Capítulo 4

Algoritmos de análisis sintáctico para LIG

Describiremos varios algoritmos para el análisis de gramáticas lineales de índices. Además del interés que presenta por sí mismo el análisis sintáctico de este tipo de gramáticas, su estudio presenta como atractivo adicional el que muchos autores prefieran realizar el análisis sintáctico de las gramáticas de adjunción de árboles indirectamente a través de LIG, esto es, traduciendo la gramática original a una gramática lineal de índices y aplicando un algoritmo de análisis sintáctico a esta última.

La mayor parte de los algoritmos de análisis sintáctico de LIG están basados en algoritmos de análisis independientes del contexto extendidos para permitir el tratamiento de las pilas de índices asociadas a los no-terminales de la gramática. Asimismo, es de destacar que prácticamente todos estos algoritmos utilizan técnicas de programación dinámica, obteniendo una complejidad temporal polinómica de orden $\mathcal{O}(n^6)$, donde n es la longitud de la cadena de entrada.

La aportación de este capítulo es múltiple, destacando la definición de nuevos algoritmos para el análisis de LIG, como es el caso del algoritmo de tipo Earley ascendente, del algoritmo de tipo Earley sin la propiedad del prefijo válido y de los algoritmos de tipo Earley que preservan dicha propiedad. Mediante estos nuevos algoritmos podemos aplicar al caso de LIG las mismas estrategias de análisis descritas para el caso de TAG en el capítulo 3. Ello permite mantener la estrategia de análisis cuando se trata de analizar una TAG previamente compilada a LIG. Además de presentar estos nuevos algoritmos, se proporciona un camino evolutivo que los relaciona y permite derivar uno de otro y se presenta una nueva definición de bosque compartido de análisis sintáctico de LIG. Por último, se realiza una descripción conjunta de la mayor parte de los algoritmos de análisis existentes para LIG. El contenido de este capítulo está basado en [15, 19].

4.1. Algoritmo de tipo CYK

A continuación mostramos una extensión para LIG del algoritmo CYK de análisis sintáctico basado en el algoritmo descrito por Vijay-Shanker y Weir en [213], al que incorpora algunas correcciones. 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 [85] 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 se utiliza un conjunto de ítems de la

forma

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

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 $(B, p, q) = (-, -, -)$.

Estos ítems son ligeramente diferentes de los propuestos por Vijay-Shanker y Weir en [213], pues estos últimos presentaban la forma $[A, \gamma, i, j \mid B, \eta, p, q]$, con $\eta \in V_I$. El elemento η es redundante ya que por la propiedad de independencia del contexto de las gramáticas lineales de índices (definición 2.1, página 33) sabemos que si $A[\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \dots a_p \ B[\ ] \ a_{q+1} \dots a_j$ entonces para cualquier α se cumple que $A[\alpha\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \dots a_p \ B[\alpha] \ a_{q+1} \dots a_j$. La descomposición de α que realizan Vijay-Shanker y Weir al considerar $\alpha = \alpha'\eta$ es innecesaria y el almacenamiento de η lo único que provoca es un aumento en el número de ítems necesarios para representar una derivación, puesto que deberemos utilizar un ítem $[A, \gamma, i, j \mid B, \eta, p, q]$ diferente para cada valor de η, cuando con un solo ítem $[A, \gamma, i, j \mid B, p, q]$ es suficiente.

Esquema de análisis sintáctico 4.1 El sistema de análisis $\mathbb{P}_{\text{CYK}}$ que se corresponde con el algoritmo de análisis de tipo CYK para una gramática lineal de índices $\mathcal{L}$ y una cadena de entrada $a_1 \dots a_n$ se define como sigue:

$$\mathcal{I}_{\text{CYK}} = \left\{ \ [A, \gamma, i, j \mid B, p, q] \mid A, B \in V_N, \ \gamma \in V_I, \ 0 \leq i \leq j \ , \ \ (p, q) \leq (i, j) \ \right\}$$

$$\mathcal{H}_{\text{CYK}} = \left\{ \ [a, i-1, i] \mid a = a_i, \ 1 \leq i \leq n \ \right\}$$

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

$$\mathcal{D}^{[\circ\circ\gamma][\ ][\circ\circ]}_{\text{CYK}} = \frac{\begin{array}{l}[B, -, i, k \mid -, -, -], \\ [C, \eta, k, j \mid D, p, q]\end{array}}{[A, \gamma, i, j \mid C, k, j]} \quad A[\circ\circ\gamma] \rightarrow B[\ ] \ C[\circ\circ] \in P$$

$$\mathcal{D}^{[\circ\circ\gamma][\circ\circ][\ ]}_{\text{CYK}} = \frac{\begin{array}{l}[B, \eta, i, k \mid D, p, q], \\ [C, -, k, j \mid -, -, -]\end{array}}{[A, \gamma, i, j \mid B, i, k]} \quad A[\circ\circ\gamma] \rightarrow B[\circ\circ] \ C[\ ] \in P$$

$$\mathcal{D}^{[\circ\circ][\ ][\circ\circ]}_{\text{CYK}} = \frac{\begin{array}{l}[B, -, i, k \mid -, -, -], \\ [C, \eta, k, j \mid D, p, q]\end{array}}{[A, \eta, i, j \mid D, p, q]} \quad A[\circ\circ] \rightarrow B[\ ] \ C[\circ\circ] \in P$$

$$\mathcal{D}^{[\circ\circ][\circ\circ][\ ]}_{\text{CYK}} = \frac{\begin{array}{l}[B, \eta, i, k \mid D, p, q], \\ [C, -, k, j \mid -, -, -]\end{array}}{[A, \eta, i, j \mid D, p, q]} \quad A[\circ\circ] \rightarrow B[\circ\circ] \ C[\ ] \in P$$

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

$$\mathcal{D}_{\text{CYK}}^{[\circ\circ][\circ\circ\gamma][\,]} = \frac{\begin{array}{l}[B,\gamma,i,k \mid D,p,q],\\ [C,-,k,j \mid -,-,-],\\ [D,\eta,p,q \mid E,r,s]\end{array}}{[A,\eta,i,j \mid E,r,s]} \quad A[\circ\circ] \rightarrow B[\circ\circ\gamma]\; C[\,] \in P$$

$$\mathcal{D}_{\text{CYK}} = \mathcal{D}_{\text{CYK}}^{\text{Scan}} \cup \mathcal{D}_{\text{CYK}}^{[\circ\circ\gamma][\,][\circ\circ]} \cup \mathcal{D}_{\text{CYK}}^{[\circ\circ\gamma][\circ\circ][\,]} \cup \mathcal{D}_{\text{CYK}}^{[\circ\circ][\,][\circ\circ]} \cup \mathcal{D}_{\text{CYK}}^{[\circ\circ][\circ\circ][\,]} \cup \mathcal{D}_{\text{CYK}}^{[\circ\circ][\,][\circ\circ\gamma]} \cup \mathcal{D}_{\text{CYK}}^{[\circ\circ][\circ\circ\gamma][\,]}$$

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

§

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 capítulo. Por consiguiente, no nos volveremos a referir explícitamente a ellas.

Los pasos $\mathcal{D}_{\text{CYK}}^{\text{Scan}}$ 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 una producción para generar el ítem correspondiente al lado izquierdo de dicha producción.

La complejidad espacial del algoritmo con respecto a la longitud n de la cadena de entrada es $\mathcal{O}(n^4)$ puesto que cada ítem almacena cuatro posiciones de la cadena de entrada. La complejidad temporal con respecto a la cadena de entrada es $\mathcal{O}(n^6)$ y viene dada por los pasos deductivos $\mathcal{D}_{\text{CYK}}^{[\circ\circ][\,][\circ\circ\gamma]}$ y $\mathcal{D}_{\text{CYK}}^{[\circ\circ][\circ\circ\gamma][\,]}$. 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 [214] 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. Schabes extiende en [170] el algoritmo CYK para permitir el cálculo de probabilidades requerido para el tratamiento de LIG estocásticas.

4.2. 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. Para evitar esta limitación vamos a considerar la extensión del algoritmo Earley ascendente al caso de las gramáticas lineales de índices. Debemos reseñar que no conocemos ninguna adaptación anterior de este algoritmo para LIG.

Como primer paso para la definición de un algoritmo de tipo Earley ascendente para LIG debemos proceder a la introducción de 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 utilizada, 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. En consecuencia, la aparición de diferentes $\boldsymbol{A}$ en un ítem o paso deductivo indica que en todos los casos de trata de elementos LIG con el mismo no-terminal A pero que puede que tengan asociadas distintas pilas de índices.

Los ítems utilizados en el algoritmo de análisis sintáctico de tipo Earley ascendente para LIG tienen la forma

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

y representan alguno de los siguientes tipos de derivaciones:

- $A[\gamma] \Rightarrow \Upsilon_1\Upsilon_2 \stackrel{*}{\Rightarrow} a_{i+1}\dots a_p\ B[\,]\ a_{q+1}\dots a_j\ \Upsilon_2$ si y sólo si $(B,p,q) \neq (-,-,-)$, donde $B[\,]$ es un descendiente dependiente de $A[\gamma]$.
- $\Upsilon_1 \stackrel{*}{\Rightarrow} a_{i+1}\dots a_j$ si y sólo si $\gamma = -$ y $(B,p,q) = (-,-,-)$. Si Υ_1 incluye al hijo dependiente entones las pilas asociadas a $\boldsymbol{A}$ y al hijo dependiente están vacías.

Podemos observar que los ítems del nuevo esquema de análisis sintáctico, que denominaremos **buE**, son un refinamiento de los ítems de **CYK**. Sobre los pasos deductivos aplicaremos 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 tipos de pasos Init y Comp. Finalmente se realizará una *extensión* del dominio de las producciones para permitir gramáticas lineales de índices con producciones de longitud arbitraria.

Esquema de análisis sintáctico 4.2 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 $\mathcal{L}$ y una cadena de entrada $a_1\dots a_n$ se define como sigue:

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

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

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

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

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

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

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

$$\mathcal{D}_{\text{buE}} = \mathcal{D}_{\text{buE}}^{\text{Init}} \cup \mathcal{D}_{\text{buE}}^{\text{Scan}} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}[\,]} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}[\circ\circ\gamma][\circ\circ]} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}[\circ\circ][\circ\circ]} \cup \mathcal{D}_{\text{buE}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}$$

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

§

Proposición 4.1 $\mathbf{CYK} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{CYK'} \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{ECYK} \overset{\mathbf{ext}}{\Longrightarrow} \mathbf{buE}$.

Demostración:

Como primer paso definiremos el sistema de análisis $\mathbb{P}_{\mathrm{CYK}'}$ para una gramática lineal de índices $\mathcal{L}$ y una cadena de entrada $a_1 \dots a_n$.

$$\mathcal{I}_{\mathrm{CYK}'} = \left\{ \begin{array}{ll} [\boldsymbol{A} \to \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q] \mid & \boldsymbol{A} \to \Upsilon_1\Upsilon_2 \in P,\ B \in V_N,\ \gamma \in V_I, \\ & 0 \le i \le j\ ,\ \ (p,q) \le (k,j) \end{array} \right\}$$

