Capítulo 6

Autómatas lineales de índices

La utilización de autómatas para realizar el análisis sintáctico es interesante porque permite separar el problema de la definición de un algoritmo de análisis sintáctico del problema de la ejecución del mismo. En el caso particular de las gramáticas independientes del contexto es posible optar por este diseño modular puesto que es posible definir un algoritmo de análisis sintáctico como un conjunto de transiciones de un autómata a pila, probablemente no determinista, el cual puede ser interpretado eficientemente mediante las técnicas de tabulación disponibles. Este enfoque presenta ventajas evidentes, entre la cuales cabe citar la simplificación de las pruebas de corrección de los algoritmos y la disponibilidad de un entorno homogéneo para la comparación experimental de su comportamiento.

En este capítulo adaptaremos este enfoque a los lenguajes de adjunción de árboles, proporcionando modelos de autómata con los que describir los algoritmos de análisis, y técnicas de tabulación con las que pueden ser ejecutados eficientemente. Con este fin, en el presente capítulo extenderemos los autómatas a pila tradicionales para dar lugar a los autómatas lineales de índices, con tres variantes diferentes: los autómatas lineales de índices derechos, izquierdos y universales. Estos modelos de autómata permiten definir estrategias de diferentes tipos para gramáticas de adjunción de árboles, incluyendo estrategias de tipo Earley, y pueden ser ejecutados en tiempo polinomial. Finalmente, definiremos una variante bidireccional de los autómatas lineales de índices.

6.1 Autómatas a pila

Los autómatas a pila son máquinas abstractas que reconocen exactamente la clase de los lenguajes independientes del contexto. En esta sección introducimos este tipo de autómatas, así como su técnica de tabulación, puesto que servirá de base a los modelos de autómata para lenguajes de adjunción de árboles que serán presentados en éste y subsiguientes capítulos.

Presentamos dos definiciones diferentes pero equivalentes de los autómatas a pila (*Push-Down Automata*, PDA) [42]. En primer lugar presentaremos la definición clásica, que considera un autómata a pila como una máquina abstracta que consta de tres componentes: una cadena de entrada, un control finito y una pila. Seguidamente presentaremos una definición más moderna en la cual se suprimen las referencias al control finito para centrarse en el componente fundamental de este tipo de autómatas: la pila.

6.1.1 Definición clásica con estados

En la definición clásica [42], los autómatas a pila son considerados tuplas $(Q, V_T, V_S, \delta, q_0, \$_0, Q_F)$, donde:

- Q es un conjunto finito de estados.
- V_T es un conjunto finito de símbolos terminales.
- V_S es un conjunto finito de símbolos de pila.
- $q_0 \in Q$ es el estado inicial.
- $\$_0 \in V_S$ es el símbolo inicial de la pila.
- $Q_F \subseteq Q$ es el conjunto de estados finales.
- δ es una relación de $Q \times (V_T \cup \{\epsilon\}) \times (V_S \cup \{\epsilon\})$ en subconjuntos finitos de $Q \times V_S^*$ que define los movimientos o transiciones válidos del autómata. Una transición

  $$(q', \beta) \in \delta(q, a, Z)$$

  donde $q, q' \in Q$, $a \in V_T \cup \{\epsilon\}$, $Z \in V_S \cup \{\epsilon\}$, $\beta \in V_S^*$, se interpreta como sigue: si el autómata se encuentra en el estado q, el siguiente elemento a leer en la cadena de entrada es a y el símbolo en la cima de la pila es Z, entonces puede pasar al estado q' y reemplazar la cima de la pila por β.

La *configuración* de un autómata a pila en un momento dado viene definida por el triple (q, α, w), donde q indica el estado en el que se encuentra el autómata, α el contenido de la pila y w la parte de la cadena de entrada que resta por leer. El cambio de una configuración a otra viene determinado por la aplicación de una transición, de tal modo que si $(q, \alpha Z, aw)$ es una configuración y $(q', \beta) \in \delta(q, a, Z)$, entonces el autómata pasará a la nueva configuración $(q', \alpha\beta, w)$, hecho que denotamos mediante $(q, \alpha Z, aw) \vdash (q', \alpha\beta, w)$. Denotamos por $\overset{*}{\vdash}$ el cierre reflexivo y transitivo de $\vdash$. A la hora de representar una secuencia de derivaciones de tamaño m, utilizaremos la notación $\overset{m}{\vdash}$.

