operadores matriciales de máximo $\vee /$ y mínimo $\wedge /$ se aplican sobre el conjunto índice universal correspondiente: ($i=1..e$). También se verifica la siguiente desigualdad

$$\bigvee / (i=1..e) (\bigwedge / (j=1..s) S_{ij}) \leq \bigwedge / (j=1..s) (\bigvee / (i=1..e) S_{ij})$$

■ Coincidencia de operaciones

$$R \circ S = R \wedge S \quad \text{si } R \text{ es de orden } n \times 1 \text{ y } S \text{ es de orden } 1 \times 1$$

En el apartado siguiente se considera la representación matricial de una inferencia borrosa y se estudia la complejidad de diversos algoritmos borrosos equivalentes. La transformación entre algoritmos se lleva a cabo teniendo en cuenta las anteriores propiedades de las relaciones binarias borrosas. En general la complejidad del algoritmo, expresado en forma matricial, dependerá grandemente del tipo de parentización utilizado, siendo por tanto de interés, la obtención de la complejidad para diferentes formas de cómputo del algoritmo borroso correspondiente.

3. Complejidad de algoritmos borrosos expresados en forma matricial

Si dos variables lingüísticas X e Y están relacionadas condicionalmente a través de las etiquetas lingüísticas A y B , su implicación se puede representar mediante la regla elemental de inferencia borrosa

R: If X is A Then Y is B

donde X y A son subconjuntos borrosos del universo U de n dimensiones, mientras que los subconjuntos borrosos Y y B pertenecen al universo V de m dimensiones. Dicha implicación se puede caracterizar mediante la relación binaria borrosa mínimo [LEE 90] [KOSKO 91], definida a través de la matriz característica:

$$S = [b(v,1) \wedge a(1,u)] \quad v=1,\dots,m \quad u=1,\dots,n$$

que se puede poner en la forma:

$$S = B \circ A^T$$

La relación funcional borrosa entre una entrada (conjunto borroso) X y su correspondiente salida (conjunto borroso) Y , se puede también caracterizar a través de una composición máximo-mínimo [LEE 90] [KOSKO 91], expresada matricialmente mediante:

$$Y = S \circ X$$

luego

$$Y = (B \circ A^T) \circ X = B \circ (A^T \circ X)$$

donde $Y < m \times 1 >$, $B < m \times 1 >$, $A < n \times 1 >$ y $X < n \times 1 >$ son los correspondientes vectores columna característicos, con las dimensiones indicadas.

El número de operaciones mín-máx necesarias para computar la primera forma parentética de la expresión anterior es: $(m)(n) + m(2n-1)$. Esta primera forma, tiene la ventaja de poder caracterizar la relación de implicación S de forma genérica, diferente de la composición máx-min: $B \circ A^T$, y de hacer la ejecución del algoritmo independiente del la relación de implicación elegida [LIM 90]. Para la segunda forma de parentización [WATANABE 90], el número de operaciones necesarias es: $2n-1 + m$; en donde

se comprueba la importancia de la forma parentética, a la hora de evaluar un producto matricial múltiple.

En un controlador elemental de una sola entrada X y una sola salida Y , existirán r reglas del tipo:

R_i : If X is A_i Then Y is B_i

combinadas mediante la conectiva Also (Or). La expresión matricial resultante de la salida Y se puede caracterizar mediante la operación máximo::

$$Y = \vee / (i=1..r) (B_i \circ (A_i^T \circ X))$$

Para computar la anterior expresión será necesario realizar un número de operaciones mín-máx: $r(2n-1+m)+(r-1)m$. Por tanto su complejidad computacional es del orden de $r(n+m)$.

Si se aplican las propiedades asociativa y distributiva para calcular el valor de Y , éste se puede poner en la forma:

$$Y = (\vee / (i=1..r) (B_i \circ A_i^T)) \circ X$$

Si se descompone esta expresión en una etapa de compilación (no dependiente de X) y en otra de ejecución (dependiente de X), el número de operaciones máx-mín para calcular la anterior expresión es:

Durante la compilación del algoritmo: $r(m)n+(r-1)(m)n$

Durante la ejecución del algoritmo: $m(2n-1)$

Esta segunda forma de cómputo [MANZOUL 88] tiene como ventaja que el número de operaciones en la fase de ejecución no depende del número de reglas r .

Si se considera una regla R de inferencia borrosa, compuesta de múltiples antecedentes A_k asociados a las variables de entrada X_k :

R : If $(X_1$ is $A_1)$ And ... $(X_e$ is $A_e)$ Then Y is B

donde $X_1, \dots, X_e$ y $A_1, \dots, A_e$ son subconjuntos borrosos de dimensión n . La expresión matricial de la salida Y se puede caracterizar mediante:

$$Y = B \circ (\wedge / (k=1..e) (A_k^T \circ X_k))$$

Para computar la anterior expresión será necesario realizar un número de operaciones min-max: $e(2n-1)+(e-1)+m$. Por tanto su complejidad computacional es del orden de $e(n)+m$.

Al aplicar las propiedades asociativa y distributiva, se puede poner el valor de Y , en la forma [KOSKO 91]:

$$Y = \wedge / (k=1..e) ((B \circ A_k^T) \circ X_k)$$

El número de operaciones máx-mín para calcular la anterior expresión es:

Durante la compilación del algoritmo: $e(m)n$

Durante la ejecución del algoritmo: $e(m)(2n-1)+m(e-1) = m(2(e)n-1)$

En general un controlador borroso estará en general definido por un conjunto de r inferencias borrosas R_i , combinadas mediante la conectiva Or , con múltiples antecedentes (e entradas borrosas de dimensión n) y múltiples consecuentes (s salidas borrosas de dimensión m):

R_i : If (X_1 is A_{i1}) And ... And (X_e is A_{ie}) Then
(Y_1 is B_{i1}) And ... And (Y_s is B_{is})

($i=1, \dots, r$)

Cada una de estas inferencias se puede a su vez descomponer [LEE 90] en un conjunto de s inferencias de simple consecuente:

R_{ij} : If (X_1 is A_{i1}) And ... And (X_e is A_{ie}) Then (Y_j is B_{ij})

($i=1, \dots, r$) ($j=1, \dots, s$)

Las correspondientes salidas Y_j vendrán definidas por la siguiente expresión matricial equivalente:

$$Y_j = \bigvee_{(i=1..r)} (B_{ij} \circ (\bigwedge_{(k=1..e)} (A_{ik}^T \circ X_k))) \quad (j=1, \dots, s)$$

Para computar esta expresión será necesario realizar un número de operaciones min-max: $r(e(2n-1) + (e-1)) + s(r)m + s(r-1)m$. Por tanto su complejidad computacional es del orden de $r(e(n) + (s)m)$.

Otras dos expresiones equivalentes a la anterior, aunque computacionalmente menos eficientes, son:

$$Y_j = \bigvee_{(i=1..r)} \bigwedge_{(k=1..e)} (B_{ij} \circ (A_{ik}^T \circ X_k)) \quad (j=1, \dots, s)$$

$$Y_j = \bigvee_{(i=1..r)} \bigwedge_{(k=1..e)} ((B_{ij} \circ A_{ik}^T) \circ X_k) \quad (j=1, \dots, s)$$

Estas dos últimas formas de evaluar la salida tienen sin embargo, la ventaja de realizar el proceso de multiplicación matricial de forma uniforme, independiente del número de salidas y reglas, lo que es de gran interés, para poder realizar su computación en un procesador modular de inferencias de antecedente y consecuente simples [LIM 90].

Si las entradas X_k son valores exactos $u(k)$ (representables también mediante vectores característicos unitarios), el algoritmo de inferencia borroso se reduce en este caso a:

$$Y_j = \bigvee_{(i=1..r)} (B_{ij} \circ (\bigwedge_{(k=1..e)} a_{ik}(u(k))) \quad (j=1, \dots, s)$$

El número de operaciones máx-mín. necesario para computar la anterior expresión es: $r(e-1) + s(r)m + s(r-1)m$. Su complejidad computacional es por tanto del orden de $r(e + (s)m)$. Si los consecuentes B_{ij} son además conjuntos borrosos unitarios, la complejidad computacional se reduce a: $r(e + s)$.

En el siguiente apartado se considera la estructura general de un procesador borroso, y se analizan, de forma sistemática, las principales opciones de diseño.

4. Factores generales de diseño de un procesador borroso

Para realizar el diseño de la arquitectura de un procesador específico es conveniente disponer de una descripción de alto nivel de sus principales componentes. Es posible modelizar un procesador de control borroso genérico, mediante una tupla PB de 9 componentes básicos:

$$PB = (ENT, SAL, UCP, BOR, DBOR, UI, URA, MEM, BUS)$$

donde ENT es el conjunto de entradas y SAL el de salidas; UCP es la unidad local de control del procesador borroso, que se comunica también con el procesador central (host); BOR es el borrosificador de las entradas y DBOR el desborrosificador de las salidas; UI es la unidad de inferencia en donde se realizan los correspondientes productos matriciales; URA es la unidad lógica de detección de reglas activas, aquellas que se deben computar en la unidad de inferencia; MEM es el conjunto de bancos locales de memoria de datos de entrada y salida, de resultados intermedios, y del programa borroso (funciones de pertenencia, reglas de inferencia, y opcionalmente matrices características de relaciones de implicación); finalmente BUS es el conjunto de elementos de comunicación de datos y señales de control.

La forma de configurar y dimensionar los bloques anteriores puede ser muy diversa, dependiendo del tipo de aplicación y de los criterios de diseño empleados. Un factor primordial de diseño, es el grado de generalidad del procesador, considerándose usualmente tres tipos de chips borrosos:

I) Procesador superespecífico, dedicado a una aplicación concreta, así como destinado a un tamaño del problema a tratar [YAMAKAWA 88]. Se utiliza generalmente este tipo para algunas aplicaciones de control, en donde el tiempo de computación es un parámetro especialmente crítico.

II) Procesador específico independiente del tamaño y forma del problema [LIM 90] [IKEDA 91]. Este requerimiento implica la necesidad de descomposición o reconfiguración del algoritmo a computar, para que pueda ejecutarse en los procesadores existentes. En este caso puede ser necesaria una unidad de agregación que coordine los resultados parciales proporcionados o solicitados por las diferentes unidades de proceso. Este tipo de procesadores presentan un cierto grado de tolerancia a fallos, debido a su gran modularidad.

III) Procesador específico reconfigurable, intermedio entre los dos tipos anteriores, capaz de resolver un conjunto limitado de instancias del problema, mediante la reconfiguración de sus componentes y/o conexiones [ISIK 88] [WATANABE 90].

Otro factor fundamental, en el diseño de un procesador borroso, es la especificación del valor (o rango de variación) de sus parámetros básicos

a) Número de bits de resolución de las entradas y salidas: (b_x, b_y) .

b) Número de entradas X_k y salidas Y_j : (e, s) .

c) Número de reglas R_i del algoritmo borroso: (r) .

También es necesario considerar inicialmente los parámetros internos de borrosificación de las variables y etiquetas lingüísticas asociadas:

a) Número de elementos de los dominios de borrosificación de entrada U y salida V : (n, m) .

b) Número de bits utilizados para cuantificar los niveles discretos de la función de pertenencia: (b_μ) .

Es conveniente también tener en cuenta que las etiquetas lingüísticas se suelen caracterizar mediante números o intervalos borrosos [NOVAK 89], con un cierto grado de solapamiento máximo (GSM) entre ellos, cuyo valor es generalmente dos, en la mayor parte de las aplicaciones. Se puede reducir bastante el tamaño del almacenamiento necesario y simplificar su recuperación, al utilizar un esquema de compactación de la memoria [CHIUE 92], almacenando diversos números borrosos (etiquetas que no se solapan) en un mismo bloque. Utilizando este esquema, y si el GSM es dos, se puede almacenar el conjunto de todos los números borrosos (etiquetas) asociados a una variable de entrada o salida, utilizando únicamente dos memorias de tamaño $(b\mu n)$ o $(b\mu m)$ respectivamente.

Otro factor, relacionado con el anterior, que permite también reducir considerablemente la computación de un algoritmo borroso, es la detección de las reglas activas en cada instante, mediante un circuito lógico multivaluado URA (citado con anterioridad) [IKEDA 91] [CHIUE 92]. Esta unidad permite computar de forma borrosa en la UI, únicamente las reglas que contribuyen en un momento dado, de forma no nula, a la salida correspondiente. Siendo el número máximo de reglas simultáneamente activas igual a $(GSM)^c$.

La realización de un algoritmo borroso, en una estructura paralela VLSI, requiere en general la aplicación de un cierto grado de secuencialización, debido al número considerable de operaciones a realizar para procesar un instancia del mismo.

En el algoritmo borroso, de múltiples antecedentes y consecuentes (sección 3):

$$Y_j = \vee / (i=1..r) (B_{ij} \circ (\wedge / (k=1..e) (A_{ik}^T \circ X_k))) \quad (j=1, \dots, s)$$

donde $X_k = [x_{ku}]$ ($u=1, \dots, n$) e $Y_j = [y_{jv}]$ ($v=1, \dots, m$), el número de índices involucrados es de cinco. La dimensión del grafo de dependencias de datos asociado será también cinco. El secuenciado del algoritmo, puede interpretarse, de acuerdo al método de proyección de dependencias [KUNG 88] [QUINTON 89], como la proyección de este grafo de dimensión 5 en un subespacio de dimensión menor (función de ubicación). La correspondiente función de planificación temporal, permite determinar el orden de ejecución de las operaciones en los diversos procesadores, de forma que diferentes procesos no queden asignados al mismo procesador, y que se respete el orden impuesto por las dependencias entre los datos.

Los tipos de secuencialización generalmente utilizados en un procesador borroso son:

I) A nivel de bit [TOGAI 86]. La UCP en este caso es bastante compleja, aunque se simplifica considerablemente la UI y el BUS.

II) A nivel de palabra [WATANABE 90] [LIM 90]. Equivalente a la asignación temporal de los índices de borrosificación u y v . Este tipo es comunmente empleado en los procesadores borrosos actuales.

III) A nivel de salida [LIM 90]. Equivale a la asignación temporal del índice j . Puede implicar la ejecución en paralelo de las reglas R_i , si se quiere obtener una velocidad de cómputo adecuada. La asignación de reglas a los correspondientes procesadores de la UI es conveniente que sea dinámica, de forma que sólo se procesen las reglas activas [IKEDA 91] [CHIUE 92].

IV) A nivel de entrada [LIM 90]. Equivale a la asignación temporal del índice k . La utilización de este nivel implica la adaptación simple del procesador a un algoritmo con cualquier número de antecedentes.

V) A nivel de regla [ISIK 88] [TOGAI 86]. Equivale a la asignación temporal del índice i . La utilización de este nivel implica también la adaptación simple del procesador a un algoritmo con cualquier número de reglas. Si se quiere ejecutar una instancia del algoritmo a gran velocidad, se debe calcular las salidas Y_j en paralelo. Se acelerará el cómputo correspondiente si se evalúan únicamente, en la UI, las reglas activas en cada instante.

En general, al diseñar un procesador borroso, se utilizan varios de los niveles de secuenciado arriba mencionados, siendo a veces necesaria la utilización de registros de almacenamiento intermedios [LIM 90] y/o registros de desplazamiento [TOGAY 86] para sincronizar adecuadamente el flujo de datos correspondiente.

Las etapas de un algoritmo borroso de mayor complejidad computacional, son fundamentalmente las realizadas en la UI, es decir, la multiplicación matricial (composición máximo-mínimo) en sus diversas formas: producto interno de dos vectores, producto de matriz por vector, y producto de un escalar por un vector (truncamiento).

La etapa de desborrosificación, mediante el cálculo del centro de área:

$$Cda_y = (+/(v=1..m) y(v) v) / (+/(v=1..m) y(v)) = T / S$$

tiene también una complejidad computacional importante. Dicha complejidad se puede reducir de forma considerable, al eliminar los productos correspondientes, mediante el siguiente algoritmo equivalente, expresado en forma de ecuaciones de recurrencia:

$$\begin{aligned} (v=m) \quad S(v) &= y(v) \\ (v=m-1..1) \quad S(v) &= y(v) + S(v+1) \\ (v=m) \quad T(v) &= S(v) \\ (v=m-1..1) \quad T(v) &= S(v) + T(v+1) \\ S &= S(1) \\ T &= T(1) \end{aligned}$$

El circuito de cálculo del centro del área que aparece en [WATANABE 90] utiliza implícitamente esta recurrencia. El cociente correspondiente puede calcularse con un procesador simple, compuesto de un restador, un comparador con cero y un contador síncrono; restando de forma repetida al numerador el denominador, hasta que se anule. Cuando la latencia global del sistema sea también importante será necesario utilizar un divisor más especializado.

En el siguiente apartado se presenta el diseño de un procesador sistólico del tipo II, para acelerar la ejecución del producto matricial (máximo-mínimo) de matriz por vector en la UI. Este procesador es también válido también para realizar los diversos productos vectoriales mencionados.

5. Diseño de un Procesador sistólico máx-mín

El algoritmo de multiplicación (máximo-mínimo) matriz-vector:

$$C = A \circ B = [v/(j=1..n) (a(i,j) \wedge b(j))] \quad (i=1,...,m)$$

donde $C < m \times 1 >$ $A < m \times n >$ $B < n \times 1 >$ tienen las dimensiones indicadas, se puede computar sistólicamente de diversas formas (procesadores tipo I apartado 4) [KUNG 88] [QUINTON 89]

[MEGSON 92]. El tipo de comunicación (factor primordial en una arquitectura sistólica) es el mismo que en el producto matricial clásico, siendo únicamente diferentes las operaciones en el procesador elemental (PE). No teniendo mucho sentido el rediseñar estructuras sistólicas, completamente equivalentes a las existentes [MANZOUL 88], para su aplicación a la multiplicación matricial máx-mín.

Teniendo en cuenta las propiedades asociativa y conmutativa de la operación máximo, este algoritmo se puede también ejecutar en una estructura sistólica modular, del tipo independiente del tamaño del problema a tratar y tolerante a fallos (procesadores tipo II apartado 4). Una solución sistólica eficiente de este tipo, es la indicada en [LIN 91]. También en esta referencia se hace una revisión de diferentes estructuras sistólicas existentes, para realizar el producto matriz-vector, de un tamaño genérico, suponiendo fijado el número de procesadores.