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

$$\mathcal{D}^{[\circ\circ\gamma][\,][\circ\circ]}_{\mathrm{CYK}'} = \frac{\begin{array}{c}[\boldsymbol{B} \to \Upsilon_1\bullet, -, i, k \mid -, -, -], \\ [\boldsymbol{C} \to \Upsilon_2\bullet, \eta, k, j \mid D, p, q]\end{array}}{[A[\circ\circ\gamma] \to B[\,]\ C[\circ\circ]\bullet, \gamma, i, j \mid C, k, j]}$$

$$\mathcal{D}^{[\circ\circ\gamma][\circ\circ][\,]}_{\mathrm{CYK}'} = \frac{\begin{array}{c}[\boldsymbol{B} \to \Upsilon_1\bullet, \eta, i, k \mid D, p, q], \\ [\boldsymbol{C} \to \Upsilon_2\bullet, -, k, j \mid -, -, -]\end{array}}{[A[\circ\circ\gamma] \to B[\circ\circ]\ C[\,]\bullet, \gamma, i, j \mid B, i, k]}$$

$$\mathcal{D}^{[\circ\circ][\,][\circ\circ]}_{\mathrm{CYK}'} = \frac{\begin{array}{c}[\boldsymbol{B} \to \Upsilon_1\bullet, -, i, k \mid -, -, -], \\ [\boldsymbol{C} \to \Upsilon_2\bullet, \eta, k, j \mid D, p, q]\end{array}}{[A[\circ\circ] \to B[\,]\ C[\circ\circ]\bullet, \eta, i, j \mid D, p, q]}$$

$$\mathcal{D}^{[\circ\circ][\circ\circ][\,]}_{\mathrm{CYK}'} = \frac{\begin{array}{c}[\boldsymbol{B} \to \Upsilon_1\bullet, \eta, i, k \mid D, p, q], \\ [\boldsymbol{C} \to \Upsilon_2\bullet, -, k, j \mid -, -, -]\end{array}}{[A[\circ\circ] \to B[\circ\circ]\ C[\,]\bullet, \eta, i, j \mid D, p, q]}$$

$$\mathcal{D}^{[\circ\circ][\,][\circ\circ\gamma]}_{\mathrm{CYK}'} = \frac{\begin{array}{c}[\boldsymbol{B} \to \Upsilon_1\bullet, -, i, k \mid -, -, -], \\ [\boldsymbol{C} \to \Upsilon_2\bullet, \gamma, k, j \mid D, p, q], \\ [\boldsymbol{D} \to \Upsilon_3\bullet, \eta, p, q \mid E, r, s]\end{array}}{[A[\circ\circ] \to B[\,]\ C[\circ\circ\gamma]\bullet, \eta, i, j \mid E, r, s]}$$

$$\mathcal{D}^{[\circ\circ][\circ\circ\gamma][\,]}_{\mathrm{CYK}'} = \frac{\begin{array}{c}[\boldsymbol{B} \to \Upsilon_1\bullet, \gamma, i, k \mid D, p, q], \\ [\boldsymbol{C} \to \Upsilon_2\bullet, -, k, j \mid -, -, -], \\ [\boldsymbol{D} \to \Upsilon_3\bullet, \eta, p, q \mid E, r, s]\end{array}}{[A[\circ\circ] \to B[\circ\circ\gamma]\ C[\,]\bullet, \eta, i, j \mid E, r, s]}$$

$$\mathcal{D}_{\mathrm{CYK}'} = \mathcal{D}^{\mathrm{Scan}}_{\mathrm{CYK}'} \cup \mathcal{D}^{[\circ\circ\gamma][\,][\circ\circ]}_{\mathrm{CYK}'} \cup \mathcal{D}^{[\circ\circ\gamma][\circ\circ][\,]}_{\mathrm{CYK}'} \cup \mathcal{D}^{[\circ\circ][\,][\circ\circ]}_{\mathrm{CYK}'} \cup \mathcal{D}^{[\circ\circ][\circ\circ][\,]}_{\mathrm{CYK}'} \cup \mathcal{D}^{[\circ\circ][\,][\circ\circ\gamma]}_{\mathrm{CYK}'} \cup \mathcal{D}^{[\circ\circ][\circ\circ\gamma][\,]}_{\mathrm{CYK}'}$$

$$\mathcal{F}_{\mathrm{CYK}'} = \{\ [\boldsymbol{S} \to \Upsilon\bullet, -, 0, n \mid -, -, -]\ \}$$

Para demostrar que $\mathbf{CYK} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{CYK'}$, definiremos la siguiente función

$$f([\boldsymbol{A} \to \Upsilon\bullet, \gamma, i, j \mid C, p, q]) = [A, \gamma, i, j \mid C, p, q]$$

de la cual se obtiene directamente que $\mathcal{I}_{\mathrm{CYK}} = f(\mathcal{I}_{\mathrm{CYK}'})$ y que $\Delta_{\mathrm{CYK}} = f(\Delta_{\mathrm{CYK}'})$ por inducción en la longitud de las secuencias de derivación. En consecuencia, $\mathbb{P}_{\mathrm{CYK}} \xrightarrow{\mathbf{ir}} \mathbb{P}_{\mathrm{CYK}'}$, con lo que hemos probado lo que pretendíamos.

Definiremos ahora el sistema de análisis sintáctico $\mathbb{P}_{\text{ECYK}}$ para una gramática lineal de índices $\mathcal{L}$ cuyas producciones tienen a lo sumo dos elementos en el lado derecho y una cadena de entrada $a_1 \dots a_n$:

$$\mathcal{I}_{\text{ECYK}} = \mathcal{I}_{\text{CYK}'} = \mathcal{I}_{\text{buE}} =$$
$$\mathcal{D}^{\text{Init}}_{\text{ECYK}} = \mathcal{D}^{\text{Init}}_{\text{buE}}$$
$$\mathcal{D}^{\text{Scan}}_{\text{ECYK}} = \mathcal{D}^{\text{Scan}}_{\text{buE}}$$
$$\mathcal{D}^{\text{Comp}[\,]}_{\text{ECYK}} = \mathcal{D}^{\text{Comp}[\,]}_{\text{buE}}$$
$$\mathcal{D}^{\text{Comp}[\circ\circ\gamma][\circ\circ]}_{\text{ECYK}} = \mathcal{D}^{\text{Comp}[\circ\circ\gamma][\circ\circ]}_{\text{buE}}$$
$$\mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ]}_{\text{ECYK}} = \mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ]}_{\text{buE}}$$
$$\mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}_{\text{ECYK}} = \mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}_{\text{buE}}$$
$$\mathcal{F}_{\text{ECYK}} = \mathcal{F}_{\text{buE}}$$

Para demostrar que **CYK′** $\overset{\textbf{sr}}{\Longrightarrow}$ **ECYK**, deberemos demostrar que para todo sistema de análisis $\mathbb{P}_{\text{CYK}'}$ y $\mathbb{P}_{\text{ECYK}}$ se cumple que $\mathcal{I}_{\text{CYK}'} \subseteq \mathcal{I}_{\text{ECYK}}$ y $\overset{*}{\vdash}_{\text{CYK}'} \subseteq \overset{*}{\vdash}_{\text{ECYK}}$. Lo primero es cierto por definición, puesto que $\mathcal{I}_{\text{CYK}'} = \mathcal{I}_{\text{ECYK}}$. Para lo segundo debemos mostrar que $\mathcal{D}_{\text{CYK}'} \supseteq \overset{*}{\vdash}_{\text{ECYK}}$. Consideremos caso por caso:

- Un paso deductivo $\mathcal{D}^{\text{Scan}}_{\text{CYK}'}$ es equivalente a la secuencia de pasos deductivos constituida por la aplicación de un paso $\mathcal{D}^{\text{Init}}_{\text{ECYK}}$ y un paso $\mathcal{D}^{\text{Scan}}_{\text{ECYK}}$:

$$\frac{}{[A[\,] \to \bullet a, -, j, j \mid -, -, -]}$$

$$\frac{[A[\,] \to \bullet a, -, j, j \mid -, -, -],\ [a, j, j+1]}{[A[\,] \to a\bullet, -, j, j+1 \mid -, -, -]}$$

- Un paso deductivo $\mathcal{D}^{[\circ\circ\gamma][\,][\circ\circ]}_{\text{CYK}'}$ es equivalente a la secuencia de pasos deductivos constituida por la aplicación de un paso $\mathcal{D}^{\text{Init}}_{\text{ECYK}}$, un paso $\mathcal{D}^{\text{Comp}[\,]}_{\text{ECYK}}$ y un paso $\mathcal{D}^{[\circ\circ\gamma][\circ\circ]}_{\text{ECYK}}$:

$$\frac{}{[A[\circ\circ\gamma] \to \bullet B[\,]\, C[\circ\circ], -, i, i \mid -, -, -]}$$

$$\frac{[A[\circ\circ\gamma] \to \bullet B[\,]\, C[\circ\circ], -, i, i \mid -, -, -],\ [\boldsymbol{B} \to \Upsilon_1 \bullet, -, i, k \mid -, -, -]}{[A[\circ\circ\gamma] \to B[\,] \bullet C[\circ\circ], -, i, k \mid -, -, -]}$$

$$\frac{[A[\circ\circ\gamma] \to B[\,] \bullet C[\circ\circ], -, i, k \mid -, -, -],\ [\boldsymbol{C} \to \Upsilon_2 \bullet, \eta, k, j \mid D, p, q]}{[A[\circ\circ\gamma] \to B[\,]\, C[\circ\circ]\bullet, \gamma, i, j \mid C, k, j]}$$

- Un paso deductivo $\mathcal{D}^{[\circ\circ\gamma][\circ\circ][\,]}_{\text{CYK}'}$ es equivalente a la secuencia formada por un paso $\mathcal{D}^{\text{Init}}_{\text{ECYK}}$, un paso $\mathcal{D}^{\text{Comp}[\circ\circ\gamma][\circ\circ]}_{\text{ECYK}}$ y un paso $\mathcal{D}^{[\,]}_{\text{ECYK}}$:

$$\frac{}{[A[\circ\circ\gamma] \to \bullet B[\circ\circ]\, C[\,], -, i, i \mid -, -, -]}$$

$$\frac{[A[\circ\circ\gamma] \to \bullet B[\circ\circ]\, C[\,], -, i, i \mid -, -, -],\ [\boldsymbol{B} \to \Upsilon_1 \bullet, \eta, i, k \mid D, p, q]}{[A[\circ\circ\gamma] \to B[\circ\circ] \bullet C[\,], \gamma, i, k \mid B, i, k]}$$

$$\frac{[A[\circ\circ\gamma] \to B[\circ\circ] \bullet C[\,], -, i, k \mid B, i, k],\ [\boldsymbol{C} \to \Upsilon_2 \bullet, -, k, j \mid -, -, -]}{[A[\circ\circ\gamma] \to B[\circ\circ]\, C[\,]\bullet, \gamma, i, j \mid B, i, k]}$$

- Un paso $\mathcal{D}_{\text{CYK}'}^{[\circ\circ][\,][\circ\circ]}$ es equivalente a una secuencia de pasos formada por un paso $\mathcal{D}_{\text{ECYK}}^{\text{Init}}$, un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\,]}$ y un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\circ\circ][\circ\circ]}$:

$$\frac{}{[A[\circ\circ] \to \bullet B[\,]\, C[\circ\circ], -, i, i \mid -, -, -]}$$

$$\frac{[A[\circ\circ] \to \bullet B[\,]\, C[\circ\circ], -, i, i \mid -, -, -],\ [\boldsymbol{B} \to \Upsilon_1 \bullet, -, i, k \mid -, -, -]}{[A[\circ\circ] \to B[\,] \bullet C[\circ\circ], -, i, k \mid -, -, -]}$$

$$\frac{[A[\circ\circ] \to B[\,] \bullet C[\circ\circ], -, i, k \mid -, -, -],\ [\boldsymbol{C} \to \Upsilon_2 \bullet, \eta, k, j \mid D, p, q]}{[A[\circ\circ] \to B[\,]\, C[\circ\circ] \bullet, \eta, i, j \mid D, p, q]}$$

- Un paso $\mathcal{D}_{\text{CYK}'}^{[\circ\circ][\,][\circ\circ]}$ es equivalente a una secuencia de pasos formada por un paso $\mathcal{D}_{\text{ECYK}}^{\text{Init}}$, un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\circ\circ][\circ\circ]}$ y un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\,]}$:

$$\frac{}{[A[\circ\circ] \to \bullet B[\circ\circ]; C[\,], -, i, i \mid -, -, -]}$$

$$\frac{[A[\circ\circ] \to \bullet B[\circ\circ]\, C[\,], -, i, i \mid -, -, -],\ [\boldsymbol{B} \to \Upsilon_1 \bullet, \eta, i, k \mid D, p, q]}{[A[\circ\circ] \to B[\,] \bullet C[\circ\circ], \eta, i, k \mid D, p, q]}$$

$$\frac{[A[\circ\circ] \to B[\,] \bullet C[\circ\circ], \eta, i, k \mid D, p, q],\ [\boldsymbol{C} \to \Upsilon_2 \bullet, -, k, j \mid -, -, -]}{[A[\circ\circ] \to B[\,]\, C[\circ\circ] \bullet, \eta, i, j \mid D, p, q]}$$

