http://dx.doi.org/10.12795/PH.1997.v11.i02.10

SISTEMAS DE GRAMÁTICAS Y PROCESAMIENTO MODULAR

Carlos Martín Vide

The new mathematical theory of grammar systems is an example of multi-agent system trying to formally model the cooperative interaction of several processors towards the common aim to generate/accept a language. Given the fact that natural languages seem to be modular structures, the theory of grammar systems appears as a good candidate to adequately give account of them. Grammar systems allow to generate a non-context-free language by means of only regular rules, which is a spectacular feature from the viewpoint of classical formal language theory. Moreover, grammar systems show advantages in terms of computational complexity. We introduce here in the theory.

1. Idea de un sistema de gramáticas

Sabemos que en la teoría clásica de lenguajes formales cada gramática y cada autómata operan individualmente, de manera que una gramática [respectivamente, un autómata] genera [respectivamente, reconoce] un lenguaje. Hoy, sin embargo, conceptos como distribución, cooperación, paralelismo, comunicación van ganando terreno progresivamente en diferentes ámbitos de las ciencias de la computación. Parece incluso bastante razonable que el cerebro humano funcione de una manera distribuida, sobre todo a la vista de la gran cantidad de capacidades y habilidades que parece poner en juego simultáneamente y de la plausibilidad de que tenga una estructura modular. Los sistemas de gramáticas constituyen un modelo matemático del procesamiento tanto distribuido como paralelo de la información.

Un sistema de gramáticas es una estructura compuesta por diversas gramáticas que funcionan coordinadas, conforme a un protocolo o algoritmo especificado, para producir un lenguaje. El citado protocolo de cooperación es el elemento crucial. Esta maquinaria generativa tiene especial flexibilidad para alcanzar objetivos tales como:

a) modelizar un fenómeno real complejo,

b) afinar la capacidad generativa,

c) disminuir la complejidad descriptiva.

Las clases de sistemas de gramáticas que vamos a considerar aquí son las siguientes:

1) sistemas con componentes que funcionan secuencialmente, estando activo uno en cada momento ("cooperating distributed grammar systems", o sistemas de

gramáticas distribuidas en cooperación, introducidos en Csuhaj-Varjú & Dassow 1989, a partir de las ideas de Meersman & Rozenberg 1978), y

2) sistemas con componentes que funcionan en paralelo, sincrónicamente ("parallel communicating grammar systems", o sistemas de gramáticas comunicadas en paralelo, introducidos en Păun & Sântean 1989).

Las referencias más accesibles y actuales para iniciarse en la materia son Csuhaj-Varjú, Dassow, Kelemen & Păun 1994, Păun 1995a y Păun 1995c).

En un sistema de gramáticas distribuidas en cooperación, diversas gramáticas trabajan juntas, sucesivamente, sobre un axioma común. En cada paso de la derivación está activa una gramática, y las demás permanecen inactivas. Las condiciones bajo las cuales una gramática puede activarse o desactivarse aparecen especificadas en el protocolo de cooperación. Cada gramática ha de usar un cierto número de reglas exactamente (=k), como mínimo ese número (≥k), como máximo ese número (≤k), un número cualquiera (*) o el máximo número posible de reglas (t). El lenguaje compuesto por las cadenas terminales así generadas es el lenguaje generado por el sistema.

En un sistema de gramáticas comunicadas en paralelo, cada gramática trabaja sobre su propio axioma al mismo tiempo que las restantes trabajan sobre sus axiomas respectivos, empleando cada una de ellas una regla en cada unidad de tiempo. Hasta aquí, se trata de gramáticas separadas que funcionan separadamente. Lo que da nacimiento a un sistema es la posibilidad de comunicación. Cuando un componente del sistema introduce un símbolo de llamada, se desencadena la comunicación. Una de las gramáticas se encarga de la dirección del proceso. El lenguaje generado por ésta es el lenguaje que el sistema produce.

Las convenciones notacionales básicas que utilizaremos son las usuales: V^* es el monoide libre generado por el alfabeto V con la operación de concatenación, λ es la cadena vacía, $V^+=V^*-\{\lambda\}$, $|x|$ es la longitud de $x\in V^*$ y $|x|_U$ es el número de ocurrencias de símbolos de $U\subseteq V^*$ en x. Una gramática de Chomsky es una estructura $G=(V_N,V_T,S,P)$, donde V_N es el alfabeto no-terminal, V_T el alfabeto terminal, S el axioma y P el conjunto de producciones. Una de las referencias clásicas en la teoría de lenguajes formales, a la que nos remitimos, es Salomaa (1973).