A continuación se presenta una solución sistólica alternativa, para realizar el producto matriz-vector, del tipo independiente del problema, que tiene como ventaja el estar formada por PE's muy simples, sin señales específicas de control. Habiéndose simplificado el proceso de agregación (cómputo de resultados parciales) y el control correspondiente. Esta estructura difiere de la de Y. Lin, en que se utiliza un procesador más, para realizar únicamente la función de agregación, con la misma funcionalidad que el resto; en lugar de distribuir la función de agregación entre todos los PE's, lo que hace algo más complejas las células elementales. Este tipo de optimización tiene especialmente sentido para el caso en el que los PE's realicen operaciones de cómputo muy simples, como en este caso (máx-mín).

En la figura 1 se muestra la estructura sistólica obtenida, aplicando las técnicas de resincronización [LEISERSON 83] al sistema semistólico correspondiente. Habiéndose indicado la operación de máximo mediante ($\vee$) y la de mínimo mediante ($\wedge$), y los registros D mediante ($\blacksquare$). Se ha supuesto que en el caso particular indicado: el número de filas de la matriz es $m=3$, el número de columnas es $n=9$, el número de procesadores útiles (descontando el de agregación) disponibles es $p=3$, y el número de bits es $b = 4$.

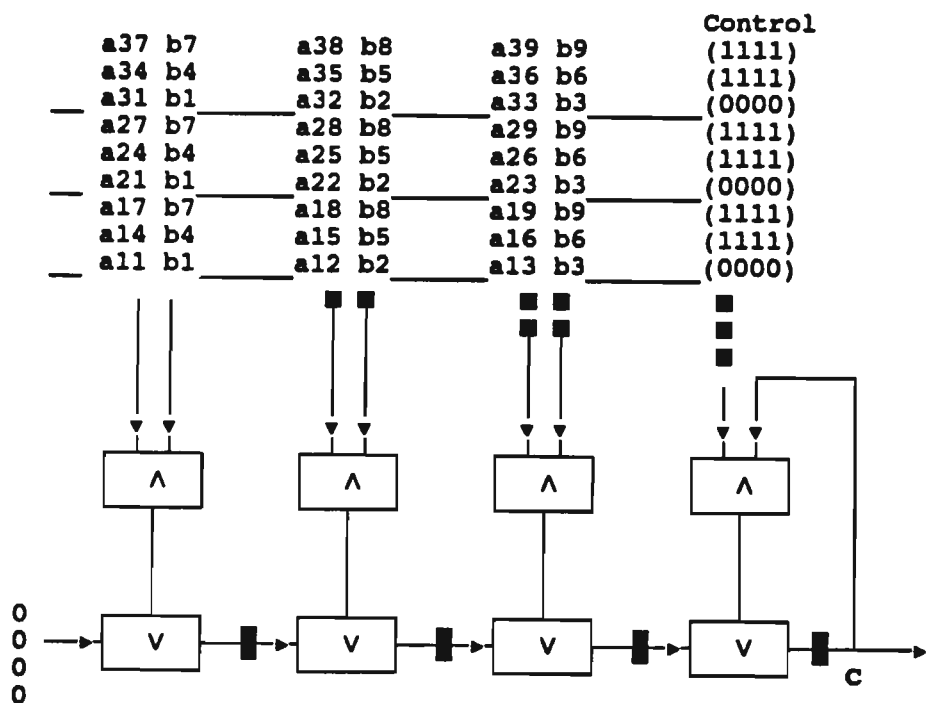


Figura 1. Estructura sistólica para realizar el producto matricial máx-mín de un vector por una matriz.

La señal de control, suponiendo que n es múltiplo de p , consistirá, en general, en un valor mínimo (0000) seguido de $(n/p)-1$ valores máximos (1111). El número total de ciclos de reloj, para computar producto total, es $m(n/p)+p+1$.

Esta estructura sistólica realiza de forma independiente, el producto escalar de los vectores fila de A por el vector columna B , y permite, variando las señales de control y adaptando la configuración de los datos de entrada (A y B), multiplicar matrices de cualquier orden. Esta estructura tiene también un cierto grado de tolerancia a fallos, al ser todos los procesadores elementales idénticos, y permitir ejecutar el algoritmo con un número de procesadores menor.

6. Conclusiones

En este artículo se ha descrito una notación matricial pseudo-APL para la especificación en alto nivel de un algoritmo borroso. Este lenguaje se ha utilizado también para la transformación sistemática entre diversos algoritmos equivalentes, y se ha realizado de forma simple el análisis de la complejidad de los mismos. Se ha descrito, de forma genérica, la estructura y diferentes opciones de diseño de un procesador borroso. Por último se ha mostrado una estructura sistólica para acelerar el producto de matriz-vector, en la unidad de inferencia correspondiente. Esta estructura es capaz de evaluar matrices de cualquier orden y es también tolerante a fallos.

REFERENCIAS

- [BACKUS 78] J. Backus, Can programming be liberated from the von Neumann style? A functional style and its algebra of programs, Commun. ACM vol 21 No 8, pp. 613-641, August 1978.
- [CHIUEH 92] Tzi-cker Chiueh, "Optimization of fuzzy logic inference architecture", Computer IEEE, pp. 67-71, May 1992
- [COSNARD 92] M. Cosnard, M. Nivat, Y. Robert, Algorithmique parallèle, Masson, 1992.
- [DAVIO 83] M. Davio, J. P. Deschamps and A. Thaise, Digital systems with algorithm implementation, John Wiley & Sons, 1983.
- [HORIIE 87] S. Horiie, S. Nishida and T. Sakaguchi, "A design method of systolic arrays under the constraints of the number of the processors", Proc. ICASSP'87 IEEE, Vol. 2, 764-767, 1987.
- [ISIK 88] Can Isik, "Inference hardware for fuzzy rule-based systems", Fuzzy Computing. Theory, Hardware and Applications, pp. 185-194, Elsevier Science Publisher B.V. (North-Holland), 1988.
- [IKEDA 91] H. Ikeda, Y. Hiramoto, N. kisu and H. Takahashi, "A fuzzy processor for a sophisticated automatic transimission control", IFSA'91, Brussels, Vol. Eng., pp. 53-56, 1991.
- [JAGADIS 87] H. V. Jagadish, S. K. Rao and T. Kailath, "Array architectures for iterative algorithms", Proc. of the IEEE, Vol. 75, NO. 9 pp. 1304-1321, 1987.

- [KLIR 88] George J. Klir and Tina A. Folger, Fuzzy sets, uncertainty, and information, Prentice-Hall, 1988.
- [KOSKO 91] Bart Kosko, Neural networks and fuzzy systems, Prentice Hall, 1991.
- [KUNG 88] S. Y. Kung, VLSI array processors, Prentice Hall, 1988.
- [LEE 90] C. C. Lee, "Fuzzy logic in control systems: Fuzzy Logic Controller" Part I & II, IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 2 March/April, pp. 404-435, 1990.
- [LEISER 83] C. E. Leiserson and J. B. Saxe. "Optimizing synchronous systems", Journal of VLSI Computer Systems, 1, pp. 41-67, April 1983.
- [LIM 90] Meng-Hiot Lim, Yoshiyasu Takefuji, "Implementing fuzzy rule-based systems on silicon chips", IEEE Expert, pp. 31-45, February 1990.
- [LIN 91] Yen-Chun Lin, "Array size anomaly of problem-size independent systolic arrays for matrix-vector multiplication", Parallel Computing, 17, 515-522 (1991).
- [MANOHAR 88] Swaminathan Manohar and Gerard Baudet, "A pragmatic approach to systolic design", Proc. International Conference on Systolic Arrays, pp. 463-472, May 1988.
- [MANZOUL 88] M. A. Manzoul and H. A. Serrate, "Fuzzy systolic arrays", Proc. 18th Intern. Symp. Multiple-valued Logics, Spain, 1988.
- [MEGSON 92] G. M. Megson, An introduction to systolic algorithm design, Oxford University Press, 1992.
- [NELIS 87] M. Nelis and Ed. F. Deprettere, "A systematic method for mapping algorithms of arbitrarily large dimensions onto fixed size systolic arrays, Proc. 1987 IEEE International Symposium on Circuits and Systems, Vol 2, pp. 559-563, May 1987.
- [NOVAK 89] Vilem Novák, Fuzzy sets and their application, Adam Hilger, Bristol, 1989.
- [PATYRA 91] M. J. Patyra, VLSI implementation of fuzzy logic circuits, Proc. IFSA'91, pp. 171-174, 1991.
- [QUINTON 89] P. Quinton, Y. Robert, Algorithmes et architectures systoliques, Ed. Masson, 1989.
- [RAJOPAD 86] S. V. Rajopadhye, Synthesizing systolic arrays with control signals from recurrence equations, Distributed Computing, 3, 88-105, 1989.
- [RAO 85] S. K. Rao, Regular iterative algorithms and their implementations on processor arrays, PhD Thesis, Stanford University, USA, 1985.
- [TEICH 91] Jürgen Teich and Lothar Thiele, "Control Generation in the design of processors arrays, Journal of VLSI Signal Processing, 3, 77-92 (1991)
- [TOGAI 86] M. Togai and H. Watanabe, "Expert system on a chip: an engine for real-time

approximate reasoning", IEEE Expert, Vol. 1, No. 3, pp. 55-62, 1986.

- [WATANA 90] H. Watanabe, W. D. Dettloff, and K. E. Yount, "A VLSI fuzzy logic controller with reconfigurable, cascable architecture, IEEE Journal of Solid-State Circuits, pp. 376-381, Vol. 25, No. 2, April 1990.
- [YAMAKAW 88] Takeshi Yamakawa, "Intrinsic fuzzy electronic circuits for sixth generation computer", Fuzzy Computing. Theory, Hardware and Applications, pp. 157-171, Elsevier Science Publisher B.V. (North-Holland), 1988.

UNA APLICACION DE LAS CADENAS INCIERTAS DE KAUFMANN Y GIL ALUJA FRENTE A LAS CADENAS DE MARKOV AL CONTROL DE GESTION DE TESORERIA DE LAS EMPRESAS¹

Autores: Enrique López González y Cristina Mendaña Cuervo
Departamento de Dirección y Economía de la empresa
Universidad de León
Campus de Vegazana s/n
E 24071 León (España)
Tfno.: (987) 20 10 52
Fax: (987) 21 01 98

Resumen: En el presente trabajo se realiza un análisis comparativo entre las cadenas inciertas de Kaufmann y Gil Aluja y las cadenas estocásticas de Markov en su aplicación al control de gestión de la liquidez, interesándose por contrastar la validez de estos instrumentos operativos, en orden a evaluar distintas alternativas de sistemas de control de la tesorería de la empresa, atendiendo al ambiente aleatorio o incierto en que se desarrolla.

Palabras Clave: Kaufmann and Gil Aluja Fuzzy Chains, Markov Chains, Fuzzy Cash Management.

1. Introducción

La complejidad de los problemas que la gestión de las empresas plantea hace cada vez más difícil tomar decisiones sin la ayuda de medios adecuados de tratamiento de la información y de técnicas que la ciencia va aportando.

En este sentido, la teoría markoviana es una de las aplicaciones más elegantes de la teoría de probabilidades. No obstante, existen situaciones donde no podemos emplear probabilidades, como cuando nos situamos en la incertidumbre en lugar de la aleatoriedad, pero aún en este caso es posible el tratamiento de la información.

De acuerdo con lo anterior, nuestro trabajo va a consistir en llevar a cabo un estudio comparativo entre las cadenas inciertas de Kaufmann y Gil Aluja y las cadenas estocásticas, referido a la problemática que presenta para el control de gestión empresarial la tesorería, esto es, la administración de las disponibilidades de efectivo de las empresa, donde se presentan tres problemas principales: primero, no mantener saldos de liquidez ociosos (es decir, produciendo una rentabilidad por debajo de una alternativa mejor conseguible); segundo, no financiar descubierto a tasas más altas que otras posibles; y tercero, evitar que

¹Investigación financiada con una Ayuda de la Junta de Castilla y León. (P.I. n° 1402/90).

de forma simultánea se esté utilizando una línea de financiación a tipo alto y se tengan recursos excedentes a tipos menores.

En la Figura 1 se ilustra un sistema típico de gestión de la liquidez. Diariamente pueden tener lugar flujos monetarios positivos y negativos. En cuanto a los primeros, pueden servir para cubrir los segundos cuando éstos se produzcan, pero puede existir un periodo de tiempo entre unos y otros en el cual sería absurdo mantener saldos de caja ociosos, tal como se mencionó anteriormente.

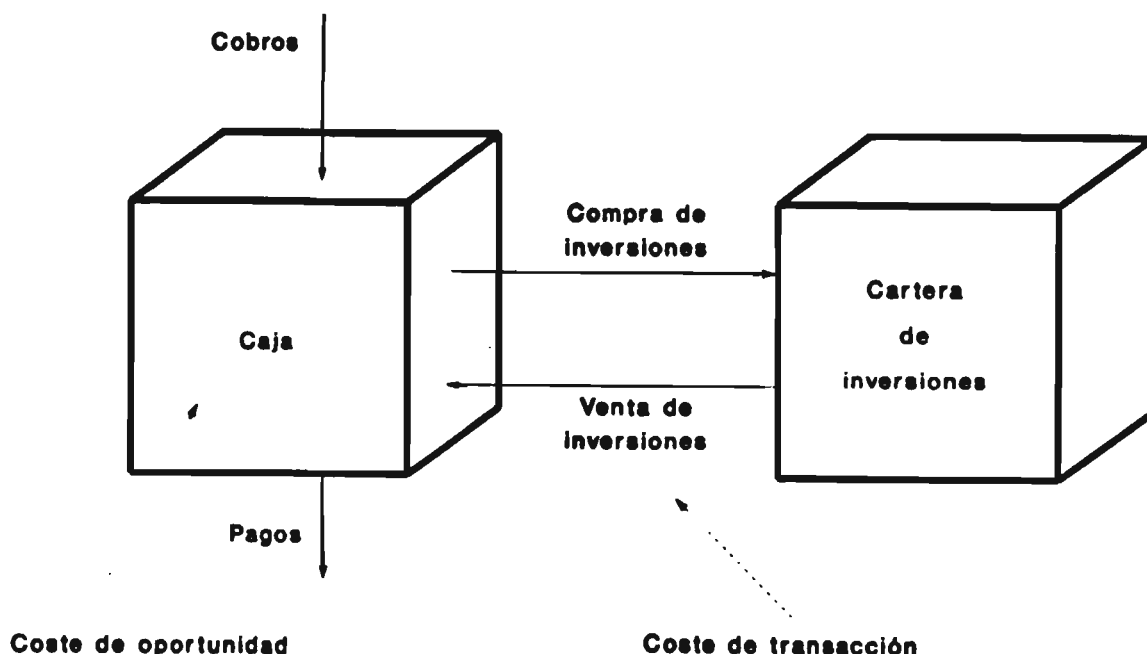


Fig. 1. Sistema de gestión de la liquidez.

En este sistema hay dos costes relevantes a considerar: (1) el coste de oportunidad (esto es, la pérdida de réditos) por mantener valores en efectivo en lugar de tenerlos en la Cartera de valores y (2) el coste de transacción que ocurre al mudar valores de la Cartera a la Caja o viceversa y que están relacionados con las comisiones percibidas por el agente o corredor que en último término realiza la operación de compra-venta. En consecuencia, el control de gestión de la liquidez supone la resolución de dos cuestiones interdependientes: la fijación del nivel óptimo de liquidez y la determinación de la periodicidad y montante de las transferencias entre la Caja y la Cartera de Inversiones (activos líquidos), pues, la mayor parte de las empresas registran entradas y salidas de caja a lo largo del tiempo que no observan un ritmo regular o predecible.

Nuestro estudio comenzará entonces por la situación donde se consideran los flujos de caja como una variable aleatoria, planteándose el interés de la aplicación de las cadenas de Markov. No obstante, si nuestro conocimiento del entorno es impreciso, como puede ocurrir en la toma de decisiones de gestión de la liquidez, el modelo debería incluir la noción de cadenas con datos borrosos en lugar de probabilidades.

Por consiguiente, el propósito de nuestro trabajo radicará en el análisis de estos instrumentos operativos en orden a evaluar distintas alternativas de sistemas de control de la tesorería de las empresas.

2. Aplicación de las cadenas de Markov al control de gestión de la liquidez

Dado el carácter restrictivo de suponer conocido el comportamiento de los flujos de caja, la literatura contable relacionada se ha centrado en la consideración de los mismos como variables aleatorias, de ahí que los modelos propugnados en orden a determinar cuando y por cuanto deben realizarse las transferencias entre la Caja y la Cartera de Inversiones se basan principalmente en establecer unos niveles límite de caja que sirvan de guía para el responsable de la gestión de liquidez (controller).

En este sentido, a modo ilustrativo, podemos considerar que los límites superior e inferior de control del saldo de caja, tal como se muestra en la Figura 2, son X e Y , mientras que Z designa el saldo de caja considerado como objetivo a alcanzar. De esta forma, cuando el saldo de caja alcanza un nivel X , entonces se deben invertir $(X - Z)$ unidades monetarias en valores; cuando el saldo desciende hasta Y , es preciso vender $(Z - Y)$ unidades monetarias de valores o activos líquidos obteniéndose así la suma de dinero correspondiente.

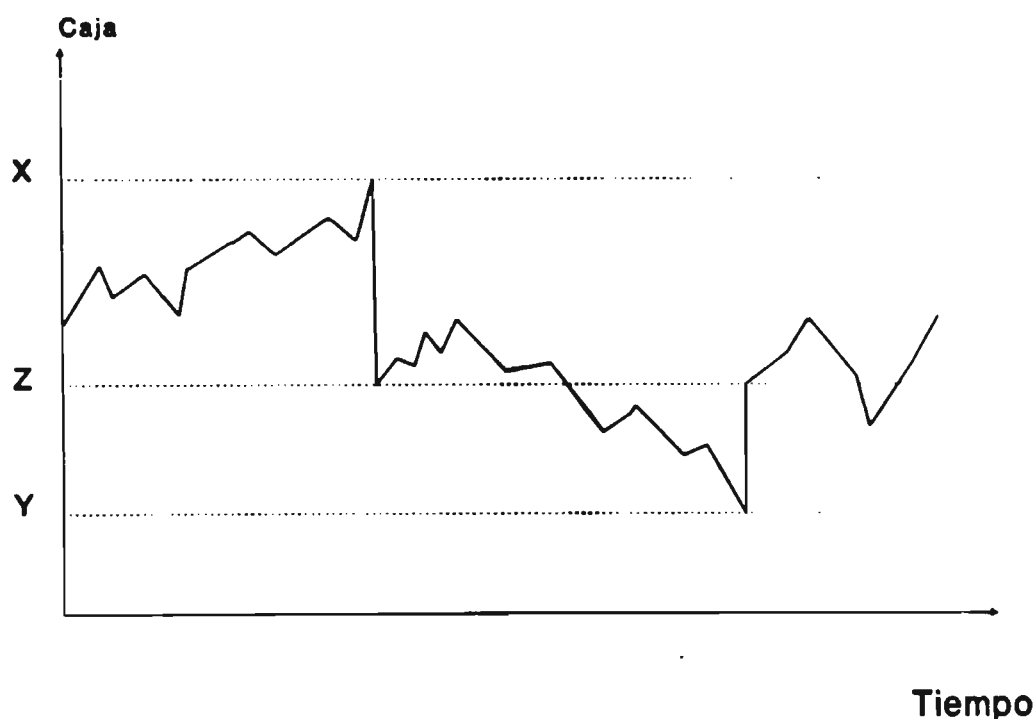


Fig. 2. Perfil de evolución de los flujos de caja.