- Un paso $\mathcal{D}_{\text{CYK}'}^{[\circ\circ][\,][\circ\circ\gamma]}$ es equivalente a la secuencia de pasos formada por un paso $\mathcal{D}_{\text{ECYK}}^{\text{Init}}$, un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\,]}$ y un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}$:

$$\frac{}{[A[\circ\circ] \to \bullet B[\,]\, C[\circ\circ\gamma], -, i, i \mid -, -, -]}$$

$$\frac{[A[\circ\circ] \to \bullet B[\,]\, C[\circ\circ\gamma], -, i, i \mid -, -, -],\ [\boldsymbol{B} \to \Upsilon_1 \bullet, -, i, k \mid -, -, -],}{[A[\circ\circ] \to B[\,] \bullet C[\circ\circ\gamma], -, i, k \mid -, -, -]}$$

$$\frac{\begin{array}{l}[A[\circ\circ] \to B[\,] \bullet C[\circ\circ\gamma], -, i, k \mid -, -, -],\\ [\boldsymbol{C} \to \Upsilon_2 \bullet, \gamma, k, j \mid D, p, q],\\ [\boldsymbol{D} \to \Upsilon_3 \bullet, \eta, p, q \mid E, r, s]\end{array}}{[A[\circ\circ] \to B[\,] \bullet C[\circ\circ\gamma], \eta, i, j \mid E, r, s]}$$

- Un paso deductivo $\mathcal{D}_{\text{CYK}'}^{[\circ\circ][\circ\circ\gamma][\,]}$ es equivalente a la secuencia de pasos deductivo constituida por un paso $\mathcal{D}_{\text{ECYK}}^{\text{Init}}$, un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}$ y un paso $\mathcal{D}_{\text{ECYK}}^{\text{Comp}[\,]}$:

$$\frac{}{[A[\circ\circ] \to \bullet B[\circ\circ\gamma]; C[\,], -, i, i \mid -, -, -]}$$

$$\frac{\begin{array}{l}[A[\circ\circ] \to \bullet B[\circ\circ\gamma]; C[\,], -, i, i \mid -, -, -, -],\\ [\boldsymbol{B} \to \Upsilon_1 \bullet, \gamma, i, k \mid D, p, q],\\ [\boldsymbol{D} \to \Upsilon_2 \bullet, \eta, p, q \mid E, r, s]\end{array}}{[A[\circ\circ] \to B[\circ\circ\gamma] \bullet C[\,], \eta, i, k \mid E, r, s]}$$

$$\frac{[A[\circ\circ] \to B[\circ\circ\gamma] \bullet C[\,], \eta, i, k \mid E, r, s],\ [\boldsymbol{C} \to \Upsilon_3 \bullet, -, k, j \mid -, -, -],}{[A[\circ\circ] \to B[\circ\circ\gamma]\, C[\,] \bullet, \eta, i, j \mid E, r, s]}$$

El esquema de análisis sintáctico **ECYK** está definido para gramáticas lineales de índices en las cuales ningún ninguna producción puede tener más de dos elementos en su lado derecho mientras que el esquema de análisis **buE** está definido para cualquier LIG. Es fácil mostrar que $\mathbf{ECYK} \overset{\mathbf{ext}}{\Longrightarrow} \mathbf{buE}$ puesto que $\mathbf{ECYK}(\mathcal{L}) = \mathbf{buE}(\mathcal{L})$ es cierto para toda gramática lineal de índices ya que por definición $\mathbb{P}_{\mathrm{ECYK}} = \mathbb{P}_{\mathrm{buE}}$. □

La complejidad espacial con respecto a la cadena de entrada del algoritmo definido por el esquema de análisis **buE** es $\mathcal{O}(n^4)$ puesto que cada ítem almacena 4 posiciones de la cadena de entrada. La complejidad temporal con respecto a la cadena de entrada es $\mathcal{O}(n^6)$ y viene dada por los pasos $\mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$. Aunque dichos pasos manipulan 7 posiciones de la cadena de entrada, mediante aplicación parcial cada uno de ellos puede descomponerse en una secuencia de pasos, cada uno de ellos manipulando a lo sumo 6 posiciones con respecto a la cadena de entrada.

4.3. Algoritmo de tipo Earley sin la propiedad del prefijo válido

El algoritmo descrito por el esquema de análisis sintáctico **buE** es totalmente ascendente en el sentido de que 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 ítems generados mediante la utilización de predicción, que permite determinar qué producciones son candidatas a formar parte de la derivación atendiendo a la 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 $[\boldsymbol{A} \rightarrow \bullet\Upsilon, -, i, i \mid -, -, -]$ para todas las posibles $\boldsymbol{A} \rightarrow \Upsilon \in P$ y todas las posibles posiciones i y j de la cadena de entrada, se generarán únicamente aquellos ítems que involucren producciones cuyo esqueleto independiente del contexto sea relevante durante el proceso de análisis. Dicha tarea será encomendada al conjunto de pasos deductivos Pred.

En lo que respecta a los ítems, los del esquema **E** se definen como los del esquema **buE**.

Esquema de análisis sintáctico 4.3 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 $\mathcal{L}$ y una cadena de entrada $a_1 \ldots a_n$ se define como sigue:

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

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

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Scan}} = \mathcal{D}_{\mathrm{buE}}^{\mathrm{Scan}}$$

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

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\,]} = \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}[\,]}$$
$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]} = \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]}$$
$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ]} = \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}[\circ\circ][\circ\circ]}$$
$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]} = \mathcal{D}_{\mathrm{buE}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$$
$$\mathcal{D}_{\mathrm{E}} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Init}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Scan}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\,]} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ]} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$$
$$\mathcal{F}_{\mathrm{E}} = \mathcal{F}_{\mathrm{buE}}$$

§

Proposición 4.2 $\mathbf{buE} \stackrel{\mathbf{df}}{\Longrightarrow} \mathbf{E}$.

Demostración:

Para demostrar que el esquema de análisis sintáctico **E** es el resultado de aplicar un filtrado dinámico al esquema de análisis **buE**, debemos demostrar que $\mathcal{I}_{\mathrm{buE}} \supseteq \mathcal{I}_{\mathrm{E}}$ y que $\vdash_{\mathrm{buE}} \supseteq \vdash_{\mathrm{E}}$ para los sistemas de análisis sintáctico $\mathbb{P}_{\mathrm{buE}}$ y $\mathbb{P}_{\mathrm{E}}$. Lo primero es cierto por definición puesto que $\mathcal{I}_{\mathrm{buE}} = \mathcal{I}_{\mathrm{E}}$. Respecto a lo segundo es suficiente con mostrar que $\vdash_{\mathrm{buE}} \supseteq \mathcal{D}_{\mathrm{E}}$.

Los pasos deductivos en $\mathcal{D}_{\mathrm{E}}^{\mathrm{Scan}}$, $\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\,]}$, $\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]}$, $\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ]}$ y $\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$ son idénticos a sus homónimos del sistema de análisis sintáctico $\mathbb{P}_{\mathrm{buE}}$. El conjunto de pasos deductivos $\mathcal{D}_{\mathrm{E}}^{\mathrm{Init}}$ es un subconjunto de $\mathcal{D}_{\mathrm{bu}}^{\mathrm{Init}}$. Con respecto a los pasos deductivos en $\mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}}$ tenemos que dado un paso deductivo

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

existe un paso deductivo

$$\frac{}{[\boldsymbol{B} \rightarrow \bullet \Upsilon_3, -, j, j \mid -, -, -]} \in \mathcal{D}_{\mathrm{buE}}^{\mathrm{Pred}}$$

y por tanto existe la inferencia

$$[\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \boldsymbol{B}\ \Upsilon_2, \gamma, i, j \mid C, p, q] \vdash_{\mathrm{buE}} [\boldsymbol{B} \rightarrow \bullet \Upsilon_3, -, j, j \mid -, -, -]$$

□

El algoritmo descrito por el esquema de análisis sintáctico **E**, que mantiene una complejidad espacial $\mathcal{O}(n^4)$ y una complejidad temporal $\mathcal{O}(n^6)$ con respecto a la cadena de entrada, está muy relacionado con el algoritmo de tipo Earley descrito por Schabes y Shieber en [175] aunque este último 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 [175] 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. Lo que realmente ocurre es que dichos autores optaron, a la hora de definir la traducción de TAG a LIG, 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 la tabla 4.1 se muestran las producciones propuestas por Schabes y Shieber junto con su equivalente que utiliza el conjunto de nodos de la gramática de adjunción original como conjunto de no-terminales. Más precisamente, por cada nodo elemental η definimos dos no-terminales η^t y η^b.

Producción	Original	Equivalente
dom. inmediata (espina)	$b[\circ\circ\eta] \rightarrow t[\eta_1] \dots t[\circ\circ\eta_s] \dots t[\eta_n]$	$\eta^b[\circ\circ] \rightarrow \eta_1^t[\,] \dots \eta_s^t[\circ\circ] \dots \eta_n^t[\,]$
dom. inmediata (no espina)	$b[\eta] \rightarrow t[\eta_1] \dots t[\eta_n]$	$\eta^b[\,] \rightarrow \eta_1^t[\,] \dots \eta_n^t[\,]$
No adjunción	$t[\circ\circ\eta] \rightarrow b[\circ\circ\eta]$	$\eta^t[\circ\circ] \rightarrow \eta^b[\circ\circ]$
Adjunción predicativa	$t[\circ\circ\eta] \rightarrow t[\circ\circ\eta\eta_r]$	$\eta^t[\circ\circ] \rightarrow \eta_r^t[\circ\circ\eta]$
Adjunción de modificador	$b[\circ\circ\eta] \rightarrow t[\circ\circ\eta\eta_r]$	$\eta^b[\circ\circ] \rightarrow \eta_r^t[\circ\circ\eta]$
Pie	$b[\circ\circ\eta\eta_f] \rightarrow b[\circ\circ\eta]$	$\eta_f^b[\circ\circ\eta] \rightarrow \eta^b[\circ\circ]$
Sustitución	$t[\eta] \rightarrow t[n_r]$	$\eta^t[\,] \rightarrow n_r^t[\,]$

Tabla 4.1: Producciones propuestas por Schabes y Shieber

4.4. Algoritmos de tipo Earley con la propiedad del prefijo válido

Por la definición del formalismo de las gramáticas lineales de índices sabemos que una producción con $A[\circ\circ\gamma]$ como elemento del lado izquierdo sólo será útil en una derivación si se cumple la condición

$$S[\,] \stackrel{*}{\Rightarrow} w\, A[\alpha\gamma]\, \Upsilon$$

donde $\alpha \in V_I^*$ y $w \in V_T^*$. La comprobación de esta condición en los pasos Pred restringiría mucho más el número de ítems que contienen producciones con punto de la forma $\boldsymbol{A} \rightarrow \bullet\Upsilon$. Un resultado importante es que aquellos algoritmos que no verifiquen el cumplimiento de dicha condición en el momento de realizar la predicción no poseerán la propiedad del prefijo válido[1]. En consecuencia, tanto el algoritmo descrito por **E** como el algoritmo descrito por Schabes y Shieber no poseen dicha propiedad.

Para obtener un algoritmo de tipo Earley con la propiedad del prefijo válido es preciso modificar los pasos Pred de modo que 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

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

que representan invariablemente alguno de los siguientes tipos de derivaciones:

- $S[\,] \stackrel{*}{\Rightarrow} a_1 \dots a_h\, E[\alpha]\, \Upsilon_4 \stackrel{*}{\Rightarrow} a_1 \dots a_h \dots a_i\, A[\alpha\gamma]\, \Upsilon_3\Upsilon_4 \stackrel{*}{\Rightarrow} a_1 \dots a_h \dots a_i \dots a_p\, B[\alpha]\, a_{q+1} \dots a_j\, \Upsilon_2\Upsilon_3\Upsilon_4$ si y sólo si $(B, p, q) \neq (-, -, -)$, donde $A[\alpha\gamma]$ es un descendiente dependiente de $E[\alpha]$ y $B[\alpha]$ es un descendiente dependiente de $A[\alpha\gamma]$. 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.

- $S[\,] \stackrel{*}{\Rightarrow} a_1 \dots a_h\, E[\alpha]\, \Upsilon_4 \stackrel{*}{\Rightarrow} a_1 \dots a_h \dots a_i\ A[\alpha\gamma]\ \Upsilon_3\Upsilon_4 \stackrel{*}{\Rightarrow} a_1 \dots a_h \dots a_i \dots a_j\ \Upsilon_2\Upsilon_3\Upsilon_4$ si y sólo si $(E, h) \neq (-, -)$ y $(B, p, q) = (-, -, -)$, donde $A[\alpha\gamma]$ es un descendiente dependiente de $E[\alpha]$ y Υ_1 no contiene el hijo dependiente de $A[\alpha\gamma]$. Este tipo de derivación se refiere a la predicción del no-terminal A con una pila de índices no vacía.

