Nuevos algoritmos tabulares para el análisis de LIG*

Miguel A. Alonso, Jorge Graña y **Manuel Vilares**
Departamento de Computación, Universidad de La Coruña
Campus de Elviña s/n, 15071 La Coruña, España
{alonso,grana,vilares}@dc.fi.udc.es
http://coleweb.dc.fi.udc.es/

Resumen

A partir de un algoritmo de tipo CYK se desarrolla una serie de nuevos algoritmos tabulares para el análisis de Gramáticas Lineales de Índices que incluye algoritmos ascendentes y algoritmos de tipo Earley con y sin la propiedad del prefijo válido, creando un camino evolutivo continuo en el que cada algoritmo puede ser obtenido mediante transformaciones simples del algoritmo precedente. Los nuevos algoritmos creados permiten establecer un paralelismo con los algoritmos disponibles para Gramáticas de Adjunción de Árboles.

1 Introducción

Las Gramáticas de Índices [1] son una extensión de las gramáticas independientes del contexto en las cuales cada símbolo no-terminal tiene asociado una pila de índices. Si restringimos la forma de las producciones de tal modo que la pila asociada al no-terminal del lado izquierdo de una producción (denominado *padre*) sólo pueda transmitirse a un no-terminal del lado derecho (denominado *hijo dependiente*) y los demás no-terminales estén asociados con pilas de tamaño acotado, obtenemos las Gramáticas Lineales de Índices (LIG) [6]. Además del interés que este tipo de gramáticas presentan por sí mismos como formalismo descriptivo, en lingüística computacional presentan una relevancia especial puesto que son habitualmente utilizadas como formalismo intermedio a través del cual se realiza el análisis sintáctico de las Gramáticas de Adjunción de Árboles (TAG) [12, 9, 10, 13].

Formalmente, una LIG es un quíntupla (V_T, V_N, V_I, S, P), donde V_T es un conjunto finito de terminales, V_N es un conjunto finito de no-terminales, V_I es un conjunto finito de índices (elementos que se almacenan en las pilas asociadas a los no-terminales), $S \in V_N$ es el axioma de la gramática y P es un conjunto finito de producciones que tienen alguna de las formas siguientes:

$$A_0[\circ\circ\gamma] \to A_1[\,]\ldots A_{i-1}[\,]\, A_i[\circ\circ]\, A_{i-1}[\,]\, \ldots A_m[\,]$$
$$A_0[\circ\circ] \to A_1[\,]\ldots A_{i-1}[\,]\, A_i[\circ\circ]\, A_{i-1}[\,]\, \ldots A_m[\,]$$
$$A_0[\circ\circ] \to A_1[\,]\ldots A_{i-1}[\,]\, A_i[\circ\circ\gamma]\, A_{i-1}[\,]\, \ldots A_m[\,]$$
$$A_0[\,] \to a$$

donde $A_j \in V_N$ para todo $0 \leq j \leq m$, A_i es el hijo dependiente, $\gamma \in V_I$, $\circ\circ$ representa la parte de la pila transmitida del padre al hijo dependiente, m es la longitud de la producción y $a \in V_T \cup \{\epsilon\}$.

La relación de derivación $\Rightarrow$ se define en LIG como $\Upsilon \Rightarrow \Upsilon'$

- si $\Upsilon = \Upsilon_1 A[\alpha\gamma]\ \Upsilon_4$ y existe una producción $A[\circ\circ\gamma] \to \Upsilon_2 A'[\circ\circ\gamma']\ \Upsilon_3$ de tal modo que $\Upsilon' = \Upsilon_1\Upsilon_2 A'[\alpha\gamma']\ \Upsilon_3\Upsilon_4$
- o bien si $\Upsilon = \Upsilon_1 A[\,]\ \Upsilon_4$ y existe una producción $A[\,] \to a$ de modo que $\Upsilon' = \Upsilon_1\ a\ \Upsilon_4$

donde $A \in V_N$, $\alpha \in V_I^*$ y $\gamma, \gamma' \in V_I \cup \{\epsilon\}$. Denotamos por $\stackrel{*}{\Rightarrow}$ el cierre reflexivo y transitivo de $\Rightarrow$. El lenguaje generado por una LIG queda definido por todos los $w \in V_T^*$ tales que $S[\,] \stackrel{*}{\Rightarrow} w$.

*Financiado en parte por FEDER (1FD97-0047-C04-02) y por la Xunta de Galicia (XUGA20402B97).

En cuanto al análisis sintáctico de este tipo de gramáticas, la *propiedad de independencia del contexto de LIG* desempeña un papel fundamental, como se verá en las siguientes secciones. Dicha propiedad establece que si $A[\gamma] \stackrel{*}{\Rightarrow} uB[\ ]w$, donde $u,v,w \in V_T^*$, $A,B \in V_N$, $B[\ ]$ es el descendiente dependiente de $A[\gamma]$ y $\gamma \in V_I \cup \{\epsilon\}$, entonces para cualquier $\beta \in V_I^*$ se cumple que $A[\beta\gamma] \stackrel{*}{\Rightarrow} uB[\beta]w$ y para cualquier $\Upsilon_1, \Upsilon_2 \in (V_N[V_I^*] \cup V_T)^*$ se cumple que $\Upsilon_1 A[\beta\gamma]\Upsilon_2 \stackrel{*}{\Rightarrow} \Upsilon_1 uB[\beta]w\Upsilon_2$. Si además $B[\beta] \stackrel{*}{\Rightarrow} v$, donde $v \in V_T^*$, entonces $\Upsilon_1 A[\beta\gamma]\Upsilon_2 \stackrel{*}{\Rightarrow} \Upsilon_1 uB[\beta]w\Upsilon_2 \stackrel{*}{\Rightarrow} \Upsilon_1 uvw\Upsilon_2$.

1.1 Esquemas de análisis sintáctico

Describiremos los algoritmos de análisis mediante esquemas de análisis sintáctico [11], una estructura para realizar descripciones de alto nivel de algoritmos de análisis sintáctico que permite estudiar fácilmente las relaciones entre diferentes algoritmos mediante el análisis de las relaciones formales entre los esquemas de análisis subyacentes.