Por tanto, al controller le interesa conocer la probabilidad estimada para vender valores de la Cartera de Inversiones, por ejemplo, dos días después de haber inspeccionado el estado del saldo de caja. Esta pregunta es de su incumbencia ya que según donde se encuentre el saldo de caja deberá o no vender valores para obtener efectivo y, así, poder planificar entonces cuáles serán aquellos valores que deberá vender.

Para resolver la cuestión anterior presenta gran utilidad la noción de cadena de Markov que es un modelo de proceso aleatorio en el transcurso del tiempo y cuya característica fundamental radica en que la probabilidad de que el sistema bajo estudio esté en una condición particular depende sólo de su condición actual.

Construyamos entonces el modelo considerado arriba para el problema del control de gestión de la liquidez mediante una cadena de Markov; para ello, supongamos un sistema S que puede tomar N estados $E_1, E_2, \dots, E_N$ (N finito). Se admite que es posible medir o calcular la probabilidad de ocurrencia de que S esté en el Estado E_i , $i = 1, 2, \dots, N$ y esto para todos los momentos en tiempo discreto $n = 0, 1, 2, 3, \dots$ y se escribe para estas probabilidades $p_i(n)$, $i = 1, 2, \dots, N$, $n = 0, 1, 2, 3, \dots$. Las $p_i(n)$ forman los elementos del conjunto de "vectores de estado" que describen el sistema para todos los momentos del futuro considerados, esto es, la probabilidad de que el sistema se encuentre en el estado E_i en la fecha n .

$$[p(n)] = \begin{array}{c} E_1 \quad E_2 \quad \dots \quad E_N \\ \boxed{p_1(n)} \quad \boxed{p_2(n)} \quad \dots \quad \boxed{p_N(n)} \end{array} \quad (1)$$

Se establece la hipótesis de que el paso de un estado E_i a un estado E_j sólo depende de E_i y de E_j . Se supone que es posible medir o calcular la probabilidad de todo par (E_i, E_j) , $i, j = 1, 2, \dots, N$, que llamaremos p_{ij} sabiendo que el sistema se hallará en el estado E_j en el momento $n+1$ y está en el estado E_i en el momento n . Se establecen entonces como ecuaciones que rigen el sistema:

$$p_j(n+1) = \sum_{i=1}^N p_i(n) \cdot p_{ij} \quad j = 1, 2, \dots, N \quad (2)$$

Se supone aquí que las p_{ij} no dependen de n (cadena estacionaria). Podemos representar (2) también bajo la forma matricial:

$$\begin{array}{c} \begin{array}{c} E_1 \quad E_2 \quad \dots \quad E_N \\ \boxed{p_1(n+1)} \quad \boxed{p_2(n+1)} \quad \dots \quad \boxed{p_N(n+1)} \end{array} = \\ = \begin{array}{c} E_1 \quad E_2 \quad \dots \quad E_N \\ \boxed{p_1(n)} \quad \boxed{p_2(n)} \quad \dots \quad \boxed{p_N(n)} \end{array} \circ \end{array}$$

$$\begin{array}{c} \circ \quad \begin{array}{c} E_1 \quad E_2 \quad \dots \quad E_N \\ \begin{array}{|c|c|} \hline p_{11} & p_{12} \\ \hline p_{21} & p_{22} \\ \hline \vdots & \vdots \\ \hline p_{N1} & p_{N2} \end{array} \quad \dots \quad \begin{array}{|c|} \hline p_{1N} \\ \hline p_{2N} \\ \hline \vdots \\ \hline p_{NN} \end{array} \end{array} \quad (3)$$

en donde $\circ$ significa la composición "suma-producto" (filas por columnas).

Se puede escribir también de forma simplificada:

$$[p(n+1)] = [p(n)] \cdot [M] \quad (4)$$

La matriz de transición formada por las probabilidades p_{ij} , y que se designa por $[M]$ se llamará "matriz estocástica" y poseerá las propiedades que se señalan a continuación:

$$p_i(n) \geq 0, \quad j = 1, 2, \dots, N \quad n = 0, 1, 2, 3, \dots \quad (5)$$

$$\sum_{j=1}^N p_j(n) = 1, \quad n = 0, 1, 2, 3, \dots \quad (6)$$

$$p_{ij} \geq 0, \quad i, j = 1, 2, \dots, N \quad (7)$$

$$\sum_{j=1}^N p_{ij} = 1, \quad i = 1, 2, \dots, N \quad (8)$$

En particular, supongamos que la matriz de la Figura 3 describe el comportamiento del balance de caja. Los números de la columna vertical (filas) son la cantidad de dinero (unidades monetarias) en la contabilidad de caja en la mañana del día n . Los números de la fila superior (columnas) son el balance de caja en unidades monetarias (u.m.) en la mañana del día $n+1$. La tabla señala, por ejemplo, que si el balance de caja es 2 (20.000 u.m.) en el día n , será 1 (10.000 u.m.) el día $n+1$ con probabilidad de 0.177 (el número marcado en la fila 2, columna 1, es 0.177). De esta forma, la Figura 3 muestra la matriz de transición del proceso, basado en el sistema de control descrito en la Figura 2, donde se supone una cota superior de 5 y una cota inferior de 1, con el punto de retorno igual a 2. En dicha matriz las filas 2, 3 y 4 muestran los cálculos o medidas de la probabilidad de que el sistema se mueva de un estado a otro. Para deducir las probabilidades de los estados 1 ó 5 hay que recordar que cuando el controller llega el día n y encuentra el sistema en el estado 1 ó 5, inmediatamente lo transfiere al estado 2, de ahí que los datos de las filas 1, 5 y 2 son los mismos.

	1	2	3	4	5
1	.177	.588	.235	0	0
2	.177	.588	.235	0	0
3	0	.29	.184	.526	0
4	0	0	.324	.541	.135
5	.177	.588	.235	0	0

Fig.3. Matriz de transición del balance de caja.

La matriz estocástica de la Figura 3 es la matriz de transición de un paso para este problema, facilitándonos conocer la probabilidad de que el sistema se mueva de un estado a otro en un paso. De esta forma, para obtener la matriz de transición de dos pasos basta con multiplicar la de un paso por sí misma; con lo cual podemos generalizarlo en forma simbólica expresando que si $[M]$ es la matriz de transición de un paso, la matriz de transición de n pasos será $[M]^n$, que significa el producto de n factores iguales a $[M]$.

La matriz de transición de n pasos facilita al controller contestar a la pregunta planteada anteriormente, esto es, conocer la probabilidad de que el sistema esté en un estado específico después de n transiciones. Para lo cual si en la expresión (4) hacemos $n = 0$, obtenemos:

$$[p(1)] = [p(0)] \circ [M]$$

En forma semejante, $[p(2)] = [p(1)] \circ [M]$ y por sustitución,

$$[p(2)] = [p(1)] \circ [M] = [p(0)] \circ [M] \circ [M] = [p(0)] \circ [M]^2$$

Por lo tanto, se podrá escribir la expresión (4) como sigue:

$$[p(n)] = [p(0)] \circ [M]^n \quad (9)$$

en donde $p(0)$ es el vector de probabilidades de encontrar el sistema en un estado particular en el instante 0, esto es, cuando se inspecciona el sistema por primera vez.

En nuestro caso de aplicación, supóngase que el controller llega el lunes por la mañana y comprueba que el sistema está en el estado 3 (30.000 u.m.) y desea saber la probabilidad de encontrar el sistema en el estado 1 el miércoles por la mañana, ya que entonces deberá vender valores para obtener dinero y le convendrá determinar por adelantado cuáles valores de su cartera deberá vender.

De acuerdo con la expresión (9), obtenemos

$$[p(2)] = [p(0)] \circ [M]^2 =$$

$$= [0 \ 0 \ 1 \ 0 \ 0] \circ \begin{array}{|c|c|c|c|c|} \hline .135 & .518 & .223 & .124 & 0 \\ \hline .135 & .518 & .223 & .124 & 0 \\ \hline .051 & .224 & .272 & .381 & .071 \\ \hline .024 & .173 & .267 & .463 & .073 \\ \hline .135 & .518 & .223 & .124 & 0 \\ \hline \end{array} =$$

$$= [.051 \ .224 \ .272 \ .381 \ .071]$$

Puesto que el dato de la columna 1 es 0.051, el controller sabe que la probabilidad de tener que transferir fondos de la cartera a la caja es sólo de 0.051 con lo cual no parece prudente dedicar mucho tiempo a determinar qué valores se venderían si hubiese necesidad de ello el miércoles.

Por otro lado, la expresión (9) también puede utilizarse para especificar la probabilidad de que el sistema se mueva de un estado a otro en cualquier número de transiciones si las probabilidades de un paso no cambian con el tiempo, lo cual nos conduce a la consideración de las probabilidades estacionarias, es decir, la propiedad ergódica en una cadena de Markov (KAUFMANN, 1966).

De esta forma, si $\lim_{n \rightarrow \infty} [M]^n = [M]$

donde $[M]$ es una matriz estocástica sin ningún elemento nulo, se dice que el sistema es ergódico, que posee un régimen permanente. La matriz $[M]$ que posee esta propiedad se denomina *matriz ergódica*. Hay una prueba en dos partes que dice si una cadena de Markov es ergódica: (a) debe haber una probabilidad positiva de que se pueda pasar de un estado i a otro cualquiera j si hay suficientes transiciones y (b) debe existir al menos un estado con probabilidad positiva de que el sistema permanezca en él después de una transición.

Si $[M]$ es tal que todas sus líneas son idénticas entonces se dice que el sistema es "completamente ergódico", y en este caso, para n suficientemente grande, el estado del sistema no depende del estado inicial. En efecto:

$$[p(n)] = [p(0)] \circ [M]^n$$

$$\lim_{n \rightarrow \infty} [p(n)] = [p(0)] \circ [M^\infty] \quad (10)$$

$$= [p]$$

en donde $[p]$ es una de las filas idénticas de $[M^\infty]$.

Para calcular las probabilidades asintóticas $[p]$ se disponen de varios métodos, uno de los más sencillos, especialmente si se puede utilizar la opción de multiplicación de matrices incluida en la mayoría de las Hojas Electrónicas de Cálculo, consiste en calcular $[M]^n$ hasta que $[M]^{n+1} = [M]^n$.

$$\begin{array}{c}
 [M] \\
 \begin{array}{|c|c|c|c|c|}
 \hline
 .177 & .588 & .235 & 0 & 0 \\
 \hline
 .177 & .588 & .235 & 0 & 0 \\
 \hline
 0 & .29 & .184 & .526 & 0 \\
 \hline
 0 & 0 & .324 & .541 & .135 \\
 \hline
 .177 & .588 & .235 & 0 & 0 \\
 \hline
 \end{array}
 \end{array}
 \circ
 \begin{array}{c}
 [M] \\
 \begin{array}{|c|c|c|c|c|}
 \hline
 .177 & .588 & .235 & 0 & 0 \\
 \hline
 .177 & .588 & .235 & 0 & 0 \\
 \hline
 0 & .29 & .184 & .526 & 0 \\
 \hline
 0 & 0 & .324 & .541 & .135 \\
 \hline
 .177 & .588 & .235 & 0 & 0 \\
 \hline
 \end{array}
 \end{array}
 =
 \begin{array}{c}
 [M]^2 \\
 \begin{array}{|c|c|c|c|c|}
 \hline
 .135 & .518 & .223 & .124 & 0 \\
 \hline
 .135 & .518 & .223 & .124 & 0 \\
 \hline
 .051 & .224 & .272 & .381 & .071 \\
 \hline
 .024 & .173 & .267 & .463 & .073 \\
 \hline
 .135 & .518 & .223 & .124 & 0 \\
 \hline
 \end{array}
 \end{array}$$

.....

$$[M]^{16} =$$

.083	.348	.248	.284	.038
.083	.348	.248	.284	.038
.083	.347	.248	.284	.038
.083	.347	.248	.284	.038
.083	.348	.248	.284	.038

$$[M]^{17} =$$

.083	.347	.248	.284	.038
.083	.347	.248	.284	.038
.083	.347	.248	.284	.038
.083	.347	.248	.284	.038
.083	.347	.248	.284	.038

En nuestro caso de aplicación, tenemos

$$[p] = [.083 \ .347 \ .248 \ .284 \ .038]$$

Podemos observar que $p(1) = 0.083$ y como hay 260 días laborales en el año, entonces el número de veces en las que el sistema estará en el estado 1 en dicho año será de $0,083 \times 260 = 21.58$ veces. Por consiguiente, "en promedio" habrá que mover valores de la Cartera a la Caja 21.58 veces por año y una vez cada $260/21,58 \approx 12$ días. Un análisis similar para el estado 5 produce $0.038 \times 260 = 9.88$ movimientos de la Caja a la Cartera por año.

Esta información presenta gran interés en la planificación de la periodicidad y montante de las transacciones entre la Caja y la Cartera de inversiones.

No obstante, a pesar del atractivo de la aplicación de las cadenas de Markov, conviene recordar que para poder aplicar una probabilidad es necesario tener, en primer lugar, una sucesión de fenómenos que se hayan repetido en unas determinadas condiciones y, en segundo lugar, poder aplicar los resultados obtenidos sobre otro fenómeno sometido a las mismas condiciones que las anteriores. En este sentido, existen situaciones en que no es posible utilizar probabilidades, cuando existe incertidumbre en lugar de azar. Si nuestro conocimiento del entorno es impreciso, como ocurre en la toma de decisiones de gestión de la liquidez, el modelo debería incluir la noción de cadenas con datos borrosos en lugar de probabilidades.

2. Aplicación de las cadenas inciertas de Kaufmann y Gil Aluja al control de gestión de la liquidez

En un ambiente de incertidumbre el modelo debe incluir la noción de nivel de presunción, esto es, en lugar de una ley de probabilidad se deben tomar leyes de posibilidad. A este respecto, la propuesta de KAUFMANN y GIL ALUJA (1991) está basada en la teoría markoviana, aunque presenta modificaciones importantes, pues, al situarse en ambiente de incertidumbre se no emplean probabilidades sino datos borrosos.

Construyamos entonces el modelo considerado anteriormente para el problema de la gestión de liquidez, ahora en ambiente de incertidumbre, mediante una "cadena incierta de Kaufmann y Gil Aluja". Para ello, siguiendo a dichos autores, supongamos un sistema S cuyo conjunto de estados es $E = \{E_1, E_2, \dots, E_N\}$ (N finito). Se admite que es posible valorar en $[0, 1]$ la posibilidad de ocurrencia de que S se encuentre en el estado E_i , $i = 0, 1, 2, \dots, N$ y esto para todas las fechas $n = 0, 1, 2, 3, \dots$ y que se designan estas posibilidades $q_i(n)$, $i = 0, 1, 2, \dots, N$, $n = 0, 1, 2, 3, \dots$. Las $q_i(n)$ forman los elementos del conjunto de vectores de estado que describen el sistema para todos los momentos futuros considerados.

$$[q(n)] = \begin{array}{c} E_1 \quad E_2 \quad \dots \quad E_N \\ \boxed{q_1(n)} \quad \boxed{q_2(n)} \quad \dots \quad \boxed{q_N(n)} \end{array} \quad (11)$$

Se establece la hipótesis que las $q_i(n)$ forman una ley de posibilidad para todo n , es decir, que se tiene que:

$$\bigvee_{i=1}^N q_i(n) = 1 \quad (12)$$

En otras palabras, por lo menos uno de los estados tiene la posibilidad 1.

Se parte también de la hipótesis que la posibilidad de paso de un estado E_i , a un estado E_j sólo depende de E_i y de E_j . Se supone que se puede valorar la posibilidad de todo par (E_i, E_j) , $i, j = 1, 2, \dots, N$, es decir, q_{ij} , sabiendo que el sistema se hallará en el estado E_j en el momento $n+1$ y está en el estado E_i en el momento n . Se establece entonces como ecuación que rige el sistema:

$$q_j(n+1) = \bigvee_{i=1}^N (q_i(n) \wedge q_{ij}(n)), \quad j = 1, 2, \dots, N \quad (13)$$

Se supone aquí que las q_{ij} no dependen de n (cadena estacionaria). Por otra parte:

$$\forall i \in \{1, 2, \dots, N\} : \exists j \in \{1, 2, \dots, N\}$$

$$\bigvee_{i=1}^N q_{ij} = 1 \quad (14)$$

Esto es, cada fila de la relación contiene por lo menos un 1.

De esta manera, las q_{ij} forman una relación borrosa normal que se va a denominar "matriz incierta" por comparación con la "matriz estocástica" y, a su vez, el sistema construido en base a la misma se denominará "cadena incierta" (sobreentendiéndose con filas normales).