[1] La propiedad del prefijo válido se discute en la sección 3.4.

- $S[\,] \stackrel{*}{\Rightarrow} a_1 \dots a_i A[\,]\Upsilon_4 \stackrel{*}{\Rightarrow} a_1 \dots a_i \dots a_j\, \Upsilon_2\Upsilon_4$ si y sólo si $(E,h) = (-,-)$, $\gamma = -$ y $(B,p,q) = (-,-,-)$. Si Υ_1 incluye al hijo dependiente de $A[\,]$ entones las pilas asociadas a $A[\,]$ y al hijo dependiente están vacías. Este tipo de derivación se refiere a la predicción o compleción del no-terminal A con una pila de índices vacía.

Observamos que el nuevo conjunto de ítems así definido es un refinamiento de los ítems del esquema **E**, de tal modo que el elemento γ se utiliza para almacenar la cima de la pila de índices predicha en el caso de que el ítem represente una predicción (recordemos que en el esquema **E** los ítems resultado de una predicción tenían $\gamma = -$). Por otra parte, 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 necesitaríamos almacenar una cuádrupla (E,η,h,k). Sin embargo, por la propiedad de independencia del contexto de LIG (definición 2.1, página 33) no es necesario almacenar η puesto que la derivación será válida independiente del resto de la pila de índices. Respecto al índice k, no es necesario puesto que al ser todos los ítems predichos en el esquema **E** de la forma $[\boldsymbol{A} \to \bullet\Upsilon, h, h \mid B, p, q]$, 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 de análisis sintáctico 4.4 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 $\mathcal{L}$ y una cadena de entrada $a_1 \dots a_n$ se define como sigue:

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

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

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

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

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

$$\mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ]}_{\text{Earley}_1} = \frac{[E,h \mid A[\circ\circ] \to \Upsilon_1 \bullet B[\circ\circ]\ \Upsilon_2, \gamma, i, j \mid -,-,-]}{[E,h \mid \boldsymbol{B} \to \bullet\Upsilon_3, \gamma, j, j \mid -,-,-]} \quad \begin{array}{l}\boldsymbol{B} \in \{B[\circ\circ\gamma], B[\circ\circ]\} \text{ sii } \gamma \neq - \\ \boldsymbol{B} = B[\,] \text{ sii } \gamma = -\end{array}$$

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

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

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

$$\mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ]}_{\text{Earley}_1} = \frac{[E,h \mid A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ]\ \Upsilon_2, \gamma, i, j \mid -, -, -], \quad [E,h \mid \boldsymbol{B} \rightarrow \Upsilon_3 \bullet, \gamma, j, k \mid C, p, q]}{[E,h \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ] \bullet \Upsilon_2, \gamma, i, k \mid C, p, q]}$$

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

$$\begin{aligned}\mathcal{D}_{\text{Earley}_1} = \ & \mathcal{D}^{\text{Init}}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Scan}}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}[\,]}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}[\circ\circ\gamma][\circ\circ]}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ]}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ\gamma]}_{\text{Earley}_1} \cup \\ & \mathcal{D}^{\text{Comp}[\,]}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}[\circ\circ\gamma][\circ\circ]}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ]}_{\text{Earley}_1} \cup \mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}_{\text{Earley}_1}\end{aligned}$$

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

§

Proposición 4.3 *El algoritmo representado por el esquema de análisis sintáctico* **Earley$_1$** *es correcto y completo.*

Demostración:

La demostración de la corrección se obtiene verificando mediante inducción en la longitud de las secuencias deductivas que los ítems siempre mantienen la invariante. El caso base viene definido por el conjunto de pasos deductivos Init, que trivialmente satisfacen la invariante. El paso de inducción se realiza verificando que, para cada paso deductivo, si se cumple la invariante para los ítems antecedentes entonces también se cumple para el ítem consecuente.

La completud se demuestra verificando que dada una derivación más a la izquierda de la gramática existe una secuencia deductiva en el esquema de análisis que representa dicha derivación. Para ello aplicamos inducción sobre la longitud de la derivación. Como caso base tenemos que una derivación de longitud 1 se corresponde con la aplicación de un paso Init donde $\Upsilon = \epsilon$. Como hipótesis de inducción suponemos que para toda derivación más a la izquierda de longitud l existe una secuencia deductiva que la representa. Como paso de inducción tenemos que la secuencia deductiva de toda derivación más a la izquierda de longitud $l+1$ se obtiene a partir de la secuencia deductiva correspondiente a la derivación de longitud l más un conjunto de pasos deductivos de tipo Pred y/o Scan y/o Comp. □

Proposición 4.4 *El algoritmo representado por el esquema de análisis sintáctico* **Earley$_1$** *satisface la propiedad del prefijo válido.*

Demostración:

La prueba de esta proposición es un corolario de la prueba de la corrección y completud: si un ítem $[E, h \mid \boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q]$ ha sido generado entonces es que existe una derivación $S[\,] \stackrel{*}{\Rightarrow} a_1 \ldots a_j \Upsilon_2 \Upsilon_3 \Upsilon_4$ y por tanto $S[\,] \stackrel{*}{\Rightarrow} a_1 \ldots a_j w$, donde w se obtiene de una derivación $\Upsilon_2 \Upsilon_3 \Upsilon_4 \stackrel{*}{\Rightarrow} w$. □

Proposición 4.5 $\mathbf{E} \stackrel{\mathbf{sr}}{\Longrightarrow} \mathbf{E}' \stackrel{\mathbf{ir}}{\Longrightarrow}\stackrel{\mathbf{df}}{\Longrightarrow} \mathbf{Earley_1}$.

Demostración:

Como primer paso definiremos el esquema de análisis sintáctico $\mathbf{E}'$ que se obtiene a partir de $\mathbf{E}$ rompiendo el conjunto de pasos deductivos $\mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}}$ en cuatro conjuntos $\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\,]}$, $\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ\gamma][\circ\circ]}$, $\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ][\circ\circ]}$y $\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ][\circ\circ\gamma]}$. El sistema de análisis correspondiente se describe a continuación:

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

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Init}} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Init}}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Scan}} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Scan}}$$

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

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ\gamma][\circ\circ]} = \frac{\begin{array}{l}[A[\circ\circ\gamma] \rightarrow \Upsilon_1 \bullet B[\circ\circ]\ \Upsilon_2, -, i, j \mid -, -, -], \\ [\boldsymbol{E} \rightarrow \bullet \Upsilon_3, -, h, h \mid -, -, -]\end{array}}{[\boldsymbol{B} \rightarrow \bullet \Upsilon_4, -, j, j \mid -, -, -]}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ][\circ\circ]} = \frac{[A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ]\ \Upsilon_2, -, i, j \mid -, -, -]}{[\boldsymbol{B} \rightarrow \bullet \Upsilon_3, -, j, j \mid -, -, -]}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ][\circ\circ\gamma]} = \frac{[A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ\gamma]\ \Upsilon_2, -, i, j \mid -, -, -]}{[\boldsymbol{B} \rightarrow \bullet \Upsilon_3, -, j, j \mid -, -, -]}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\,]} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\,]}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\circ\circ][\circ\circ]} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ]}$$

$$\mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$$

$$\mathcal{D}_{\mathrm{E}'} = \mathcal{D}_{\mathrm{E}'}^{\mathrm{Init}} \cup \mathcal{D}_{\mathrm{E}'}^{\mathrm{Scan}} \cup \mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}} \cup \mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\,]} \cup \mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\circ\circ\gamma][\circ\circ]} \cup \mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\circ\circ][\circ\circ]} \cup \mathcal{D}_{\mathrm{E}'}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$$

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

Las condiciones a verificar son que $\mathcal{I}_{\mathrm{E}} \subseteq \mathcal{I}_{\mathrm{E}'}$ y que $\stackrel{*}{\vdash}_{\mathrm{E}} \subseteq \stackrel{*}{\vdash}_{\mathrm{E}'}$. La primera condición se verifica por la propia definición de los ítems mientras que la segunda se obtiene fácilmente puesto que los diferentes pasos predicitvos lo único que hacen es especificar las diferentes formas que puede tener la producción que aparece en el ítem antecedente de cada uno de ellos. En el caso concreto de los pasos en $\mathcal{D}_{\mathrm{E}'}^{\mathrm{Pred}[\circ\circ\gamma][\circ\circ]}$, el segundo ítem antecedente es

totalmente redundante en este esquema de análisis puesto que su existencia se deriva de la existencia del primer ítem antecedente.

Para demostrar que el esquema de análisis $\mathbf{Earley_1}$ es derivable del esquema de análisis $\mathbf{E}'$ mediante refinamiento de los ítems y un filtro dinámico definiremos la siguiente función f de tal modo que

$$f([E,h \mid \boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid -,-,-]) = [\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2, -, i, j \mid B, p, q]$$

mientras que en cualquier otro caso

$$f([E,h \mid \boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q]) = [\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2, \gamma, i, j \mid B, p, q]$$

De la definición de f se obtiene directamente que $\mathcal{I}_{\mathrm{E}'} = f(\mathcal{I}_{\mathrm{Earley}_1})$ y que $\Delta_{\mathrm{E}'} = f(\Delta_{\mathrm{Earley}_1})$ por inducción en la longitud de las secuencias de derivación. En consecuencia, $\mathbb{P}_{\mathrm{E}'} \xrightarrow{\mathbf{ir}} \mathbb{P}_{\mathrm{Earley}_1}$, con lo que hemos probado lo que pretendíamos.

Con respecto al filtrado dinámico, se cumple que $\vdash_{\mathrm{E}'} \supseteq \mathcal{D}_{\mathrm{Earley}_1}$ por la propia definición de los pasos deductivos en $\mathbf{Earley_1}$. □

La complejidad espacial con respecto a la longitud n de la cadena de entrada del algoritmo descrito por el esquema de análisis $\mathbf{Earley_1}$ 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 $\mathcal{D}_{\mathrm{Earley}_1}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$. Los ítem involucrados en dichos pasos hacen referencia a 8 posiciones de la cadena de entrada. Mediante aplicación parcial podemos reducir la complejidad a $\mathcal{O}(n^7)$, que sin embargo resulta superior a la complejidad $\mathcal{O}(n^6)$ obtenida para los algoritmos presentados anteriormente. Para rebajar la complejidad temporal del algoritmo deberemos recurrir a una técnica más sofisticada, similar a la utilizada por Nederhof en [125] para reducir la complejidad de su algoritmo de tipo Earley con la propiedad del prefijo válido para el análisis de TAG. En el caso que nos ocupa, dividiremos cada paso deductivo de $\mathcal{D}_{\mathrm{Earley}_1}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]}$ en dos pasos de tal forma que la complejidad de cada uno de ellos sea a lo sumo $\mathcal{O}(n^6)$:

$$\mathcal{D}_{\mathrm{Earley}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]^0} = \frac{\begin{array}{l}[A,i \mid \boldsymbol{B} \rightarrow \Upsilon_3 \bullet, \gamma, j, k \mid C, p, q], \\ [E,h \mid \boldsymbol{C} \rightarrow \Upsilon_4 \bullet, \gamma', p, q \mid D, r, s]\end{array}}{[[\boldsymbol{B} \rightarrow \Upsilon_3 \bullet, \gamma, j, k \mid D, r, s]]}$$

$$\mathcal{D}_{\mathrm{Earley}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]^1} = \frac{\begin{array}{l}[[\boldsymbol{B} \rightarrow \Upsilon_3 \bullet, \gamma, j, k \mid D, r, s]], \\ [E,h \mid A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ\gamma]\ \Upsilon_2, \gamma', i, j \mid -,-,-], \\ [E,h \mid \boldsymbol{C} \rightarrow \Upsilon_4 \bullet, \gamma', p, q \mid D, r, s]\end{array}}{[E,h \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ\gamma] \bullet \Upsilon_2, \gamma', i, k \mid D, r, s]}$$