El *lenguaje aceptado por estado final* por un autómata a pila viene determinado por el conjunto de cadenas $w \in V_T^*$ tal que $(q_0, \$_0, w) \overset{*}{\vdash} (p, \alpha, \epsilon)$, donde $p \in Q_F$ y $\alpha \in V_S^*$.

El *lenguaje aceptado por pila vacía* por un autómata a pila viene determinado por el conjunto de cadenas $w \in V_T^*$ tal que $(q_0, \$_0, w) \overset{*}{\vdash} (p, \epsilon, \epsilon)$ para cualquier $p \in Q$.

Dado un autómata a pila que reconoce un determinado lenguaje por estado final, es posible construir otro autómata a pila que reconoce el mismo lenguaje por pila vacía y viceversa [42].

6.1.2 Definición sin estados

El control finito de un autómata a pila es un elemento prescindible, puesto que el estado de una configuración dada puede almacenarse en el elemento situado en la cima de la pila [11]. Como consecuencia obtenemos una definición alternativa equivalente [51, 11, 26] según la cual un autómata a pila es una tupla $(V_T, V_S, \Theta, \$_0, \$_f)$, donde

- V_T es un conjunto finito de símbolos terminales.
- V_S es un conjunto finito de símbolos de pila.
- $\$_0 \in V_S$ es el símbolo inicial de la pila.
- $\$_f \in V_S$ es el símbolo final de la pila.

- Θ es un conjunto de transiciones, cada una de las cuales pertenece a uno de los tres tipos siguientes, donde $C, F, G \in V_S$, $\xi \in V_S^*$ y $a \in V_T \cup \{\epsilon\}$:

  SWAP: Transiciones de la forma $C \overset{a}{\longmapsto} F$ que reemplazan el elemento C de la cima de la pila por el elemento F mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila ξC es una pila ξF.

  PUSH: Transiciones de la forma $C \overset{a}{\longmapsto} CF$ que apilan un nuevo elemento F en la pila mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila ξC es una pila ξCF.

  POP: Transiciones de la forma $CF \overset{a}{\longmapsto} G$ que eliminan los dos elementos C y F de la cima de la pila y los sustituyen por G mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila ξCF es una pila ξG, con lo cual el tamaño de la pila decrece en una unidad.

Con objeto de simplificar la notación, denotaremos mediante $\longmapsto$ las transiciones aplicadas sobre la palabra vacía. Según la nueva definición, la *configuración* de un autómata a pila en un momento dado viene determinada por el par (ξ, w), donde ξ es el contenido de la pila y w es la parte de la cadena de entrada que resta por leer. Una configuración (ξ, aw) deriva una configuración (ξ', w), hecho que denotamos mediante $(\xi, aw) \vdash (\xi', w)$, si y sólo si existe una transición que aplicada a ξ devuelve ξ' y consume a de la cadena de entrada. En caso de ser necesario identificar una derivación d concreta, utilizaremos la notación $\vdash_d$. Denotamos por $\overset{*}{\vdash}$ el cierre reflexivo y transitivo de $\vdash$.

Decimos que una cadena w es aceptada por un autómata a pila si $(\$_0, w) \overset{*}{\vdash} (\$_0\$_f, \epsilon)$. El *lenguaje aceptado* por un autómata a pila viene determinado por el conjunto de cadenas $w \in V_T^*$ tales que $(\$_0, w) \overset{*}{\vdash} (\$_0\$_f, \epsilon)$.

6.1.3 Esquemas de compilación de gramáticas independientes del contexto

Un *esquema de compilación* es un conjunto de reglas que permite, a partir de una gramática independiente del contexto y de una estrategia de análisis sintáctico, construir un autómata a pila que describa los cálculos que se pueden realizar con dicha gramática utilizando la estrategia de análisis elegida.

Los esquemas de compilación que se van a mostrar se basan en el paradigma de llamada/retorno [29], utilizando para ello los seis tipos de reglas mostrados en la tabla 6.1. A toda regla [**CALL**] le corresponde una regla [**RET**] y viceversa. Las reglas [**INIT**], [**CALL**] y [**SEL**] definen las transiciones del autómata encargadas de la fase predictiva del algoritmo de análisis mientras que las reglas [**RET**] y [**PUB**] definen las transiciones encargadas de propagar la información en la fase ascendente. Por este motivo la fase descendente o predictiva de una estrategia de análisis, cuando es implantada en un autómata a pila, recibe el nombre de *fase de llamada*, mientras que la fase ascendente recibe el nombre de *fase de retorno*.