Un *sistema de análisis sintáctico* para una gramática $\mathcal{G}$ y una cadena de entrada $a_1 \dots a_n$ es un triple $\mathbb{P} = \langle \mathcal{I}, \mathcal{H}, \mathcal{D} \rangle$, donde $\mathcal{I}$ es un conjunto de *ítems* que representan resultados intermedios del proceso de análisis, $\mathcal{H}$ es un conjunto inicial de ítems denominado *hipótesis* que representan la cadena que va a ser analizada y $\mathcal{D}$ es un conjunto de reglas de inferencia de la forma $\frac{\eta_1,\dots,\eta_k}{\xi}\ cond$, denominadas *pasos deductivos*, mediante las cuales se derivan nuevos ítems ξ a partir de los ítems η_i existentes si las condiciones *cond* son satisfechas. La presencia de un conjunto $\mathcal{F} \subseteq \mathcal{I}$ de *ítems finales* indica el reconocimiento de la cadena de entrada.

Un *esquema de análisis sintáctico* es un sistema de análisis $\mathbb{P}$ parametrizado para cualquier gramática y cadena de entrada. Un esquema de análisis puede ser generalizado a partir de otro mediante *refinamiento de los ítems* (rompiendo un ítem individual en varios), *refinamiento de los pasos deductivos* (descomponiendo un paso deductivo simple en una secuencia de pasos) y *extensión* (considerando una clase más amplia de gramáticas).

Las operaciones de *filtrado* permiten disminuir el número de ítems y pasos deductivos de un esquema de análisis mediante la eliminación de pasos redundantes (*filtrado estático*) y la utilización de información contextual para determinar la validez de los ítems (*filtrado dinámico*). La operación de *contracción de pasos* permite que una secuencia de pasos deductivos sea reemplazada por un solo paso.

2 Algoritmo de tipo CYK

Como punto de partida tomaremos una extensión para LIG del algoritmo CYK. Nos basaremos en una versión modificada del algoritmo descrito por Vijay-Shanker y Weir en [12]. Asumiremos que cada producción posee dos elementos en el lado derecho o bien un único elemento que debe ser un terminal. Este condicionante puede verse como una trasposición de la forma normal de Chomsky al caso de las gramáticas lineales de índices.

El algoritmo trabaja reconociendo de forma ascendente la parte de la cadena de entrada cubierta por cada posible elemento gramatical. Para ello utiliza un conjunto de ítems

$$[A, \gamma, i, j \mid B, p, q]$$

que representan uno de los siguientes tipos de derivaciones:

- $A[\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \dots a_p\ B[\ ]\ a_{q+1} \dots a_j$ si y sólo si $(B,p,q) \neq (-,-,-)$ y donde $B[\ ]$ es un descendiente dependiente de $A[\gamma]$.
- $A[\ ] \stackrel{*}{\Rightarrow} a_{i+1} \dots a_j$ si y sólo si $\gamma = -$ y $(B,p,q) = (-,-,-)$.

donde $-$ se utiliza para denotar que el valor de un componente está indefinido.

Estos ítems se corresponden con los propuestos para la tabulación de los autómatas lineales de índices [8] y los autómatas ascendentes con dos pilas [5] y son ligeramente diferentes de los propuestos por Vijay-Shanker y Weir en [12] para su algoritmo de tipo CYK, que presentaban la forma $[A, \gamma, i, j \mid B, \eta, p, q]$, con $\eta \in V_I$. El elemento η es redundante puesto que por la propiedad de independencia del contexto de las gramáticas lineales de índices tenemos que si $A[\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \dots a_p\ B[\ ]\ a_{q+1} \dots a_j$ entonces para cualquier β se cumple que $A[\beta\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \dots a_p\ B[\beta]\ a_{q+1} \dots a_j$.

Esquema 1 *El sistema de análisis $\mathbb{P}_{\mathrm{CYK}}$ que se corresponde con el algoritmo de análisis de tipo CYK para una gramática lineal de índices $\mathcal{G}$ y una cadena de entrada $a_1 \dots a_n$ se define como sigue:*

$$\mathcal{I}_{\mathrm{CYK}} = \left\{ \begin{array}{l} [A,\gamma,i,j \mid B,p,q] \mid A,B \in V_N,\ \gamma \in V_I, \\ 0 \le i \le j\ ,\ \ (p,q) \le (k,j) \end{array} \right\}$$

$$\mathcal{H}_{\mathrm{CYK}} = \{\ [a,i-1,i] \mid a = a_i,\ 1 \le i \le n\ \}$$

$$\mathcal{D}^{\mathrm{Scan}}_{\mathrm{CYK}} = \frac{[a,j,j+1]}{[A,-,j,j+1 \mid -,-,-]}\ \ A[\,] \to a \in P$$

$$\mathcal{D}^{[\circ\circ\gamma][\,][\circ\circ]}_{\mathrm{CYK}} = \frac{\begin{array}{c}[B,-,i,k \mid -,-,-],\\ [C,\eta,k,j \mid D,p,q]\end{array}}{[A,\gamma,i,j \mid C,k,j]}$$

tal que $A[\circ\circ\gamma] \to B[\,]\ C[\circ\circ] \in P$.

$$\mathcal{D}^{[\circ\circ\gamma][\circ\circ][\,]}_{\mathrm{CYK}} = \frac{\begin{array}{c}[B,\eta,i,k \mid D,p,q],\\ [C,-,k,j \mid -,-,-]\end{array}}{[A,\gamma,i,j \mid B,i,k]}$$

tal que $A[\circ\circ\gamma] \to B[\circ\circ]\ C[\,] \in P$.

$$\mathcal{D}^{[\circ\circ][\,][\circ\circ]}_{\mathrm{CYK}} = \frac{\begin{array}{c}[B,-,i,k \mid -,-,-],\\ [C,\eta,k,j \mid D,p,q]\end{array}}{[A,\eta,i,j \mid D,p,q]}$$

tal que $A[\circ\circ] \to B[\,]\ C[\circ\circ] \in P$.