El primer paso generará un pseudo-ítem intermedio de la forma $[[\boldsymbol{B} \rightarrow \Upsilon_3 \bullet, \gamma, j, k \mid D, r, s]]$ que será utilizado como antecedente del segundo paso deductivo y que representa una derivación

$$B[\gamma'\gamma] \overset{*}{\Rightarrow} a_{j+1} \ldots a_p\, C[\gamma']\, a_{s+1} \ldots a_q \overset{*}{\Rightarrow} a_{j+1} \ldots a_p \ldots a_r\ D[\,]\ a_{s+1} \ldots a_q \ldots a_k$$

para algún γ', p y q. Los pasos de $\mathcal{D}_{\mathrm{Earley}}^{\mathrm{Comp}[\circ\circ][\circ\circ\gamma]^1}$ combinan dicho pseudo-ítem con el ítem $[E,h \mid A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ\gamma]\ \Upsilon_2, \gamma', i, j \mid -,-,-]$ que representa una derivación

$$S[\,] \overset{*}{\Rightarrow} a_1 \ldots a_h\, E[\alpha]\, \Upsilon_5 \overset{*}{\Rightarrow} a_1 \ldots a_h \ldots a_i\ A[\alpha\gamma']\ \Upsilon_3\Upsilon_5 \overset{*}{\Rightarrow} a_1 \ldots a_h \ldots a_i \ldots a_j\ B[\alpha\gamma'\gamma]\ \Upsilon_2\Upsilon_3\Upsilon_5$$

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

$$S[\,] \overset{*}{\Rightarrow} a_1 \ldots a_h\, E[\alpha]\, \Upsilon_5 \overset{*}{\Rightarrow} a_1 \ldots a_h \ldots a_p\ C[\alpha\gamma']\ \Upsilon_4\Upsilon_5 \overset{*}{\Rightarrow} a_1 \ldots a_h \ldots a_p \ldots a_r\ D[\alpha]\ a_{s+1} \ldots a_q\ \Upsilon_4\Upsilon_5$$

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

$$\begin{aligned} S[\,] &\stackrel{*}{\Rightarrow} a_1 \ldots a_h\, E[\alpha]\, \Upsilon_5 \\ &\stackrel{*}{\Rightarrow} a_1 \ldots a_h \ldots a_i\, A[\alpha\gamma']\, \Upsilon_3\Upsilon_5 \\ &\stackrel{*}{\Rightarrow} a_1 \ldots a_h \ldots a_i \ldots a_j\, B[\alpha\gamma'\gamma]\, \Upsilon_2\Upsilon_3\Upsilon_5 \\ &\stackrel{*}{\Rightarrow} a_1 \ldots a_h \ldots a_i \ldots a_j \ldots a_p\, C[\alpha\gamma']\, a_{q+1} \ldots a_k\, \Upsilon_2\Upsilon_3\Upsilon_5 \\ &\stackrel{*}{\Rightarrow} a_1 \ldots a_h \ldots a_i \ldots a_j \ldots a_p \ldots a_r\, D[\alpha]\, a_{s+1} \ldots a_{q+1} \ldots a_k\, \Upsilon_2\Upsilon_3\Upsilon_5 \end{aligned}$$

Esquema de análisis sintáctico 4.5 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 $\mathcal{L}$ y una cadena de entrada $a_1 \ldots a_n$ se define como sigue:

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

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

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

$$\mathcal{D}^{\text{Init}}_{\text{Earley}} = \mathcal{D}^{\text{Init}}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Scan}}_{\text{Earley}} = \mathcal{D}^{\text{Scan}}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Pred}[\,]}_{\text{Earley}} = \mathcal{D}^{\text{Pred}[\,]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Pred}[\circ\circ\gamma][\circ\circ]}_{\text{Earley}} = \mathcal{D}^{\text{Pred}[\circ\circ\gamma][\circ\circ]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ]}_{\text{Earley}} = \mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ\gamma]}_{\text{Earley}} = \mathcal{D}^{\text{Pred}[\circ\circ][\circ\circ\gamma]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Comp}[\,]}_{\text{Earley}} = \mathcal{D}^{\text{Comp}[\,]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Comp}[\circ\circ\gamma][\circ\circ]}_{\text{Earley}} = \mathcal{D}^{\text{Comp}[\circ\circ\gamma][\circ\circ]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ]}_{\text{Earley}} = \mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ]}_{\text{Earley}_1}$$

$$\mathcal{D}^{\text{Comp}[\circ\circ][\circ\circ\gamma]^0}_{\text{Earley}} = \frac{\begin{array}{c}[A, i \mid \boldsymbol{B} \rightarrow \Upsilon_3\bullet\ , \gamma, j, k \mid C, p, q], \\ [E, h \mid \boldsymbol{C} \rightarrow \Upsilon_4\bullet, \gamma', p, q \mid D, r, s]\end{array}}{[[\boldsymbol{B} \rightarrow \Upsilon_3\bullet\ , \gamma, j, k \mid D, r, s]]}$$

$$\mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]^1} = \frac{\begin{array}{l}[[\boldsymbol{B} \rightarrow \Upsilon_3 \bullet\ , \gamma, j, k \mid D, r, s]],\\ [E, h \mid A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ\gamma]\ \Upsilon_2\ , \gamma', i, j \mid -, -, -],\\ [E, h \mid \boldsymbol{C} \rightarrow \Upsilon_4 \bullet, \gamma', p, q \mid D, r, s]\end{array}}{[E, h \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ\gamma] \bullet \Upsilon_2, \gamma', i, k \mid D, r, s]}$$

$$\begin{aligned}\mathcal{D}_{\text{Earley}} = \quad & \mathcal{D}_{\text{Earley}}^{\text{Init}} \cup \mathcal{D}_{\text{Earley}}^{\text{Scan}} \cup \mathcal{D}_{\text{Earley}}^{\text{Pred}[\,]} \cup \mathcal{D}_{\text{Earley}}^{\text{Pred}[\circ\circ\gamma][\circ\circ]} \cup \mathcal{D}_{\text{Earley}}^{\text{Pred}[\circ\circ][\circ\circ]} \cup \mathcal{D}_{\text{Earley}}^{\text{Pred}[\circ\circ][\circ\circ\gamma]} \cup \\ & \mathcal{D}_{\text{Earley}}^{\text{Comp}[\,]} \cup \mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ\gamma][\circ\circ]} \cup \mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ][\circ\circ]} \cup \mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]^0} \cup \mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]^1}\end{aligned}$$

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

§

Proposición 4.6 $\mathbf{Earley_1} \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{Earley}$.

Demostración:

Para demostrar que el esquema de análisis **Earley** puede ser obtenido mediante un refinamiento de los pasos deductivos del esquema $\mathbf{Earley_1}$ debemos probar que para todo sistema de análisis $\mathbb{P}_{\text{Earley}_1}$ y $\mathbb{P}_{\text{Earley}}$ se cumple que $\mathbb{P}_{\text{Earley}_1} \overset{\mathbf{sr}}{\Longrightarrow} \mathbb{P}_{\text{Earley}}$. Ello conlleva demostrar que $\mathcal{I}_{\text{Earley}_1} \subseteq \mathcal{I}_{\text{Earley}}$ y que $\overset{*}{\vdash}_{\text{Earley}_1} \subseteq \overset{*}{\vdash}_{\text{Earley}}$. Lo primero se obtiene directamente puesto que $\mathcal{I}_{\text{Earley}_1} = \mathcal{I}_{\text{Earley}}$ por definición de los sistemas de análisis. Lo segundo se obtiene demostrando que $\mathcal{D}_{\text{Earley}_1} \subseteq \overset{*}{\vdash}_{\text{Earley}}$. El únicos conjunto de pasos deductivos de $\mathbb{P}_{\text{Earley}_1}$ que no se han incorporado directamente en $\mathbb{P}_{\text{Earley}}$ es $\mathcal{D}_{\text{Earley}_1}^{\text{Comp}[\circ\circ][\circ\circ\gamma]}$ pero todo paso perteneciente a ese conjunto es equivalente a la la aplicación de un paso deductivo de $\mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]^0}$ seguido de un paso deductivo del conjunto $\mathcal{D}_{\text{Earley}}^{\text{Comp}[\circ\circ][\circ\circ\gamma]^1}$:

$$\frac{[A, i \mid \boldsymbol{B} \rightarrow \Upsilon_3 \bullet\ , \gamma, j, k \mid C, p, q],\ [E, h \mid \boldsymbol{C} \rightarrow \Upsilon_4 \bullet, \gamma', p, q \mid D, r, s]}{[[\boldsymbol{B} \rightarrow \Upsilon_3 \bullet\ , \gamma, j, k \mid D, r, s]]}$$

$$\frac{\begin{array}{l}[[\boldsymbol{B} \rightarrow \Upsilon_3 \bullet\ , \gamma, j, k \mid D, r, s]],\\ [E, h \mid A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ\gamma]\ \Upsilon_2\ , \gamma', i, j \mid -, -, -],\\ [E, h \mid \boldsymbol{C} \rightarrow \Upsilon_4 \bullet, \gamma', p, q \mid D, r, s]\end{array}}{[E, h \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ\gamma] \bullet \Upsilon_2, \gamma', i, k \mid D, r, s]}$$

□

4.5. El bosque de análisis

Los algoritmos anteriores tal y como han sido descritos son realmente reconocedores y no analizadores sintácticos puesto que no construyen una representación de los árboles derivados o *bosque compartido de análisis sintáctico*, que debe satisfacer las siguientes propiedades [34]:

1. Debe contener en forma compacta todos los árboles de análisis. En particular, su tamaño debe ser de orden polinomial con respecto a la longitud de la cadena de entrada.

2. La recuperación de cada árbol de análisis individual debe poder realizarse en tiempo lineal con respecto al tamaño del bosque de análisis.

Billot y Lang [30] definen el bosque compartido correspondiente a una gramática independiente del contexto $\mathcal{G} = (V_T, V_N, P, S)$ y a una cadena de entrada $a_1 \dots a_n$ como una gramática independiente del contexto en la cual los no-terminales son de la forma $\langle A, i, j\rangle$, donde $A \in V_N$ y $i, j \in 0..n$, y las producciones presentan al forma

$$\langle A_0, j_0, j_m\rangle \rightarrow w_0\ \langle A_1, j_0, j_1\rangle\ w_1\ \langle A_2, j_1, j_2\rangle\ \dots\ w_{m-1}\ \langle A_m, j_{m-1}, j_m\rangle\ w_m$$

donde $A_0 \rightarrow w_0 A_1 w_1 A_2 \dots w_{m-1} A_m w_m \in P$ y $w_i \in V_T^*$, mediante las cuales se expresa que A_0 reconoce la subcadena $a_{j_0+1} \dots a_{j_m}$ mediante la aplicación de la producción $A_0 \rightarrow w_0 A_1 w_1 A_2 \dots w_{m-1} A_m w_m$ y donde cada A_i reconoce una subcadena $a_{j_{i-1}+1} \dots a_{j_i}$.

Podemos extender el concepto de bosque de análisis compartido para definir el concepto de *bosque de LIG* (*LIGed forest*) [215, 31]. Dado el bosque compartido de la gramática independiente del contexto que constituye el esqueleto de una LIG, siempre que una producción LIG $A_0[\circ\circ\gamma] \rightarrow A_1[\,] \dots A_d[\circ\circ\gamma'] \dots A_m[\,]$ participe en una derivación se añadirá la producción

$$\langle A_0, j_0, j_m\rangle[\circ\circ\gamma] \rightarrow \langle A_1, j_0, j_1\rangle\ \dots\ \langle A_d, j_{d-1}, j_d\rangle[\circ\circ\gamma']\ \dots\ \langle A_m, j_{m-1}, j_m\rangle$$

al bosque de LIG para indicar que A_0 reconoce la subcadena $a_{j_0+1} \dots a_{j_m}$ mediante la aplicación de la producción $A_0[\circ\circ\gamma] \rightarrow A_1 \dots A_d[\circ\circ\gamma'] \dots A_m$ donde cada A_i reconoce una subcadena $a_{j_{i-1}+1} \dots a_{j_i}$ y la pila de índices es transmitida de A_0 a A_d pero reemplazando la cima γ por γ'. Desgraciadamente, la gramática lineal de índices construida de esta manera no satisface una de las propiedades que debe poseer un bosque de análisis compartido, puesto que no es posible extraer cada uno de los árboles de análisis individuales en tiempo lineal con respecto al tamaño del bosque. Vijay-Shanker y Weir [215] intentan resolver este problema mediante la definición de un autómata finito no-determinista que se encarga de comprobar si un símbolo $\langle A, i, j\rangle[\alpha]$ del bosque deriva una cadena de terminales. Nederhof define en [126] un autómata finito similar al de Vijay-Shanker y Weir junto con un método de transformación gramatical, de tal modo que la gramática lineal de índices obtenida garantiza que todos los símbolos LIG que puedan aparecer en una derivación derivan a su vez una cadena de terminales.