En primer lugar definiremos el esquema de compilación correspondiente a una estrategia genérica basada en el paradigma llamada/retorno, parametrizada con respecto a la información que se predice y propaga en las fases de llamada y de retorno, respectivamente. Utilizaremos la siguiente notación:

- $A_{r,s}$ para referirnos al elemento de la producción r que ocupa la posición s, de modo que para una producción r tenemos que $A_{r,0} \to A_{r,1} \ldots A_{r,n_r}$.

- $\nabla_{r,s}$ para indicar el reconocimiento parcial de una producción, notacionalmente equivalente a una producción con punto $A_{r,0} \to A_{r,1} \ldots A_{r,s} \bullet A_{r,s+1} \ldots A_{r,n_r}$.

Regla	Tarea
[INIT]	inicia los cálculos a partir de la pila inicial.
[CALL]	requiere el análisis de un no terminal de una producción.
[SEL]	selecciona una producción.
[PUB]	determina que una producción ha sido completamente analizada.
[RET]	continua el proceso de análisis después de terminar una producción.
[SCAN]	reconoce los terminales que componen la cadena de entrada.

Tabla 6.1: Reglas de los esquemas de compilación de gramáticas independientes del contexto

- $\overrightarrow{A_{r,s}}$ para referirnos a la predicción de información con respecto a $A_{r,s}$.
- $\overleftarrow{A_{r,s}}$ para representar la información propagada ascendentemente con respecto a $A_{r,s}$.

Con el fin de simplificar al máximo la definición de los esquemas de compilación y sin pérdida de generalidad, supondremos que se cumplen las siguientes condiciones sobre la gramática independiente del contexto:

- El axioma sólo aparece en el lado izquierdo de la producción unitaria 0, que tiene la forma $S \to X$, con $X \in V_N \cup V_T$
- Los terminales o la palabra vacía sólo aparecen en producciones unitarias de la forma $A_{r,0} \to a$, donde $a \in V_T \cup \{\epsilon\}$.

El esquema de compilación genérico de una gramática independiente del contexto en un autómata a pila queda definido por los elementos inicial $\$_0$ y final $\overleftarrow{S}$ y por las siguientes reglas:

$$
\begin{array}{lll}
\textbf{[INIT]} & \$_0 \longmapsto \$_0 \ \nabla_{0,0} & \\
\textbf{[CALL]} & \nabla_{r,s} \longmapsto \nabla_{r,s} \ \overrightarrow{A_{r,s+1}} & \\
\textbf{[SEL]} & \overrightarrow{A_{r,0}} \longmapsto \nabla_{r,0} & r \neq 0 \\
\textbf{[PUB]} & \nabla_{r,n_r} \longmapsto \overleftarrow{A_{r,0}} & \\
\textbf{[RET]} & \nabla_{r,s} \ \overleftarrow{A_{r,s+1}} \longmapsto \nabla_{r,s+1} & \\
\textbf{[SCAN]} & \overrightarrow{A_{r,0}} \overset{a}{\longmapsto} \overleftarrow{A_{r,0}} & A_{r,0} \to a
\end{array}
$$

La primera columna indica el nombre de la regla, la segunda las transiciones generadas por la misma y la tercera las condiciones, generalmente referidas a la forma de las producciones, que se deben cumplir para que la regla sea aplicable.

A continuación presentamos tres versiones concretas del esquema de compilación genérico. La primera se corresponde con una estrategia de análisis descendente en la cual los no terminales

Estrategia	$\overrightarrow{A_{r,s+1}}$	$\overleftarrow{A_{r,s+1}}$
Ascendente	$\Box$	$A_{r,s+1}$
Earley	$\overline{A_{r,s+1}}$	$\overline{\overline{A_{r,s+1}}}$
Descendente	$A_{r,s+1}$	$\Box$

Tabla 6.2: Parámetros del esquema de compilación genérico de CFG en PDA

se predicen en la fase de llamada pero no se propagan en la fase ascendente. La segunda se corresponde con una estrategia mixta de tipo Earley [38] en la que los no terminales se predicen en la fase de llamada y se propagan en la fase de retorno. La tercera y última se corresponde con una estrategia ascendente en la cual no hay ningún tipo de predicción en la fase de llamada mientras que en la fase de retorno se propagan los no terminales analizados.