$$\mathcal{D}^{[\circ\circ][\circ\circ][\,]}_{\mathrm{CYK}} = \frac{\begin{array}{c}[B,\eta,i,k \mid D,p,q],\\ [C,-,k,j \mid -,-,-]\end{array}}{[A,\eta,i,j \mid D,p,q]}$$

tal que $A[\circ\circ] \to B[\circ\circ]\ C[\,] \in P$.

$$\mathcal{D}^{[\circ\circ][\,][\circ\circ\gamma]}_{\mathrm{CYK}} = \frac{\begin{array}{c}[B,-,i,k \mid -,-,-],\\ [C,\gamma,k,j \mid D,p,q],\\ [D,\eta,p,q \mid E,r,s]\end{array}}{[A,\eta,i,j \mid E,r,s]}$$

tal que $A[\circ\circ] \to B[\,]\ C[\circ\circ\gamma] \in P$.

$$\mathcal{D}^{[\circ\circ][\circ\circ\gamma][\,]}_{\mathrm{CYK}} = \frac{\begin{array}{c}[B,\gamma,i,k \mid D,p,q],\\ [C,-,k,j \mid -,-,-],\\ [D,\eta,p,q \mid E,r,s]\end{array}}{[A,\eta,i,j \mid E,r,s]}$$

tal que $A[\circ\circ] \to B[\circ\circ\gamma]\ C[\,] \in P$.

$$\begin{array}{rl}\mathcal{D}_{\mathrm{CYK}} = & \mathcal{D}^{\mathrm{Scan}}_{\mathrm{CYK}} \cup \mathcal{D}^{[\circ\circ\gamma][\,][\circ\circ]}_{\mathrm{CYK}} \cup \mathcal{D}^{[\circ\circ\gamma][\circ\circ][\,]}_{\mathrm{CYK}} \cup \\ & \mathcal{D}^{[\circ\circ][\,][\circ\circ]}_{\mathrm{CYK}} \cup \mathcal{D}^{[\circ\circ][\circ\circ][\,]}_{\mathrm{CYK}} \cup \mathcal{D}^{[\circ\circ][\,][\circ\circ\gamma]}_{\mathrm{CYK}} \cup \\ & \mathcal{D}^{[\circ\circ][\circ\circ\gamma][\,]}_{\mathrm{CYK}}\end{array}$$

$$\mathcal{F}_{\mathrm{CYK}} = \{\ [S,-,0,n \mid -,-,-]\ \}$$

La definición de las hipótesis realizada en este sistema de análisis sintáctico se corresponde con la estándar y es la misma que se utilizará en los restantes sistemas de análisis del artículo. Por consiguiente, no nos volveremos a referir explícitamente a ellas.

Los pasos $\mathcal{D}^{\mathrm{Scan}}_{\mathrm{CYK}}$ son los encargados de iniciar el procesamiento ascendente de la cadena de entrada. Los demás pasos se encargan de combinar los ítems correspondientes a los elementos del lado derecho de las producciones involucradas para generar el ítem correspondiente al lado izquierdo de la producción, propagando la información necesaria acerca de la pila de índices.

La complejidad espacial del algoritmo con respecto a la cadena de entrada es $\mathcal{O}(n^4)$ puesto que cada ítem almacena cuatro posiciones de dicha cadena. La complejidad temporal con respecto a la cadena de entrada es $\mathcal{O}(n^6)$ y viene dada por los pasos deductivos $\mathcal{D}^{[\circ\circ][\,][\circ\circ\gamma]}_{\mathrm{CYK}}$ y $\mathcal{D}^{[\circ\circ][\circ\circ\gamma][\,]}_{\mathrm{CYK}}$. Aunque dichos pasos manipulan en principio 7 posiciones de la cadena de entrada, mediante aplicación parcial cada uno de ellos se puede descomponer en una sucesión de pasos que manipulan a lo sumo 6 posiciones de la cadena de entrada.

Vijay-Shanker y Weir describen en [13] un algoritmo de tipo CYK generalizado para gramáticas lineales de índices que manipulan más de un índice de la pila de índices en cada producción. La misma generalización puede ser aplicada al esquema de análisis sintáctico propuesto.

3 Algoritmo de tipo Earley ascendente

El algoritmo de tipo CYK presenta una limitación muy importante: sólo es aplicable a gramáticas lineales de índices cuyas producciones tienen a lo sumo dos elementos en el lado derecho. Como primer paso para eliminar esta limitación procederemos a introducir un punto en las producciones, que nos permitirá distinguir la parte de la producción ya reconocida de aquella que resta por reconocer. Con respecto a la notación, utilizaremos $\boldsymbol{A}$ para referirnos al elemento LIG de una producción constituido por el no-terminal A y una pila de índices asociada, cuando la forma de dicha pila sea irrelevante en el contexto de utilización. Los ítems

$$[\boldsymbol{A} \to \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q]$$

del nuevo esquema de análisis sintáctico, que denominaremos **buE**, son un refinamiento de los ítems de **CYK** y representan alguno de los siguientes tipos de derivaciones:

- $A[\gamma] \Rightarrow \Upsilon_1\Upsilon_2 \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p \; B[\,] \; a_{q+1} \ldots a_j \; \Upsilon_2$ si y sólo si $(B, p, q) \neq (-,-,-)$, donde $B[\,]$ es un descendiente dependiente de $A[\gamma]$.
- $\Upsilon_1 \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_j$ si y sólo si $\gamma = -$ y $(B, p, q) = (-,-,-)$. Si Υ_1 incluye al hijo dependiente entones su pila asociada está vacía.

Los pasos deductivos sufrirán también un refinamiento puesto que los pasos de tipo $\mathcal{D}_{\text{CYK}}^{[\circ\circ\gamma][\,][\circ\circ]}$, $\mathcal{D}_{\text{CYK}}^{[\circ\circ\gamma][\circ\circ][\,]}$, $\mathcal{D}_{\text{CYK}}^{[\circ\circ][\,][\circ\circ]}$, $\mathcal{D}_{\text{CYK}}^{[\circ\circ][\circ\circ][\,]}$, $\mathcal{D}_{\text{CYK}}^{[\circ\circ][\,][\circ\circ\gamma]}$ y $\mathcal{D}_{\text{CYK}}^{[\circ\circ][\circ\circ\gamma][\,]}$ serán separados en diferentes pasos de tipo Init y Comp. Finalmente, la extensión del dominio permitirá tratar gramáticas lineales de índices con producciones de longitud arbitraria.