Los enfoques anteriores siguen el criterio expuesto por Lang en [108] de considerar que la salida de un analizador sintáctico debe ser la intersección entre la gramática de entrada y la cadena a analizar. Boullier presenta en [32] un enfoque alternativo según el cual el bosque de análisis compartido para una LIG $\mathcal{L} = (V_T, V_N, V_I, P, S)$ y una cadena de entrada w se define mediante una *gramática de derivación lineal*, una gramática independiente del contexto que reconoce el lenguaje definido por las secuencias de producciones LIG de $\mathcal{L}$ que podrían ser utilizadas para derivar w. Con anterioridad a la construcción de la gramática de derivación lineal es preciso obtener el cierre transitivo de una serie de relaciones sobre $V_N \times V_N$.

Con el fin de evitar el uso de estructuras de datos adicionales, tales como máquinas de estado finito o relaciones precalculadas, nos hemos inspirado en la utilización de gramáticas independientes del contexto como representación del bosque de análisis compartido para gramáticas de adjunción de árboles [215] para tratar de capturar la independencia al contexto de la aplicación de producciones en el caso de LIG. Dada una gramática lineal de índices $\mathcal{L} = (V_T, V_N, V_I, P, S)$ y una cadena de entrada $w = a_1 \dots a_n$, el bosque compartido para $\mathcal{L}$ y w es una gramática independiente del contexto $\mathcal{L}^w = (V_T, V_N^w, P^w, S^w)$ en la que los elementos en V_N^w tienen la forma $\langle A, \gamma, i, j, B, p, q\rangle$, donde $A, B \in V_N$, $\gamma \in V_I$ y $i, j, p, q \in 0 \dots n$. El axioma S^w es el no-terminal $\langle S, -, 0, n, -, -, -, \rangle$. Las producciones en P^w se construyen tal y como se muestra a continuación:

Caso 1a: si $A[\,] \Rightarrow a_j$ entonces se añade la producción

$$\langle A, -, j-1, j, -, -, - \rangle \rightarrow a_j$$

Caso 1b: Si $A[\,] \Rightarrow \epsilon$ entonces se añade la producción

$$\langle A, -, j, j, -, -, - \rangle \rightarrow \epsilon$$

Caso 2a: Si $A[\circ\circ\gamma] \rightarrow B[\,]\, C[\circ\circ] \in P$, $B[\,] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_k$ y $C[\alpha\eta] \stackrel{*}{\Rightarrow} a_{k+1} \ldots a_p\, D[\alpha]\, a_{q+1} \ldots a_j$ entonces se añade la producción

$$\langle A, \gamma, i, j, C, k, j \rangle \rightarrow \langle B, -, i, k, -, -, - \rangle\ \langle C, \eta, k, j, D, p, q \rangle$$

Caso 2b: Si $A[\circ\circ\gamma] \rightarrow B[\circ\circ]\, C[\,] \in P$, $B[\alpha\eta] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p\, D[\alpha]\, a_{q+1} \ldots a_k$ y $C[\,] \stackrel{*}{\Rightarrow} a_{k+1} \ldots a_j$ entonces se añade la producción

$$\langle A, \gamma, i, j, B, i, k \rangle \rightarrow \langle B, \eta, i, k, D, p, q \rangle\ \langle C, -, k, j, -, -, - \rangle$$

Caso 3a: Si $A[\circ\circ] \rightarrow B[\,]\, C[\circ\circ] \in P$, $B[\,] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_k$ y $C[\alpha\eta] \stackrel{*}{\Rightarrow} a_{k+1} \ldots a_p\, D[\alpha]\, a_{q+1} \ldots a_j$ entonces se añade la producción

$$\langle A, \eta, i, j, D, p, q \rangle \rightarrow \langle B, -, i, k, -, -, - \rangle\ \langle C, \eta, k, j, D, p, q \rangle$$

Caso 3b: Si $A[\circ\circ] \rightarrow B[\circ\circ]\, C[\,] \in P$, $B[\alpha\eta] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p\, D[\alpha]\, a_{q+1} \ldots a_k$ y $C[\,] \stackrel{*}{\Rightarrow} a_{k+1} \ldots a_j$ entonces se añade la producción

$$\langle A, \eta, i, j, D, p, q \rangle \rightarrow \langle B, \eta, i, k, D, p, q \rangle\ \langle C, -, k, j, -, -, - \rangle$$

Caso 4a: Si $A[\circ\circ] \rightarrow B[\,]\, C[\circ\circ\gamma] \in P$, $B[\,] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_k$, $C[\alpha\eta\gamma] \stackrel{*}{\Rightarrow} a_{k+1} \ldots a_p\, D[\alpha\eta]\, a_{q+1} \ldots a_j$ y $D[\alpha\eta] \stackrel{*}{\Rightarrow} a_{p+1} \ldots a_r\, E[\alpha]\, a_{s+1} \ldots a_q$ entonces se añade la producción

$$\langle A, \eta, i, j, E, r, s \rangle \rightarrow \langle B, -, i, k, -, -, - \rangle\ \langle C, \gamma, k, j, D, p, q \rangle\ \langle D, \eta, p, q, E, r, s \rangle$$

donde las derivaciones que comienzan en $\langle D, \eta, p, q, E, r, s \rangle$ posibilitan la recuperación de la parte restante de la pila de índices perteneciente a A.

Caso 4b: Si $A[\circ\circ] \rightarrow B[\circ\circ\gamma]\, C[\,] \in P$, $B[\alpha\eta\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p\, D[\alpha\eta]\, a_{q+1} \ldots a_k$, $C[\,] \stackrel{*}{\Rightarrow} a_{k+1} \ldots a_j$ y $D[\alpha\eta] \stackrel{*}{\Rightarrow} a_{p+1} \ldots a_r\, E[\alpha]\, a_{s+1} \ldots a_q$ entonces se añade la producción

$$\langle A, \eta, i, j, E, r, s \rangle \rightarrow \langle B, \gamma, i, k, D, p, q \rangle\ \langle C, -, k, j, -, -, - \rangle\ \langle D, \eta, p, q, E, r, s \rangle$$

donde las derivaciones que comienzan en $\langle D, \eta, p, q, E, r, s \rangle$ posibilitan la recuperación de la parte restante de la pila de índices perteneciente a A.

Caso 5: Si $A[\circ\circ\gamma] \rightarrow B[\circ\circ] \in P$ y $B[\alpha\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p\, D[\alpha]\, a_{q+1} \ldots a_j$ entonces se añade la producción

$$\langle A, \gamma, i, j, B, i, j \rangle \rightarrow \langle B, \eta, i, j, D, p, q \rangle$$

Caso 6: Si $A[\circ\circ] \rightarrow B[\circ\circ] \in P$ y $B[\alpha\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p\, D[\alpha]\, a_{q+1} \ldots a_j$ entonces se añade la producción

$$\langle A, \gamma, i, j, D, p, q \rangle \rightarrow \langle B, \gamma, i, j, D, p, q \rangle$$

Caso 7: Si $A[\circ\circ] \rightarrow B[\circ\circ\gamma] \in P$, $B[\alpha\eta\gamma] \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_p\, D[\alpha\eta]\, a_{q+1} \ldots a_j$ y $D[\alpha\eta] \stackrel{*}{\Rightarrow} a_{p+1} \ldots a_r\, E[\alpha]\, a_{s+1} \ldots a_q$ entonces se añade la producción

$$\langle A, \eta, i, j, E, r, s\rangle \rightarrow \langle B, \gamma, i, j, D, p, q\rangle\ \langle D, \eta, p, q, E, r, s\rangle$$

donde las derivaciones que comienzan en $\langle D, \eta, p, q, E, r, s\rangle$ posibilitan la recuperación de la parte restante de la pila de índices perteneciente a A.

Es importante reseñar que estamos asumiendo gramáticas lineales de índices cuyas producciones tienen a lo sumo dos elementos en el lado derecho. Este hecho no representa un problema puesto que cualquier producción LIG $A_0[\circ\circ\gamma] \rightarrow A_1 \ldots a_d[\circ\circ\gamma'] \ldots A_m[\,]$ puede ser implícitamente binarizada en un conjunto de producciones LIG

$$\begin{gathered}
\nabla_0[\,] \rightarrow \epsilon \\
\nabla_1[\circ\circ] \rightarrow \nabla_0[\circ\circ]\ A_1[\,] \\
\vdots \\
\nabla_{d-1}[\circ\circ] \rightarrow \nabla_{d-2}[\circ\circ]\ A_{d-1}[\,] \\
\nabla_d[\circ\circ\gamma] \rightarrow \nabla_{d-1}[\,]\ A_d[\circ\circ\gamma'] \\
\nabla_{d+1}[\circ\circ] \rightarrow \nabla_d[\circ\circ]\ A_{d+1}[\,] \\
\vdots \\
\nabla_m[\circ\circ] \rightarrow \nabla_{m-1}[\circ\circ]\ A_m[\,] \\
A_0[\circ\circ] \rightarrow \nabla_m[\circ\circ]
\end{gathered}$$

donde los ∇_i son símbolos no-terminales nuevos que representan el reconocimiento parcial de la producción original. De hecho, un símbolo ∇_i es equivalente a una producción con punto que tenga el punto situado justo antes del no-terminal A_{i+1} ó con el punto al final del lado derecho en el caso de ∇_m.

Existe una correspondencia directa entre el caso 1 y los pasos de tipo Scan del esquema de análisis sintáctico **CYK**, y entre los casos 2, 3 y 4 y el resto de pasos del mismo esquema. Existe también una correspondencia similar para los demás esquemas de análisis, donde el caso 1 se corresponde con los pasos de tipo Scan y los demás casos se corresponden con los diferentes pasos de compleción.

Es interesante señalar que el conjunto de no-terminales es un subconjunto del conjunto de ítems de los esquemas de análisis **CYK**, **buE** y **E**. El caso del esquema de análisis **Earley** es ligeramente diferente, puesto que cada no-terminal $\langle A, \gamma, i, j, B, p, q\rangle$ representa la clase de ítems $[E, h \mid A, \gamma, i, j \mid D, p, q]$ para cualquier valor de E y de h.

Al igual que ocurre cuando se utilizan gramáticas independientes del contexto para representar el bosque de análisis compartido en el caso de TAG [215], las derivaciones de $\mathcal{L}^w$ codifican las derivaciones de la cadena de entrada w según $\mathcal{L}$ pero el conjunto específico de cadenas terminales generadas por $\mathcal{L}^w$ no es relevante. Lo importante es que $L(\mathcal{L}^w)$, el lenguaje generado por $\mathcal{L}^w$, es no vacío si y sólo si w pertenece a $L(\mathcal{L})$, el lenguaje generado por $\mathcal{L}$. Si podamos $\mathcal{L}^w$ reteniendo los símbolos útiles de tal modo que podamos garantizar que cada no-terminal genera una cadena terminal, las derivaciones de w en la LIG original pueden ser obtenidas a partir de las derivaciones de w en $\mathcal{L}^w$.

El número de posibles producciones en $\mathcal{L}^w$ es $\mathcal{O}(n^7)$. Esta complejidad puede ser reducida a $\mathcal{O}(n^6)$ mediante la transformación de las producciones que presentan la forma $A[\circ\circ] \rightarrow$

$B[\,]\ C[\circ\circ\gamma]$ en dos producciones

$$A[\circ\circ] \rightarrow B[\,]\ C^{\gamma}[\circ\circ]$$
$$C^{\gamma}[\circ\circ] \rightarrow C[\circ\circ\gamma]$$

donde C^{γ} es un nuevo no-terminal. De modo análogo, las producciones $A[\circ\circ] \rightarrow B[\circ\circ\gamma]\ C[\,]$ se pueden transformar en

$$A[\circ\circ] \rightarrow B^{\gamma}[\circ\circ]\ C[\,]$$
$$B^{\gamma}[\circ\circ] \rightarrow B[\circ\circ\gamma]$$

donde B^{γ} es un nuevo no-terminal.

4.6. Comparación entre los algoritmos de análisis sintáctico para LIG y TAG

Podemos apreciar que existe una gran similitud entre los esquemas de análisis **CYK**, **buE** y **E** definidos para gramáticas lineales de índices en este capítulo y los esquemas homónimos definidos para gramáticas de adjunción de árboles en el capítulo 3, así como entre el esquema **Earley** para LIG y el esquema **Nederhof** para TAG.

En lo que respecta a los ítems tenemos que en ambos casos se almacenan las mismas posiciones de la cadena de entrada y una producción con punto. Las diferencias surgen en aquella información referida al esqueleto independiente del contexto que es preciso guardar.