En la tabla 6.2 se muestran los valores que deben tomar los parámetros de predicción y propagación de información para transformar el esquema de compilación genérico en esquemas correspondientes a las tres estrategias de análisis citadas, donde $\Box$ representa un símbolo de pila nuevo. En el caso de estrategias Earley es necesario distinguir la llamada de un no terminal $A_{r,s+1}$ de su retorno, para lo cual utilizamos los símbolos $\overline{A_{r,s+1}}$ y $\overline{\overline{A_{r,s+1}}}$, respectivamente.

El esquema de compilación descendente de una gramática independiente del contexto en un autómata a pila queda definido por los elementos inicial $\$_0$ y final $\Box$ y por el siguiente conjunto de reglas de compilación:

$$
\begin{array}{lll}
\textbf{[INIT]} & \$_0 \longmapsto \$_0\ \nabla_{0,0} & \\
\textbf{[CALL]} & \nabla_{r,s} \longmapsto \nabla_{r,s}\ \ A_{r,s+1} & \\
\textbf{[SEL]} & A_{r,0} \longmapsto \nabla_{r,0} & r \neq 0 \\
\textbf{[PUB]} & \nabla_{r,n_r} \longmapsto \Box & \\
\textbf{[RET]} & \nabla_{r,s}\ \ \Box\ \longmapsto \nabla_{r,s+1} & \\
\textbf{[SCAN]} & A_{r,0} \overset{a}{\longmapsto} \Box & A_{r,0} \to a
\end{array}
$$

Ejemplo 6.1 Consideremos la gramática independiente del contexto definida por las producciones:

$$
\begin{array}{ll}
(0) & S \to X \\
(1) & X \to AXB \\
(2) & X \to \epsilon \\
(3) & A \to a \\
(4) & B \to b
\end{array}
$$

que genera el lenguaje $\{a^n b^n \mid n \geq 0\}$. La tabla 6.3 muestra el conjunto de transiciones del autómata a pila que se obtiene al aplicar el esquema de compilación descendente mientras que la tabla 6.4 muestra la secuencia de configuraciones que sigue el autómata para analizar correctamente la cadena de entrada *aabb*. La primera columna muestra la transición aplicada, la segunda el tipo de la regla que generó dicha transición, la tercera el contenido de la pila y la cuarta la parte que resta por leer de la cadena de entrada. Las configuraciones marcadas

(a)	**[INIT]**	$\$_0 \longmapsto \$_0\ \nabla_{0,0}$
(b)	**[CALL]**	$\nabla_{0,0} \longmapsto \nabla_{0,0}\ \ X$
(c)	**[RET]**	$\nabla_{0,0}\ \Box\ \ \longmapsto \nabla_{0,1}$
(d)	**[PUB]**	$\nabla_{0,1} \longmapsto \Box$
(e)	**[SEL]**	$X \longmapsto \nabla_{1,0}$
(f)	**[CALL]**	$\nabla_{1,0} \longmapsto \nabla_{1,0}\ A$
(g)	**[RET]**	$\nabla_{1,0}\ \Box\ \ \longmapsto \nabla_{1,1}$
(h)	**[CALL]**	$\nabla_{1,1} \longmapsto \nabla_{1,1}\ X$
(i)	**[RET]**	$\nabla_{1,1}\ \Box\ \ \longmapsto \nabla_{1,2}$
(j)	**[CALL]**	$\nabla_{1,2} \longmapsto \nabla_{1,2}\ B$
(k)	**[RET]**	$\nabla_{1,2}\ \Box\ \ \longmapsto \nabla_{1,3}$
(l)	**[PUB]**	$\nabla_{1,3} \longmapsto \Box$
(m)	**[SCAN]**	$X \overset{\epsilon}{\longmapsto} \Box$
(n)	**[SCAN]**	$A \overset{a}{\longmapsto} \Box$
(p)	**[SCAN]**	$B \overset{b}{\longmapsto} \Box$

Tabla 6.3: Transiciones del autómata a pila con estrategia descendente

con $*$ son aquellas en las que hay más de una opción para proseguir el análisis. Se trata de configuraciones resultado de aplicar una transición de tipo **[CALL]**. Si el elemento apilado coincide con el lado izquierdo de varias producciones, debe proseguirse el análisis por cada una de ellas. En la tabla 6.4 sólo se muestran las configuraciones que llevan al reconocimiento de la cadena de entrada. El resto de las posibles configuraciones llevan al autómata a detenerse en configuraciones que no son finales. ¶

El esquema de compilación para una estrategia de tipo Earley de una gramática independiente del contexto en un autómata a pila queda definido por los elementos inicial $\$_0$ y final $\overline{\overline{S}}$ y por el siguiente conjunto de reglas de compilación:

