LENGUAJES FORMALES BORROSOS

Claudio Moraga

European Centre for Soft Computing
Mieres, Asturias

Resumen
Se presenta un estudio sobre lenguajes formales borrosos obtenidos a partir de lenguajes de Chomsky, lenguajes de Petri y lenguajes de Lindenmayer, utilizando alfabetos nítidos, pero producciones borrosas acompañadas de t-normas y t-conormas. Se muestra que mediante una generalización de los cortes alfa es posible derivar lenguajes nítidos asociados a un mismo lenguaje borroso, que no solo son distintos como conjuntos de palabras, sino que además suelen ser de distinto tipo dentro de sus familias de origen.
Palabras clave: Lenguajes borrosos, cortes alfa, grado de pertenencia.

Abstract
This paper studies fuzzy formal languages based on Chomsky, Petri and Lindenmayer languages, by using crisp alphabets but fuzzy productions accompanied by t-norms and t-conorms. It is shown that a generalization of alpha cuts leads to the derivation of crisp languages associated to a same fuzzy formal language, which not only are different with respect to the words that they contain, but also they are quite often of a different type within their families of origin.
Keywords: Fuzzy languages, alpha cuts, membership degree.

1. Introducción

Un lenguaje formal nítido es un conjunto de palabras construidas sobre un alfabeto dado, siguiendo reglas propias del lenguaje. Los lenguajes formales son mayormente infinitos y estudian fundamentalmente la estructura del lenguaje. Los lenguajes formales, a diferencia de los lenguajes

Recibido: 06/09/06. Aceptado: 26/10/06.

naturales, no requieren de una semántica. Un lenguaje formal borroso, es un conjunto borroso de palabras construidas sobre un alfabeto dado, pero cada palabra tiene en este caso un *grado de pertenencia* al lenguaje. Los lenguajes naturales son un buen ejemplo de lenguajes borrosos que evolucionan con el pasar del tiempo. Mas allá de las palabras que oficialmente pertenecen al lenguaje, cada generación acuña palabras propias, a las que el uso insistente les otorga de facto un alto grado de pertenencia al lenguaje (hasta que pasan de moda y son reemplazadas por otras). Los lenguajes naturales tienen también aspectos borrosos desde el punto de vista semántico, por cuanto permiten expresar predicados vagos. Por ejemplo, al decir "la humedad relativa hoy es *baja*", es el uso del lenguaje en un contexto, un tiempo y un lugar dados, el que le da significado al predicado vago "*baja*" y distintos porcentajes de humedad relativa, tendrán posiblemente también distintos grados de pertenencia al conjunto borroso que representa dicho predicado vago.

En el área de lenguajes formales hay tres familias de lenguajes, que son representativas de distintas opciones para generar las palabras pertenecientes a dichos lenguajes. Se trata de los lenguajes formales introducidos por N. Chomsky [Cho 59], [Cho 63], [Cho 64], [Cho 65], los lenguajes formales generados con Redes de Petri [Pet 73] , [Hac 75] y los lenguajes de Lindenmayer [Lin 68], [Lin 71], [Her 74], cuyas raices se encuentran en la botánica. Los lenguajes de Chomsky se caracterizan por *secuencias* de transformaciones elementales que llevan de un símbolo auxiliar inicial a una palabra del lenguaje. En el caso de los lenguajes generados por Redes de Petri, las transformaciones elementales se aplican *de manera concurrente* siguiendo la dinámica de la red. Finalmente, en el caso de los lenguajes de Lindenmayer, las transformaciones elementales se aplican *simultáneamente* a todos los símbolos de una palabra para generar una nueva palabra.

Los primeros en introducir el concepto de lenguaje borroso fueron E.T. Lee y L. Zadeh [LeZ 69], [Zad 71], en el contexto de lenguajes naturales. En este caso se destaca la flexibilidad del lenguaje natural para expresar predicados vagos y se analizan las posibilidades de su procesamiento formal. En el contexto de los lenguajes formales posiblemente fue la tesis doctoral de E.T. Lee [Lee 72] la que abrió el camino para la investigación de los lenguajes formales borrosos. Si bien los mayores esfuerzos de investigación se han centrado en los lenguajes formales borrosos derivados de la jerarquía de Chomsky (p. ej. [Asv 03], [Pet 05] entre los mas recientes), cabe mencionar [Per 90] que estudia algunos aspectos de lenguajes formales borrosos basados en Redes de Petri, como también [MSM 97], que incursiona en los lenguajes borrosos de Lindenmayer.

Un lenguaje formal nítido, como se ha dicho anteriormente, es un conjunto de palabras sobre un alfabeto dado. A todo lenguaje formal se le asocia una estructura generadora, la cual especifica el conjunto finito de símbolos a usar, normalmente llamado el alfabeto, posiblemente un conjunto finito de símbolos auxiliares, y un conjunto finito de reglas para generar las palabras. Se ha hecho énfasis en que la estructura generadora consiste en conjuntos finitos (de símbolos y reglas); no obstante lo cual ésta puede dar lugar a un lenguaje formal infinito. En el caso de los lenguajes de Chomsky, la estructura generadora es una gramática mientras que en los lenguajes de Petri, es una red. Los lenguajes de Lindenmayer tienen como estructura generadora un "sistema-L". Los sistemas-L son análogos a las gramáticas de Chomsky, excepto por la forma fundamentalmente distinta de utilizar las reglas.

Para generar un lenguaje formal borroso a partir de uno nítido hay dos alternativas básicas: una de ellas consiste en reemplazar los conjuntos nítidos de símbolos existentes en la estructura generadora del lenguaje, por conjuntos borrosos. La otra preserva los conjuntos nítidos de símbolos, pero considera un conjunto borroso de reglas, adjudicando a cada una de ellas una ponderación o "grado de fuerza" en el intervalo [0,1]. En ambos casos hay que extender la estructura generadora con operaciones adecuadas al entorno borroso elegido. Estas operaciones serán normalmente t-normas y t-conormas [Men 43], [ScS 83], [ATV 83], [KMP 00]. Dado que hay un número infinito de t-normas y conormas, entonces también habrá un número infinito de lenguajes borrosos que se podrán asociar a uno nítido. En lo que sigue se utilizará el símbolo $\otimes$ para representar una t-norma y el símbolo $\oplus$ para representar una t-conorma.

El resto de este artículo está estructurado como sigue: primero se analizarán algunas propiedades de lenguajes borrosos derivados de lenguajes de Chomsky y se discutirán los correspondientes autómatas borrosos. Seguidamente se analizarán lenguajes borrosos basados en redes borrosas de Petri. En la cuarta sección se discutirán ciertos aspectos de lenguajes borrosos obtenidos de lenguajes de Lindenmayer. En los tres casos se analizarán lenguajes nítidos que pueden ser derivados de un lenguaje formal borroso. Algunas consideraciones generales cerrarán el trabajo.