Podemos representar de nuevo la expresión (13) ahora bajo forma matricial:

$$\begin{array}{c}
 \begin{array}{ccc}
 E_1 & E_2 & \\
 \hline
 q_1(n+1) & q_2(n+1) & \dots\dots q_N(n+1)
 \end{array}
 = \\
 \\
 = \begin{array}{ccc}
 E_1 & E_2 & \\
 \hline
 q_1(n) & q_2(n) & \dots\dots q_N(n)
 \end{array}
 * \begin{array}{ccc}
 E_1 & E_2 & \\
 \hline
 q_{11} & q_{12} & \dots q_{1N} \\
 E_2 & q_{21} & q_{22} & \dots q_{2N} \\
 \vdots & \vdots & \ddots \vdots \\
 E_N & q_{N1} & q_{N2} & \dots q_{NN}
 \end{array}
 \quad (15)
 \end{array}$$

en donde * significa la composición maxmin (filas por columnas), o también

$$\mu_N(x, z) = \bigvee_y (\mu_N(x, y) \wedge \mu_N(y, z)) \quad (16)$$

Dado que las $q_i(n)$ y q_{ij} son valuaciones, tendrá que cumplirse que:

$$q_j(n) \geq 0, \quad \begin{array}{l} j = 1, 2, \dots N \\ n = 0, 1, 2, 3, \dots \end{array} \quad (17)$$

$$\bigvee_{j=1}^N q_j(n) = 1, \quad n = 0, 1, 2, 3, \dots \quad (18)$$

$$q_{ij} \geq 0 \quad (19)$$

$$\bigvee_{j=1}^N q_{ij}(n) = 1, \quad i = 1, 2, \dots N \quad (20)$$

Se escribirá también (15) bajo la forma simplificada:

$$[q(n+1)] = [q(n)] * [\underline{N}] \quad (21)$$

en donde $[\underline{N}]$ es la matriz formada por las valuaciones q_{ij} introducida en (15). De esta manera $[\underline{N}]$ es una relación borrosa de filas normales según la terminología de la teoría de los subconjuntos borrosos. En el supuesto de que las q_{ij} dependan del tiempo y sean $q_{ij}(n)$ entonces $[\underline{N}(n)]$ representa una cadena llamada "no estacionaria".

Podemos ampliar nuestra consideración, por el interés pedagógico que presenta, a la noción de relaciones Φ -borrosas normales.

Una relación Φ -borrosa normal posee valuaciones constituídas por intervalos de confianza de $[0, 1]$ en lugar de estimaciones en $[0, 1]$. Por ello se afectará a cada transición un intervalo de confianza:

$$[a_1, a_2] \subset [0, 1] \quad 0 \leq a_1 \leq a_2 \leq 1 \quad (22)$$

pero, dado que la relación es normal, existirá siempre por lo menos un 1 en cada fila.

Conviene recordar también que:

$$[a, a] = a \quad a \in [0, 1] \quad (23)$$

$$[a_1, a_2] (\wedge) [b_1, b_2] = [a_1 \wedge b_1, a_2 \wedge b_2] \quad (24)$$

$$[a_1, a_2] (\vee) [b_1, b_2] = [a_1 \vee b_1, a_2 \vee b_2] \quad (25)$$

La composición maxmin se realizará con los intervalos de confianza de la misma manera que con los valores de $[0, 1]$, respetando (23), (24) y (25). Los intervalos de confianza de $[0, 1]$ son conmutativos, asociativos, distributivos, etc..., tienen las mismas propiedades algebraicas que los valores de $[0, 1]$. En definitiva, una cadena Φ -borrosa normal puede tratarse de la misma manera que una cadena borrosa normal, aunque para diferenciarlas a efectos simbólicos, se exprese por $[N]$.

En particular, supongamos que la matriz de la Figura 4 describe el comportamiento estimado como posible del balance de caja. Los números de la primera columna vertical (filas) son la cantidad de dinero (u.m.) en la contabilidad de caja en la mañana del día n . Los números de la fila superior (columnas) son el balance de caja en unidades monetarias en la mañana del día $n+1$. La tabla señala que, por ejemplo, si el balance de caja es 2 en el día n , la posibilidad de que sea 1 el día $n+1$ es de $[0.2, 0.4]$ (el número marcado en la fila 2, columna 1, es de $[0.2, 0.4]$).

	1	2	3	4	5
1	[.2, .4]	1	[.3, .5]	0	0
2	[.2, .4]	1	[.3, .5]	0	0
3	0	[.4, .7]	.35	1	0
4	0	0	.6	1	[.1, .4]
5	[.2, .4]	1	[.3, .5]	0	0

Fig. 4. Relación Φ -borrosa normal del balance de caja.

De esta forma, la Figura 4 muestra la matriz incierta de transición donde se supone a efectos del sistema de control de gestión de la liquidez, una cota superior de 5 y una cota inferior de 1, con el punto de retorno igual a 2. En dicha matriz, las filas 2, 3 y 4 muestran las valuaciones (opinión subjetiva) de que el sistema se mueva de un estado a otro y las filas 1 y 5 se deducen de forma semejante que en el caso descrito en el epígrafe anterior.

Esta relación Φ -borrosa normal facilita al controller contestar a la pregunta sobre el nivel de presunción de que el sistema se encuentre en un estado específico en un momento futuro cercano, pues, al utilizar valuaciones, resultará interesante conocer, a partir de estos datos subjetivos, los resultados para los que la presunción es más elevada.

Para ello, de forma similar a lo considerado en la expresión (10), se calculará $[\tilde{N}]^n$ hasta que $[\tilde{N}]^{n+1} = [\tilde{N}]^n$, donde se tiene

$$\lim_{n \rightarrow \infty} [\tilde{N}]^n = [\tilde{N}^*]$$

No obstante, con la composición maxmin es conveniente adoptar como relación Φ -borrosa normal límite el cierre transitivo $[\hat{N}]$, es decir:

$$[\hat{N}] = [\tilde{N}] \cup [\tilde{N}]^2 \cup [\tilde{N}]^3 \dots \quad (26)$$

ya que, exceptuando el caso en que $[\tilde{N}]$ es reflexiva no siempre se cumple que $[\tilde{N}]^{n+1} \supset [\tilde{N}]^n$ y el camino más "fuerte" en el grafo valuado se obtiene considerando el cierre transitivo.

En nuestro caso de aplicación tenemos:

$$[\tilde{N}]$$

.2,.4	1	.3,.5	0	0
.2,.4	1	.3,.5	0	0
0	.4,.7	.35	1	0
0	0	.6	1	.1,.4
.2,.4	1	.3,.5	0	0

$$[\tilde{N}]$$

*

.2,.4	1	.3,.5	0	0
.2,.4	1	.3,.5	0	0
0	.4,.7	.35	1	0
0	0	.6	1	.1,.4
.2,.4	1	.3,.5	0	0

=

$$[\tilde{N}]^2$$

[.2, .4]	1	[.3, .5]	[.3, .5]	0
[.2, .4]	1	[.3, .5]	[.3, .5]	0
[.2, .4]	[.4, .7]	.6	1	[.1, .4]
[.1, .4]	[.4, .7]	.6	1	[.1, .4]
[.2, .4]	1	[.3, .5]	[.3, .5]	0

$$[\underset{\sim}{N}]^3 = \begin{array}{|c|c|c|c|c|} \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \\ \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline \end{array}$$

$$[\underset{\sim}{N}]^4 = \begin{array}{|c|c|c|c|c|} \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \\ \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline \end{array}$$

Nos detenemos: $[\underset{\sim}{N}]^4 = [\underset{\sim}{N}]^3$ y se obtiene

$$[\underset{\sim}{N}] = \begin{array}{|c|c|c|c|c|} \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \\ \hline [.2, .4] & 1 & [.3, .5] & [.3, .5] & [.1, .4] \\ \hline \end{array}$$

En nuestro caso de aplicación, supóngase entonces que el controller llega el lunes por la mañana y comprueba que el sistema está en el estado 3 y desea saber cual es la posibilidad de encontrar el sistema en el estado 1 el miércoles por la mañana, ya que entonces deberá vender valores para obtener dinero y a él le convendrá planificar cuáles valores de su cartera deberá vender.

De esta forma, tenemos:

$$\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 1 & 0 & 0 \end{array} * [\underset{\sim}{N}] = \\ \\ = \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \hline [.2, .4] & [.4, .7] & .6 & 1 & [.1, .4] \end{array} \end{array}$$

Por tanto, si el estado inicial del sistema se encuentra en 3 en un momento futuro cercano el máximo de presunción se encontrará en 4. No obstante, centrándonos en la cuestión planteada arriba, podemos observar que se supone una posibilidad de $[0.2, 0.4]$ que el sistema esté en el estado 1 el miércoles.

Por otro lado, conviene señalar que, al utilizar la concepción del máximo de presunción, carece de sentido la noción de estado persistente. En otras palabras, no podemos decir que el número de veces en las que el sistema estará en el estado 1 durante el año sea de $260 \times [0.2, 0.4] = [52, 104]$.

Sin embargo, sí es posible, en lugar de pretender llegar al caso límite, plantear la situación en el futuro después de algunas etapas relativamente cortas. Si las matrices correspondientes tienen un número reducido de estados, estos cálculos pueden tener una significación real mientras que en el caso contrario se necesitará la intervención de una bola de cristal.

4. Análisis comparativo

El objetivo del presente epígrafe consiste en analizar con mayor detalle las principales diferencias que presentan las cadenas inciertas de Kaufmann y Gil Aluja frente a las cadenas de Markov.

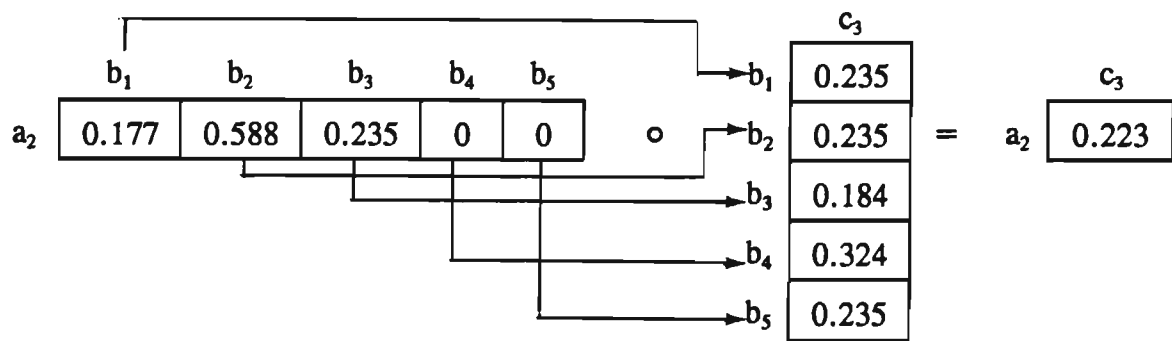
A este respecto, las consideraciones efectuadas anteriormente nos ha permitido observar la necesidad de disponer de técnicas operativas distintas atendiendo al ambiente aleatorio o incierto en que se desarrolla el ámbito de aplicación de las mismas, lo cual supone tener en cuenta una ley de probabilidad en el primer caso mientras que se deben tomar leyes de posibilidad en el segundo.

No obstante, existe un gran paralelismo entre la teoría de probabilidades y la de posibilidades: en la primera, el axioma fundamental es la aditividad en la medida; en la segunda, el axioma fundamental es la monotonía en la inclusión para toda valuación.

A modo de ejemplo ilustrativo podríamos retomar el problema del controller que llega el lunes por la mañana y comprueba que el sistema está en el estado 2 (20.000 u.m.) y le interesa conocer bien la probabilidad (en el caso de considerar el flujo de caja como una variable aleatoria) o bien la posibilidad (si el flujo de caja es una variable incierta) de encontrar el sistema en el estado 3 el miércoles por la mañana.

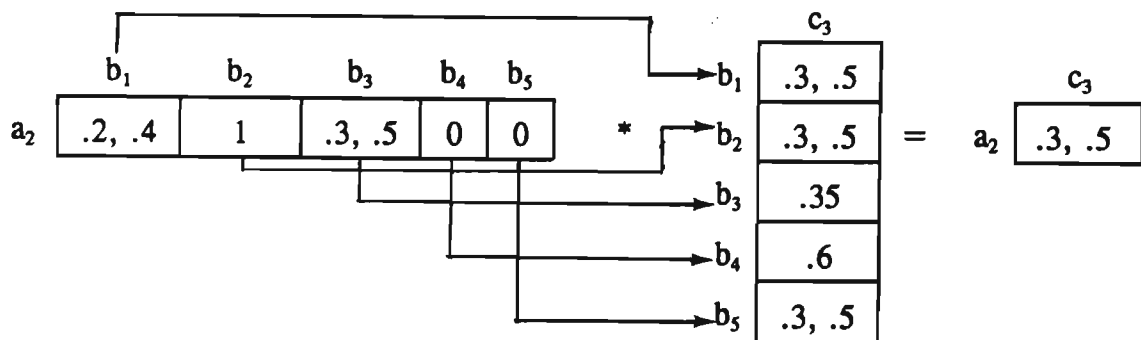
En el primer caso, el estudio de la aplicación de las cadenas de Markov nos ha permitido observar como el cálculo de las probabilidades de transición de dos, tres o, de hecho de n pasos, consiste en hacer una lista de todas las rutas, encontrar la probabilidad de cada una y sumarmas todas, lo cual, como ya se mencionó y es de sobra conocido, nos acerca al concepto de multiplicación de matrices (composición suma-producto) como base conceptual del encadenamiento markoviano.

En nuestro ejemplo de aplicación, la probabilidad de que el controller encuentre el sistema en el estado 3 después de dos pasos, si el estado inicial era 2, es de 0.223 que se obtiene como sigue a continuación:



$$P(a_2 c_3) = [(a_2 b_1) \cdot (b_1 c_3)] + [(a_2 b_2) \cdot (b_2 c_3)] + [(a_2 b_3) \cdot (b_3 c_3)] + [(a_2 b_4) \cdot (b_4 c_3)] \\ + [(a_2 b_5) \cdot (b_5 c_3)] = [0.177 \cdot 0.235] + [0.588 \cdot 0.235] + [0.235 \cdot 0.184] + \\ + [0 \cdot 0.324] + [0 \cdot 0.235] \approx 0.223$$

Sin embargo, en un ambiente de incertidumbre el encadenamiento con datos borrosos se efectuará mediante la composición maxmin, tomando las valuaciones mínimas y después la el máximo de los mínimos. En nuestra aplicación, si al controller le interesara conocer cuál es la posibilidad de que después de dos pasos el sistema se encuentre en el estado 3, si el estado inicial fuera 2, operaría como sigue a continuación:



$$\mu(a_2 c_3) = [(a_2 b_1) \wedge (b_1 c_3)] \vee [(a_2 b_2) \wedge (b_2 c_3)] \vee [(a_2 b_3) \wedge (b_3 c_3)] \vee [(a_2 b_4) \wedge (b_4 c_3)] \\ \vee [(a_2 b_5) \wedge (b_5 c_3)] = [(0.2, 0.4) \wedge (0.3, 0.5)] \vee [1 \wedge (0.3, 0.5)] \vee \\ \vee [(0.3, 0.5) \wedge 0.35] \vee [0 \wedge 0.6] \vee [0 \wedge (0.3, 0.5)] = [0.3, 0.5]$$

5. Conclusiones

En este trabajo se estudian técnicas operativas interesadas en facilitar alcanzar los dos objetivos principales en la toma de decisiones de gestión de la liquidez: la fijación del nivel óptimo de liquidez y la determinación de la periodicidad y el montante de las transferencias entre la caja y la cartera de valores. No obstante, este estudio se ocupó especialmente del segundo objetivo. De esta forma, se ha analizado en primer lugar la utilidad que presenta la aplicación de las cadenas de Markov cuando se consideran los flujos de tesorería como una variable aleatoria.

Sin embargo, existen situaciones en que no es posible utilizar probabilidades, es decir, cuando existe incertidumbre en lugar de aleatoriedad, como sucede en el control de gestión de tesorería borrosa. De ahí que el modelo debe incluir la noción de cadenas con datos borrosos en lugar de probabilidades desarrollada por KAUFMANN y GIL ALUJA (1991). Por consiguiente, en este estudio se ha contrastado la utilidad que presentan las "cadenas inciertas de Kaufmann y Gil Aluja", donde el encadenamiento se hace mediante operadores asociados máximo-mínimo (la decisión más procedente), mientras que el encadenamiento markoviano se realizaba a través de operadores asociados suma-producto (el que decide se considera que es objetivo y siempre el más objetivo).

6. Referencias bibliográficas

- CHANAS, S. and KOLODZIEJYK, W.: "Maximun flow in a network with fuzzy arc capacities". Fuzzy Sets & Systems. Vol. 8, N° 2, August 1982, pp. 165-174.
- CZOGALA, E. and ZIMMERMANN, H.J.: "The aggregation operations for decision-making in probabilistic fuzzy environment". Fuzzy Sets & Systems. Vol. 13, N° 3, August 1984, pp. 223-240.
- DELGADO, M. et al.: "Some structural and optimization problems on fuzzy graphs". In KULIKOWSKI, R. and RUDNICKI, J. (eds.): "System Analysis and Computer Science". Omnitech Press. Poland, 1990, pp. 47-63.
- FEDRIZZI, M. et. al. (eds.): "Interactive Fuzzy Optimization". Springer-Verlag. Berlin, 1991.
- GIL ALUJA, J.: "El estudio dinámico de la elección de inversiones". Técnica Contable, Febrero 1967, págs. 41-50 y 66.
- GIL LAFUENTE, A.M.: "El análisis financiero en la incertidumbre". Ariel. Barcelona, 1990.
- ITAKURA, H. and NISHIKAWA, Y.: "Fuzzy network technique for technological forecasting". Fuzzy Sets & Systems. Vol 14, N° 2, November 1984, pp. 99-114.
- KAUFMANN, A.: "Métodos y Modelos de la programación matemática". CECSA. México, 1966.
- KAUFMANN, A. y GIL ALUJA, J.: "Introducción de la teoría de los subconjuntos borrosos a la gestión de las empresas". Milladoiro. Santiago de Compostela, 1986.
- KAUFMANN, A. y GIL ALUJA, J.: "Técnicas operativas de gestión para el tratamiento de la incertidumbre". Hispano Europea. Barcelona, 1987.
- KAUFMANN, A. y GIL ALUJA, J.: "Modelos para la investigación de efectos olvidados". Milladoiro. Santiago de Compostela, 1988.
- KAUFMANN, A. y GIL ALUJA, J.: "Las matemáticas del azar y de la incertidumbre". Centro de Estudios Ramón Areces. Madrid, 1990.
- KAUFMANN, A. y GIL ALUJA, J.: "Nuevas técnicas para la dirección estratégica". Universidad de Barcelona. Barcelona, 1991.
- ZIMMERMANN, H.J.: "Using fuzzy sets in operations research". European Journal of Operations Research, Vol. 13, N° 3. 1983, pp. 201-216.