[INIT]	$\$_0 \longmapsto \$_0\ \nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \longmapsto \nabla_{r,s}\ \ \overline{A_{r,s+1}}$	
[SEL]	$\overline{A_{r,0}} \longmapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \longmapsto \overline{\overline{A_{r,0}}}$	
[RET]	$\nabla_{r,s}\ \ \overline{\overline{A_{r,s+1}}}\ \ \longmapsto \nabla_{r,s+1}$	
[SCAN]	$\overline{A_{r,0}} \overset{a}{\longmapsto} \overline{\overline{A_{r,0}}}$	$A_{r,0} \to a$

		$\$_0$						$aabb$	
(a)	**[INIT]**	$\$_0$	$\nabla_{0,0}$					$aabb$	
(b)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	X				$aabb$	$*$
(e)	**[SEL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,0}$				$aabb$	
(f)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,0}$	A			$aabb$	
(n)	**[SCAN]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,0}$	$\Box$			abb	
(g)	**[RET]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$				abb	
(h)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	X			abb	$*$
(e)	**[SEL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,0}$			abb	
(f)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,0}$	A		abb	
(n)	**[SCAN]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,0}$	$\Box$		bb	
(g)	**[RET]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,1}$			bb	
(h)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,1}$	X		bb	$*$
(m)	**[SCAN]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,1}$	$\Box$		bb	
(i)	**[RET]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,2}$			bb	
(j)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,2}$	B		bb	
(p)	**[SCAN]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,2}$	$\Box$		b	
(k)	**[RET]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\nabla_{1,3}$			b	
(l)	**[PUB]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,1}$	$\Box$			b	
(i)	**[RET]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,2}$				b	
(j)	**[CALL]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,2}$	B			b	
(p)	**[SCAN]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,2}$	$\Box$				
(k)	**[RET]**	$\$_0$	$\nabla_{0,0}$	$\nabla_{1,3}$					
(l)	**[PUB]**	$\$_0$	$\nabla_{0,0}$	$\Box$					
(c)	**[RET]**	$\$_0$	$\nabla_{0,1}$						
(d)	**[PUB]**	$\$_0$	$\Box$						

Tabla 6.4: Configuraciones del autómata a pila descendente durante el análisis de $aabb$

El esquema de compilación ascendente de una gramática independiente del contexto en un autómata a pila queda definido por los elementos inicial $\$_0$ y final S y por el siguiente conjunto de reglas de compilación:

[INIT]	$\$_0 \longmapsto \$_0\ \nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \longmapsto \nabla_{r,s}\ \Box$	
[SEL]	$\Box \longmapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \longmapsto A_{r,0}$	
[RET]	$\nabla_{r,s}\ A_{r,s+1} \longmapsto \nabla_{r,s+1}$	
[SCAN]	$\Box \overset{a}{\longmapsto} A_{r,0}$	$A_{r,0} \to a$

6.1.4 Técnicas de tabulación de autómatas a pila

La independencia del contexto de las transiciones de los autómatas a pila permite tabular la ejecución de los mismos. En esta sección presentamos dos técnicas diferentes para la tabulación de los autómatas a pila, una propuesta por Lang [50, 51], la otra propuesta por Nederhof [57].

La técnica de Lang

En una gramática independiente del contexto, si $B \overset{*}{\Rightarrow} \delta$ entonces $\alpha B \beta \overset{*}{\Rightarrow} \alpha \delta \beta$ para todo $\alpha, \beta \in (V_N \cup V_T)^*$. Esta misma independencia del contexto se traslada a los autómatas a pila, de tal modo que si

$$(B, a_{i+1} \dots a_j \dots a_n) \overset{*}{\vdash} (BC, a_{j+1} \dots a_n)$$

entonces también se cumple que

$$(\xi B, a_{i+1} \dots a_j \dots a_n) \overset{*}{\vdash} (\xi BC, a_{j+1} \dots a_n)$$

para todo $\xi \in V_S^*$. Denominaremos *derivaciones independientes del contexto* a este tipo de derivaciones. Observamos que este tipo de derivaciones presenta gran semejanza con las transiciones PUSH.