En las gramáticas de adjunción todo nodo forma parte de un árbol elemental y por lo tanto conocemos directamente el nodo raíz y el nodo pie del árbol al que pertenece. La operación de adjunción de un árbol β en un nodo N^{γ} se traduce en que el análisis continúa en la raíz del árbol β, propagando a través de la espina la pila de adjunciones no terminadas, la más reciente de las cuales es la que ha sido realizada sobre N^{γ}. El árbol γ será retomado en dicho nodo cuando se alcance el nodo pie de β. En consecuencia, un ítem representando el estado del proceso de análisis en un nodo arbitrario M^{β} tendría la forma siguiente:

$$[M^{\beta} \rightarrow \delta \bullet \nu, N^{\gamma}, i, j \mid \mathbf{F}^{\beta}, p, q]$$

Vemos que la información proporcionada por la aparición de $\mathbf{F}^{\beta}$ es redundante y puede ser eliminada. Aunque aparentemente la información proporcionada N^{γ} es necesaria, podemos ver que es redundante si consideramos que en lugar de N^{γ} podría aparecer cualquier otro nodo de cualquier otro árbol elemental que admita la adjunción de β. Si eliminamos N^{γ} de los ítems ganaremos en compartición y reduciremos la complejidad con respecto al tamaño de la gramática, que consideramos constante. Efectivamente, la desaparición de dicho elemento se suple por la adición de condiciones de aplicación que comprueban que se cumplan las restricciones de adjunción.

Con respecto a los ítems utilizados en aquellos algoritmos de análisis sintáctico de TAG que preservan la propiedad del prefijo válido, tras las consideraciones anteriores tendrían que tener la forma

$$[\mathbf{R}^{\beta}, h \mid M^{\beta} \rightarrow \delta \bullet \nu, i, j \mid p, q]$$

donde observamos que $\mathbf{R}^{\beta}$ es redundante, puesto que el nodo padre del árbol β es siempre conocido.

Con respecto a los pasos deductivos, la tabla 4.2 muestra la equivalencia entre los pasos utilizados por los algoritmos para el análisis de LIG y aquellos utilizados para el análisis de TAG. Vemos que la mayor diferencia radica en que en el caso de LIG debemos diferenciar entre aquellas predicciones y compleciones del esqueleto independiente del contexto que se realizan fuera de la espina y aquellas que se realizan en la espina. Esto se debe a que en el caso de LIG

TAG	LIG
Init	Init
Scan	Scan
Pred (fuera de la espina)	Pred[ ]
Comp (fuera de la espina)	Comp[ ]
Pred (en la espina)	Pred[$\circ\circ$][$\circ\circ$]
Comp (en la espina)	Comp[$\circ\circ$][$\circ\circ$]
AdjPred	Pred[$\circ\circ$][$\circ\circ\gamma$]
FootPred	Pred[$\circ\circ\gamma$][$\circ\circ$]
FootComp	Comp[$\circ\circ\gamma$][$\circ\circ$]
AdjComp	Comp[$\circ\circ$][$\circ\circ\gamma$]

Tabla 4.2: Correspondencia de los pasos deductivos para TAG y LIG

la pila de índices debe propagarse explícitamente a través de la espina, mientras que en el caso de TAG la pila de adjunciones es propagada implícitamente a través de la espina y por eso no es necesario diferenciar ambos casos.

Como caso especialmente interesante tenemos el algoritmo para el análisis sintáctico de LIG definido por el esquema **Earley** y el algoritmo para el análisis sintáctico de TAG definido por el esquema **Nederhof**. En ambos casos se implementa una estrategia que aplica predicción tanto sobre el esqueleto independiente del contexto como sobre la pila de índices o adjunciones. En ambos casos se obtiene inicialmente un algoritmo con complejidad superior a $\mathcal{O}(n^6)$ pero se reduce a esta última mediante la división del paso más complejo en una serie de pasos que se combinan entre sí. El interés de estos algoritmos viene determinado porque si bien **Nederhof** representa el algoritmo para el análisis de TAG tal y como fue descrito por Nederhof [125], la estrategia adoptada en el esquema **Earley** para LIG muestra un carácter más general, aplicable no sólo al caso de LIG sino también a la tabulación de diversos modelos e autómatas para esta clase de lenguajes [53, 14] y a la tabulación de algoritmos para TAG. Aplicada a este caso, el conjunto de pasos $\mathcal{D}^{\text{AdjComp}}_{\text{Nederhof}}$ se descompondría en:

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

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

a partir de los cuales podemos definir un nuevo esquema de análisis sintáctico $\mathbb{P}_{\text{Nederhof}'}$ para TAG.

Esquema de análisis sintáctico 4.6 El sistema de análisis $\mathbb{P}_{\text{Nederhof}'}$ que se corresponde con una variación el algoritmo de análisis sintáctico de tipo Earley para TAG propuesto por Nederhof, dada una gramática de adjunción de árboles $\mathcal{T}$ y una cadena de entrada $A_1 \dots a_n$ se define como sigue:

$$\mathcal{I}_{\text{Nederhof}'} = \mathcal{I}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{Init}}_{\text{Nederhof}'} = \mathcal{D}^{\text{Init}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{Scan}}_{\text{Nederhof}'} = \mathcal{D}^{\text{Scan}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{Pred}}_{\text{Nederhof}'} = \mathcal{D}^{\text{Pred}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{Comp}}_{\text{Nederhof}'} = \mathcal{D}^{\text{Comp}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{AdjPred}}_{\text{Nederhof}'} = \mathcal{D}^{\text{AdjPred}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{FootPred}}_{\text{Nederhof}'} = \mathcal{D}^{\text{FootPred}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{FootComp}}_{\text{Nederhof}'} = \mathcal{D}^{\text{FootComp}}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{AdjComp}^0}_{\text{Nederhof}'} = \mathcal{D}^{\text{AdjComp}^0}_{\text{Nederhof}}$$

$$\mathcal{D}^{\text{AdjComp}^1}_{\text{Nederhof}'} = \frac{\begin{array}{c}[[M^\gamma \rightarrow \upsilon\bullet, j, m \mid p, q]],\\ [h, M^\gamma \rightarrow \upsilon\bullet, k, l \mid p, q],\\ [h, N^\gamma \rightarrow \delta \bullet M^\gamma\nu, i, j \mid p', q']\end{array}}{[h, N^\gamma \rightarrow \delta M^\gamma \bullet \nu, i, m \mid p \cup p', q \cup q']} \quad \beta \in \text{adj}(M^\gamma)$$

$$\begin{aligned}\mathcal{D}_{\text{Nederhof}'} = \; & \mathcal{D}^{\text{Init}}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{Scan}}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{Pred}}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{Comp}}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{AdjPred}}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{FootPred}}_{\text{Nederhof}'} \\ & \cup \mathcal{D}^{\text{FootComp}}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{AdjComp}^0}_{\text{Nederhof}'} \cup \mathcal{D}^{\text{AdjComp}^1}_{\text{Nederhof}'}\end{aligned}$$

$$\mathcal{F}_{\text{Nederhof}'} = \mathcal{F}_{\text{Nederhof}}$$

§

Proposición 4.7 $\mathbf{E} \overset{\mathbf{df}}{\Longrightarrow} \mathbf{Ear}' \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{Earley}' \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{Nederhof}'$.

4.7. Otros algoritmos de análisis sintáctico para LIG

A diferencia del caso de las gramáticas de adjunción de árboles, donde encontramos un elevado número de algoritmos diferentes, en el caso de las gramáticas lineales de índices prácticamente todos los algoritmos de análisis sintáctico descritos en la literatura son variaciones de los algoritmos CYK, con la salvedad de los algoritmos que se describen a continuación.

4.7.1. Algoritmo bidireccional

Schneider define en [181, 182] una extensión del algoritmo de tipo *head-corner* para gramáticas independientes del contexto presentado por Sikkel en [189]. El resultado es un algoritmo bidireccional que procede ascendentemente guiado por los hijos dependientes de las producciones de la gramática. Los resultados intermedios del proceso de análisis se almacenan en ítems con producciones anotadas por dos puntos, de la forma

$$[\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2 \bullet \Upsilon_3, \gamma, i, j \mid B, p, q]$$

que representan los siguientes tipos de derivaciones:

- $\Upsilon_2 \overset{*}{\Rightarrow} \Upsilon_{2'}\ E[\gamma]\ \Upsilon_{2''} \overset{*}{\Rightarrow} a_{i+1} \ldots a_p\ B[\,]\ a_{q+1} \ldots a_j$ si y sólo si $(B, p, q) \neq (-, -, -)$ y donde $B[\,]$ es un descendiente dependiente de $E[\gamma]$.
- $\Upsilon_2 \overset{*}{\Rightarrow} a_{i+1} \ldots a_j$ si y sólo si $\gamma = -$ y $(B, p, q) = (-, -, -)$.

La cadena de entrada $a_1 \ldots a_n$ ha sido reconocida cuando se obtiene un ítem de la forma $[\boldsymbol{S} \rightarrow \bullet\Upsilon\bullet\ , -, 0, n \mid -, -, -]$. A continuación mostramos el esquema de análisis correspondiente a este algoritmo.

Esquema de análisis sintáctico 4.7 El sistema de análisis $\mathbb{P}_{\text{HC}}$ que se corresponde con el algoritmo de análisis bidireccional ascendente para una gramática lineal de índices $\mathcal{L}$ y una cadena de entrada $a_1 \ldots a_n$ se define como sigue:

$$\mathcal{I}_{\text{HC}} = \left\{ [\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2 \bullet \Upsilon_3, \gamma, i, j \mid B, p, q] \mid \begin{array}{l} \boldsymbol{A} \rightarrow \Upsilon_1\Upsilon_2\Upsilon_3 \in P,\ B \in V_N,\ \gamma \in V_I, \\ 0 \leq i \leq j\ ,\ \ (p, q) \leq (i, j) \end{array} \right\}$$

$$\mathcal{D}_{\text{HC}}^{\text{Scan}} = \frac{[a, j, j+1]}{[A[\,] \rightarrow \bullet a \bullet, -, j, j+1 \mid -, -, -]}$$

$$\mathcal{D}_{\text{HC}}^{\text{LComp}[\,]} = \frac{\begin{array}{l}[\boldsymbol{A} \rightarrow \Upsilon_1\ B[\,] \bullet \Upsilon_2 \bullet \Upsilon_3, \gamma, k, j \mid C, p, q], \\ [\boldsymbol{B} \rightarrow \bullet\Upsilon_4\bullet, -, i, k \mid -, -, -]\end{array}}{[\boldsymbol{A} \rightarrow \Upsilon_1 \bullet B[\,]\ \Upsilon_2 \bullet \Upsilon_3, \gamma, i, j \mid C, p, q]}$$

$$\mathcal{D}_{\text{HC}}^{\text{RComp}[\,]} = \frac{\begin{array}{l}[\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2 \bullet B[\,]\ \Upsilon_3, \gamma, i, k \mid C, p, q], \\ [\boldsymbol{B} \rightarrow \bullet\Upsilon_4\bullet, -, k, j \mid -, -, -]\end{array}}{[\boldsymbol{A} \rightarrow \Upsilon_1 \bullet \Upsilon_2\ B[\,] \bullet \Upsilon_3, \gamma, i, j \mid C, p, q]}$$

$$\mathcal{D}_{\text{HC}}^{\text{HC}[\circ\circ\gamma][\circ\circ]} = \frac{[\boldsymbol{B} \rightarrow \bullet\Upsilon_3\bullet, \eta, i, j \mid C, p, q]}{[A[\circ\circ\gamma] \rightarrow \Upsilon_1 \bullet B[\circ\circ] \bullet \Upsilon_2, \gamma, i, j \mid B, i, j]}$$

$$\mathcal{D}_{\text{HC}}^{\text{HC}[\circ\circ][\circ\circ]} = \frac{[\boldsymbol{B} \rightarrow \bullet\Upsilon_3\bullet, \eta, i, j \mid C, p, q]}{[A[\circ\circ] \rightarrow \Upsilon_1 \bullet B[\circ\circ] \bullet \Upsilon_2, \eta, i, j \mid C, p, q]}$$

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

$$\mathcal{D}_{\text{HC}} = \mathcal{D}_{\text{HC}}^{\text{Scan}} \cup \mathcal{D}_{\text{HC}}^{\text{LComp}[\,]} \cup \mathcal{D}_{\text{HC}}^{\text{RComp}[\,]} \cup \mathcal{D}_{\text{HC}}^{\text{HC}[\circ\circ\gamma][\circ\circ]} \cup \mathcal{D}_{\text{HC}}^{\text{HC}[\circ\circ][\circ\circ]} \cup \mathcal{D}_{\text{HC}}^{\text{HC}[\circ\circ][\circ\circ\gamma]}$$

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

§

4.7.2. Algoritmo de reconocimiento de Boullier

Boullier presenta en [31] un reconocedor para gramáticas lineales de índices que divide el proceso de análisis en las dos fases siguientes:

1. Análisis de la gramática independiente del contexto que constituye el esqueleto de la gramática lineal de índices original con el fin de obtener un bosque compartido de todos los posibles análisis.