2. Lenguajes borrosos de Chomsky

Definición 1:

Una gramática borrosa G es una estructura generadora de 7 elementos (N, T, S, P, ω, ⊗, ⊕), donde (N, T, S, P) es una gramática de Chomsky y ω: P → [0,1] asocia a cada producción en P una ponderación en el intervalo unitario. El lenguaje borroso L(G) generado con la gramática borrosa G es $\{(w, \mu_L(w)) \mid w \in T^*, S \Rightarrow^* w, \mu_L(w) = \otimes_i p_i\}$[1], donde $\mu_L(w)$ denota el grado de pertenencia de la palabra w al lenguaje L(G) y se obtiene aplicando la t-norma ⊗ a todas las ponderaciones de las producciones involucradas en la generación de w a partir de S. Si la gramática es ambigua, vale decir, si alguna palabra w puede derivarse de S utilizando distintas secuencias de producciones de P, se utilizará la t-conorma ⊕ para calcular el grado de pertenencia final de tal palabra al lenguaje, aplicándola sobre los grados de pertenencia preliminares que puedan haberse obtenido. (Si la gramática no es ambigua se omitirá la t-conorma ⊕).

Definición 2:

Un autómata finito borroso $\mathcal{A}$ está especificado por $(\Sigma, Q, F, q_0, \delta, \varphi, \otimes)$, donde $(\Sigma, Q, F, q_0, \delta)$ representa un autómata finito (nítido) y $\varphi: (\Sigma \times Q \times Q) \rightarrow [0,1]$ asocia una ponderación en el intervalo unidad a cada transición del autómata.

Es simple de ver que un autómata finito borroso $\mathcal{A}$ puede ser utilizado para generar un lenguaje regular borroso $L(\mathcal{A})$ si se elige Σ como el alfabeto del lenguaje, si se considera que una secuencia de símbolos de Σ correspondiente a una trayectoria o cadena de transiciones desde q_0 hasta algún estado de F constituye una palabra del lenguaje y, finalmente, si se aplica la t-norma ⊗ a las ponderaciones de las respectivas transiciones para producir una ponderación global, la cual será considerada como el grado de pertenencia de la palabra generada al lenguaje $L(\mathcal{A})$. Este proceso es reversible en el siguiente sentido: un autómata finito borroso $\mathcal{A}$ acepta un lenguaje regular borroso $\mathcal{L}$ si y solo si $(\Sigma, Q, F, q_0, \delta)$ acepta el lenguaje regular nítido deducible de $\mathcal{L}$ mediante la asignación

[1] La notación T^* representa el conjunto de todas las palabras que se pueden construir con el alfabeto T, incluyendo la palabra vacía. Una palabra vacía es aquélla que no tiene ningún símbolo. La notación T^+ se usará para representar el conjunto de todas las palabras que se pueden construir con el alfabeto T, pero sin incluir la palabra vacía. El símbolo $\Rightarrow^*$ representa una secuencia de derivaciones elementales.

de la ponderación "1" a todas sus palabras y, además, por cada palabra w del lenguaje, existe una trayectoria en el autómata, que acepta dicha palabra y cuya ponderación global es igual a $\mu_L(w)$. Nótese que debido a la asociatividad y conmutatividad de las t-normas, es posible que dos autómatas borrosos diferentes acepten un mismo lenguaje regular borroso. (Ver ejemplo en la figura 1. Ambos autómatas aceptan la palabra $w = a^n b$ con $\mu_L(w) = 0.35 \cdot (0.9)^{n-1}$ aun cuando las trayectorias correspondientes son formalmente distintas.)

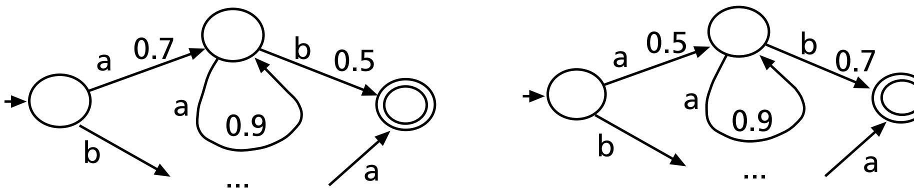


Fig. 1. Vista parcial de dos autómatas finitos borrosos diferentes que aceptan el mismo lenguaje regular borroso

Proposición 1:

Para cada autómata finito borroso $\mathcal{A}$, existe una gramática borrosa lineal hacia la derecha G_D y una gramática borrosa lineal hacia la izquierda G_I, tales que $L(\mathcal{A}) = L(G_D) = L(G_I)$. Las gramáticas lineales se obtienen de la misma manera que en el caso nítido y adicionalmente se asocia a cada producción, la ponderación de la transición correspondiente del autómata borroso y tanto en el autómata borroso como en la gramática borrosa lineal G_D o G_I se usa la misma t-norma. Recuérdese que G_D genera una palabra de izquierda a derecha, mientras que G_I genera la misma palabra de la derecha a la izquierda. En el automata borroso esto es equivalente a recorrer, en el primer caso, la trayectoria correspondiente desde el estado inicial al final mientras que en el segundo, la trayectoria se recorre desde el estado final al inicial. Debido a la asociatividad y conmutatividad de las t-normas, la ponderación global de una trayectoria es independiente del sentido en el cual se recorra. La figura 2 ilustra un ejemplo.

En el caso de los lenguajes regulares nítidos, lo anterior es bien conocido (ver e.g. [Pfe 00]). La validez de la proposición 1 se basa, consecuentemente, en lo sabido para lenguajes nítidos y en las propiedades de asociatividad y conmutatividad de las t-normas.

Proposición 2:

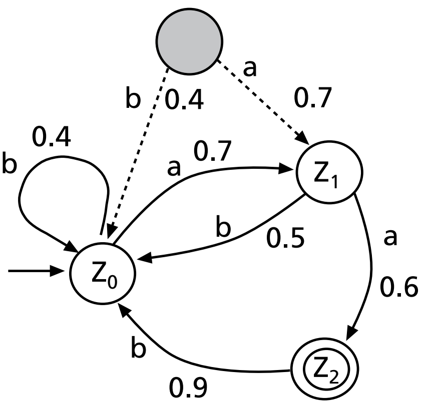
Para cada gramática regular borrosa G existe un autómata finito borroso *no-determinístico* $\mathcal{A}$, tal que L($\mathcal{A}$) = L(G). Ver ejemplo en figura 3. A diferencia del caso nítido, no rige que para cada autómata borroso finito no determinístico y cualquier t-norma, exista uno borroso finito determinístico equivalente, con la misma t-norma.