Para representar una derivación independiente del contexto sólo precisamos almacenar los símbolos de pila B y C más las posiciones de la cadena de entrada i y j, puesto que la derivación es independiente de ξ. La técnica de tabulación de autómatas a pila propuesta por Lang [50, 51] se basa precisamente en reemplazar la manipulación de pilas por la manipulación de ítems de la forma

$$[B, i, C, j]$$

que representan de forma compacta el conjunto de derivaciones independientes del contexto que comparten los elementos de la cima de la pila. Los ítems se combinan mediante *reglas de inferencia* de la forma $\frac{ants}{cons}\ trans$, donde *cons* es el ítem consecuente que se obtiene al aplicar la transición *trans* sobre los ítems antecedentes *ants*. A continuación mostramos las reglas de inferencia para los tres tipos de transiciones:

Transiciones SWAP: la aplicación de una transición $C \overset{a}{\longmapsto} F$ produce la derivación $(\xi C, a_j \dots a_n) \vdash (\xi F, a_k \dots a_n)$ en un autómata a pila, donde $k = j$ si $a = \epsilon$ y $k = j{+}1$ si $a = a_{j+1}$. Dada la derivación independiente del contexto $(\xi' B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi' BC, a_{j+1} \dots a_n)$ que da lugar a la configuración $(\xi' BC, a_{j+1} \dots a_n)$, tras la aplicación de la transición obtendremos la derivación independiente del contexto $(\xi' B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi' BF, a_{k+1} \dots a_n)$. La correspondiente regla de inferencia es

$$\frac{[B, i, C, j]}{[B, i, F, k]} \quad C \overset{a}{\longmapsto} F$$

Transiciones PUSH: la aplicación de una transición $C \overset{a}{\longmapsto} CF$ produce la derivación $(\xi C, a_{j+1} \dots a_n) \vdash (\xi CF, a_{k+1} \dots a_n)$, donde $k = j$ si $a = \epsilon$ y $k = j + 1$ si $a = a_{j+1}$. Esta derivación es por sí misma una derivación independiente del contexto, por lo que la correspondiente regla de inferencia es

$$\frac{[B, i, C, j]}{[C, j, F, k]} \quad C \overset{a}{\longmapsto} CF$$

Transiciones POP: la aplicación de una transición $CF \overset{a}{\longmapsto} G$ produce la derivación $(\xi CF, a_{k+1} \dots a_n) \vdash (\xi G, a_{l+1} \dots a_n)$. La configuración $(\xi CF, a_{k+1} \dots a_n)$ refleja una derivación independiente del contexto $(\xi C, a_{j+1} \dots a_n) \overset{*}{\vdash} (\xi CF, a_{k+1} \dots a_n)$, pero además necesitamos la derivación independiente del contexto $(\xi' B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi' BC, a_{j+1} \dots a_n)$ que colocó C en la cima de la pila para obtener la derivación independiente del contexto $(\xi' B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi' BG, a_{l+1} \dots a_n)$ resultante de la aplicación de la transición, donde $l = k$ si $a = \epsilon$ y $l = k + 1$ si $a = a_{k+1}$. La correspondiente regla de inferencia es

$$\frac{\begin{array}{c}[C, j, F, k] \\ [B, i, C, j]\end{array}}{[B, i, G, l]} \quad CF \overset{a}{\longmapsto} G$$

Las reglas de inferencia comienzan a aplicarse a partir de un ítem inicial $[-, -, \$_0, 0]$ hasta que no se pueden obtener más ítems. La presencia de un ítem final de la forma $[\$_0, 0, \$_f, n]$ indica el reconocimiento de la cadena de entrada $a_1 \dots a_n$. La manipulación de configuraciones mediante la aplicación de transiciones es equivalente a la manipulación de ítems mediante las reglas de inferencia correspondientes a cada transición [50, 51].

La técnica de Nederhof

Aunque Nederhof no ha presentado una técnica específica para la tabulación de autómatas a pila, la misma se obtiene como un caso especial de la técnica de tabulación propuesta por dicho autor para los autómatas lineales de índices [57].

A diferencia de la técnica de Lang, que traslada la propiedad de independencia del contexto de las gramáticas independientes del contexto a derivaciones con la forma de transiciones de tipo PUSH, la técnica de tabulación de Nederhof traslada dicha propiedad a derivaciones con la forma de transiciones SWAP, puesto que si tenemos una derivación

$$(B, a_{i+1} \dots a_{j+1} \dots a_n) \overset{*}{\vdash} (C, a_{j+1} \dots a_n)$$

entonces también se cumple que

$$(\xi B, a_{i+1} \dots a_{j+1} \dots a_n) \overset{*}{\vdash} (\xi C, a_{j+1} \dots a_n)$$

para todo $\xi \in V_S^*$. Al igual que en la técnica de Lang, estas derivaciones también se denominan *derivaciones independientes del contexto*.