2. Comprobación de que las condiciones impuestas por la gramática lineal de índices se cumplen a través de las espinas definidas en el bosque de análisis resultante de la primera fase.

La segunda fase constituye la parte fundamental del algoritmo y está basada en el hecho de que si consideramos la evolución individual de cada espina vemos que todo índice que es apilado en un punto debe ser sacado de la pila en otro punto de la espina y viceversa, cada vez que un índice es sacado de una pila en un punto de la espina tenemos la seguridad de que ha sido apilado en algún punto anterior de la misma. Por lo tanto, para verificar las condiciones impuestas por una LIG sobre las espinas no es necesario reconstruir las pilas de índices sino verificar que se cumplen los pares apilar/sacar. En el caso del reconocimiento la tarea se ve simplificada por el hecho de que no nos interesa indicar todas las posibles espinas entre dos elementos LIG sino simplemente verificar la existencia de una de tales espinas.

Para ello utilizaremos las relaciones $\leadsto$, $\overset{\gamma}{\prec}$ y $\overset{\gamma}{\succ}$ que se definen en la tabla 4.3, donde o_1 se refiere al elemento LIG constituido por A y su pila de índices asociada y o_2 se refiere a B y su pila de índices. Estas relaciones indican, para cada producción de la gramática, la operación que se debe aplicar a la pila asociada al elemento del lado izquierdo para obtener la pila asociada al hijo dependiente, de tal modo que $\leadsto$ indica que no se hacen cambios, $\overset{\gamma}{\prec}$ que se apila el índice γ y $\overset{\gamma}{\succ}$ que se saca de la pila el índice γ.

$$
\begin{array}{lcl}
\leadsto & = & \{(o_1, o_2) \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ]\ \Upsilon_2 \in P\} \\
\overset{\gamma}{\prec} & = & \{(o_1, o_2) \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ\gamma]\ \Upsilon_2 \in P\} \\
\overset{\gamma}{\succ} & = & \{(o_1, o_2) \mid A[\circ\circ\gamma] \rightarrow \Upsilon_1\ B[\circ\circ]\ \Upsilon_2 \in P\}
\end{array}
$$

Tabla 4.3: Relaciones binarias definidas por el reconocedor de Boullier

El algoritmo de reconocimiento de Boullier utiliza esta tabla para inicializar las relaciones y aplica las reglas de composición descritas en la tabla 4.4 para obtener los valores finales de dichas relaciones. Utilizando estas relaciones podemos calcular la relación $\approx$ que calcula porciones válidas de una espina y que se define recursivamente como

$$\approx \quad = \quad \leadsto \cup \overset{\gamma}{\prec} \overset{*}{\approx} \overset{\gamma}{\succ}$$

En este punto estamos en disposición de calcular la relación $\bowtie$ de espinas válidas que se define como

$$\bowtie \quad = \quad \{(o_1, o_2) \mid o_1 \overset{+}{\approx} o_2\}$$

tal que o_1 es de la forma $A[\alpha]$ y aparece en la parte izquierda de una producción y o_2 es de la forma $B[\alpha']$ y aparece en la parte derecha de una producción.

si	$o_1 \overset{\gamma}{\prec} o_2$	y	$o_2 \leadsto o_3$	entonces	$o_1 \overset{\gamma}{\prec} o_3$
si	$o_1 \overset{\gamma}{\prec} o_2$	y	$o_2 \overset{\gamma}{\succ} o_3$	entonces	$o_1 \leadsto o_3$
si	$o_1 \leadsto o_2$	y	$o_2 \overset{\gamma}{\prec} o_3$	entonces	$o_1 \overset{\gamma}{\prec} o_3$
si	$o_1 \leadsto o_2$	y	$o_2 \leadsto o_3$	entonces	$o_1 \leadsto o_3$
si	$o_1 \leadsto o_2$	y	$o_2 \overset{\gamma}{\succ} o_3$	entonces	$o_1 \overset{\gamma}{\succ} o_3$
si	$o_1 \overset{\gamma}{\succ} o_2$	y	$o_2 \leadsto o_3$	entonces	$o_1 \overset{\gamma}{\succ} o_3$

Tabla 4.4: Reglas de composición de las relaciones del reconocedor de Boullier

Esta última relación se utiliza para eliminar del bosque de análisis obtenido en la primera fase aquellos nodos que no cumplen las condiciones impuestas sobre las pilas de índices por la gramática. Si el nodo correspondiente al axioma de la gramática no es eliminado entonces podemos afirmar que la cadena de entrada pertenece al lenguaje definido por la gramática. El algoritmo trabaja con una complejidad temporal $\mathcal{O}(n^6)$, donde n es la longitud de la cadena de entrada, si se restringe la forma de las gramáticas de tal modo que la parte derecha de cada producción posea a lo sumo dos elementos.

4.7.3. Algoritmo de análisis sintáctico de Boullier

Boullier presenta en [32] una extensión del algoritmo precedente que lo convierte en un algoritmo de análisis sintáctico, ya que permite obtener todos los análisis posibles de una cadena de entrada de acuerdo con una gramática lineal de índices, manteniendo la complejidad $\mathcal{O}(n^6)$ para gramáticas con a lo sumo dos elementos en la parte derecha de las producciones.

El algoritmo se basa en la construcción de una *gramática de derivación lineal*, esencialmente una gramática independiente del contexto que define un lenguaje cuyas cadenas son las secuencias de producciones que constituyen derivaciones de una gramática lineal de índices para una cadena de entrada dada.

Para ello necesitamos definir las relaciones $\underset{1}{\leadsto}$, $\underset{1}{\overset{\gamma}{\prec}}$ y $\underset{1}{\overset{\gamma}{\succ}}$ de la tabla 4.5, que como podemos observar son muy similares a las relaciones $\leadsto$, $\overset{\gamma}{\prec}$ y $\overset{\gamma}{\succ}$. Adicionalmente, definimos la relación $\underset{+}{\leadsto}$ que selecciona pares de no-terminales que forman parte de una espina y cuyas pilas de índices están vacías. A partir de estas relaciones definimos las relaciones $\underset{+}{\overset{\gamma}{\succ}}$ y $\approx$ como

$$\underset{+}{\overset{\gamma}{\succ}} = \underset{1}{\overset{\gamma}{\succ}} \cup \underset{+}{\leadsto} \underset{1}{\overset{\gamma}{\succ}}$$

$$\approx = \bigcup_{\gamma \in V_I} \underset{1}{\overset{\gamma}{\prec}} \underset{+}{\overset{\gamma}{\succ}}$$

por lo que se cumple que

$$\underset{+}{\leadsto} = \underset{1}{\leadsto} \cup \approx \cup \underset{1}{\leadsto} \underset{+}{\leadsto} \cup \approx \underset{+}{\leadsto}$$

Dada una gramática lineal de índices (V_N, V_T, V_I, P, S) definimos su gramática de derivación lineal como la gramática independiente del contexto $D = (V_N^D, V_T^D, P^D, S^D)$ donde:

- $V_N^D = \{[A] \mid A \in V_N\} \cup \{[A\rho B] \mid A, B \in V_N,\ \rho \in \mathcal{R}\}$, donde $\mathcal{R} = \{\underset{1}{\overset{\gamma}{\prec}}, \underset{1}{\leadsto}, \underset{1}{\overset{\gamma}{\succ}}, \underset{+}{\leadsto}, \approx, \underset{+}{\overset{\gamma}{\succ}}\}$

$$
\begin{array}{lcl}
\underset{1}{\diamond} & = & \{(A,B) \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ]\ \Upsilon_2 \in P\} \\
\overset{\gamma}{\underset{1}{\prec}} & = & \{(A,B) \mid A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ\gamma]\ \Upsilon_2 \in P\} \\
\overset{\gamma}{\underset{1}{\succ}} & = & \{(A,B) \mid A[\circ\circ\gamma] \rightarrow \Upsilon_1\ B[\circ\circ]\ \Upsilon_2 \in P\} \\
\underset{+}{\diamond} & = & \{(A,B) \mid A[\,] \overset{+}{\Rightarrow} \Upsilon_1\ B[\,]\ \Upsilon_2 \text{ donde } B[\,] \text{ es el descendiente dependiente de } A[\,]\}
\end{array}
$$

Tabla 4.5: Relaciones binarias definidas por el analizador sintáctico de Boullier

- $V_T^D = P$
- $S^D = [S]$
- $$
\begin{array}{rcl}
P^D & = & \{[A] \rightarrow r \mid r : A[\,] \rightarrow a \in P\} \cup \\
& & \{[A] \rightarrow r[A\underset{+}{\diamond}B] \mid r : B[\,] \rightarrow a \in P\} \cup \\
& & \{[A\underset{+}{\diamond}C] \rightarrow [\Upsilon_1\Upsilon_2]r \mid r : A[\circ\circ] \rightarrow \Upsilon_1\ C[\circ\circ]\ \Upsilon_2 \in P\} \cup \\
& & \{[A\underset{+}{\diamond}C] \rightarrow [A \approx C]\} \cup \\
& & \{[A\underset{+}{\diamond}C] \rightarrow [B\underset{+}{\diamond}C][\Upsilon_1\Upsilon_2]r \mid r : A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ]\ \Upsilon_2 \in P\} \cup \\
& & \{[A\underset{+}{\diamond}C] \rightarrow [B\underset{+}{\diamond}C][A \approx B]\} \cup \\
& & \{[A \approx C] \rightarrow [B\overset{\gamma}{\underset{+}{\succ}}C][\Upsilon_1\Upsilon_2]r \mid r : A[\circ\circ] \rightarrow \Upsilon_1\ B[\circ\circ\gamma]\ \Upsilon_2 \in P\} \cup \\
& & \{[A\overset{\gamma}{\underset{+}{\succ}}c] \rightarrow [\Upsilon_1\Upsilon_2]r \mid r : A[\circ\circ\gamma] \rightarrow \Upsilon_1\ C[\circ\circ]\ \Upsilon_2 \in P\} \cup \\
& & \{[A\overset{\gamma}{\underset{+}{\succ}}c] \rightarrow [\Upsilon_1\Upsilon_2]r[A\underset{+}{\diamond}B] \mid r : B[\circ\circ\gamma] \rightarrow \Upsilon_1\ C[\circ\circ]\ \Upsilon_2 \in P\}
\end{array}
$$

  donde r identifica una producción y $[\Upsilon_1\Upsilon_2]$ denota, o bien al no-terminal $[X]$ cuando $\Upsilon_1\Upsilon_2 = X[\,]$, o bien a la cadena vacía ϵ cuando $\Upsilon_1\Upsilon_2 \in V_T^*$

Denominamos derivación lineal a una derivación a derechas en la cual primero se derivan los hijos no dependiente y por último el hijo dependiente. Las producciones P^D definen todas las posibles *derivaciones lineales* de la LIG original.

El algoritmo de análisis sintáctico de Boullier consta de los siguientes pasos:

1. Construcción de un bosque de análisis para la gramática independiente del contexto que constituye el esqueleto independiente del contexto de la LIG original.

2. Construcción del bosque de análisis para la LIG a partir del bosque de análisis obtenido para su esqueleto independiente del contexto. Dicho bosque de análisis constituye a su vez una LIG, de igual modo que el bosque de análisis de una CFG constituye una CFG.

3. Construcción de la gramática de derivación lineal a partir de la LIG que representa el bosque de análisis obtenido en el paso anterior.

Puesto que cada derivación de la gramática lineal de índices original es una cadena de la gramática de derivación lineal, la extracción de análisis individuales de una LIG ha sido reducida a la derivación de cadenas en una gramática independiente del contexto. En consecuencia, cada derivación se puede obtener en tiempo lineal.