Definición 3:

Sea G = (N, T, S, P, ω, ⊗) una gramática borrosa no ambigua y sea P_w el subconjunto de producciones de P que se utilizan para generar la palabra w. Entonces

$$(L(G), \#\alpha) = \{w \in T^* \mid \mu_L(w) \ \# \ \alpha\}$$
$$(L(G); \#\alpha) = \{w \in T^* \mid \forall p \in P_w, \ \omega(p) \ \# \ \alpha\}$$

donde # ∈ {≥, >, =, <, ≤} cuando α ∈ [0,1]. Si α es un subconjunto de [0,1] entonces # puede representar ∈ o ∉.



e.g. w = babaa
con G_D: p_2, p_1, p_4, p_1, p_3
$\mu_L(w) = 0.4 \otimes 0.7 \otimes 0.5 \otimes 0.7 \otimes 0.6$
con G_I: $p'_1, p'_2, p'_3, p'_2, p'_4$
$\mu_L(w) = 0.6 \otimes 0.7 \otimes 0.5 \otimes 0.7 \otimes 0.4$

$G_D = (T,N,S,P,\omega,\otimes)$
con: $T = \{a,b\}$, $N = \{z_0,z_1,z_2\}$, $S = z_0$
P = { p_1: $z_0 \to az_1$ $\omega(p_1) = 0.7$
p_2: $z_0 \to bz_0$ $\omega(p_2) = 0.4$
p_3: $z_1 \to az_2 \mid a$ $\omega(p_3) = 0.6$
p_4: $z_1 \to bz_0$ $\omega(p_4) = 0.5$
p_5: $z_2 \to bz_0$ $\omega(p_5) = 0.9$ }
$G_I = (T,N,S',P',\omega',\otimes)$
con: $T = \{a,b\}$, N = {z_0,z_1,z_2, ●}, $S' = z_2$
P' = { p'_1: z_2 ♦ z_1a $\omega'(p'_1) = 0.6$
p'_2: z_1 ♦ $z_0a \mid a$ $\omega'(p'_2) = 0.7$
p'_3: z_0 ♦ z_1b $\omega'(p'_3) = 0.5$
p'_4: z_0 ♦ $z_0b \mid b$ $\omega'(p'_4) = 0.4$
p'_5: z_0 ♦ z_2b $\omega'(p'_5) = 0.9$ }

Fig. 2: Gramáticas borrosas lineal hacia la derecha y lineal hacia la izquierda deducidas de un autómata finito borroso y de su equivalente con un estado inicial sin aristas entrantes. Derivación de la palabra "babaa" y cálculo de su grado consistente de pertenencia al lenguaje.

Sea $G = (T,N,S,P,\omega, \otimes)$,

con: $T = \{a,b\}$, $N = \{z_0,z_1,z_2\}$, $S = z_0$

$$
\begin{array}{ll}
P = \{p_1 : z_0 \to az_1 & \omega(p_1) = 0.7 \\
\quad p_2 : z_0 \to bz_2 & \omega(p_2) = 0.8 \\
\quad p_3 : z_0 \to b & \omega(p_3) = 0.6 \\
\quad p_4 : z_1 \to az_2 & \omega(p_4) = 0.6 \\
\quad p_5 : z_1 \to a & \omega(p_5) = 0.7 \\
\quad p_6 : z_2 \to bz_0 & \omega(p_6) = 0.9 \;\}
\end{array}
$$

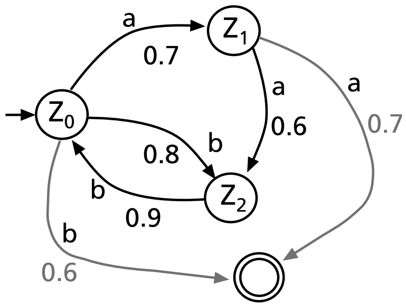


Fig. 3: Una gramática borrosa lineal hacia la derecha y un autómata borroso finito no-determinístico que acepta el lenguaje L(G).

Es simple notar que # representa una generalización del concepto de corte alfa en conjuntos borrosos. En el caso de $(L(G), \#\alpha)$, se trata de un corte alfa generalizado sobre el conjunto borroso de palabras que constituyen el lenguaje, mientras que en el caso $(L(G); \#\alpha)$, es sobre el conjunto borroso de las producciones $\{(p, \omega(p)) \mid p \in P\}$. Nótese que tanto $(L(G), \#\alpha)$ como $(L(G); \#\alpha)$ son lenguajes formales *nítidos*, deducidos del lenguaje borroso L(G). Por otra parte, si bien no es el caso en la teoría de conjuntos borrosos, también es posible utilizar los cortes alfa (generalizados o no), para generar subconjuntos –(sub-lenguajes, en el caso del presente trabajo) – borrosos, de la siguiente manera:

$$(L(G), \#\alpha)_{\text{borroso}} = \{(w, \mu_L(w)) \mid w \in T^* \text{ , } S \Rightarrow^* w, \mu_L(w) \# \alpha\}$$

y de forma análoga para $(L(G); \#\alpha)_{\text{borroso}}$

Ejemplo 1:

Sea $G = (N,T,S,P,\omega,\otimes)$, donde (N,T,S,P) es una gramática regular, $N = \{S,A,B\}$, $T = \{a,b,c\}$, y $\otimes$ es la t-norma producto. Finalmente, sean las siguientes producciones y sus respectivas ponderaciones:

$p_1 : S \to aS$	$\omega(p_1) = 0.7$	$p_2 : S \to aA$	$\omega(p_2) = 0.7$
$p_3 : A \to bA$	$\omega(p_3) = 0.5$	$p_4 : A \to bB$	$\omega(p_4) = 0.5$
$p_5 : B \to cB$	$\omega(p_5) = 0.3$	$p_6 : B \to c$	$\omega(p_6) = 0.3$

Resulta evidente que $S \Rightarrow^* a^ib^jc^k$, con $i, j, k \geq 1$ y que $L(G) = \{a^ib^jc^k \mid i, j, k \geq 1\}$. Además $\mu_L(a^ib^jc^k) = (0.7)^i \cdot (0.5)^j \cdot (0.3)^k$. Dado que los números

3, 5 y 7 son primos y el producto de $0.7 \cdot 0.5 \cdot 0.3 = 0.105$, entonces $(0.105)^n$ con $n \geq 1$, tiene una única descomposición en términos de sus factores primos menores que 1, la cual es $(0.7)^n \cdot (0.5)^n \cdot (0.3)^n$. De esto se sigue que:

$$L(G, \in \{(0.105)^n \mid n \geq 1\}) = \{a^n b^n c^n \mid n \geq 1\}$$

De manera similar, $(0.35)^n$ tiene una única descomposición en términos de sus factores primos menores que 1: $(0.7)^n \cdot (0.5)^n$, con lo cual

$$L(G, \in \{(0.35)^n \cdot (0.3)^m \mid m \geq 0, n \geq 1\}) = \{a^n b^n c^m \mid m \geq 0, n \geq 1\}$$

Nótese que en la jerarquía de Chomsky, $\{a^n b^n c^n \mid n \geq 1\}$ es un lenguaje de contexto sensitivo y $\{a^n b^n c^m \mid m \geq 0, n \geq 1\}$ es un lenguaje de contexto libre mientras que la gramática G es solo regular.