Esquema 2 *El sistema de análisis $\mathbb{P}_{\text{buE}}$ que se corresponde con el algoritmo de análisis de tipo Earley ascendente para una gramática lineal de índices $\mathcal{G}$ y una cadena de entrada $a_1 \ldots a_n$ se define como sigue:*

$$\mathcal{I}_{\text{buE}} = \left\{ \begin{array}{l} [\boldsymbol{A} \to \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q] \mid \\ \boldsymbol{A} \to \Upsilon_1\Upsilon_2 \in P, \;\; B \in V_N, \;\; \gamma \in V_I, \\ 0 \leq i \leq j \;, \;\;\; (p,q) \leq (i,j) \end{array} \right\}$$

$$\mathcal{D}_{\text{buE}}^{\text{Init}} = \frac{}{[\boldsymbol{A} \to \bullet\Upsilon, -, i, i \mid -,-,-]}$$

$$\mathcal{D}_{\text{buE}}^{\text{Scan}} = \frac{\begin{array}{c}[A[\,] \to \bullet a, -, j, j \mid -,-,-], \\ [a, j, j+1]\end{array}}{[A[\,] \to a \bullet, -, j, j+1 \mid -,-,-]}$$

$$\mathcal{D}_{\text{buE}}^{\text{Comp}^1} = \frac{\begin{array}{c}[\boldsymbol{A} \to \Upsilon_1 \bullet B[\,] \, \Upsilon_2, \gamma, i, k \mid C, p, q], \\ [\boldsymbol{B} \to \Upsilon_3 \bullet, -, k, j \mid -,-,-]\end{array}}{[\boldsymbol{A} \to \Upsilon_1 \; B[\,] \bullet \Upsilon_2, \gamma, i, j \mid C, p, q]}$$

$$\mathcal{D}_{\text{buE}}^{\text{Comp}^2} = \frac{\begin{array}{c}[A[\circ\circ\gamma] \to \Upsilon_1 \bullet B[\circ\circ] \, \Upsilon_2, -, i, k \mid -,-,-], \\ [\boldsymbol{B} \to \Upsilon_3 \bullet, \eta, k, j \mid C, p, q]\end{array}}{[A[\circ\circ\gamma] \to \Upsilon_1 \; B[\circ\circ] \bullet \Upsilon_2, \gamma, i, j \mid B, k, j]}$$

$$\mathcal{D}_{\text{buE}}^{\text{Comp}^3} = \frac{\begin{array}{c}[A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ] \, \Upsilon_2, -, i, k \mid -,-,-], \\ [\boldsymbol{B} \to \Upsilon_3 \bullet, \eta, k, j \mid C, p, q]\end{array}}{[A[\circ\circ] \to \Upsilon_1 \; B[\circ\circ] \bullet \Upsilon_2, \eta, i, j \mid C, p, q]}$$

$$\mathcal{D}_{\text{buE}}^{\text{Comp}^4} = \frac{\begin{array}{c}[A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ\gamma] \, \Upsilon_2, -, i, k \mid -,-,-], \\ [\boldsymbol{B} \to \Upsilon_3 \bullet \,, \gamma, k, j \mid C, p, q], \\ [\boldsymbol{C} \to \Upsilon_4 \bullet, \eta, p, q \mid D, r, s]\end{array}}{[A[\circ\circ] \to \Upsilon_1 \; B[\circ\circ\gamma] \bullet \Upsilon_2, \eta, i, j \mid D, r, s]}$$

$$\begin{array}{rl}\mathcal{D}_{\text{buE}} = & \mathcal{D}_{\text{buE}}^{\text{Init}} \cup \mathcal{D}_{\text{buE}}^{\text{Scan}} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}^1} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}^2} \cup \\ & \mathcal{D}_{\text{buE}}^{\text{Comp}^3} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}^4}\end{array}$$

$$\mathcal{F}_{\text{buE}} = \{ \;\; [\boldsymbol{S} \to \Upsilon \bullet \,, -, 0, n \mid -,-,-] \;\; \}$$

La complejidad espacial con respecto a la cadena de entrada es $\mathcal{O}(n^4)$ puesto que cada ítem almacena 4 posiciones de dicha cadena. La complejidad temporal con respecto a la cadena de entrada es $\mathcal{O}(n^6)$ y viene dada por el conjunto de pasos $\mathcal{D}_{\text{buE}}^{\text{Comp}^4}$.

4 Algoritmo de tipo Earley

El algoritmo descrito por el esquema de análisis sintáctico **buE** no toma en consideración si la parte de la cadena de entrada que se reconoce en cada ítem es derivable del axioma de la gramática. Los algoritmos de tipo Earley limitan el número de pasos deductivos aplicables mediante la predicción de las producciones que son candidatas a formar parte de la derivación atendiendo a su derivabilidad a partir del axioma.

En primer lugar, consideraremos que la predicción se realiza únicamente atendiendo al esqueleto independiente del contexto de la gramática lineal de índices, obteniendo un esquema de análisis que denominaremos **E** y que se deriva del esquema **buE** mediante la aplicación de un filtrado dinámico:

- El paso deductivo Init sólo contendrá producciones cuyo lado izquierdo se refiera al axioma de la gramática.
- En lugar de generar ítems de la forma $[A \to \bullet\Upsilon, -, i, i \mid -, -, -]$ para todas las posibles $A \to \Upsilon \in P$ y todas las posibles posiciones i y j de la cadena de entrada, un conjunto de pasos deductivos Pred se ocupará de generar únicamente aquellos ítems que involucren producciones cuyo esqueleto independiente del contexto sea relevante durante el proceso de análisis.