TRATAMIENTO FUZZY DE LOS INTANGIBLES DE EMPRESA

Angel NÚÑEZ GÓMEZ y José SANCHO GARCÍA
Departamento de Ingeniería Química y Nuclear
Universidad Politécnica de Valencia
Apartado 22012 - 46071 VALENCIA

Resumen: El objetivo de este trabajo es identificar un conjunto de indicadores útiles para evaluar el estado real de los recursos intangibles de las empresas, así como iniciar su formalización matemática en base a técnicas de *fuzzy sets*, incluyendo un somero análisis en torno a la aplicabilidad de tales técnicas. De manera que, en última instancia, la caracterización cuantitativa, o semicuantitativa, al menos, del estado de intangibles de dichas empresas pueda servir para efectuar una revisión crítica del mismo, y pueda llegar a constituir un instrumento efectivo de ayuda para una eventual reestructuración positiva de estos elementos organizativos, y, con ella, una clara mejora de su potencial competitivo.

Palabras Clave: Intangibles de empresa, Entorno fuzzy, Indicadores de estado.

1. Introducción.

La productividad y la calidad constituyen los pilares básicos sobre los que se asienta la competitividad de la empresa moderna. La primera viene determinada, a su vez, por la tecnología disponible y una preparación y capacitación adecuadas del personal, en tanto que la caracterización de la calidad se ha venido efectuando, de forma exclusiva, en función de las cualidades del producto o del servicio suministrado por la empresa. Más recientemente, sin embargo, y a medida que la competitividad se va convirtiendo en un imperativo incluso para las empresas de pequeño tamaño, el concepto tradicional de *relación calidad/precio* de sus productos no puede seguir siendo el único criterio a considerar. Así, el criterio de "satisfacción" —tanto del cliente como del productor— conduce a los análisis de la calidad hacia la esfera de lo intangible. Al mismo tiempo, para concurrir con éxito en los difíciles mercados actuales, las empresas, tanto de producción como de prestación de servicios, necesitan activar todas sus potencialidades, muy en particular sus *recursos intangibles*, esto es, aquellos factores que no siendo materiales ni siquiera contables, ejercen una notable influencia sobre la *calidad integral* de la propia empresa.

Como fenómeno fácilmente constatable, cada vez se percibe un mayor grado de acuerdo sobre la necesidad de tomar en consideración esta clase de recursos, y de ponderarlos convenientemente dentro de los marcos de organización y de los procedimientos de gestión vigentes en el entorno empresarial, con el objetivo final de buscar vías racionales que permitan su optimización, o bien, en una primera instancia, posibiliten su mejora. Esta coincidencia no sólo se produce en el ámbito académico, en el que se desenvuelven los estudios de la economía empresarial o el diseño de métodos de organización de la producción, sino que también afecta a un número creciente de responsables, de empresarios y de directores, quienes desde muy diferentes sectores económicos y actividades prácticas, reconocen e insisten en la importancia de disponer de equipos humanos suficientemente *motivados*. Este hecho ha contribuido, sin duda, al auge que recientemente experimenta la investigación y el desarrollo de las denominadas técnicas de "gestión de recursos humanos".

Pero si nos centramos en los instrumentos disponibles para evaluar la situación real de los recursos intangibles dentro de un contexto empresarial bien definido, su misma naturaleza inmaterial dificulta, de entrada, la posibilidad de llevar a cabo su "medida" mediante procedimientos válidos, y por tanto, capaces de reflejar su auténtica dimensión de una forma que no sea meramente cualitativa. Y el salto hasta los métodos cuantitativos —o semicuantitativos, como mínimo— de evaluación exige la posibilidad previa de llevar a cabo algún tipo de medida, de la misma manera que su análisis en profundidad, necesario para establecer mejores y más eficientes pautas de actuación, difícilmente podrá conseguirse sin el concurso de métodos adecuados y eficaces para su formalización matemática. En definitiva, sólo a través de la identificación de los indicadores más representativos de ese "estado de intangibles", de su correcta captación, y de su posterior incorporación a modelos formales de procesamiento y análisis, podrá establecerse una "ingeniería de intangibles" que resulte bien fundamentada y rigurosa desde un punto de vista científico.[1]

Las estructuras de datos que han de servir para recoger las medidas de los diferentes indicadores del estado de intangibles han de provenir, obviamente, del conjunto de fuentes (subjetivas) mejor conocedoras —las únicas disponibles, tal vez— de los detalles de organización y funcionamiento interno de la empresa: el personal perteneciente a la misma, como conjunto de expertos en un entorno *fuzzy*. Es conveniente notar que la sistematización de las opiniones de los propios trabajadores de la empresa, correctamente diseñada e interpretada, contiene un enorme potencial de información para poder extraer conclusiones en este terreno, y que puede hacerse fluir en sentido inverso al tradicional, es decir, desde los empleados a los directivos, con lo que se evita el riesgo de incurrir en interpretaciones incorrectas de objetivos cruciales para la propia empresa, por ejemplo en las formas de comunicar informaciones relativas a la posición de la empresa (productividad, volúmenes de ventas, beneficios, situación financiera) o de establecer relaciones laborales entre diferentes estamentos, y que pueden producir efectos contrarios a los deseados: es el caso del empleo de circuitos cerrados de televisión en los comedores de grandes empresas, o las visitas de los responsables de producción a las cadenas de montaje para tomar un contacto más directo con las dificultades del trabajo, y que podrían percibirse como actitudes "paternalistas" o inoportunas. La propia percepción del trabajador respecto de estas prácticas, subjetiva, contribuye así a la naturaleza *esencialmente incierta* de la evaluación.

Ocurre, además, que la atmósfera de opiniones sobre la situación de los elementos intangibles no sólo es reflejo de dicha situación, sino que también la condiciona, pues las actitudes laborales resultan, en buena medida, consecuencia directa de aquellas opiniones. De modo que nuestro problema, tal y como se plantea, quedaría enmarcado en el análisis de decisiones en entornos *fuzzy*, y con la dificultad añadida de que el propio sistema bajo control —considerado como el conjunto de opiniones de los miembros de la empresa acerca del estado de sus intangibles— presenta una naturaleza *imprecisa*, así como los objetivos y las restricciones que afectan a las decisiones en este contexto. Esto, sin embargo, no invalida en principio la aplicabilidad de los conceptos y las técnicas vinculados a las representaciones *fuzzy*. [2]

2. Alcance de las evaluaciones.

Haremos aquí unas breves precisiones adicionales, antes de delimitar el alcance y la utilidad que cabe esperar de una evaluación exhaustiva de los recursos intangibles, y de matizar los objetivos, más modestos, de este trabajo, con el que se pretende efectuar una primera aproximación al problema.

En primer lugar, entenderemos por opinión un juicio *libremente expresado* que conlleva imprecisión, con independencia del grado de consciencia de ésta por parte del emisor. No deben existir, por tanto, condicionantes previos que puedan desvirtuar esta premisa, por lo que supondremos que hay garantía de anonimato y sentido de la responsabilidad en la expresión de las opiniones. En este sentido, es conveniente que sean conocidos los fines que se persiguen al recabar las opiniones. Además, nos ocuparemos de empresas normalmente constituidas, cuyos objetivos económicos estén perfectamente definidos y que mantengan un núcleo significativo de empleados fijos. En este sentido, si bien el tamaño de la empresa no se considera un requisito excluyente, la necesidad de obtener resultados representativos implica la consideración de un número de opiniones estadísticamente fiable. Resulta por ello preferible restringir el análisis a las empresas de tamaño medio-grande. Así, los elementos que van a componer el conjunto $X = \{x\}$ de expertos en intangibles *pertenecen* a la empresa, siendo esta relación de pertenencia de naturaleza no difusa, y X un conjunto de cardinal finito.

Los indicadores de estado son aquellos elementos que contienen alguna información útil para hacer explícitas las opiniones de los expertos y poder representarlas de manera formal. Su conjunto constituye pues el objeto concreto sobre el que se centra toda la indagación y captación de las opiniones. Debido a su gran dispersión, se han agrupado en varias dimensiones o campos, los cuales se indican en el apartado siguiente, de forma que sea posible desarrollar cuestionarios detallados y homogéneos dentro de cada campo.

Cabe señalar que la aplicabilidad y la utilidad que en última instancia puede esperarse de una evaluación de este tipo, basada en conceptos y técnicas *fuzzy*, con sus proposiciones previas y mediante el desarrollo de funciones definidas formalmente, es —a juicio de los autores— de índole problemática, por lo que se ha extremado la cautela

y el rigor a la hora de adelantar proposiciones, y aun a riesgo de que no puedan extraerse, por el momento, gran cantidad de conclusiones. Sirvan, de cualquier forma, como indagación preliminar y, por descontado, sometida a discusión y abierta a cualquier sugerencia. La escasez de desarrollos previos en esta cuestión justificaría, por otro lado, la mencionada toma de precauciones.

En cuanto al alcance de las evaluaciones, podemos situarlo a dos niveles distintos. En un primer nivel, más bien elemental, entraría la captación de los indicadores y un primer análisis estadístico de los resultados de las encuestas de opinión cumplimentadas por los expertos. La confección y desarrollo detallado de los cuestionarios juega aquí un importante papel. En esta fase, la contextualización, esto es, la referencia exclusiva a un entorno empresarial bien concreto, es una necesidad, por lo que en el diseño de los cuestionarios no pueden perderse de vista las características específicas de la empresa. En el segundo nivel, en cambio, se trataría de dar mayor generalidad al análisis, buscando pautas comunes, perfiles estándar de empresas de probada competitividad, etc, de modo que pudieran obtenerse resultados de validez general —y no aplicables sólo a una determinada empresa— lo que sin duda constituiría una mejor ayuda para decidir sobre la conveniencia y la forma de introducir cambios en las estrategias de gestión de los recursos intangibles. Si se permite un símil médico, el primer nivel que se ha indicado correspondería a la "exploración" —radiografías, análisis— del estado de los intangibles, mientras que en el segundo nivel comenzaría la fase de diagnóstico y estudio del posible "tratamiento".

Para llevar a cabo la evaluación en el primero de sus niveles, puede ser suficiente un análisis estadístico convencional, de tipo enteramente numérico incluso, para lo que puede resultar de ayuda algún paquete informatizado de análisis estadístico multipropósito, de los existentes en el mercado, y que permiten una pronta y eficiente determinación de gran diversidad de parámetros y operadores estadísticos.[3] Pero lo máximo que puede verse a este nivel es qué aspectos intangibles son susceptibles de mejora y en qué proporción. Incluso haría falta repetir la evaluación para poder confirmar posibles mejoras en el estado de intangibles. Debido a estas limitaciones, cobra interés la posibilidad de desarrollar métodos generales de análisis que permitan llevar las evaluaciones hasta el segundo de los niveles descritos, simplificando al mismo tiempo la captación de los datos. Y aquí es donde ha de justificarse un tratamiento *fuzzy* riguroso para esta aplicación.

No es ocioso insistir en que, de todos modos, incluso si las evaluaciones se llevan a cabo en su segundo nivel, tales instrumentos no proporcionan la receta definitiva ni la garantía de una gestión óptima de intangibles. Es suficiente con admitir que supondrían una ayuda efectiva en esa dirección.

En cuanto a los objetivos que nos hemos propuesto para la presente comunicación, además de proponer la especial adecuación de la matemática borrosa para tratar de profundizar en este problema, quedan restringidos a la identificación de los campos más pertinentes de indicadores, al estudio de su forma de captación, y al inicio de la formalización matemática que permita su posterior tratamiento, mediante la proposición de las bases científicas que parecen más apropiadas, y que también quedan sometidas a discusión.

3. Identificación y captación de indicadores.

Hemos indicado que los detalles que han de constituir el objeto del análisis, y configurar la captación de los "datos de campo", se extienden sobre un conjunto muy amplio de posibles indicadores de estado. Es por ello que se propone, de forma tentativa y no exhaustiva, basarlos en el despliegue de un buen número de ítems o cuestiones concretas, agrupadas en cinco grandes dimensiones o campos de sondeo, de tal forma que las piezas de información relativas a cada campo mantengan un cierto grado de homogeneidad. Estas cinco dimensiones son como sigue:

- (A) Sistema vigente de incentivos y recompensas.
- (B) Gestión interna de la información (grado de transparencia, fluidez) y sus criterios y formas de suministro (selección, valoración, vías de difusión).
- (C) Valoración de la calidad y eficiencia de la comunicación, dentro de la propia empresa y también en su proyección exterior (imagen, estrategias de publicidad).
- (D) Características del "trato" profesional interpersonal: formas de relación humana, tanto interna como a través del entorno social inmediato de la empresa; y tanto en el sentido vertical (de orden o de jerarquía) como en el horizontal (compañerismo, expectativas mutuas sobre el comportamiento ajeno en la asunción de tareas y responsabilidades).
- (E) Nivel de satisfacción o realización personal que se alcanza en el trabajo (ambiente agradable, confort, situaciones de estrés, etc.)

Para cada una de estas dimensiones, o subconjuntos de elementos intangibles, ha de confeccionarse un cuestionario distinto, compuesto preferentemente de proposiciones de sentido afirmativo, sobre las que cada experto deberá pronunciarse, expresando su mayor o menor *grado de conformidad* (acuerdo o desacuerdo) con lo indicado o sugerido en ellas.[4] Puede así definirse al menos un conjunto borroso en X asociado a cada dimensión de intangibles, como serie de pares ordenados experto/opinión, opinión que reflejará, de forma cuantitativa, el grado de conformidad con el sistema de gestión en vigor relativo a dicha dimensión.

La necesidad de realizar encuestas es un imperativo del proceso de captación de los datos que reflejarán la situación de intangibles, tal y como es percibida por los expertos. En ellas, es conveniente que cada dimensión sea cubierta por un número suficientemente amplio de proposiciones, de manera que puedan recogerse la mayor cantidad posible de matices. Así por ejemplo, dentro de la dimensión (A) son relevantes todos aquellos indicadores relacionados con remuneraciones y percepciones de toda especie, y con la forma en que son recibidas: mientras la cuantía del salario es un elemento tangible, no lo es la sensación del perceptor sobre su suficiencia, su estructura, o sobre su puntualidad. Este ejemplo puede ser ilustrativo de la tenue frontera que a veces se encuentra entre lo estrictamente material y lo intangible.

También se ha considerado conveniente introducir clasificaciones precisas *a priori* de los expertos, siguiendo dos criterios fundamentales que pueden tomarse simultáneamente: departamental y estamental. La razón es que parece lógico poder discernir las distintas procedencias —departamentos o secciones, categorías laborales y niveles de responsabilidad— de las valoraciones implícitas en las opiniones manifestadas.

No se considera, en cambio, el problema de la selección ni del reconocimiento de los expertos, debido a que entendemos no puede llevarse a cabo en ausencia total de referencias previas que permitieran establecer criterios de discriminación. Así pues, salvo evidencia en contrario, hay que suponer que todos los trabajadores que conozcan su empresa, aunque sea parcialmente, son expertos en intangibles. Otro tanto sucede con el caso del posible concurso de opiniones recabadas entre clientes, que también se ha obviado en principio a pesar de que pueden contener información aprovechable respecto de ciertos elementos intangibles. Sin embargo, siempre cabe la posibilidad de tomarlos como un grupo más de expertos, si bien habría que restringirlo a una parte de las dimensiones propuestas.

4. Bases para una formalización fuzzy.

Trataremos en este apartado de explicitar algunas de las bases conceptuales ya expuestas más arriba, dando con ello inicio a la caracterización matemática del estado de intangibles empresariales.

4.1 Conjuntos y funciones de pertenencia.

Consideremos, en la línea clásica de Bellman y Zadeh,[2] las siguientes definiciones y proposiciones formales:

Definición 1. Sea $X = \{x\}$ el conjunto de expertos en intangibles. A la expresión de sus opiniones respecto de la dimensión (A) de indicadores, es posible asociar un conjunto borroso A , como serie de pares ordenados

$$(1) \quad A = \{ (x, \mu_A(x)) \} , \quad x \in X$$

donde $\mu_A(x)$, que es la función *grado de pertenencia de x en A*, mide en nuestro caso el *grado de conformidad*, expresado por cada experto x , respecto del conjunto de indicadores de estado que configuran la dimensión (A). En términos generales, $\mu_A: X \rightarrow M$ es una función entre X y el espacio de pertenencia M . Tal espacio de pertenencia es aquí el espacio de opiniones expresadas de los expertos, siempre respecto de una dimensión dada de indicadores de estado. Es evidente que en M no sólo existen dos puntos para cada experto, puesto que toda opinión es matizable.

Hipótesis 1. Podemos entonces suponer que M es el intervalo $]0,1[$, del que se han excluido los puntos extremos, al considerar que no es posible una conformidad absoluta respecto de un grupo amplio de indicadores, cual es el comprendido en cada dimensión. Sí es posible, por el contrario, manifestar una opinión absolutamente conforme con una o algunas de las proposiciones que integren el cuestionario correspondiente a esa dimensión, pero no a todas. De hecho, tan sólo algunas de las proposiciones deben admitir respuestas categóricas.

Hipótesis 2. Debido a que es necesario valorar las respuestas concretas a cada una de las proposiciones o items, el correspondiente grado de conformidad (elemental) tiene que expresarse asignándole un valor. Supongamos que tal valor es un número entero comprendido entre 0 y $n_{\max}$, donde este valor máximo puede ser 10 ó bien un número menor, en función de los tramos de conformidad que se establezcan.

Por consiguiente, cada conjunto inicial de respuestas admite sólo ciertos valores para expresar la conformidad: está discretizado. Pero al combinar todas las respuestas pertenecientes a una misma dimensión, se obtiene un conjunto borroso cuyo grado de pertenencia carece de valores "prohibidos" dentro del intervalo $]0,1[$.

Definición 2. Llamemos N al número de proposiciones que integran la dimensión (A) de indicadores. En base a sus respuestas, valoradas con la serie de enteros $n_i(x)$, podemos construir la función $\mu_A(x)$ como

$$(2) \quad \mu_A(x) = \frac{\sum_{i=1}^N n_i(x)}{N \cdot n_{\max}}, \quad x \in X$$

Observaciones. Como X es un conjunto de cardinal finito, y la función de pertenencia ha de estimarse a partir de las valoraciones efectuadas por sus elementos, el conjunto borroso A ha de quedar definido necesariamente de forma incompleta (parcial).

Las propiedades y las reglas de composición a que deban someterse las funciones de pertenencia así definidas, constituyen cuestiones clave sobre las que es necesario profundizar como condición de utilidad de este tipo de análisis. Veamos algunos aspectos de estas cuestiones en el siguiente subapartado.