Observación 1:
Si en vez de considerar la analogía de las ponderaciones de las reglas de producción como "grado de fuerza", lo que lógicamente condujo al uso de t-normas para calcular ponderaciones globales —("la fuerza de una cadena depende de su eslabón mas débil")— mientras que las t-conormas se reservaron para resolver ambigüedades, se usara la analogía "costo", parecería lógico calcular ponderaciones globales mediante t-conormas, mientras que las ambigüedades serían resueltas por t-normas, en el sentido de elegir la alternativa no mas costosa que la de menor costo relativo. Esto sugiere que dadas las ponderaciones de las transformaciones elementales requeridas por el proceso de generar palabras, es posible generar *pares* de lenguajes borrosos, usando ambas analogías antes mencionadas. Un estudio de la relación en que están estos pares de lenguajes borrosos y sus posibles consecuencias, es un campo abierto que hasta hoy no ha sido abordado. En todo caso resulta fácil de entender que para una estructura generadora y ponderaciones dadas, el lenguaje borroso obtenido calculando el grado de pertenencia de las palabras mediante una t-norma será un sublenguaje borroso de aquél obtenido calculando el grado de pertenencia de las palabras mediante una t-conorma.[2] Es importante destacar que la proposición 1 es válida, cualquiera que sea la analogía que se use, por cuanto su vigencia se basa en que la operación utilizada

[2] Recuérdese que si P y Q son conjuntos borrosos, $P \Subset Q$, si y solo si $\forall x \in P$, $\mu_P(x) \leq \mu_Q(x)$

para el cálculo de ponderaciones globales es cerrada en el intervalo [0,1], asociativa y conmutativa, propiedades que son características tanto de las t-normas como de las t-conormas. Mas aún, esta última observación atrae la atención sobre las operaciones llamadas *normas* [KlY 95] y *uninormas* [YaR 96], (de las cuales t-normas y t-conormas son casos especiales), que también satisfacen las condiciones requeridas, abriéndose así la posibilidad de generar nuevos lenguajes borrosos asociados a un mismo sistema generador.

3. Lenguajes borrosos basados en redes de Petri

Definición 4:

Una red de Petri (RP) es un grafo dirigido cuyos nodos se pueden separar en dos conjuntos disjuntos no vacíos, de plazas (P) y transiciones (T), tales que $P \cap T = \varnothing$. El conjunto de las aristas será denominado A. Las aristas conectan plazas con transiciones o transiciones con plazas, lo cual formalmente se expresa como $A \subseteq P{\times}T \cup T{\times}P$. Una función $\mu : P \to \_ \cup \{0\}$ asocia a cada plaza un número de marcas (o la deja vacía). El conjunto ordenado por plazas, de las cantidades de marcas que ellas tienen, se denomina el marcado de la red. El comportamiento dinámico de la red se manifiesta por cambios en el marcado. (Cabe hacer notar que el marcado de una RP identifica el estado de la red, lo cual permite reconocer una cierta analogía con los autómatas finitos. La capacidad de modelado de una RP es, sin embargo, muy superior a la de los autómatas finitos. Basta señalar que todo lenguaje regular puede ser generado por una RP, (ver definición 5), mientras que no todo lenguaje generado por una RP es aceptado por un autómata finito [Pet 77])

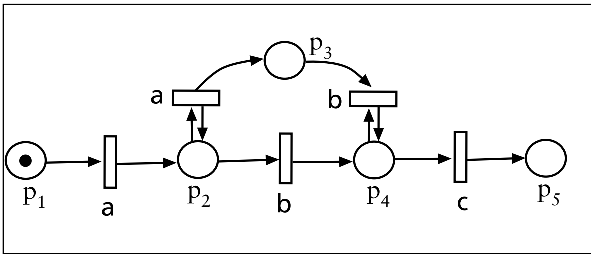
Una transición se denomina *preparada*, cuando todas las plazas que inciden en tal transición tienen por lo menos una marca. En tal caso la transición puede *activarse*. Cuando esto sucede, cada una de las plazas incidentes se desprenden de una marca y cada una de las plazas que emergen de la transición, incrementa en una marca su contingente. Resulta evidente que cada vez que una transición se activa, puede cambiar la marcación de la red.

En un momento dado puede ocurrir que en una RP haya varias transiciones que estén preparadas. El formalismo de las RP no especifica el *orden temporal* según el cual se deban activar las transiciones preparadas. Todo posible orden temporal, incluyendo activaciones simultáneas, es aceptable. Se dice que las transiciones se activan de manera *concurrente*.

Definición 5:

Una RP generadora de lenguaje es un quinteto (RP, ρ, Σ, M_0, M_F) donde RP es una red de Petri según la definición 4; $\rho : T \rightarrow \Sigma$ es una función rotuladora que asocia a cada transición un símbolo del alfabeto finito, no vacío, Σ, M_0 representa la marcación inicial y M_F es un conjunto de marcaciones finales. Una vez alcanzada una marcación final, se da término a la palabra generada. El proceso generador es muy simple: se comienza con un símbolo vacío y partiendo con M_0, cada vez que una transición se activa, se agrega a la palabra en generación el símbolo asociado a dicha transición. Véase un ejemplo en la figura 4.

Ejemplo 2:



Marcación inicial:
{1, 0, 0, 0, 0}
Marcación final:
{0, 0, 0, 0, 1}
Lenguaje generado:

$L(PR) = \{a^n b^n c \mid n > 0\}$

Fig. 4: Una red de Petri y su lenguaje generado

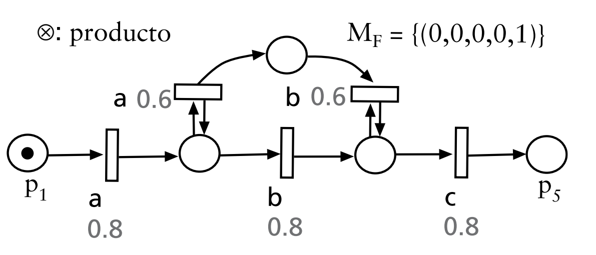
Definición 6:

Una red de Petri generadora de un lenguaje borroso está especificada por el septeto (RP, ρ, Σ, M_0, M_F, ω, $\otimes$), donde (RP, ρ, Σ, M_0, M_F) es una red de Petri generadora de lenguaje, (ver definición 5), $\omega : T \rightarrow [0,1]$ asigna a cada transición una ponderación ("grado de fuerza") y $\otimes$ es una t-norma para calcular la ponderación de la palabra generada, que será interpretada como el grado de pertenencia de la palabra al lenguaje, sobre la base de las ponderaciones de las transiciones activadas. (Tal como en el caso de las gramáticas ambiguas de Chomsky, si una red de Petri puede generar una misma palabra mediante distintas secuencias de transiciones activadas, entonces hay que especificar una t-conorma $\oplus$ para calcular el grado de pertenencia final de la palabra correspondiente, sobre la base de las ponderaciones preliminares.)