Esquema 3 *El sistema de análisis $\mathbb{P}_{\mathrm{E}}$ que se corresponde con el algoritmo de análisis de tipo Earley para una gramática lineal de índices $\mathcal{G}$ y una cadena de entrada $a_1 \dots a_n$ se define como sigue:*

$$\mathcal{I}_{\mathrm{E}} = \mathcal{I}_{\mathrm{buE}}$$

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Init}} = \frac{}{[S \to \bullet\Upsilon, -, 0, 0 \mid -, -, -]}$$

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}} = \frac{[A \to \Upsilon_1 \bullet B\, \Upsilon_2, \gamma, i, j \mid C, p, q]}{[B \to \bullet\Upsilon_3, -, j, j \mid -, -, -]}$$

$$\mathcal{D}_{\mathrm{E}} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Init}} \cup \mathcal{D}_{\mathrm{buE}}^{\mathrm{Scan}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}} \cup \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}^1} \cup \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}^2} \cup \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}^3} \cup \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}^4}$$

$$\mathcal{F}_{\mathrm{E}} = \mathcal{F}_{\mathrm{buE}}$$

Este esquema, cuya complejidad espacial es $\mathcal{O}(n^4)$ y la temporal $\mathcal{O}(n^6)$, está muy relacionado con el algoritmo de tipo Earley descrito por Schabes y Shieber en [10] aunque aquel sólo es aplicable a una clase específica de gramáticas lineales de índices obtenida a partir de una gramática de adjunción de árboles. Sin embargo, ambos comparten una característica muy importante, como es que el tipo de predicción realizado es muy poco potente puesto que no toma en consideración el contenido de la pila de índices. Aunque aparentemente el algoritmo propuesto por Schabes y Shieber en [10] utiliza información de la pila de índices para realizar la predicción, un análisis más profundo nos lleva a la conclusión de que esto no es realmente así, puesto que dichos autores optaron por almacenar el esqueleto independiente del contexto de los árboles elementales en las pilas de índices, reduciendo el conjunto de no-terminales de la LIG resultante a $\{t, b\}$. En efecto, una producción $b[\circ\circ\eta] \to t[\eta_1] \dots t[\circ\circ\eta_s] \dots t[\eta_n]$ es equivalente a $\eta^b[\circ\circ] \to \eta_1^t[\,] \dots \eta_s^t[\circ\circ] \dots \eta_n^t[\,]$ y una producción $b[\circ\circ\eta] \to t[\eta_1] \dots t[\eta_n]$ es equivalente a $\eta^b[\,] \to \eta_1^t[\,] \dots \eta_n^t[\,]$.

5 Algoritmo de tipo Earley que preserva la VPP

Los analizadores sintácticos que satisfacen la *propiedad del prefijo válido* (VPP) garantizan que, en tanto que leen la cadena de entrada de izquierda a derecha, las subcadenas leídas son prefijos válidos del lenguaje definido por la gramática. Más formalmente, un analizador sintáctico satisface la propiedad del prefijo válido si al leer la subcadena $a_1 \dots a_k$ de la cadena de entrada $a_1 \dots a_k a_{k+1} \dots a_n$ garantiza que hay una cadena $b_1 \dots b_m$, donde b_i no tiene porqué formar parte de la cadena de entrada, tal que $a_1 \dots a_k b_1 \dots b_m$ es una cadena válida del lenguaje.

En el caso de LIG, para mantener la propiedad del prefijo válido es necesario comprobar cada vez que se predice una regla si se cumple que $S[\,] \stackrel{*}{\Rightarrow} w\ A[\alpha\gamma]\ \Upsilon$, donde $A[\circ\circ\gamma]$ es el lado izquierdo de dicha regla.

Para obtener un algoritmo de tipo Earley con la propiedad del prefijo válido es preciso modificar el esquema **E** de tal modo que los pasos Pred predigan información acerca de las pilas de índices. Para ello también será necesario modificar la forma de los ítems para permitir seguir el rastro de las pilas de índices que se van prediciendo. Definiremos por tanto un nuevo conjunto de ítems de la forma

$$[E,h \mid A \to \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q]$$

que representan alguno de los siguientes tipos de derivaciones:

- $E[\ ] \stackrel{*}{\Rightarrow} a_{h+1} \dots a_i \ A[\gamma] \ \Upsilon_3 \stackrel{*}{\Rightarrow} a_{h+1} \dots a_i \dots a_p \ B[\ ] \ a_{q+1} \dots a_j \ \Upsilon_2 \Upsilon_3$ si y sólo si $(B,p,q) \neq (-,-,-)$, donde $A[\gamma]$ es un descendiente dependiente de $E[\ ]$ y $B[\ ]$ es un descendiente dependiente de $A[\gamma]$. Este tipo de derivación se corresponde con la compleción del hijo dependiente de una regla que tiene el no-terminal A como lado izquierdo. Además, la pila asociada a dicho no-terminal no debe estar vacía.

- $E[\ ] \stackrel{*}{\Rightarrow} a_{h+1} \dots a_i \ A[\gamma] \ \Upsilon_3 \stackrel{*}{\Rightarrow} a_{h+1} \dots a_i \dots a_j \ \Upsilon_2 \Upsilon_3$ si y sólo si $(E,h) \neq (-,-)$ y $(B,p,q) = (-,-,-)$, donde $A[\gamma]$ es un descendiente dependiente de $E[\ ]$ y Υ_1 no contiene el hijo dependiente de $A[\gamma]$. Este tipo de derivación se refiere a la predicción del no-terminal A con una pila de índices no vacía.

- $\Upsilon_1 \stackrel{*}{\Rightarrow} a_{i+1} \dots a_j$ si y sólo si $(E,h) = (-,-)$, $\gamma = -$ y $(B,p,q) = (-,-,-)$. Si Υ_1 incluye al hijo dependiente de A entonces la pila asociada a dicho hijo dependiente está vacía. Este tipo de derivación se refiere a la predicción o compleción del no-terminal A con una pila de índices vacía.