4.2 Subconjuntos y funciones de valoración.

Retomemos ahora el tema de la división y clasificación de los expertos, como hemos adelantado en el apartado anterior. Centrándonos sobre la clasificación estamental, y manteniendo la referencia a la dimensión (A) de indicadores de estado, podemos introducir las siguientes definiciones:

Definición 3. Dividamos el conjunto X en tres subconjuntos P , T , y G , de forma que el primero contenga los trabajadores que se encuentran en la primera línea de la producción (no necesariamente cualificados), el segundo el personal técnico, y el tercero el de gestión propiamente dicha. Se tiene:

$$(3) \quad X = P \cup T \cup G$$

En estos subconjuntos, pueden definirse sendas funciones, que denominaremos de *valoración del experto*, soportadas en $]0,1[$.

Definición 4. Las funciones de valoración del experto cumplen las siguientes condiciones: son monótonas en $]0,1[$ y están acotadas en ese intervalo. Estas funciones pueden definirse, para una o varias dimensiones, a través del operador

$$(4) \quad e_P(x) = \inf (1, \mu_A(x) + \mu_B(x)) \quad , \quad x \in P$$

donde la valoración de los expertos pertenecientes al estamento P se extiende a las dimensiones (A) y (B). Observar que cuando esta extensión abarca sólo una dimensión, la forma de la función de valoración se reduce a la de la función de pertenencia. En cambio, a medida que la extensión abarca más dimensiones se irá reduciendo la información útil que se obtiene de cada estamento.

Las ventajas que percibimos para esta definición podemos resumirlas del siguiente modo:

- 1ª) Puede obtenerse información de conjunto sobre el grado de conformidad en más de una dimensión, por estamentos laborales.
- 2ª) Cabe analizar posibles descompensaciones o desequilibrios entre las distintas dimensiones y/o los distintos estamentos.
- 3ª) Nos permite ordenar todos los datos, y por tanto, introducir la dinámica de los α -corte.

5. Observaciones finales.

Debe existir alguna forma de correlación entre el estado de los intangibles de la empresa y sus posibilidades de alcanzar mejores resultados, también en aspectos estrictamente económicos. La aprehensión de las estructuras íntimas de esa correlación pasa por la posibilidad de escrutar el estado de dichos recursos intangibles de forma cuantitativa, buscando reglas de composición generales para las funciones expertas, y pautas comunes que puedan resultar características de las empresas más competitivas. Dentro de los problemas de reconocimiento de pautas, existen abundantes aportaciones que pueden orientar la búsqueda de los procedimientos específicos más adecuados.[5]

A lo largo de esta primera aproximación al problema, se ha presentado una serie de proposiciones en torno a la mejor forma de obtener y analizar la información que es relevante para caracterizar el estado de intangibles. Se han adelantado, además definiciones que pueden ser de utilidad para construir funciones de pertenencia y de valoración, en un intento de mostrar la especial adecuación de la lógica borrosa para abordar en profundidad este problema.

Como ventajas que pueden derivarse del tratamiento *fuzzy* del problema de la correcta gestión de intangibles, cabe citar las siguientes:

- 1ª) Posibilidad de dar generalidad a las evaluaciones de intangibles, lo que permitiría una descontextualización, siquiera parcial, de muchos de los aspectos de la evaluación: en caso contrario, la necesidad de mantener una referencia exclusiva en todo momento a las características específicas de una determinada empresa supondría una limitación excesiva.
- 2ª) Obtención de funciones estándar para empresas de probada competitividad, como paso previo a la búsqueda de correlaciones generales entre la situación de intangibles y los resultados realmente alcanzados por la empresa.
- 3ª) Ayuda en las decisiones de implantar cambios en las estrategias de gestión pertinentes, y también en la verificación de su efectividad.

En última instancia, y como mínimo, la valoración de los elementos intangibles de la empresa, obtenida a través de encuestas de opinión de expertos, y adecuadamente sistematizada a través modelos basados en variables y funciones borrosas, debe permitir el desarrollo de instrumentos sólidos de análisis y diagnosis del estado real de los recursos intangibles empresariales.

La profundización en este tipo de análisis y las posibilidades de generalización de sus resultados, resultarán piezas clave para la búsqueda de estrategias óptimas de gestión de los recursos intangibles, a la vez que facilitarían y simplificarían la realización de las encuestas. Por último, la posible implantación en el ordenador de tales instrumentos, junto a una adecuada interpretación de sus diagnósticos, pueden llegar a constituir estimables ayudas para la toma de decisiones hacia una gestión de calidad que incluya la totalidad de capitales y recursos con que cuenta la empresa actual, independientemente de su tamaño, como imprescindible condición de futuro en una etapa histórica postindustrial.

6. Referencias.

- [1] Josep BURCET, "Hacia una ingeniería de los intangibles", Actas del 1er. Congreso Nacional de Ingeniería y Consultoría, 1989.
- [2] R.E. BELLMAN and L.A. ZADEH, "Decision-Making in a Fuzzy Environment", in *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, Edited by R.R Yager et al., John Wiley & Sons, Inc., 1987.
- [3] *SAS User's Guide: Basics*, Version 5 Edition, by SAS - Institute, Inc., Cary, NC (USA), 1985.
- [4] "Resultados de las encuestas de evaluación de la Docencia", Curso 1991-1992, Instituto de Ciencias de la Educación, Universidad Politécnica de Valencia, 1992.
- [5] James C. BEZDEK, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

CONJUNTOS DIFUSOS Y RECONOCIMIENTO DE RASGOS DEL TERRENO.

R.Conejo Muñoz
R.Morales Bueno
J.L.Pérez de la Cruz
J.A.Urbano Montero

Dpto.Lenguajes y Ciencias de la Computación
Facultad de Informática
Málaga (España)

Resumen: En el presente trabajo realizamos una primera aproximación al problema de la percepción automática de rasgos topográficos de un terreno natural (montes, valles, llanuras, etc.). Tales rasgos son de naturaleza esencialmente vaga, por lo que hemos de recurrir para describirlos a la teoría de conjuntos difusos. Tras introducir los conceptos nítidos necesarios, exponemos algunos algoritmos para el reconocimiento de formas topográficas, y mostramos los resultados de su implementación informática.

Palabras Clave: Geomorfología, Reconocimiento de formas, Conjuntos difusos.

1. Introducción.

Las reglas seguidas por los ingenieros para ajustar el trazado de una obra lineal suelen hacer referencia a los accidentes o rasgos morfológicos presentes en la zona intermedia entre el origen y el destino. Por ejemplo, se alude (Balaguer 86) a "evitar los cerros" o a "aprovechar las laderas de las vaguadas" o a "buscar los puertos entre dos montes"; o se dice que "en un terreno ondulado, el trazado ha de ser sumiso, ceñido al terreno".

Pero es difícil enumerar estos accidentes y rasgos globales de forma nítida: el lenguaje natural tiene para nombrarlos un gran número de expresiones, matizables y ampliables cuanto se desee. Además, las reglas hacen referencia a la importancia de estos rasgos. Un cerro puede ser "grande" o "pequeño"; una ladera, "suave" o "abrupta"; un valle, "profundo" o "poco profundo". Importancia que es función no sólo del tamaño del accidente, sino también de sus características internas.

Todo ello muestra que el problema del reconocimiento de obstáculos topográficos a partir de un mapa digital es necesario si se desea simular el razonamiento de un diseñador humano; y, además, que será preciso recurrir a técnicas apropiadas para el tratamiento de la información vaga o imprecisa, es decir, a técnicas basadas en la teoría de los conjuntos difusos. Nótese que el problema es diferente del que se presenta al identificar un objeto nítido, para el cual, en presencia de información parcial, también se han propuesto técnicas difusas.

En el presente trabajo aplicamos técnicas heurísticas para identificar los rasgos topográficos considerados como objetos difusos, distinguiendo además su grado de importancia intrínseca y extrínseca,

aplicando técnicas de representación piramidal. El sistema construido acepta como entrada una mapa digital y produce como salida varios conjuntos de rasgos identificados, correspondientes a los varios niveles piramidales y a los alfa-cortes que se soliciten. Para cada rasgo se proporcionará una descripción mediante etiquetas lingüísticas.

A continuación, en el apartado 2 exponemos las técnicas nítidas de representación de superficies topográficas en que se basa nuestro trabajo; en la parte 3 explicamos los conceptos difusos necesarios y los principios del funcionamiento global del sistema, para concluir en la parte 4 mostrando algunos resultados reales y las conclusiones que se deducen del trabajo.

2. Representación digital del terreno.

2.1. Matrices de cotas y gradientes.

Una superficie topográfica puede, en principio, representarse mediante una función que da la cota z de un punto en función de su posición sobre la esfera terrestre. Suponiendo que tratamos con fragmentos pequeños de esta, podemos emplear unos ejes cartesianos para referenciar la posición horizontal, con lo cual la superficie se representará mediante una función

$$z = z(x,y)$$

que además se suele restringir al rectángulo formado por los puntos (x,y) tales que $0 \leq x \leq X_{\max}$, $0 \leq y \leq Y_{\max}$.

Esta función se suele digitalizar recurriendo a una matriz de cotas Z en la que se almacenan las cotas de los nodos de una malla rectangular (o cuadrada) definida sobre el terreno, siendo las separaciones entre los nodos a lo largo de las direcciones de los ejes Δ_x y Δ_y . Así, la forma más simple de determinar la cota z en el punto (x,y) es seleccionar un elemento de la matriz Z

$$z = Z_{i(x),j(y)}$$

donde $i(x)$, $j(y)$ son las discretizaciones de las coordenadas x,y :

$$i(x) = \lfloor x / \Delta_x \rfloor + 1$$

$$j(y) = \lfloor y / \Delta_y \rfloor + 1$$

si suponemos que i,j varían a partir de 1. Por supuesto, se puede recurrir también a funciones de interpolación más sofisticadas que producen superficies continuas y diferenciables.

En cualquier caso, es obvio que la cota $z(x,y)$ de un punto no contiene por sí sola ninguna información acerca de la topografía, pues el origen de cotas puede tomarse arbitrariamente; la topografía en un punto, en efecto, no queda definida por su cota, sino por la relación de ésta con las próximas. La forma más sencilla de expresar esta relación es suponer que la superficie del terreno discretizada por $Z(i,j)$ es diferenciable y calcular los gradientes

$$\text{grad}(x,y) = \frac{dz}{dx} i + \frac{dz}{dy} j$$

El gradiente en un punto determina, entre otras cosas:

-la pendiente máxima del terreno en ese punto.

- la dirección de la máxima pendiente.
- la dirección de las curvas de nivel.

La expresión del gradiente en coordenadas cartesianas x, y no es la más apropiada a nuestros efectos. Es preferible la expresión dada por el módulo y el acimut de la línea de máxima pendiente creciente, normalizado al intervalo $[0, 2\pi[$. Estas dos magnitudes reales se pueden almacenar en dos matrices escalares $G(i, j)$ y $Az(i, j)$. El problema es entonces hallar el valor de estas derivadas parciales en los puntos de una malla, a partir de los valores de la función en los puntos de esta misma malla. En el ámbito del procesamiento de imágenes y visión artificial este es un problema bien conocido que se presenta cuando se quieren detectar contornos (Levine 85), para el cual en la práctica se han propuesto soluciones de eficiente implementación informática. Todas ellas pasan por definir un operador de máscara 0 que se aplica a un entorno de cada punto de la malla (la llamada "ventana") y mediante sencillas operaciones aproxima las derivadas parciales. Los operadores se suelen representar por una matriz de pesos que multiplicados por los elementos correspondientes del entorno dan el valor de la derivada.

En este trabajo se ha empleado el operador de Prewitt, dado por las máscaras

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \text{ y } \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

que es el que produce el mínimo error cuadrático (Levine 85).

2.2. Pirámides.

Las *pirámides* fueron introducidas en el procesamiento de imágenes por Tanimoto (Tanimoto y Pavlidis 75), en parte como estructura de datos, en parte como arquitectura de procesamiento paralelo (Tanimoto 84). Como estructura de datos, una pirámide puede definirse formalmente como una sucesión finita de arrays bidimensionales numéricos

$$P = (A_1, A_2, \dots, A_n)$$

tales que

1) existe un entero b tal que para todo elemento A_i de la sucesión se cumple que

$$\dim(A_{i+1}) = \dim(A_i)/b$$

2) existe una función f de b^2 argumentos tal que para todo elemento de la sucesión se cumple que

$$A_{i+1}(j, k) = f(A_i(b \times (j-1) + 1, b \times (k-1) + 1), \dots, A_i(b \times j, b \times k))$$

Es decir, cada array tiene un *tamaño de grano* b veces menor que el anterior, y el valor de cada elemento del array i -ésimo se calcula a partir de los b^2 valores correspondientes del array $i-1$ -ésimo. La función que se suele emplear para ello en proceso de imágenes es la media aritmética:

$$A_{i+1}(j, k) = \frac{A_i(b \times (j-1) + 1, b \times (k-1) + 1) + \dots + A_i(b \times j, b \times k)}{b^2}$$

En nuestro problema, cada array de la pirámide correspondería bien a una matriz de cotas, bien a una matriz de gradientes. El array inicial

contendría los valores para cada punto considerado en la discretización del terreno; los arrays siguientes contendrían valores promediados correspondientes a discretizaciones cada vez menos detalladas del mismo terreno.

Parece que cabrían aquí dos alternativas: de una parte, promediar directamente los gradientes a partir de los calculados para el nivel de detalle inicial; de otra, promediar las cotas a partir de las iniciales y conforme a ellas calcular los gradientes en cada nivel de detalle. Sin embargo, la única alternativa razonable es esta última, debido al carácter *fractal* de las superficies topográficas.

La teoría de los objetos fractales, debida a Mandelbrot (Mandelbrot 83), surgió al aplicar los potentes recursos de la topología y el análisis moderno a la descripción de las irregularidades de la Naturaleza. Los objetos fractales tienen la propiedad de ser autosimilares o autoafines. Un conjunto acotado S de un espacio euclídeo E se llama autoafín si existen un vector r y un entero N tales que

- a) S se puede particionar en N subconjuntos $S(n)$
- b) El resultado de aplicar la afinidad dada por r a cada $S(i)$ es congruente con S .

Análogamente, un conjunto acotado aleatorio S se llama estadísticamente autosimilar si existen un escalar r y un entero N tales que

- a) S se puede particionar en N subconjuntos $S(n)$.
- b) para cada $S(i)$, el conjunto obtenido multiplicando r por todos sus elementos es congruente en distribución con S .

Que los relieves naturales presentan esta propiedad es un lugar común. Como dice Mandelbrot citando al montañero inglés E. Whymper (s.XIX), "los trozos de roca a menudo presentan las formas características de los acantilados de donde han sido arrancados. ¿Por qué habría de ser de otro modo, si la masa de la montaña es más o menos homogénea? ... las mismas heladas y lluvias dan forma tanto a la masa como a sus partes".

Aceptando el carácter fractal de las superficies topográficas, es claro que propiamente no existe el gradiente de cotas, ya que la función $z(x,y)$ no es diferenciable. Lo que calculamos entonces es el gradiente que tendría una superficie diferenciable que se aproximara, a un nivel de detalle dado, a la real.

En resumen, la sucesión de arrays que se considera en la pirámide es la de arrays de cotas. Cada elemento del array se calcula en un nivel de detalle dado promediando los b^2 elementos del nivel anterior. Las matrices de gradientes se calculan aplicando los operadores de Prewitt a las cotas de cada nivel de detalle.

3. Las formas del relieve.

3.1. Elementos topográficos.

Sea una superficie topográfica dada por una función

$$z = z(x,y), \quad 0 \leq x \leq X_{\max}, \quad 0 \leq y \leq Y_{\max}$$

Un *elemento topográfico* es un par $E (F,D)$ donde F , llamado *extensión* del elemento, es un subconjunto difuso del conjunto de puntos $U = \{(x,y) ;$

$0 \leq x \leq X_{\max}$, $0 \leq y \leq Y_{\max}$ }, tal que el soporte de F es conexo; y D , llamado *descripción* del elemento, es un conjunto finito de etiquetas lingüísticas (subconjuntos difusos de R).

Un *regionamiento* topográfico es un conjunto finito de elementos topográficos de $U \{E_1, E_n\}$ tal que para todo punto $P \in E$ existe un E_i tal que P pertenece al núcleo de la extensión de E_i . No se exige que los núcleos -y menos los soportes- de los elementos sean disjuntos.

Obviamente, estamos interesados en los elementos y regionamientos que sean formalizaciones de las descripciones verbales de una superficie topográfica realizadas por un ser humano. En este trabajo consideraremos únicamente los elementos denominados *monte*, *llanura* y *valle*. Cada elemento de estos tipos quedará identificado por un conjunto de etiquetas referentes a su tamaño e importancia, y por un conjunto difuso que representa su extensión. Nótese que las ideas difusas aparecen aquí de forma natural: la cima de un monte pertenece al monte con todo derecho ($\mu=1$), pero según vamos descendiendo el grado de pertenencia disminuye ($\mu < 1$) hasta que llegamos al fondo de un valle, que no puede decirse que pertenezca al monte ($\mu=0$).

El carácter difuso de los elementos topográficos se pone de manifiesto en las definiciones que se suelen dar. El Diccionario de la Academia dice, por ejemplo, que monte es "grande elevación natural del terreno", llanura "campo o terreno dilatado, sin altos ni bajos" y valle "llanura de tierra entre montes o alturas". Los conceptos a los que se alude en las definiciones citadas son esencialmente vagos.

Por otra parte, las definiciones académicas no coinciden exactamente con la idea que se suele asociar con estas palabras; por ejemplo, en la definición de monte no se alude a la presencia de una cima, ni a la pendiente del terreno. Sin embargo, para nosotros un monte será una zona más o menos pendiente que rodea una cima. Valle será una zona que rodea un cauce. Llanura será una zona más o menos extensa de pendiente despreciable.

3.2. Reconocimiento de formas.

Es habitual en la literatura de reconocimiento de formas (Tou y Gonzalez 81) la distinción entre métodos matemáticos y métodos heurísticos. En nuestro caso, la identificación de elementos topográficos creemos que ha de basarse en métodos heurísticos. En efecto, el problema no se presta a un tratamiento matemático exacto por varias razones:

a) ante una misma superficie, es posible dar varios regionamientos diferentes igualmente válidos.

b) raramente aparecen en una superficie topográfica elementos bien aislados; más bien se puede decir que se mezclan e imbrican unos en otros.