(Recuérdese la observación 1. Aquí también se puede utilizar la analogía "costo" para la activación de las transiciones, en cuyo caso, la ponderación total se calcularía con una t-conorma y en caso de existir ambigüedades, se resolverían mediante el uso de una t-norma.)

En la figura 5 se muestra una red de Petri generadora de un lenguaje borroso y los lenguajes nítidos derivados con distintos cortes alfa.

Ejemplo 3:



$(L, > 0) = \{a^p b^p c \mid p > 0\}$

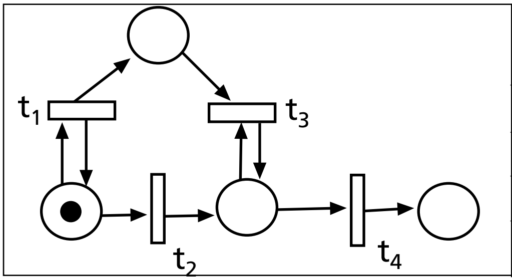
$(L, > ((0.8)^3 \cdot (0.6)^{2(n-1)})) =$

$= \{a^p b^p c \mid 0 < p \leq n\}$ para algún $n \in$ _

Fig. 5: Red de Petri generadora de un lenguaje borroso.

En el ejemplo 3 se puede apreciar que $(L, > 0)$, desde el punto de vista de los lenguajes de Chomsky es un lenguaje de contexto libre, mientras que para todo n natural, los lenguajes generados por $(L, > ((0.8)^3 \cdot (0.6)^{2(n-1)}))$ van a ser finitos y, por lo tanto regulares.

Para los lenguajes de Petri, J.L. Peterson introdujo una jerarquía, considerando el conjunto de marcaciones finales y el tipo de función rotuladora ρ [Pet 76]. En lo referente a las marcaciones finales, Peterson distingue el caso en el cual M_F especifica unívocamente un conjunto finito de marcaciones finales, y denomina los lenguajes resultantes como de "tipo L". En el caso de que M_F se considerara como un conjunto finito de *cotas inferiores* de las marcaciones finales, entonces los lenguajes resultantes se denominan de "tipo G" y, finalmente, si M_F representa el conjunto de todas las posibles marcaciones de la red con la cual se está trabajando, los lenguajes son de "tipo P". En cuanto a la función rotuladora, si ésta asigna a cada transición un símbolo explícito distinto de Σ, el lenguaje se llama "rico" y, si ρ no efectúa una rotulación rica, pero a ninguna transición le asigna el símbolo vacío λ, el lenguaje se llama "λ-libre". En todos los demás casos, el lenguaje no tiene nombre especial. Resulta evidente que la jerarquía bidimensional de Peterson es de alta complejidad. Un ejemplo del efecto del tipo de rotulación sobre el lenguaje generado se muestra en la figura 6.



Rótulos t_1 t_2 t_3 t_4	Tipo	Lenguaje
a b c d	rico	$\{a^n bc^n d \mid n \geq 0\}$
a b a b	λ-libre	$\{(a^n b)^2 \mid n > 0\}$
λ b c d	general	$\{bc^n d \mid n \geq 0\}$
$M_F = \{(0,0,0,1)\}$		

Fig. 6: Lenguajes nítidos de Petri generados con diferentes rotulaciones

Ejemplo 4:

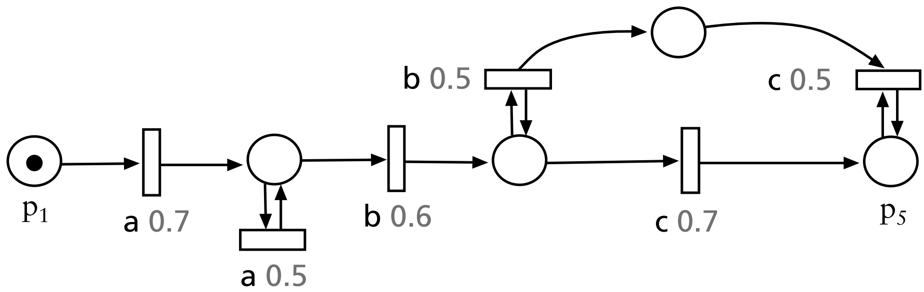


Fig. 7: Red de Petri generadora de un lenguaje borroso

De la estructura de la red de Petri se puede deducir que el grado de pertenencia de las palabras generadas $w = a^p b^q c^q$, al lenguaje L(PR), está dado por la ecuación siguiente:

$$\mu_{L(RP)}(w) = 0.7 \otimes (0.5)^{p-1} \otimes 0.6 \otimes (0.5)^{q-1} \otimes 0.7 \, f \, (0.5)^{q-1}$$
$$= 0.294 \cdot (0.5)^{p-1} \cdot (0.5)^{2(q-1)} = 0.294 \cdot (0.5)^{p+2q-3}$$

Si p = q =1 entonces $\mu_{L(RP)}$ (w) = 0.294·(0.5)0 = 0.294
Si p=2, q=1 entonces $\mu_{L(RP)}$ (w) = 0.294·(0.5)1 = 0.147
Si p=1, q=2 entonces $\mu_{L(RP)}$ (w) = 0.294·(0.5)2 = 0.0735
Si p=2, q=2 entonces $\mu_{L(RP)}$ (w) = 0.294·(0.5)3 = 0.03675
…

De lo cual se sigue que:

(L, > 0.2) = {abc}
(L, > 0.1) = {abc, a2bc}
(L, > 0.07) = {abc, a2bc, ab2c2}
(L, > 0,03) = {abc, a2bc, ab2c2, a2b2c2 }
...

Es fácil observar que el valor de $\mu_{L(RP)}$ (w) disminuye a medida que aumenta el valor de p y q, con lo cual los lenguajes obtenidos con cortes alfa crecen, incluyendo nuevas palabras, a medida que el nivel alfa desciende. Dicho de otra manera, el nivel alfa controla el crecimiento del lenguaje nítido derivado. Desde el punto de vista de Peterson, todos los lenguajes nítidos obtenidos con los cortes alfa son de tipo L; además todos son obviamente λ-libres (ya que no hay transiciones rotuladas con λ en la RP original), siendo (L, > 0.2) el único que puede ser generado por una RP con rotulación rica.

Si una red de Petri se va a utilizar para generar un lenguaje borroso y hay alguna transición rotulada con el símbolo vacío “λ", esta transición no tiene un efecto directo sobre la palabra generada; tan solo sobre el funcionamiento de la red (y solo desde este punto de vista, tiene un efecto indirecto sobre la palabra generada). Por esta razón las transiciones rotuladas λ se considerarán primeramente como si tuvieran ponderación 1. Esta situación también es válida en el caso de querer estudiar los lenguajes (L(RP), #α) derivados del lenguaje borroso. Si se desean estudiar los lenguajes derivados de tipo (L(RP); #α), entonces sí que tiene sentido adjudicar a las transiciones λ una ponderación que puede ser diferente de 1 a fin de poder desactivarlas en función del parámetro de control α y la relación # (ver ejemplo 5).