2. Sistemas de gramáticas distribuidas en cooperación

Definición 1. Un sistema de gramáticas distribuidas en cooperación (CD) es una estructura:

$$\Gamma=(V_N,V_T,S,P_1,...,P_n),$$

con $n\geq 1$, donde $V_N\cap V_T=\varnothing$, $S\in V_N$ y P_1, ..., P_n son conjuntos finitos de reglas de reescritura (usualmente independientes del contexto, aunque cabe considerar cualquier otro tipo de reglas pertenecientes a cualquier clase de mecanismo generativo) sobre $V_N\cup V_T$, a los que llamamos componentes del sistema.

Definición 2. En $(V_N\cup V_T)^*$ se define la *derivación inmediata* con respecto a P_i, con $1\leq i\leq n$, del modo usual, y se nota $\Rightarrow_i$. Las derivaciones que constan de exactamente k, a lo sumo k y al menos k pasos se notan, respectivamente, con $\Rightarrow_i^{=k}$, $\Rightarrow_i^{\leq k}$ y $\Rightarrow_i^{\geq k}$, con $k\geq 1$. Una

derivación arbitraria se expresa como $\Rightarrow_i^*$, y una derivación máxima o terminal como $\Rightarrow_i^t$. Esta última circunstancia se caracteriza formalmente así:

$$x \Rightarrow_i^t y \text{ ssi } (x \Rightarrow_i^{\geq 1} y \wedge \neg \exists z \in (V_N \cup V_T)^* \mid y \Rightarrow_i z).$$

Definición 3. El *lenguaje* generado por Γ en el modo $g \in \{\leq k, =k, \geq k \mid k \geq 1\} \cup \{*, t\}$ se define como:

$$L_g(\Gamma) = \{x \in V_T^* \mid S \Rightarrow_{i_1}^g x_1 \Rightarrow_{i_2}^g \ldots \Rightarrow_{i_m}^g x_m = x, \text{ con } m \geq 1,\ 1 \leq i_j \leq n,\ 1 \leq j \leq m\}.$$

Ejemplo 4. Considérese el siguiente sistema CD:

$$\Gamma = (\{S,A,A',B,B'\}, \{a,b,c\}, S, P_1, P_2),$$

con:

$$P_1 = \{S \rightarrow S,\ S \rightarrow AB,\ A' \rightarrow A,\ B' \rightarrow B\},$$

$$P_2 = \{A \rightarrow aA'b,\ B \rightarrow cB',\ A \rightarrow ab,\ B \rightarrow c\}.$$

Dependiendo del modo elegido de derivación, Γ genera, entre otros, los siguientes lenguajes:

$L_g(\Gamma) = \{a^n b^n c^n \mid n \geq 1\}$, con $g \in \{=2, \geq 2\}$,

$L_g(\Gamma) = \{a^n b^n c^m \mid n,m \geq 1\}$, con $g \in \{=1, \geq 1, *, t\} \cup \{\leq k \mid k \geq 1\}$,

$L_g(\Gamma) = \varnothing$, con $g \in \{=k, \geq k \mid k \geq 3\}$.

Veamos en detalle cómo funciona Γ en el modo =2. Hemos de empezar por S. Sólo se puede usar P_1. Aplicar dos veces la regla $S \rightarrow S$ no cambia nada, de modo que optamos por usar dos reglas diferentes:

$$S \Rightarrow_1 S \Rightarrow_1 AB.$$

De ahora en adelante ya no volverá a aparecer S. A AB únicamente se le puede aplicar P_2. Si usamos las reglas no-terminales, obtenemos:

$$AB \Rightarrow_2 aA'bB \Rightarrow_2 aA'bcB'.$$

Ahora sólo podemos emplear P_1:

$$aA'bcB' \Rightarrow_1 aAbcB' \Rightarrow_1 aAbcB.$$

A continuación, de nuevo P_2:

$$aAbcB \Rightarrow_2 aaA'bbcB \Rightarrow_2 aaA'bbccB'.$$

Volvemos a P_1, para obtener:

$$aaA'bbccB' \Rightarrow_1 aaAbbccB' \Rightarrow_1 aaAbbccB.$$

Si queremos terminar, hemos de utilizar otra vez P_2:

$$aaAbbccB \Rightarrow_2 aaabbbccc.$$

Así pues, hemos de usar sucesivamente P_1 y las reglas no-terminales de P_2, y acabamos la derivación con las reglas terminales de P_2. En conclusión, en efecto:

$$L_{=2}(\Gamma) = \{a^n b^n c^n \mid n \geq 1\}.$$

Ejemplo 5. El sistema de gramáticas CD:

$$\Gamma = (\{S,A\}, \{a\}, S, P_1, P_2, P_3),$$

con:

$$P_1 = \{S \rightarrow AA\},$$
$$P_2 = \{A \rightarrow S\},$$
$$P_3 = \{S \rightarrow a\},$$

genera el lenguaje:

$$L_t(\Gamma) = \{a^{2n} \mid n \geq 0\}.$$

Ejemplo 6. El sistema de gramáticas CD:

$$\Gamma = (\{S,A,A'\}, \{a,b\}, S, P_1, P_2, P_3),$$

con:

$$P_1 = \{S \rightarrow S, S \rightarrow AA, A' \rightarrow A\},$$
$$P_2 = \{A \rightarrow aA', A \rightarrow a\},$$
$$P_3 = \{A \rightarrow bA', A \rightarrow b\},$$

genera el lenguaje:

$$L_{=2}(\Gamma) = L_{\geq 2}(\Gamma) = \{xx \mid x \in \{a, b\}^+\}.$$

3. Sistemas de gramáticas comunicadas en paralelo

Definición 7. Un *sistema de gramáticas comunicadas en paralelo* (PC) es una estructura:

$$\Gamma = (V_N, K, V_T, (S_1,P_1),...,(S_n,P_n)),$$

donde V_N, K, V_T son alfabetos disjuntos, con $K = \{Q_1, ..., Q_n\}$, $S_i \in V_N$ y P_i son conjuntos finitos de reglas de reescritura (usualmente independientes del contexto, aunque cabe considerar cualquier otro tipo de reglas pertenecientes a cualquier clase de mecanismo generativo) sobre $V_\Gamma = V_N \cup K \cup V_T$, con $1 \leq i \leq n$. Los elementos de K reciben el nombre de símbolos de llamada, (S_i,P_i) son los componentes de Γ, con $1 \leq i \leq n$, y existe una correspondencia biyectiva entre los símbolos de llamada y los componentes, de tal manera que cada símbolo de llamada hace referencia al componente cuyo subíndice es el mismo. Una n-upla $(x_1,...,x_n)$, con $x_i \in V_\Gamma^*$ y $1 \leq i \leq n$ es una configuración de Γ.

Definición 8. Dado un sistema de gramáticas PC:

$$\Gamma = (V_N, K, V_T, (S_1,P_1),...,(S_n,P_n)),$$

y dadas dos configuraciones:

$$(x_1,...,x_n),$$
$$(y_1,...,y_n),$$

con $x_i, y_i \in V_\Gamma^*$, $1 \leq i \leq n$ y $x_1 \notin V_T^*$, definimos la relación de *derivación inmediata*:

$$(x_1,...,x_n) \Rightarrow (y_1,...,y_n)$$

si se da una de las siguientes situaciones:

(i) $|x_i|_K = 0$, con $1 \leq i \leq n \wedge ((x_i \Rightarrow_i y_i) \vee (x_i \in V_T^* \wedge x_i = y_i)$, con $1 \leq i \leq n)$, o bien

(ii) $\exists i$, $1 \leq i \leq n$: $|x_i|_K > 0$; si $x_i = z_1 Q_{i_1} z_2 ... z_m Q_{i_m} z_{m+1}$, $z_i \in V_\Gamma^*$, $1 \leq i \leq m+1$ y $|x_{i_j}|_K = 0$, para todo j, con $1 \leq j \leq m$, entonces $y_i = z_1 x_{i_1} z_2 ... z_m x_{i_m} z_{m+1}$ [e $y_{i_j} = S_{i_j}$, con $1 \leq j \leq m$]; si $|x_{i_j}|_K > 0$, para algún j, $1 \leq j \leq m$, entonces $y_i = x_i$; para todo i, $1 \leq i \leq n$, para el cual y_i no esté especificado arriba, tendremos que $y_i = x_i$.