La percepción de elementos topográficos es un caso claro de incertidumbre del tipo 2 (Gupta 88), incertidumbre que nace no de la conducta aleatoria de un sistema físico, sino de las características inherentes a los procesos perceptivos y cognitivos del ser humano. Desde un punto de vista general, existen ya textos (Kandel 82), (Dutta-Majumder 86) que tratan todo el campo del reconocimiento de formas desde el punto de vista de los conjuntos difusos. Dignas de mención son las

aplicaciones al reconocimiento de caracteres manuscritos (Tamura 71), (Chatterji 82), a la detección de bordes (Gupta et al.88) o a la umbralización de histogramas de imágenes (Ray y Dutta-Majumder 88). En cualquier caso, nuestro problema es diferente de los abordados en los trabajos mencionados, pues no tratamos de percibir un objeto nítido (como en el caso de la detección de bordes y regiones en una imagen) ni de determinar el grado de pertenencia de un objeto a una clase (como en el caso de los caracteres), sino de determinar el grado de pertenencia de cada punto de la imagen a un objeto determinado. Por ello, hemos tenido que desarrollar algoritmos específicos para esta tarea.

3.3. Reglas y algoritmos.

Nuestras reglas heurísticas para reconocer la extensión de un rasgo se basan en la presencia de ciertos puntos y líneas distinguidos de la superficie: los extremos locales y las líneas de máxima pendiente.

Un máximo local es un punto cuya cota es superior a la de todos los puntos que lo rodean. Todo máximo local tiene gradiente nulo; sin embargo, al emplear una malla para representar el terreno, no es posible asegurar que el gradiente calculado en un máximo local de la malla sea nulo. Por tanto, los máximos locales se deben encontrar por comparación directa con sus 4 u 8 vecinos en la malla.

En las superficies topográficas reales no suele haber mínimos locales, salvo en la frontera de la región. En efecto, la presencia de mínimos locales vulneraría el *principio de escorrentía*: a partir de cada punto de la superficie es posible trazar una curva tal que el valor de la cota z es monótono decreciente en función de la longitud del arco.

La línea de máxima pendiente queda definida en cada punto por el azimut del gradiente. Si consideramos la discretización dada por una malla, la línea de máxima pendiente se trazará pasando de cada punto a aquel de sus 4 u 8 vecinos cuya cota sea más diferente.

A partir de las líneas de máxima pendiente se define el *campo de escorrentías* de una superficie dado por una función nítida $e=e(x,y)$ calculada de la siguiente manera: supongamos un elemento C de curva de nivel, de longitud l . Consideremos la región de la superficie formada por todos los puntos P tales que existe una línea de máxima pendiente descendente que se origina en P e intersecta con C , y sea S su área. El valor del campo de escorrentías en un punto de la curva de nivel vendrá dado por dS/dl .

El valor del campo de escorrentías en los puntos de la malla se puede aproximar por el siguiente algoritmo: Inicializar a 0 el valor de e en todo nodo; partiendo de cada nodo P de la malla, marcar la línea descendente (z decreciente) de máxima pendiente que parte de P , incrementando en 1 el valor de e en todos los puntos marcados. Finalmente se habrá obtenido un campo de valores que es mayor en aquellas regiones en las que tienden a confluir las líneas de máxima pendiente descendente. En la mayoría de los terrenos reales, estas regiones ocupan una pequeña proporción de la superficie total, y en ellas el valor de e es mucho mayor que en el resto. Estas regiones son las *vaguadas* o *cauces*.

Los máximos locales y cauces se emplean como *semillas* en un algoritmo de propagación difusa de etiquetas, enmascarando el proceso

con los valores del módulo del gradiente. A cada máximo local le corresponde una etiqueta de *monte*, que se propaga a sus puntos vecinos con grado de pertenencia progresivamente decreciente hasta que se alcanza un cauce u otro monte, siempre que el valor del gradiente sea suficientemente elevado (lo cual introduce otra modulación en la función de pertenencia). Análogamente, a cada cauce le corresponde una etiqueta de *valle*, que se propaga atenuándose a sus puntos vecinos. Las etiquetas de *llanura* se asignan a las regiones conexas que se forman con puntos de escaso gradiente.

4. Resultados y conclusiones.

Los algoritmos anteriormente descritos se han implementado mediante un conjunto de programas C que corren sobre un ordenador PC compatible. Al sistema se le han proporcionado entradas reales (fig. 1) en forma de matrices de 64x64 puntos. El sistema es capaz de agregar los datos en forma piramidal (las figs. 5, 6 y 7 corresponden al nivel siguiente de agregación) y realizar un proceso de identificación en cada nivel de agregación.

En cada nivel de agregación, la salida generada consiste en una familia de conjuntos difusos que constituyen un regionamiento topográfico de la superficie. Por ejemplo, las figuras 2 y 5 muestran los alfa-cortes de la extensión de un elemento de tipo monte correspondientes a $\mu=0.9$; para el mismo elemento, las figuras 3 y 6 muestran los alfa-cortes con $\mu=0.5$, y las figuras 4 y 7 los alfa-cortes con $\mu=0.2$. Como puede observarse, los resultados están en buena concordancia con la identificación que realizaría un sujeto humano.

Además, el sistema proporciona una descripción lingüística de cada elemento. Por ahora, esta consta únicamente de tres descriptores:

- el tipo de elemento
- su tamaño relativo ("grande", "pequeño", etc.)
- su dificultad relativa ("abrupto", "suave", etc.).

Creemos que el sistema implementado muestra claramente las posibilidades de las técnicas difusas para la descripción de los rasgos de un terreno real. Nuestro trabajo futuro se desarrollará según las siguientes líneas:

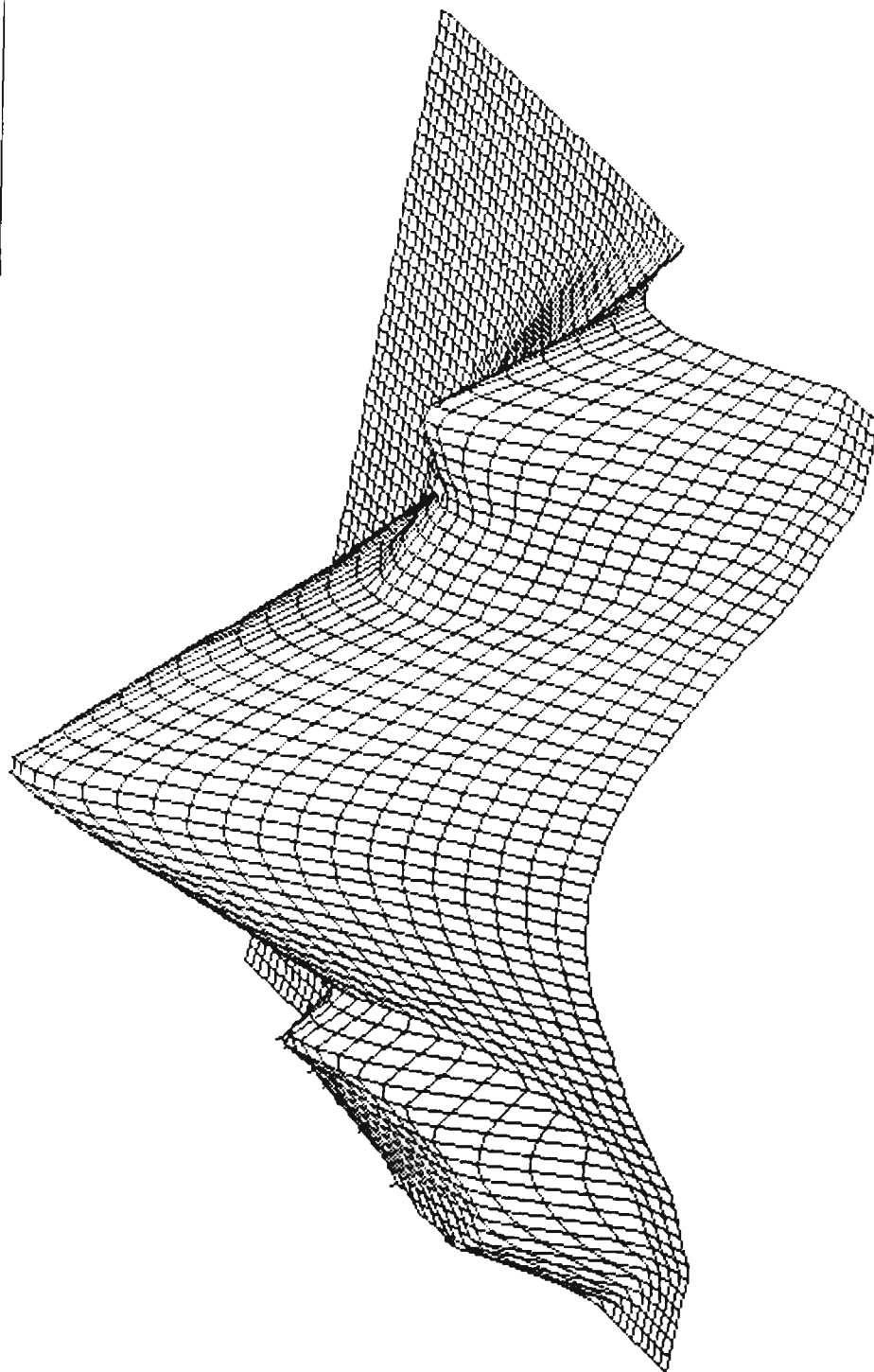
- ampliación del conjunto de *semillas* del regionamiento.
- ampliación del repertorio de elementos topográficos.
- integración de los análisis realizados en los diversos niveles de detalle.

Referencias.

- Balaguer,E. *et al.* (1986): *Trazado de carreteras*. Madrid, ETSICCP.
- Chatterji,B.N. (1982): Character Recognition using fuzzy similarity relations. En Gupta,M.M. y E.Sanchez (eds.): *Approximate Reasoning in Decision Analysis*, Amsterdam, North-Holland.
- Dutta-Majumder,D. (1986): *Fuzzy Mathematical Approach to Pattern Recognition*. Nueva York, Wiley.
- Gupta,M.M. (1988): Cognition, Perception and Uncertainty. En Gupta,M.M. y T.Yamakawa (eds.): *Fuzzy Computing: Theory, Hardware and Applications*, Amsterdam, North-Holland.
- Gupta,M.M., Knopf,G.K. y P.N.Nikiforuk: Edge Perception using Fuzzy Logic. En Gupta,M.M. y T.Yamakawa (eds.): *Fuzzy Computing: Theory, Hardware and Applications*, Amsterdam, North-Holland.
- Kandel,A. (1982): *Fuzzy Techniques in Pattern Recognition*. Nueva York, Wiley.
- Levine,M.D. (1985): *Vision in Man and Machine*. Nueva York, McGraw-Hill.
- Mandelbrot,B. (1983): *The fractal geometry of Nature*. San Francisco, Freeman & Co.
- Ray,K.S. y D.Dutta-Majumder (1988): Fuzzy Rule Based Approach to Image Segmentation. En Gupta,M.M. y T.Yamakawa (eds.): *Fuzzy Computing: Theory, Hardware and Applications*, Amsterdam, North-Holland.
- Tamura,S. *et al.* (1971): Pattern Classification based on fuzzy relations. *IEEE Transactions on Man, Machine and Cybernetics* 1 61-66
- Tanimoto,S.L. (1984): A hierarchical cellular logic for pyramid computers. *Journal of Parallel and Distributed Computing* 1 105-132
- Tanimoto,S.L. y T.Pavlidis (1975): A hierarchical data structure for picture processing. *Computer Graphics and Image Processing* 4 104-119
- Tou,J.T. y R.C.Gonzalez (1981): *Pattern Recognition Principles*. Reading, Ma., Addison-Wesley.

REPRESENTACION GRAFICA.

Nombre del fichero de datos: s2.dat
Coordenadas del punto de vista X: 700.00 Y: 300.00 Z: 1200.00