En lo referente a los lenguajes (L(RP); #α), para cada valor de α y tipo de # hay distintas transiciones que se considerarán como eliminadas de la red y será la red resultante la que genere en cada caso el lenguaje correspondiente. La figura 8 presenta una RP generadora de un lenguaje borroso, en cuyo análisis se usará la notación (rótulo, ponderación) para identificar las transiciones.

Ejemplo 5:

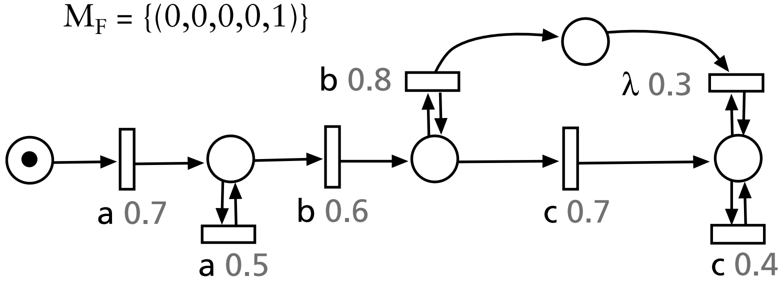


Fig. 8: Red de Petri generadora de un lenguaje borroso

(L(RP) ; > 0)	= $\{a^p b^q c^r \mid p, q, r > 0\}$	Lenguaje de tipo L, con una transición λ
(L(RP) ; > 0.3)	= $\{a^p b c^r \mid p, r > 0\}$	Lenguaje de tipo L, λ-libre
(L(RP) ; > 0.4)	= $\{a^p bc \mid p > 0\}$	Lenguaje de tipo L, λ-libre
(L(RP) ; > 0.5)	= {abc }	Lenguaje de tipo L, rico

En el caso del lenguaje (L(RP) ; > 0), la transición rotulada λ permite desocupar la plaza p_4 cada vez que la transición (b, 0.8) se activa, a fin de poder alcanzar la marcación final. Al mismo tiempo, nótese que debido al no-determinismo que es posible en una RP, toda vez que la plaza p_5 reciba una marca, tanto la transición (c, 0.4) como la transición (λ, 0.3) estarán preparadas (si la transición (b, 0.8) ha sido previamente activada), pero solo una de ellas podrá activarse. Por esta razón, las palabras de este lenguaje pueden terminar libremente con varias "c". En lo que respecta al lenguaje (L(RP) ; > 0.3), en su generación se considera la transición (λ, 0.3) como no existente. Es interesante notar que la transición (b, 0.8), pese a tener una ponderación mayor que 0.3 no se activa a fin de mantener la plaza p_4 vacía, que es condición necesaria para poder alcanzar la marcación final especificada. Es de hacer notar que aunque todos los lenguajes derivados son del mimo tipo "L", en lo que respecta a la rotulación *de la red resultante*, son de distinto tipo. Cuando α es 0, el lenguaje es de tipo general; cuando α es 0.3 o 0.4, es λ-libre y cuando α es 0.5, el lenguaje es rico.

4. Lenguajes borrosos de Lindenmayer

Arístides Lindenmayer, un estudioso de la botánica, desarrolló la familia de lenguajes formales que ahora lleva su nombre [Lin 71], basándose en formas de crecimiento de algunos organismos vegetales simples.

Definición 7:

La estructura generadora de un lenguaje de Lindenmayer ha sido llamada "sistema L". El sistema L mas simple es el sistema 0L, el cual consta de un alfabeto finito no vacío V, de una *palabra* inicial $w_0 \in V^+$, y de un conjunto finito no vacío de producciones $P \subset (V \times V^*)$, las que representan reglas para la transformación de símbolos según el siguiente esquema: Sean u y v palabras en V^*, tales que $u = u_1, u_2,...u_n$, donde $u_i \in V$, y $v = v_1,v_2,...v_n$, donde $v_i \in V^*$ con $i = 1, 2, ..., n$. Entonces $u \Rightarrow v$ –(v es derivable a partir de u)– si y solo si existen producciones p_i en P tales que $p_i(u_i) = v_i$, con $i = 1, 2, ..., n$.

Es simple observar que en los sistemas 0L, $|P| \geq |V|$, es decir, tiene que haber por lo menos tantas producciones elementales como símbolos en el alfabeto. Además, las producciones elementales pertinentes se aplican *simultáneamente* a todos los símbolos de una palabra. Por esta razón, los sistemas 0L suelen ser también llamados sistemas de reescritura o de reemplazo de palabras.

Un sistema 0L se llama determinístico, abreviado D0L, si todas las producciones en P son funciones, es decir si no hay dos producciones que tengan un mismo símbolo de V al lado izquierdo y distintas palabras de V^* al lado derecho. En tal caso $|P| = |V|$. Si las producciones tienen al lado derecho palabras de V^+, el sistema se denomina λ-libre o propagador, abreviado P0L. Un sistema DP0L es a la vez determinístico y propagador.

Definición 8:

El lenguaje generado por un sistema 0L es

$$L\ (0L) = \{w \in V^* \mid w_0 \Rightarrow^* w\}$$

Se usa el símbolo _ para el lenguaje, a fin de distinguirlo de "L" en el sistema 0L. Dado que hay un solo alfabeto V, en cada etapa de derivación se obtiene una nueva palabra del lenguaje. (Cabe notar una cierta analogía con los lenguajes de tipo P en la jerarquía de Peterson para los lenguajes de Petri). El lenguaje L obviamente hereda las características del sistema generador: así, L (D0L) es un lenguaje determinístico, L (P0L) es propagador y L (DP0L), tiene ambas propiedades.

Definición 9:

Un sistema 0L se llama especial, abreviado E0L, si existe un subalfabeto F $\not\subset$ V para construir las palabras del lenguaje. La especificación de un sistema E0L es, consecuentemente, (V, F, w_0, P).

$$\_(E0L) = \{ w \in F^* \mid w_0 \Rightarrow^* w \}$$

En el caso de la generación de un lenguaje _(E0L) cada etapa de derivación no necesariamente produce una nueva palabra del lenguaje. (En este sentido se puede apreciar una cierta analogía con los lenguajes de Chomsky).

Un lenguaje _(EDP0L) es un lenguaje de Lindenmayer, especial, determinístico y propagador.

Definición 10:

Un sistema ponderado ω0L tiene la siguiente estructura: (V, w_0, P, ω, ⊗, ⊕), donde (V, w_0, P) es un sistema 0L y ω: P → [0,1] asigna una ponderación a cada producción en P. Un sistema ω0L genera un lenguaje borroso _(ω0L) de Lindenmayer. En cada etapa de derivación, se utiliza la t-norma para calcular la ponderación (posiblemente preliminar) de cada palabra sobre la base de las ponderaciones de las producciones elementales utilizadas tanto en la etapa actual como en las anteriores. Si una palabra puede generarse con distintas secuencias de etapas de derivación, se usa la t-conorma para calcular la ponderación final de dicha palabra considerando las ponderaciones preliminares. La ponderación final se considera como el grado de pertenencia de la palabra al lenguaje. La palabra inicial tiene siempre grado de pertenencia 1 al lenguaje.