Para representar una derivación independiente del contexto sólo precisamos almacenar B, C y las posiciones i y j, por lo que se utilizarán ítems de la forma

$$[B, i, C, j]$$

Estos ítems se generan mediante reglas de inferencia, una para cada tipo de transición:

Transiciones SWAP: la aplicación de una transición $C \overset{a}{\longmapsto} F$ produce una derivación $(\xi C, a_{j+1} \dots a_n) \vdash (\xi F, a_{k+1} \dots a_n)$ en un autómata a pila, donde $k = j$ si $a = \epsilon$ y $k = j + 1$ si $a = a_{j+1}$. Dada la derivación independiente del contexto $(\xi B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi C, a_{j+1} \dots a_n)$, tras la aplicación de la transición obtendremos la derivación independiente del contexto $(\xi B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi F, a_{k+1} \dots a_n)$. La correspondiente regla de inferencia es

$$\frac{[B, i, C, j]}{[B, i, F, k]} \quad C \overset{a}{\longmapsto} F$$

Transiciones PUSH: la aplicación de una transición $C \overset{a}{\longmapsto} CF$ produce la derivación $(\xi C, a_{j+1} \dots a_n) \vdash (\xi CF, a_{k+1} \dots a_n)$, donde $k = j$ si $a = \epsilon$ y $k = j+1$ si $a = a_{j+1}$. Dada la derivación independiente del contexto $(\xi B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi C, a_{j+1} \dots a_n)$, tras la aplicación de la transición obtendremos la derivación independiente del contexto $(\xi F, a_{k+1} \dots a_n) \overset{0}{\vdash} (\xi F, a_{k+1} \dots a_n)$. La correspondiente regla de inferencia es

$$\frac{[B, i, C, j]}{[F, k, F, k]} \quad C \overset{a}{\longmapsto} CF$$

Transiciones POP: la aplicación de una transición $CF \overset{a}{\longmapsto} G$ produce una derivación $(\xi CF, a_{k+1} \dots a_n) \vdash (\xi G, a_{l+1} \dots a_n)$, donde $l = k$ si $a = \epsilon$ y $l = k+1$ si $a = a_{k+1}$. La configuración $(\xi CF, a_{k+1} \dots a_n)$ refleja una derivación constituida por

1. una derivación independiente del contexto $(\xi B, a_{i+1} \dots a_n) \overset{*}{\vdash} (\xi C, a_{j+1} \dots a_n)$;
2. un paso de derivación $(\xi C, a_{j+1} \dots a_n) \vdash (\xi CF', a_{k'+1} \dots a_n)$ resultado de la aplicación de una transición $C \overset{b}{\longmapsto} CF'$, donde $k' = j$ si $b = \epsilon$ y $k' = j+1$ si $b = a_{j+1}$;
3. una derivación independiente del contexto $(\xi CF', a_{k'+1} \dots a_n) \overset{*}{\vdash} (\xi CF, a_{k+1} \dots a_n)$.

En consecuencia, la regla de inferencia tendrá la forma

$$\frac{\begin{array}{c}[F', k', F, k] \\ [B, i, C, j]\end{array}}{[B, i, G, l]} \quad \begin{array}{l} C \overset{b}{\longmapsto} CF' \\ CF \overset{a}{\longmapsto} G \end{array}$$

Las reglas de inferencia comienzan a aplicarse a partir de un ítem inicial $[\$_0, 0, \$_0, 0]$ hasta que no se pueden obtener más ítems. La presencia de un ítem final de la forma $[\$_0, 0, \$_f, n]$ indica el reconocimiento de la cadena de entrada $a_1 \dots a_n$. Al igual que en la técnica de Lang, la manipulación de configuraciones mediante la aplicación de transiciones es equivalente a la manipulación de ítems mediante las reglas de inferencia correspondientes a cada transición.

6.2 Autómatas lineales de índices

Los autómatas lineales de índices (*Linear Indexed Automata*, LIA) [55, 57] son una extensión de los autómatas a pila en la cual cada símbolo de pila tiene asociado una pila de índices y están estrechamente relacionados con los autómatas lógicos a pila restringidos [5]. Como podremos observar, los autómatas lineales de índices están muy vinculados con las gramáticas lineales de índices [2], y por tanto, en lo que sigue adoptaremos la notación ya expuesta al introducir esta clase de gramáticas. Formalmente, un autómata lineal de índices es una tupla $(V_T, V_S, \$_0, \$_f, V_I, \mathcal{T})$, donde:

- V_T es un conjunto finito de símbolos terminales.
- V_S es un conjunto finito de símbolos de pila.
- $\$_0 \in V_S$ es el símbolo inicial de la pila.
- $\$_f \in V_S$ es el símbolos final de pila.
- V_I es un conjunto finito de índices.