situacion: (8,2) Z (6,4) X (3,1) Y "z" fin "c" CAMBIA SALIO

FIGURA 1

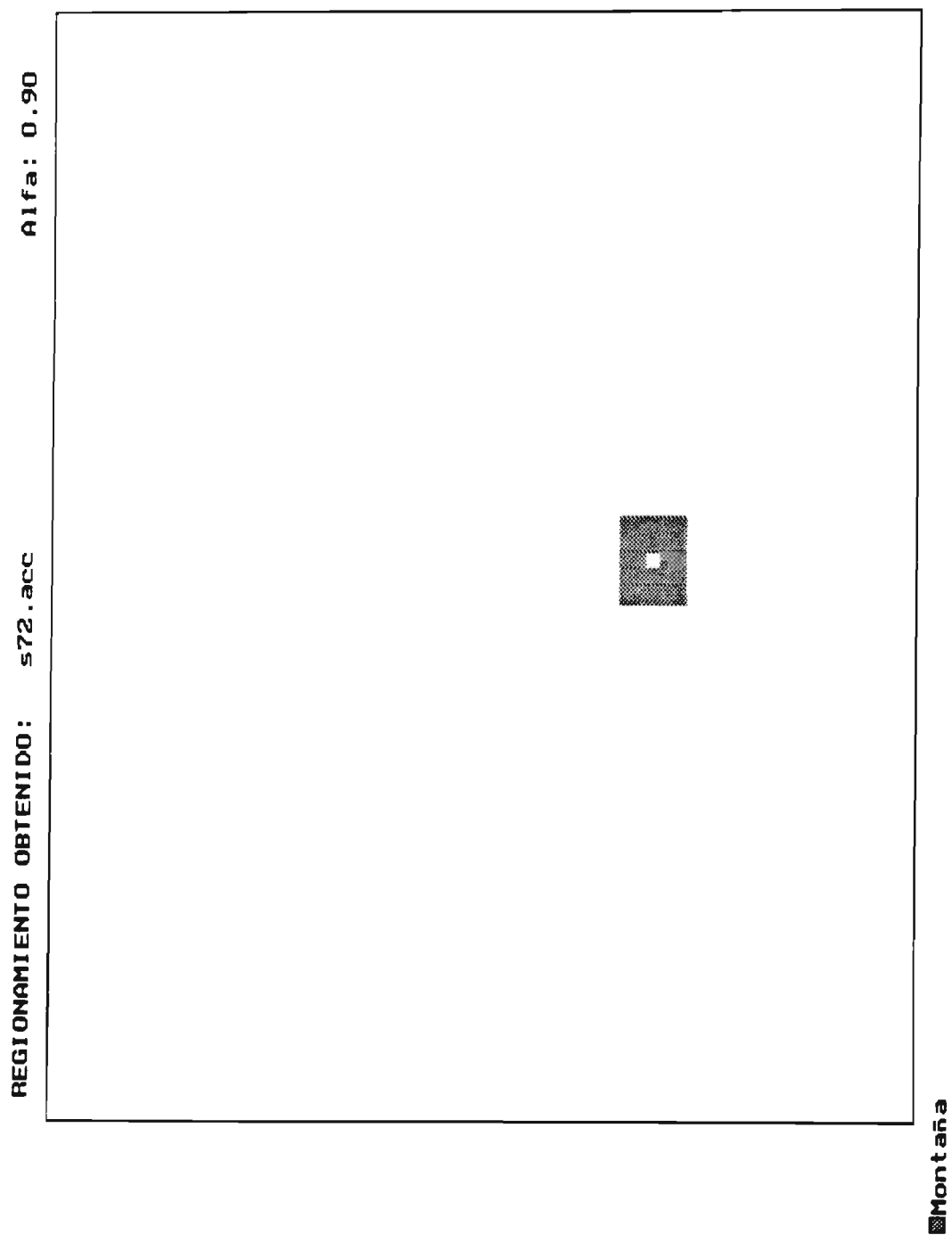


FIGURA 2

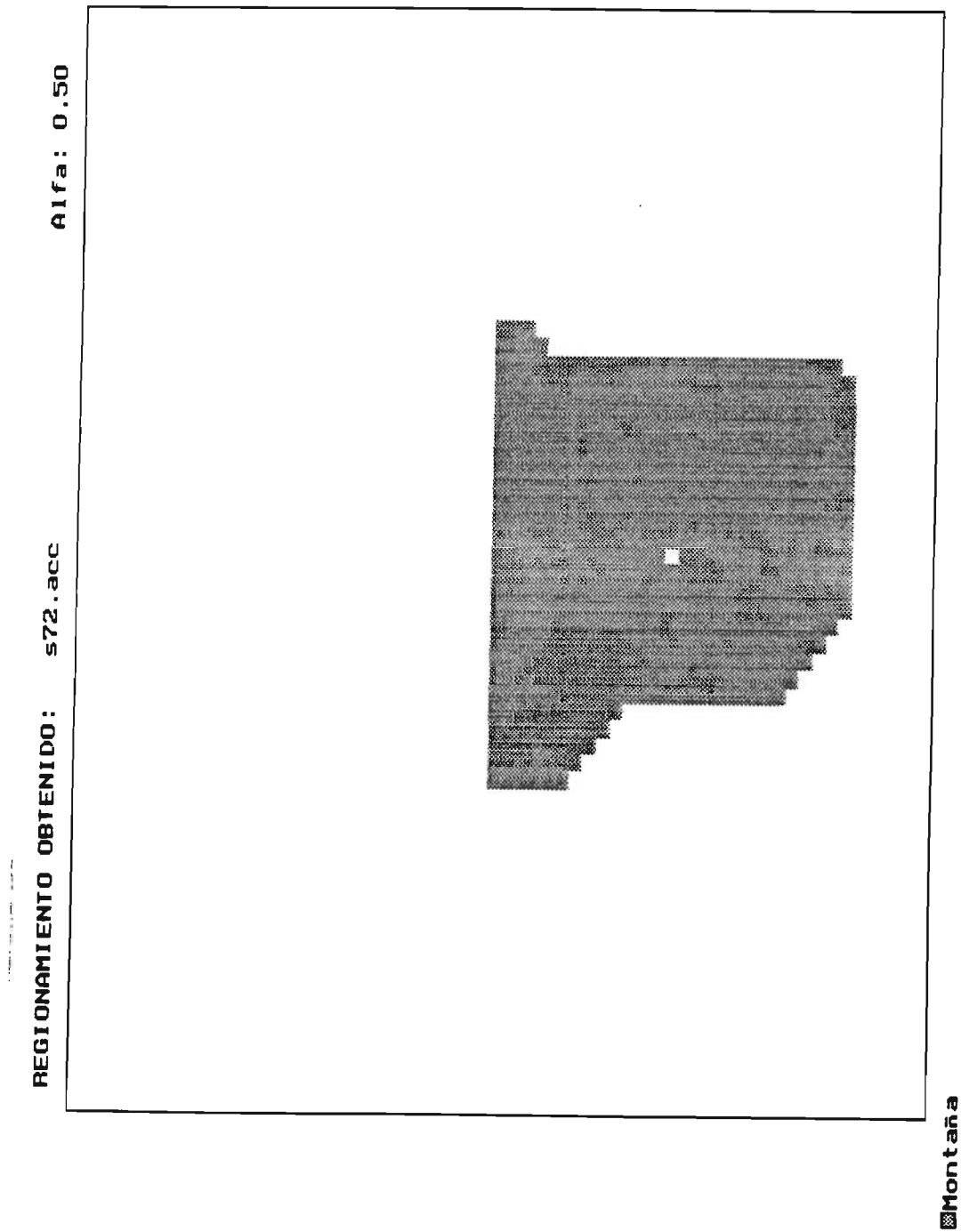
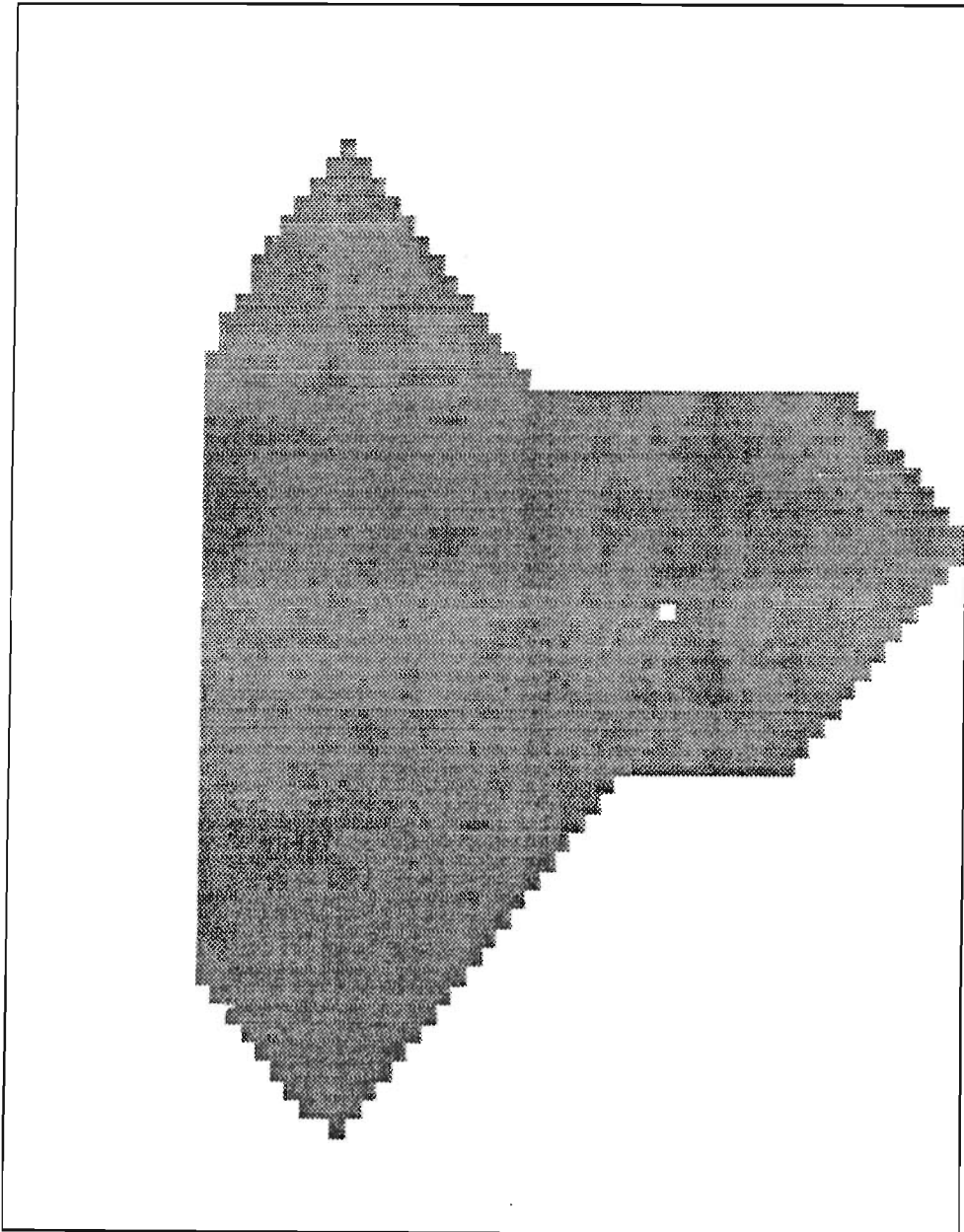


FIGURA 3

Alfa: 0.20

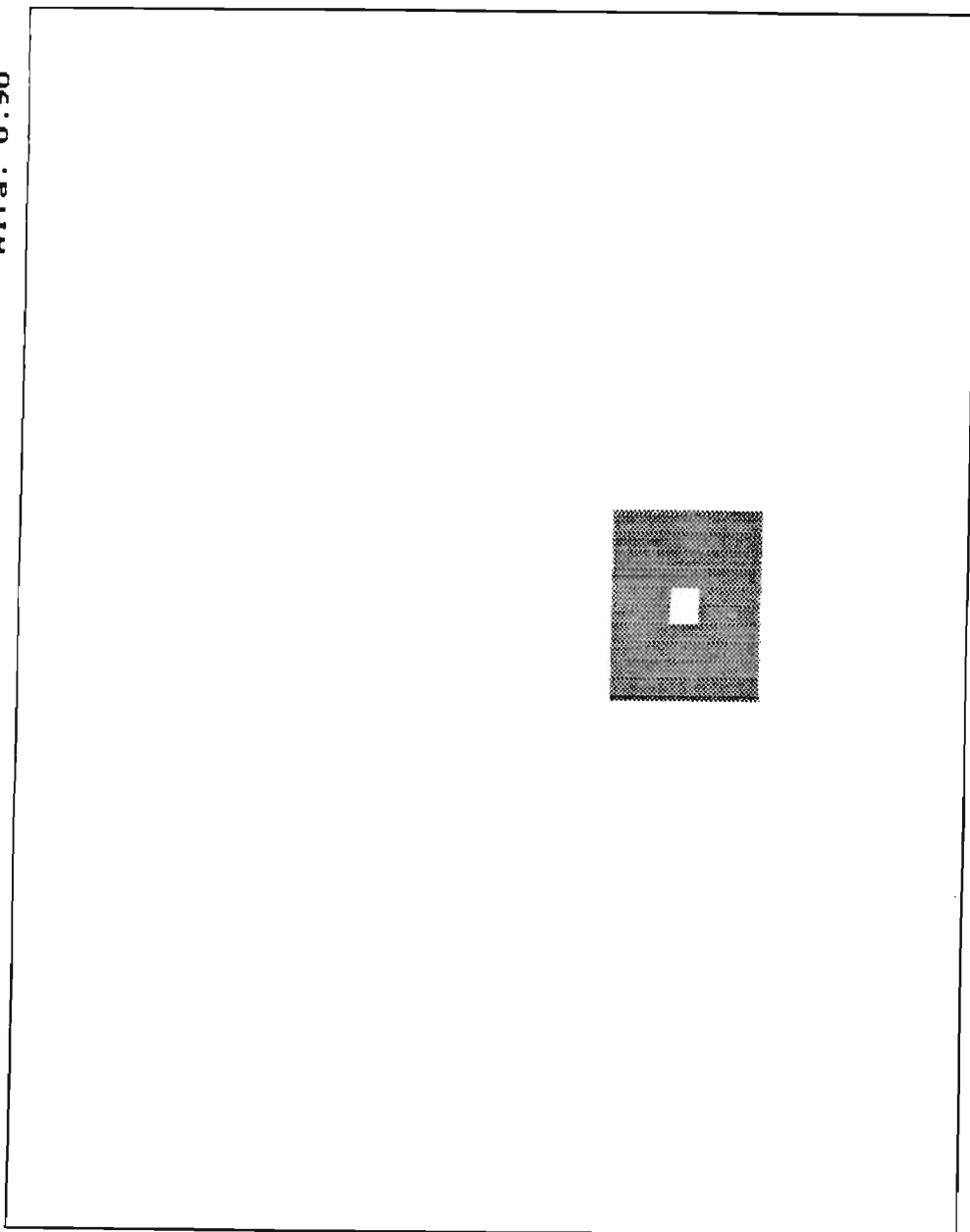
REGIONAMIENTO OBTENIDO: s72.acc



Montaña

FIGURA 4

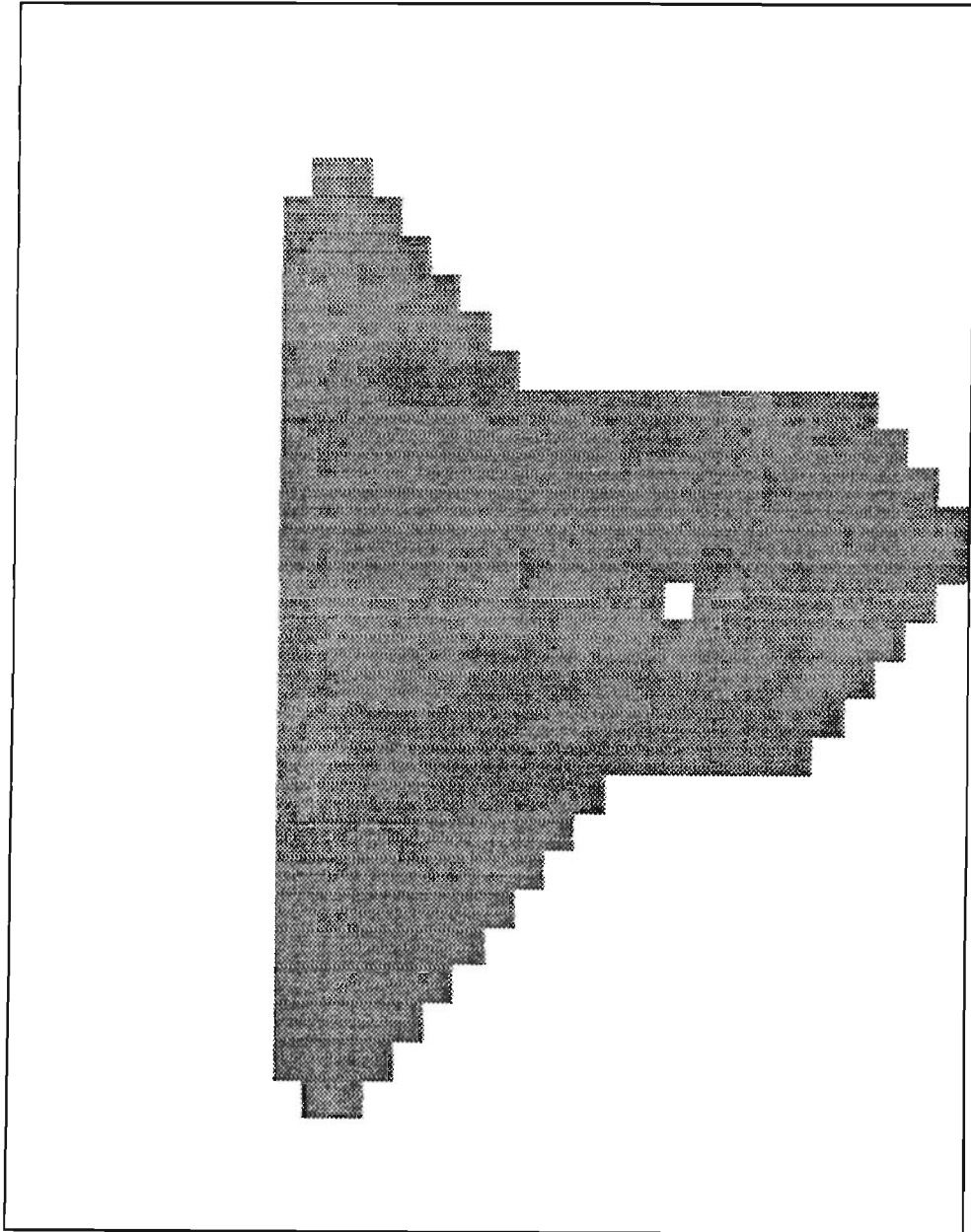
REGIONAMIENTO OBTENIDO: sred72.acc Alfa: 0.90



Montaña

FIGURA 5

REGIONAMIENTO OBTENIDO: sred72.acc Alfa: 0.50



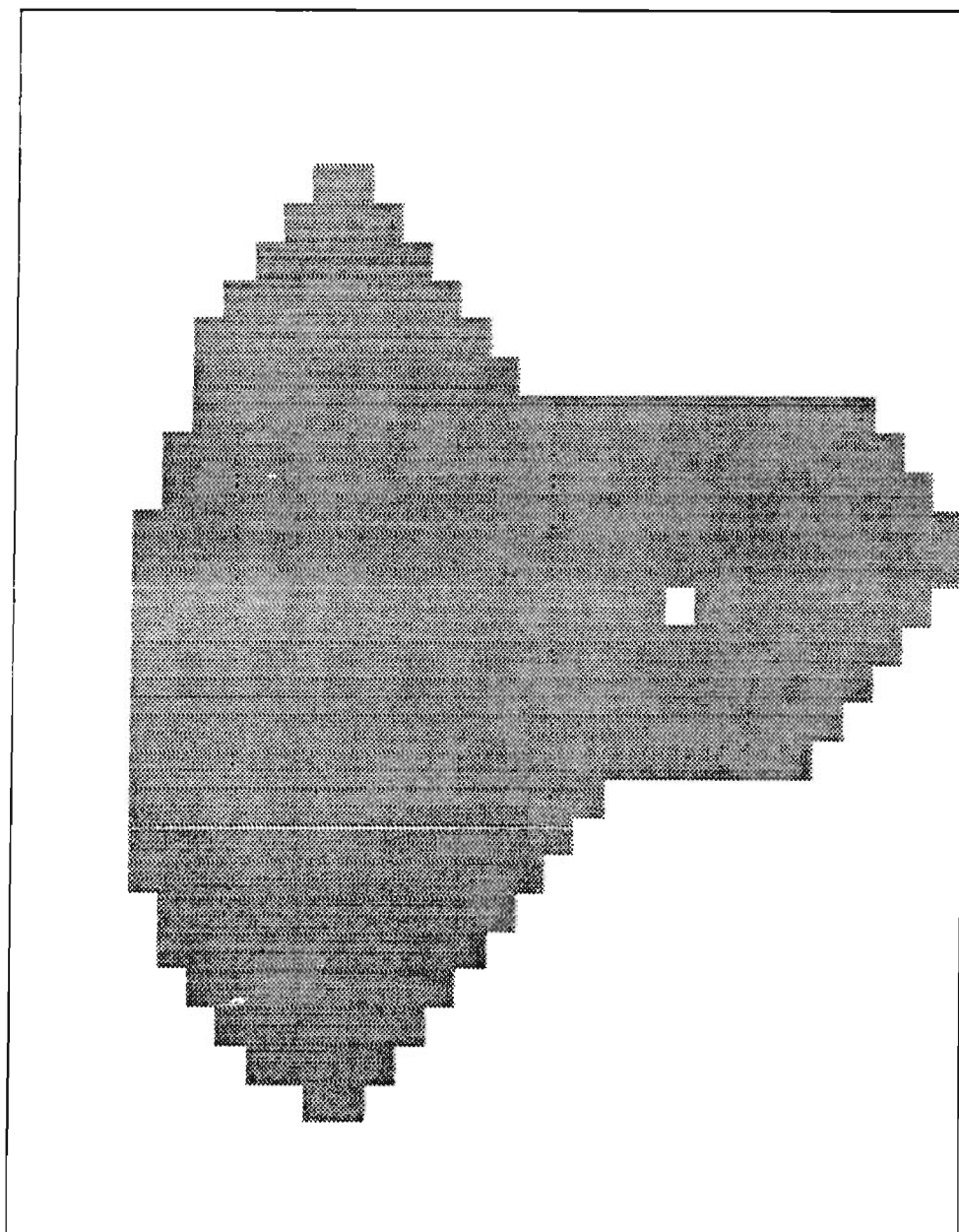
■ Montaña

FIGURA 6

Alfa: 0.20

sred72.acc

REGIONAMIENTO OBTENIDO:



Montaña

FIGURA 7

INDICE DE AUTORES

Aguilar, J.A.	; 127
Alamán, X.	; 127
Arranz, L.I.	; 193
Avellana, N.	; 289
Barriga, A.	; 301
Barro, S.	; 323
Baturone, I.	; 301
Bugarín, A.	; 323
Burillo, P.	; 135
Bustince, H.	; 135
Cadenas, J.M.	; 175
Campo, I. del	; 267
Carrabina, J.	; 289
Castro, J.L.	; 147, 283
Clares, B.	; 203
Conejo, R.	; 185, 203, 381
Cubillo, S.	; 1
Delgado, M.	; 81, 147
Díaz, J.A.	; 93
Echanobe, J.	; 73
Esteva, F.	; 111
Fernández, F.	; 339
Galán, R.	; 253
García Calves, P.	; 111
Garitagoitia, J.R.	; 73, 267
Godó, L.	; 111
González, A.	; 59
González de Mendívil, J.R.	; 73, 267
Gutiérrez, J.	; 313, 339
Herrera, F.	; 147, 175
Huertas, J.L.	; 301
Jacas, J.	; 41
Jiménez, A.	; 253
Lago, R.	; 47
Lamata, M.T.	; 161

Lisa, F.	; 289
López, E.	; 355
Luna de Toledo, M.	; 103
Matía, F.	; 253
Mayor, G.	; 17
Medina, J.M.	; 233
Mendaña, C.	; 355
Moral, S.	; 161
Morales, R.	; 185, 203, 381
Núñez, A.	; 371
Olivas, J.A.	; 47
Pablo, E. de	; 127
Pereira, R.	; 93
Pérez de la Cruz, J.L.	; 185, 203, 381
Pons, O.	; 215
Recasens, J.	; 41
Reina, R.	; 73
Requena, I.	; 81
Reyes de los Mozos, M.	; 289
Rives, J.	; 93
Ruiz, A.	; 313, 339
Ruiz, R.	; 323
Sánchez, S.	; 301
Sancho, J.	; 371
Sanz, R.	; 253
Sobrino, A.	; 23, 47
Tarela, J.M.	; 267
Torrens, J.	; 17
Triguero, F.A.	; 185
Trillas, E.	; 1
Urbano, J.A.	; 381
Valderrama, E.	; 289
Verdegay, J.L.	; 147
Vila, M.A.	; 215, 233
Zurita, J.M.	; 283