Un sistema ponderado ωE0L tiene la siguiente estructura: (V, F, w_0, P, ω, ⊗, min, ⊕), donde (V, F, w_0, P) es un sistema E0L y ω: P → [0,1] asigna una ponderación a cada producción en P. Un sistema ωE0L genera un lenguaje borroso _(ωE0L) de Lindenmayer. En cada etapa de derivación, se utiliza la t-norma ⊗ a fin de calcular para cada palabra la ponderación correspondiente a dicha etapa de derivación. Si la palabra alcanzada pertenece a F* la ponderación de la etapa se agrega con ⊗. Si la palabra alcanzada no pertenece a F* la ponderación de la etapa se agrega con la t-norma mínimo. Si una palabra de F* puede generarse con distintas secuencias de etapas de derivación, se usa la t-conorma para calcular la ponderación final de dicha palabra considerando las ponderaciones preliminares. La ponderación final se considera como el grado de pertenencia de la palabra al lenguaje.

Ejemplo 6:

Considérese el sistema ponderado ωPD0L = ({A, B, C, D}, ABCD, P, ω, producto, ⊕), con las siguientes producciones y ponderaciones:

$p_1 : A \rightarrow AA$	$\omega(p_1) = 0.95$	$p_2 : B \rightarrow BB$	$\omega(p_2) = 0.93$
$p_3 : C \rightarrow CC$	$\omega(p_3) = 0.89$	$p_4 : D \rightarrow DD$	$\omega(p_4) = 0.91$

El lenguaje nítido _(PD0L) es $\{A^kB^kC^kD^k \mid k = 2^n, n \geq 0\}$. Si se asume que A, B, C y D son los lados de un cuadrado y de longitud 1, entonces _(PD0L) representaría "el conjunto de todos los cuadrados de lado 2^n con n entero y mayor que 0".

En el lenguaje borroso _(ωPD0L) resulta evidente que cada nueva palabra tendrá un grado de pertenencia al lenguaje menor que las anteriores; vale decir, a medida que n crece, $\mu_{\_(\omega PD0L)}(w)$ decrece. _(ωPD0L) podría representar, entonces, "el conjunto de todos los cuadrados *pequeños* de lado 2^n con n entero y mayor que 0". Ver figura 9.

Nótese que al *mismo* lenguaje formal borroso podría asociarse una interpretación muy diferente: A, B, C y D podrían representar (los sonidos de) cuatro campanas; 2^k, el número de tañidos de cada campana y $\mu_{\_(\omega PD0L)}(w) = (0.95)^k(0.93)^k(0.91)^k(0.89)^k$ una medida de la intensidad acústica decreciente del mensaje del carillón.

Lo anterior permite concluir que un lenguaje formal borroso puede servir de modelo abstracto para varios fenómenos (aparentemente muy) distintos del mundo real.

Ejemplo 7:

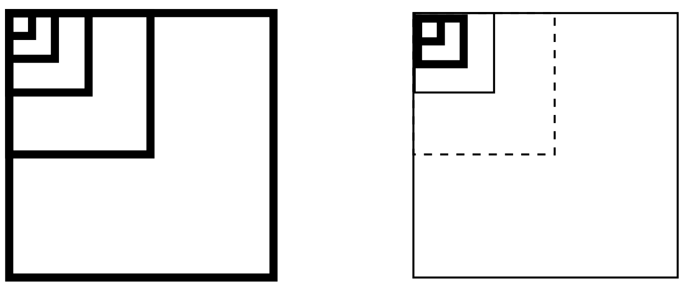


Fig. 9: Lado izquierdo: conjunto nítido de cuadrados con aristas de longitud 2^n
Lado derecho: conjunto borroso de cuadrados pequeños con aristas de longitud 2^n

Sea ω0L = ({A, B, C}, ABC, P, ω, mínimo, máximo), con las siguientes producciones y ponderaciones:

$p_1 : A \to AA$	$\omega(p_1) = 0.9$	$p_2 : A \to A$	$\omega(p_2) = 0.7$
$p_3 : B \to BB$	$\omega(p_3) = 0.9$	$p_4 : B \to B$	$\omega(p_4) = 0.7$
$p_5 : C \to CC$	$\omega(p_5) = 0.9$	$p_6 : C \to C$	$\omega(p_6) = 0.7$

El sistema ω0L es propagador, pero no-determinístico. _(ω0L) = {(w = $A^pB^qC^r$, μ_(w)) | p, q, r > 0}. Dado que por simplicidad en este ejemplo se usa la t-norma mínimo, el valor de μ_(w) dependerá de si se usó o no alguna de las producciones con subíndice par; en el primer caso la ponderación de la palabra derivada será 0.7, en el segundo, 0.9. Los lenguajes determinados por cortes alfa son los siguientes:

(£(ω0L); > 0) = {$A^pB^qC^r$ | p, q, r > 0 } que es P0L
(£(ω0L); > 0.7) = {$A^kB^kC^k$ | k = 2^r, r > 0 } que es DP0L
(£(ω0L); > 0.9) = {ABC} que es D0L

Cuando $\alpha = 0$ se utilizan todas las producciones y el lenguaje resultante es propagador, pero no determinístico. Si $\alpha = 0.7$, sólo son permitidas las producciones de subíndice impar, con lo cual el lenguaje resultante es propagador y determinístico. Al considerar $\alpha = 0.9$, ninguna producción es aplicable. El lenguaje resultante sólo consta de la palabra inicial (que por definición tiene grado de pertenencia 1 al lenguaje). El lenguaje {ABC}, sin embargo, es un lenguaje D0L, cuyo sistema generador es ({A, B, C}, ABC, { $A \to \lambda$, $B \to \lambda$, $C \to \lambda$}).

5. Conclusiones

Es posible construir lenguajes formales borrosos mediante la asignación de ponderaciones en el intervalo [0,1] a las transformaciones elementales contenidas en un sistema generador, necesarias para derivar palabras e introduciendo operaciones basadas en t-normas y t-conormas. Dos diferentes analogías para interpretar las ponderaciones de las transformaciones elementales, conducen a sendas familias de lenguajes borrosos, cuya relación aun es desconocida. Cortes alfa generalizados permiten obtener diferentes lenguajes formales nítidos (o borrosos) de cada lenguaje formal borroso. Esto, a su vez, permite visualizar (a posteriori) un lenguaje formal borroso como una agregación ponderada de diversos lenguajes formales nítidos sobre un alfabeto común.[3] Los lenguajes resultantes pueden ser, además, de diversos tipos dentro de sus respectivas jerarquías.