En otras palabras, de $(x_1,...,x_n)$ se deriva $(y_1,...,y_n)$ cuando:

(i) [paso de reescritura] no aparece en $(x_1,...,x_n)$ ningún símbolo de llamada: tenemos entonces una simple reescritura de los componentes, $x_i \Rightarrow_i y_i$, con $1 \leq i \leq n$, mediante la utilización de una regla para cada componente P_i, con excepción de los casos en que x_i es terminal, en los que $y_i = x_i$, o bien

(ii) [paso de comunicación] aparece algún símbolo de llamada: toda ocurrencia de Q_j en x_i es sustituida por su forma sentencial x_j siempre que x_j no contenga a su vez símbolos de llamada.

Es importante tener presente que la comunicación tiene siempre prioridad sobre la reescritura.

El funcionamiento de un sistema de gramáticas PC se bloquea en dos casos:

(i) cuando un componente x_i de la n-upla actual $(x_1,...,x_n)$ no es terminal, pero no se le puede aplicar ninguna regla de P_i, o

(ii) cuando aparece una llamada circular: si P_{i_1} introduce Q_{i_2}, P_{i_2} introduce Q_{i_3},..., $P_{i_{k-1}}$ introduce Q_{i_k}, y P_{i_k} introduce Q_{i_1}, no es posible la reescritura, porque la comunicación, según hemos dicho, tiene prioridad.

Definición 9. El *lenguaje* generado por un sistema de gramáticas PC:

$$\Gamma = (V_N, K, V_T, (S_1,P_1),...,(S_n,P_n))$$

se define como:

$$L(\Gamma) = \{x \in V_T^* \mid (S_1,...,S_n) \Rightarrow^* (x,y_2,...,y_n), \text{ con } y_i \in V_\Gamma^*, 2 \leq i \leq n\}.$$

Partimos, pues, de la n-upla de axiomas y avanzamos mediante sucesivos pasos de reescritura y comunicación hasta que el componente P_1 produce una cadena terminal. Este componente es el director del sistema y su lenguaje es el de éste. Por tanto, mientras que en un sistema de gramáticas CD todos los componentes ocupan la misma posición, en un sistema de gramáticas PC, en cambio, tenemos una jerarquía de dos niveles: el director del sistema y los que podemos llamar procesadores.

Hasta aquí hemos considerado sistemas de gramáticas PC en los que no se ha impuesto restricción alguna sobre el uso de los símbolos de llamada, es decir, cualquier componente P_i puede introducir cualquier símbolo Q_j.

Definición 10. Sea un sistema de gramáticas PC:

$$\Gamma = (V_N, K, V_T, (P_1, S_1), \ldots, (P_n, S_n)).$$

Si únicamente P_1 puede introducir símbolos de llamada, esto es, si:

$$P_i \subseteq (V_N \cup V_T)^* \times (V_N \cup V_T)^*,$$

para $2 \leq i \leq n$, entonces decimos que Γ es un *sistema centralizado*. En caso contrario, es un sistema no-centralizado.

Definición 11. Un sistema de gramáticas PC es un *sistema retornante* si, después de la comunicación, todo componente una cadena del cual se ha enviado a otro componente vuelve a su axioma. Si eliminamos la expresión de la definición 8 que hemos colocado entre corchetes, [e $y_{ij} = S_{ij}$, con $1 \leq j \leq m$], obtenemos un sistema no-retornante: después de la comunicación, el componente P_{ij} no vuelve a S_{ij}, sino que sólo ha enviado una copia de la cadena y continúa procesando la original en el estado en que se encuentra. Así pues, para un sistema PC Γ, tendremos un lenguaje en el modo retornante $L_r(\Gamma)$ (obtenido mediante un modo de derivación que notamos como $\Rightarrow_r$) y un lenguaje en el modo no-retornante $L_{nr}(\Gamma)$ (obtenido mediante un modo de derivación que notamos como $\Rightarrow_{nr}$).