El nuevo conjunto de ítems así definido es un refinamiento de los ítems del esquema $\mathbf{E}$: el elemento γ se utiliza para almacenar la cima de la pila de índices predicha (en el esquema $\mathbf{E}$ los ítem resultado de una predicción tenían $\gamma = -$) y el par de elementos (E,h) permite seguir la traza del ítem involucrado en la predicción. En principio podríamos suponer que para guardar dicha traza es necesario almacenar (E,η,h,k). Sin embargo, por la propiedad de independencia del contexto de LIG dicho elemento no es necesario puesto que la derivación será válida independiente del resto de la pila de índices. El índice k no es necesario puesto que al ser todos los ítems predichos en el esquema $\mathbf{E}$ de la forma $[A \to \bullet\Upsilon, h, h \mid B, p, q]$ su presencia sería redundante.

Con respecto a los pasos deductivos, será necesario adaptar los pasos de compleción para que manipulen adecuadamente los nuevos componentes E y h y refinar los pasos predictivos con el fin de diferenciar los diferentes casos que se pueden dar.

Esquema 4 *El sistema de análisis $\mathbb{P}_{\text{Earley}_1}$ que se corresponde con el algoritmo de análisis de tipo Earley que preserva la propiedad del prefijo válido para una gramática lineal de índices $\mathcal{G}$ y una cadena de entrada $a_1 \dots a_n$ se define como sigue:*

$$\mathcal{I}_{\text{Earley}_1} = \left\{ \begin{array}{l} [E,h \mid A \to \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q] \mid \\ A \to \Upsilon_1 \Upsilon_2 \in P, \ B, C \in V_N, \ \gamma \in V_I, \\ 0 \leq h \leq i \leq j \ , \ (p,q) \leq (i,j) \end{array} \right\}$$

$$\mathcal{D}^{\text{Init}}_{\text{Earley}_1} = \frac{}{[-,- \mid S \to \bullet\Upsilon, -, 0, 0 \mid -,-,-]}$$

$$\mathcal{D}^{\text{Scan}}_{\text{Earley}_1} = \frac{\begin{array}{c}[-,- \mid A[\,] \to \bullet a, -, j, j \mid -,-,-], \\ [a, j, j+1]\end{array}}{[-,- \mid A[\,] \to a\bullet, -, j, j+1 \mid -,-,-]}$$

$$\mathcal{D}^{\text{Pred}^1}_{\text{Earley}_1} = \frac{[E,h \mid A \to \Upsilon_1 \bullet B[\,]\,\Upsilon_2, \gamma, i, j \mid C, p, q]}{[-,- \mid B \to \bullet\Upsilon_3, -, j, j \mid -,-,-]}$$

$$\mathcal{D}^{\text{Pred}^2}_{\text{Earley}_1} = \frac{\begin{array}{c}[E,h \mid A[\circ\circ\gamma] \to \Upsilon_1 \bullet B[\circ\circ]\,\Upsilon_2, \gamma, i, j \mid -,-,-], \\ [M,m \mid E \to \bullet\Upsilon_3, \gamma', h, h \mid -,-,-]\end{array}}{[M,m \mid B \to \bullet\Upsilon_4, \gamma', j, j \mid -,-,-]}$$

$$\mathcal{D}^{\text{Pred}^3}_{\text{Earley}_1} = \frac{[E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ]\,\Upsilon_2, \gamma, i, j \mid -,-,-]}{[E,h \mid B \to \bullet\Upsilon_3, \gamma, j, j \mid -,-,-]}$$

$$\mathcal{D}^{\text{Pred}^4}_{\text{Earley}_1} = \frac{[E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ\gamma]\,\Upsilon_2, \gamma', i, j \mid -,-,-]}{[A,i \mid B \to \bullet\Upsilon_3, \gamma, j, j \mid -,-,-]}$$

$$\mathcal{D}^{\text{Comp}^1}_{\text{Earley}_1} = \frac{\begin{array}{c}[E,h \mid A \to \Upsilon_1 \bullet B[\,]\,\Upsilon_2, \gamma, i, j \mid C, p, q], \\ [-,- \mid B \to \Upsilon_3\bullet, -, j, k \mid -,-,-]\end{array}}{[E,h \mid A \to \Upsilon_1 \ B[\,] \bullet \Upsilon_2, \gamma, i, k \mid C, p, q]}$$

$$\mathcal{D}^{\text{Comp}^2}_{\text{Earley}_1} = \frac{\begin{array}{c}[E,h \mid A[\circ\circ\gamma] \to \Upsilon_1 \bullet B[\circ\circ]\,\Upsilon_2, \gamma, i, j \mid -,-,-], \\ [M,m \mid E \to \bullet\Upsilon_3, \gamma', h, h \mid -,-,-], \\ [M,m \mid B \to \Upsilon_4\bullet, \gamma', j, k \mid C, p, q]\end{array}}{[E,h \mid A[\circ\circ\gamma] \to \Upsilon_1 \ B[\circ\circ] \bullet \Upsilon_2, \gamma, i, k \mid B, j, k]}$$

$$\mathcal{D}^{\text{Comp}^3}_{\text{Earley}_1} = \frac{\begin{array}{c}[E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ]\,\Upsilon_2, \gamma, i, j \mid -,-,-], \\ [E,h \mid B \to \Upsilon_3\bullet, \gamma, j, k \mid C, p, q]\end{array}}{[E,h \mid A[\circ\circ] \to \Upsilon_1 \ B[\circ\circ] \bullet \Upsilon_2, \gamma, i, k \mid C, p, q]}$$

$$\mathcal{D}^{\text{Comp}^4}_{\text{Earley}_1} = \frac{\begin{array}{c}[E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ\gamma]\,\Upsilon_2\,,\gamma',i,j \mid -,-,-],\\ [A,i \mid \boldsymbol{B} \to \Upsilon_3\bullet\,,\gamma,j,k \mid C,p,q],\\ [E,h \mid \boldsymbol{C} \to \Upsilon_4\bullet,\gamma',p,q \mid D,r,s]\end{array}}{[E,h \mid A[\circ\circ] \to \Upsilon_1\; B[\circ\circ\gamma] \bullet \Upsilon_2,\gamma',i,k \mid D,r,s]}$$

$$\mathcal{D}_{\text{Earley}_1} = \mathcal{D}^{\text{Init}}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Scan}}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^1}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^2}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^3}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^4}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^1}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^2}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^3}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^4}_{\text{Earley}_1}$$

$$\mathcal{F}_{\text{Earley}_1} = \{\ [-,- \mid \boldsymbol{S} \to \Upsilon\bullet\,,-,0,n \mid -,-,-]\ \}$$