6. Discusión

Tal como se dijo al comienzo, los lenguajes formales no tienen asociada una semántica: sus palabras no tienen un significado: sólo tienen una estructura. Es muy legítimo, entonces, preguntarse cual es la razón para estudiar lenguajes formales y, mas aún, los lenguajes formales borrosos.

Una primera respuesta parcial se puede dar en el contexto de los lenguajes de programación para ordenadores. Por una parte, estos lenguajes representan una versión muy restringida de un lenguaje natural –(el inglés)– eliminando las ambigüedades y reduciendo tanto el vocabulario como las formas de expresión. Por otra parte, los lenguajes de programación de ordenadores son lenguajes formales de contexto libre y esto permite su eficiente manejo, especialmente su traducción a otro lenguaje formal que una máquina –un ordenador– puede aceptar y seguir. Lenguajes de programación que incluyan aspectos semánticos borrosos, por ejemplo, que permitan instrucciones del tipo: "incremente *un poco* el valor actual de la variable x", ofrecerían una mayor flexibilidad al programador (ver, por ejemplo, [SaG 04]). Para que esto sea posible, sin embargo, es indispensable que tal instrucción tenga una traducción unívoca en el lenguaje formal que el ordenador puede seguir. Esto a su vez requiere procesar la instrucción en el contexto de un lenguaje formal, el cual deberá ser borroso.

Recuérdese que un modelo es la representación de aspectos seleccionados, relevantes, de un sistema, con el propósito de entender mejor su influencia en el comportamiento del sistema, y de poder controlar mejor, a la vez que predecir, su funcionamiento. Desde este punto de vista, los lenguajes formales pueden considerarse como modelos de los lenguajes naturales, en los cuales el aspecto relevante considerado es la estructura, al precio de dejar de lado la semántica. Ahora bien los lenguajes formales borrosos agregan como nuevo aspecto relevante: el grado de pertenencia de cada palabra al lenguaje, ofreciendo así un modelo mas rico de los

[3] Recuérdese que si B es un conjunto borroso y B_α es un corte α, entonces

$$B = \bigcup_{\alpha \in [0,1]} \alpha B_\alpha$$

lenguajes naturales aun cuando se continúe haciendo abstracción de la semántica. Como se ilustró brevemente al término del ejemplo 6, es justamente la abstracción de la semántica la que permite que un lenguaje formal borroso pueda servir de modelo de varios sistemas aparentemente distintos, pero que poseen una misma estructura.

Referencias

[ATV 83] Alsina C., Trillas E. and Valverde L., On some logical connectives for fuzzy set theory. *Journal of Mathematical Analysis and its Applications,* **93** (1) 15-26, 1983

[Asv 03] Asveld P.R.J.: Algebraic aspects of families of fuzzy languages. *Theoretical Computer Science* **293** (2) 417-445, 2003

[Cho 59] Chomsky N.: On certain formal Properties of Grammars. *Information and Control* **2**, 137–167, 1959

[Cho 63] Chomsky N.: Formal Properties of Grammars. *Handbook of Mathematical Psychology*, Vol. 2 (Bush, Galanter, Luce, Eds.), John Wiley, 1963

[Cho 64] Chomsky N.: *Syntactic Structures*. Mouton & Co, The Hague, 1964

[Cho 65] Chomsky N.: *Aspects of the Theory of Syntax*. M.I.T. Press, Cambridge, Mass., 1965

[Hac 75] Hack M.: Petri Net languages. Computation Structures Group Memo 124. Project MAC, M.I.T., 1975

[Her 74] Herman G.T.: Closure properties of some Families of languages associated with Biological Systems. *Information and Control* **24**, 101–121, 1974

[KMP 00] Klement E., Mesiar R., Pap E.: *Triangular Norms*. Kluwer, Dordrecht, 2000

[KlY 95] Klir G., Yuan B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall, 1995

[LeZ 69] Lee E.T., Zadeh L.A. : Note on fuzzy languages. *Inform. Sci.* **1**, 421-434, 1969

[Lee 72] Lee E.T.: Fuzzy languages and their relation to automata. Dissertation, Dept. Electr. Eng. and Comp. Science, University of California, Berkeley, 1972

[Lee 95] Lee E.T.: Application of fuzzy languages to intelligent information retrieval. *Kybernetes* **24** (9), 49-55, 1995

[Lin 68] Lindenmayer A.: Mathematical Models for Cellular Interactions in Development. *Journal of Theoretical Biology*. **18**, 280–315, 1968

[Lin 71] Lindenmayer A.: Developmental Systems without Cellular Interactions, their languages and Grammars. *Journal of Theoretical Biology*. **30**, 455–484, 1971

[Men 42] Menger K.: Statistical Metrics. *Proceedings National Academy of Sciences USA* **28**, 535–537, 1942

[MSM 97]Meyer zu Bexten E., Sajadi F., Moraga C.: Properties of Lindenmayer Fuzzy Languages and α-driven Lindenmayer Languages. *Proc. 27th Intl. Symposium on Multiple-valued Logic* 195-200. IEEE-CS-Press, 1997

[Per 90] Person M.: Unscharfe Petrinetzsprachen. Diplomarbeit, Fachbereich Informatik, Universität Dortmund, 1990

[Pet 73] Peterson J.L.: Modeling of Parallel Systems. *Dissertation*, Dept. E.E., Stanford University, Stanford CA., 1973

[Pet 76] Peterson J.L.: Computation sequence sets. *Journal of Computer and Systems Science* **13** (1) 1-24, 1976

[Pet 77] Peterson J.L.: Petri Nets. *Computing Surveys* **9** (3) 223-252, 1977

[Pet 05] Petković T.: Fuzzy automata, fuzzy languages and semigroups, *Proceedings of Workshop on Automata and Semigroups* at ICALP05, Lisbon, Portugal, 78-85, 2005

[Pet 62] Petri C.A.: Kommunikation mit Automaten. *Dissertation*, Institut für Instrumentelle Mathematik, Universität Bonn, 1962

[Pfe 00] Pfenning F.: Formal Languages, Automata and Computation. Class Notes to Course 15-453 at CMU. http://www.cs.cmu.edu/~fp/courses/flac/ , 2000

[SaG 04] Sánchez Álvarez D., Gómez Skarmeta A.F.: A fuzzy language. *Fuzzy Sets and Systems* **141**, 335-390, 2004

[ScS 83] Schweizer B., Sklar A.: *Probabilistic Metric Spaces*. North–Holland, 1983

[YaR 96] Yager, R., Rybalov, A.: Uninorm aggregation operators. *Fuzzy Sets and Systems* **80** (1996) 111-120

[Zad 71] Zadeh L.A.: Fuzzy languages and their relation to human and machine intelligence. Memo 302, Electron. Research Laboratory, University of California, Berkeley, 1971. Also: *Proc. Conference on Man and Computer*, 130-165, Bordeaux, France. S. Karger, Basel 1972