Para ilustrar las definiciones, vamos a escoger a continuación un par de sistemas que ofrecen interpretaciones lingüísticamente relevantes: son conocidos lenguajes no-independientes del contexto que corresponden a ciertos fenómenos que encontramos en el lenguaje natural.

Ejemplo 12. Sea el siguiente sistema de gramáticas PC:

$$\Gamma = (\{S_1, S_2, S_3\}, K, \{a,b,c\}, (S_1, P_1), (S_2, P_2), (S_3, P_3)),$$

con:

$$P_1 = \{S_1 \rightarrow aS_1,\ S_1 \rightarrow a^3Q_2,\ S_2 \rightarrow b^2Q_3,\ S_3 \rightarrow c\},$$
$$P_2 = \{S_2 \rightarrow bS_2\},$$
$$P_3 = \{S_3 \rightarrow cS_3\}.$$

Obtenemos el siguiente lenguaje:

$$L_r(\Gamma) = L_{nr}(\Gamma) = \{a^n b^n c^n \mid n \geq 1\}.$$

Esta expresión formaliza el fenómeno lingüístico conocido como concordancias múltiples.

Examinemos brevemente cómo funciona Γ. Empezamos con (S_1, S_2, S_3). Utilizando la primera regla de P_1 y las únicas que forman parte de P_2 y P_3 para $n \geq 0$ pasos, tenemos que:

$$(S_1, S_2, S_3) \Rightarrow_r (a^n S_1, b^n S_2, c^n S_3).$$

Tomamos ahora la segunda regla de P_1:

$$(a^n S_1, b^n S_2, c^n S_3) \Rightarrow_r (a^{n+3} Q_2, b^{n+1} S_2, c^{n+1} S_3).$$

Como ha aparecido el símbolo de llamada Q_2, hemos de ejecutar un paso de comunicación. Así pues, enviamos $b^{n+1}S_2$ al primer componente para sustituir a Q_2:

$$(a^{n+3} Q_2, b^{n+1} S_2, c^{n+1} S_3) \Rightarrow_r (a^{n+3} b^{n+1} S_2, S_2, c^{n+1} S_3).$$

Continuamos de la siguiente manera:

$$(a^{n+3}b^{n+1}S_2,S_2,c^{n+1}S_3) \Rightarrow_r (a^{n+3}b^{n+3}Q_3,bS_2,c^{n+2}S_3) \Rightarrow_r (a^{n+3}b^{n+3}c^{n+2}S_3,bS_2,S_3) \Rightarrow_r (a^{n+3}b^{n+3}c^{n+3},b^2S_2,cS_3).$$

Como hay una única llamada de P_1 a P_2 y a P_3, en Γ los modos retornante y no-retornante coinciden.

Ejemplo 13. Sea el siguiente sistema de gramáticas PC:

$$\Gamma = (\{S_1,S_2\}, K, \{a,b\}, (S_1,P_1), (S_2,P_2)),$$

con:

$$P_1 = \{S_1 \rightarrow S_1, S_1 \rightarrow Q_2Q_2\},$$

$$P_2 = \{S_2 \rightarrow aS_2, S_2 \rightarrow bS_2, S_2 \rightarrow a, S_2 \rightarrow b\}.$$

Obtenemos el lenguaje:

$$L_r(\Gamma) = L_{nr}(\Gamma) = \{xx \mid x \in \{a, b\}^+\}.$$

Esta expresión formaliza el fenómeno lingüístico conocido como reduplicación.

El funcionamiento de Γ es sencillo. P_1 permanece inactivo y entretanto P_2 genera una cadena. Luego P_1 introduce Q_2Q_2, con lo que la cadena generada por P_2 es enviada a P_1 y duplicada (ha de ser terminal; si no, el sistema queda bloqueado).

Obsérvese que los sistemas de los ejemplos 12 y 13 son centralizados.

Veamos ahora un caso en el que los lenguajes retornante y no-retornante no coinciden.