La complejidad espacial con respecto a la longitud n de la cadena de entrada del algoritmo descrito por el esquema de análisis **Earley₁** es $\mathcal{O}(n^5)$ puesto que cada ítem hace uso de 5 posiciones. La complejidad temporal con respecto a la cadena de entrada es $\mathcal{O}(n^7)$ y viene dada por el conjunto de pasos deductivos $\mathcal{D}^{\text{Comp}^4}_{\text{Earley}_1}$. Para rebajar la complejidad temporal del algoritmo recurriremos a una técnica similar a la utilizada en [4, 3] para disminuir la complejidad de los modelos de autómata para el análisis de lenguajes de adjunción de árboles. En el caso que nos ocupa, dividiremos cada paso deductivo de $\mathcal{D}^{\text{Comp}^4}_{\text{Earley}_1}$ en dos pasos de tal forma que la complejidad de cada uno de ellos sea a lo sumo $\mathcal{O}(n^6)$, obteniendo el siguiente esquema de análisis.

Esquema 5 *El sistema de análisis $\mathbb{P}_{\text{Earley}}$ que se corresponde con el algoritmo de análisis de tipo Earley que preserva la propiedad del prefijo válido para una gramática lineal de índices $\mathcal{G}$ y una cadena de entrada $a_1 \ldots a_n$ se define como sigue:*

$$\mathcal{I}^1_{\text{Earley}} = \left\{ \begin{array}{l} [E,h \mid \boldsymbol{A} \to \Upsilon_1 \bullet \Upsilon_2,\gamma,i,j \mid B,p,q] \mid \\ \boldsymbol{A} \to \Upsilon_1\Upsilon_2 \in P,\ B,C \in V_N,\\ \gamma \in V_I,\ 0 \le h \le i \le j,\ (p,q) \le (i,j) \end{array} \right\}$$

$$\mathcal{I}^2_{\text{Earley}} = \left\{ \begin{array}{l} [[\boldsymbol{A} \to \Upsilon\bullet,\gamma,i,j \mid B,p,q]] \mid \\ \boldsymbol{A} \to \Upsilon \in P,\ B \in V_N,\\ \gamma \in V_I,\ i \le j\ ,\ (p,q) \le (i,j) \end{array} \right\}$$

$$\mathcal{I}_{\text{Earley}} = \mathcal{I}^1_{\text{Earley}} \cup \mathcal{I}^2_{\text{Earley}}$$

$$\mathcal{D}^{\text{Comp}^{4.0}}_{\text{Earley}} = \frac{\begin{array}{c}[A,i \mid \boldsymbol{B} \to \Upsilon_3\bullet\,,\gamma,j,k \mid C,p,q],\\ [E,h \mid \boldsymbol{C} \to \Upsilon_4\bullet,\gamma',p,q \mid D,r,s]\end{array}}{[[\boldsymbol{B} \to \Upsilon_3\bullet\,,\gamma,j,k \mid D,r,s]]}$$

$$\mathcal{D}^{\text{Comp}^{4.1}}_{\text{Earley}} = \frac{\begin{array}{c}[[\boldsymbol{B} \to \Upsilon_3\bullet\,,\gamma,j,k \mid D,r,s]],\\ [E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ\gamma]\,\Upsilon_2\,,\gamma',i,j \mid -,-,-],\\ [E,h \mid \boldsymbol{C} \to \Upsilon_4\bullet,\gamma',p,q \mid D,r,s]\end{array}}{[E,h \mid A[\circ\circ] \to \Upsilon_1\; B[\circ\circ\gamma] \bullet \Upsilon_2,\gamma',i,k \mid D,r,s]}$$

$$\mathcal{D}_{\text{Earley}} = \mathcal{D}^{\text{Init}}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Scan}}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^1}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^2}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^3}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}^4}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^1}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^2}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^3}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}^{4.0}}_{\text{Earley}} \cup \mathcal{D}^{\text{Comp}^{4.1}}_{\text{Earley}}$$

$$\mathcal{F}_{\text{Earley}} = \mathcal{F}_{\text{Earley}_1}$$

El primer paso genera un pseudo-ítem intermedio de la forma $[[\boldsymbol{B} \to \Upsilon_3\bullet\,,\gamma,j,k \mid D,r,s]]$ que a su vez es un antecedente del segundo paso y que representa una derivación

$$B[\gamma'\gamma] \stackrel{*}{\Rightarrow} a_{j+1} \ldots a_r\; D[\,]\; a_{s+1} \ldots a_k$$

para algún γ'. Los pasos de $\mathcal{D}^{\text{Comp}^{4.1}}_{\text{Earley}}$ combinan dicho pseudo-ítem con el ítem $[E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ\gamma]\,\Upsilon_2\,,\gamma',i,j \mid -,-,-]$ que representa una derivación

$$\begin{array}{rl} E[\,] & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_i\; A[\gamma']\,\Upsilon_3 \\ & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_i \ldots a_j\; B[\gamma'\gamma]\,\Upsilon_2\Upsilon_3 \end{array}$$

y con el ítem $[E,h \mid \boldsymbol{C} \to \Upsilon_4\bullet,\gamma',p,q \mid D,r,s]$ que representa una derivación

$$\begin{array}{rl} E[\,] & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_p\; C[\gamma']\,\Upsilon_5 \\ & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_p \ldots a_r\; D[\,]\; a_{s+1} \ldots a_q\; \Upsilon_5 \end{array}$$

con lo cual podemos generar un ítem de la forma $[E,h \mid A[\circ\circ] \to \Upsilon_1\; B[\circ\circ\gamma] \bullet \Upsilon_2,\gamma',i,k \mid D,r,s]$ que representa la existencia de una derivación

$$\begin{array}{rl} E[\,] & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_i\; A[\gamma']\,\Upsilon_3 \\ & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_i \ldots a_j\; B[\gamma'\gamma]\,\Upsilon_2\Upsilon_3 \\ & \stackrel{*}{\Rightarrow} a_{h+1} \ldots a_i \ldots a_j \ldots a_r\; D[\,]\; a_{s+1} \ldots a_k\; \Upsilon_2\Upsilon_3 \end{array}$$