Ejemplo 14. Sea el siguiente sistema de gramáticas PC:

$$\Gamma = (\{S_1,S_2\}, K, \{a\}, (S_1,P_1), (S_2,P_2)\},$$

con:

$$P_1 = \{S_1 \rightarrow aQ_2, S_2 \rightarrow aQ_2, S_2 \rightarrow a\},$$

$$P_2 = \{S_2 \rightarrow aS_2\}.$$

Obtenemos los siguientes lenguajes:

$$L_r(\Gamma) = \{a^{2n+1} \mid n \geq 1\},$$

$$L_{nr}(\Gamma) = \{a^{(n+1)(n+2)/2} \mid n \geq 1\}.$$

4. Capacidad generativa y complejidad descriptiva

Las reglas que hemos empleado en el ejemplo 12 son todas ellas regulares, y las del ejemplo 13 son algunas de ellas regulares y otras independientes del contexto. En ambos casos, utilizadas aisladamente, sin que formen parte de un sistema de gramáticas, no pueden generar -como es bien sabido- las estructuras lingüísticas mencionadas, que son no-independientes del contexto. En cambio, cuando tales reglas se integran en un sistema y funcionan coordinadamente de acuerdo con el protocolo de cooperación, pueden generar con una gran simplicidad las estructuras deseadas. Vemos, así, que conjuntos de reglas simples y de poca capacidad generativa, si son organizados de manera adecuada en forma de sistema, alcanzan una potencia inusual en la teoría clásica de lenguajes formales.

Disponemos, además, de algunos resultados significativos sobre la complejidad descriptiva de los dos tipos de sistemas de gramáticas, especialmente en comparación con las gramáticas independientes del contexto. Así, por ejemplo, para las medidas de complejidad Var (número de elementos no-terminales), Prod (número de producciones) y Symb (número de símbolos que se utilizan en las producciones), y para una buena cantidad de sistemas CD y PC, comprobamos que hay a menudo lenguajes independientes del contexto que se pueden generar de una manera mucho más económica con sistemas que con gramáticas aisladas. En muchos casos obtenemos resultados del estilo siguiente: existe una secuencia L_n, con $n \geq 1$, de lenguajes independientes del contexto tal que $M_{CF}(L_n) \geq n$, para $n \geq 1$, pero $M_{PC}(L_n) \leq p$, donde p es una constante y M una de las medidas de complejidad descriptiva.

Los sistemas de gramáticas están empezando a ser aplicados a la descripción de las lenguas naturales. Buenas muestras de ello son Csuhaj-Varjú 1994 y Jiménez López 1996.

REFERENCIAS

E. Csuhaj-Varjú (1994), "Grammar systems: a multi-agent framework for natural language generation", en Gh. Păun, ed., *Mathematical aspects of natural and formal languages*: 63-78. World Scientific, Singapore.

E. Csuhaj-Varjú & J. Dassow (1989), "On cooperating distributed grammar systems", *Journal of Information Processing and Cybernetics (EIK)*, *26*, 1-2: 49-63.

E. Csuhaj-Varjú, J. Dassow, J. Kelemen & Gh. Păun (1994), *Grammar systems: a grammatical approach to distribution and cooperation*. Gordon and Breach, London.

M. D. Jiménez López (1996), "Sistemas de gramáticas y lenguajes naturales: ideas intuitivas al respecto", en C. Martín Vide, ed., *Lenguajes naturales y lenguajes formales*, *XII*: 223-236. PPU, Barcelona.

R. Meersman & G. Rozenberg (1978), "Cooperating grammar systems", en *Lecture Notes in Computer Science*, vol. 64: 364-373. Springer, Berlin.

Gh. Păun (1995a), "Generating languages in a distributed way: grammar systems", en C. Martín Vide, ed., *Lenguajes naturales y lenguajes formales*, *XI*: 45-67. PPU, Barcelona.

Gh. Păun (1995b), "Grammar systems: a grammatical approach to distribution and cooperation", en Z. Fulop & F. Gecseg, eds., *Lecture Notes in Computer Science*, vol. 944: 429-443. Springer, Berlin.

Gh. Păun (1995c), "Parallel communicating grammar systems. A survey", en C. Martín Vide, ed., *Lenguajes naturales y lenguajes formales*, *XI*: 257-283. PPU, Barcelona.

Gh. Păun & L. Sântean (1989), "Parallel communicating grammar systems: the regular case", *Annals of the University of Bucharest: Series Mathematics-Informatics*, *38*: 55-63.

A. Salomaa (1973), *Formal languages*. Academic Press, New York.