6 Conclusión

Se ha descrito un conjunto de algoritmos de análisis sintáctico de TAG que constituyen una línea evolutiva continua que tiene como punto de partida el algoritmo de tipo CYK de Vijay-Shanker y Weir [12] y que a través de nuevos algoritmos de tipo Earley ascendente y de tipo Earley sin la propiedad del prefijo válido permite llegar a un nuevo algoritmo de tipo Earley que preserva dicha propiedad manteniendo la complejidad temporal $\mathcal{O}(n^6)$ con respecto a la cadena de entrada. En esta línea evolutiva podrían haberse considerado más algoritmos intermedios, pero las restricciones en el espacio disponible han aconsejado considerar solamente los hitos más importantes en dicha evolución.

Es interesante destacar la relación existente entre los algoritmos de análisis de LIG y aquellos destinados al análisis de TAG [2]. En particular, la técnica utilizada para reducir la complejidad del algoritmo de tipo Earley con la propiedad del prefijo válido puede trasponerse directamente al caso de TAG, con lo cual la fase de compleción de adjunción del algoritmo de tipo Earley para TAG con la propiedad del prefijo válido propuesto por Nederhof [7] quedaría definida por los dos pasos siguientes en lugar de los tres pasos $\mathcal{D}_{\text{Earley}}^{\text{AdjComp}^0}$, $\mathcal{D}_{\text{Earley}}^{\text{AdjComp}^1}$ y $\mathcal{D}_{\text{Earley}}^{\text{AdjComp}^2}$ descritos en [2]:

$$\mathcal{D}_{\text{Earley}}^{\text{AdjComp}^{0'}} = \frac{\begin{array}{c}[j, \top \to \mathbf{R}^\beta \bullet, j, m \mid k, l], \\ [h, M^\gamma \to \upsilon \bullet, k, l \mid p, q],\end{array}}{[[M^\gamma \to \upsilon \bullet, j, m \mid p, q]]}$$

tal que $\beta \in \text{adj}(M^\gamma)$.

$$\mathcal{D}_{\text{Earley}}^{\text{AdjComp}^{1'}} = \frac{\begin{array}{c}[[M^\gamma \to \upsilon \bullet, j, m \mid p, q]], \\ [h, M^\gamma \to \upsilon \bullet, k, l \mid p, q], \\ [h, N^\gamma \to \delta \bullet M^\gamma \nu, i, j \mid p', q']\end{array}}{[h, N^\gamma \to \delta M^\gamma \bullet \nu, i, m \mid p \cup p', q \cup q']}$$

tal que $\beta \in \text{adj}(M^\gamma)$.

En este caso explotamos la información del ítem $[h, M^\gamma \to \upsilon \bullet, k, l \mid p, q]$ en vez de utilizar un ítem redundante $[h, \mathbf{F}^\gamma \to \bot \bullet, p, q \mid p, q]$ y se preserva igualmente la complejidad $\mathcal{O}(n^6)$.

Referencias

[1] Alfred V. Aho. Indexed grammars — an extension of context-free grammars. *Journal of the Association for Computer Machinery*, 15(4):647–671, 1968.

[2] Miguel A. Alonso, David Cabrero, Eric de la Clergerie y Manuel Vilares. Tabular algorithms for TAG parsing. In *Proc. of EACL'99*, Bergen, Noruega, junio de 1999.

[3] Miguel A. Alonso, Eric de la Clergerie y David Cabrero. Tabulation of automata for tree adjoining languages. In *Proc. of MOL-6*, Orlando, FL, EE.UU., julio de 1999.

[4] Eric de la Clergerie y Miguel A. Alonso. A tabular interpretation of a class of 2-Stack Automata. In *Proc. of COLING-ACL'98*, pp. 1333–1339, Montreal, Quebec, Canadá, agosto de 1998.

[5] Eric de la Clergerie, Miguel A. Alonso y David Cabrero. A tabular interpretation of bottom-up automata for TAG. In *Proc. of TAG+4*, pp. 42–45, Filadelfia, PA, EE.UU., agosto de 1998.

[6] Gerald Gazdar. Applicability of indexed grammars to natural languages. In U. Reyle y C. Rohrer, editores, *Natural Language Parsing and Linguistic Theories*, pp. 69–94. D. Reidel Publishing Company, 1987.

[7] Mark-Jan Nederhof. Solving the correct-prefix property for TAGs. In T. Becker y H.-V. Krieger, editores, *Proc. of MOL-5*, pp. 124–130, Schloss Dagstuhl, Saarbruecken, Alemania, agosto de 1997.

[8] Mark-Jan Nederhof. Linear indexed automata and tabulation of TAG parsing. In *Proc. of TAPD'98*, pp. 1–9, París, Francia, abril de 1998.

[9] Yves Schabes. Stochastic lexicalized tree-adjoining grammars. In *Proc. of COLING'92*, pp. 426–432, Nantes, Francia, agosto de 1992.

[10] Yves Schabes y Stuart M. Shieber. An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20(1):91–124, 1994.

[11] Klaas Sikkel. *Parsing Schemata — A Framework for Specification and Analysis of Parsing Algorithms*. Springer-Verlag, Berlín/Heidelberg/Nueva York, 1997.

[12] K. Vijay-Shanker y David J. Weir. Polynomial parsing of extensions of context-free grammars. In Masaru Tomita, editor, *Current Issues in Parsing Technology*, capítulo 13, pp. 191–206. Kluwer Academic Publishers, Norwell, MA, EE.UU., 1991.

[13] K. Vijay-Shanker y David J. Weir. Parsing some constrained grammar formalisms. *Computational Linguistics*, 19(4):591–636, 1993.