- $\mathcal{T}$ es un conjunto finito de transiciones, que pueden ser de alguno de los siguientes tipos:
  - $C[\,] \overset{a}{\longmapsto} F[\,]$
  - $C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ\gamma']$
  - $C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ]\; G[\,]$
  - $C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\,]\; G[\circ\circ\gamma']$
  - $F[\circ\circ\gamma]\; G[\,] \overset{a}{\longmapsto} C[\circ\circ\gamma']$
  - $F[\,]\; G[\circ\circ\gamma] \overset{a}{\longmapsto} C[\circ\circ\gamma']$

  donde $C, F, G \in V_S$, $a \in V_T \cup \{\epsilon\}$, $\gamma, \gamma' \in V_I \cup \{\epsilon\}$ y en cada transición, bien $\gamma = \epsilon$, bien $\gamma' = \epsilon$ o bien $\gamma = \gamma' = \epsilon$.

Dada una transición, decimos que el elemento de la parte derecha que comparte la pila de índices con el elemento de la parte izquierda es su *hijo dependiente*. La relación de *descendiente dependiente* es el cierre reflexivo y transitivo de la relación de hijo dependiente.

Una *configuración* de un autómata lineal de índices es un par (Υ, w), donde $\Upsilon \in (V_S[V_I^*])^*$ y $w \in V_T^*$, tal que Υ representa el contenido de la pila del autómata y w representa la parte de la cadena de entrada que resta por leer. Una configuración (Υ_1, aw) deriva una configuración (Υ_2, w), denotado mediante $(\Upsilon_1, aw) \vdash (\Upsilon_2, w)$ si y sólo si existe una transición que transforma la pila Υ_1 en la pila Υ_2 leyendo $a \in V_T \cup \{\epsilon\}$ de la cadena de entrada. En caso de ser necesario identificar una derivación d concreta, utilizaremos la notación $\vdash_d$. Denotamos por $\overset{*}{\vdash}$ el cierre reflexivo y transitivo de $\vdash$. Decimos que una cadena de entrada w es aceptada por un autómata lineal de índices si $(\$_0[\,], w) \overset{*}{\vdash} (\$_f[\,], \epsilon)$. El lenguaje de un autómata será el conjunto de todas las palabras que acepta.

Con respecto a la notación que aplicaremos en lo que resta de capítulo, utilizaremos letras mayúsculas para referirnos a los símbolos de pila; las letras minúsculas a y b para los terminales de la cadena de entrada (la cual se representará como la secuencia $a_1 \ldots a_n$); γ y η para referirnos a elementos en V_I; y finalmente α, δ y β pare referirnos a elementos en V_I^*. Debemos recordar al lector que las letras griegas α, β y γ también se utilizan habitualmente en gramáticas de adjunción de árboles para denotar árboles iniciales, auxiliares y elementales, respectivamente. Sin embargo, la aparición de estas letras en una lista de índices o en los nodos de un árbol permite identificar claramente su significado.

6.2.1 Esquemas de compilación de gramáticas de adjunción de árboles

En el caso que nos ocupa, un esquema de compilación es un conjunto de reglas que permite, a partir de una gramática de adjunción de árboles y de una estrategia de análisis sintáctico, construir un autómata lineal de índices que describa los cálculos que se pueden realizar con dicha gramática utilizando la estrategia de análisis elegida. Los esquemas de compilación que se van a mostrar a continuación se basan en el paradigma de llamada/retorno [29], utilizando para ello los 12 tipos de reglas de compilación que se muestran en la tabla 6.5, cuyo significado intuitivo se muestra gráficamente en la figura 6.1. Las reglas de compilación **[INIT]**, **[CALL]**, **[SCALL]**, **[SEL]**, **[ACALL]** y **[FCALL]** definen las transiciones encargadas de realizar la fase predictiva del algoritmo de análisis, mientras que las reglas de compilación **[RET]**, **[SRET]**, **[PUB]**, **[ARET]** y **[FRET]** definen las transiciones encargadas de propagar la información en la fase ascendente. Estas reglas dotan a todo elemento $A[\alpha]$ del autómata de la siguiente semántica: A indica un nodo de un árbol elemental mientras que α indica la pila de adjunciones pendientes en dicho nodo.

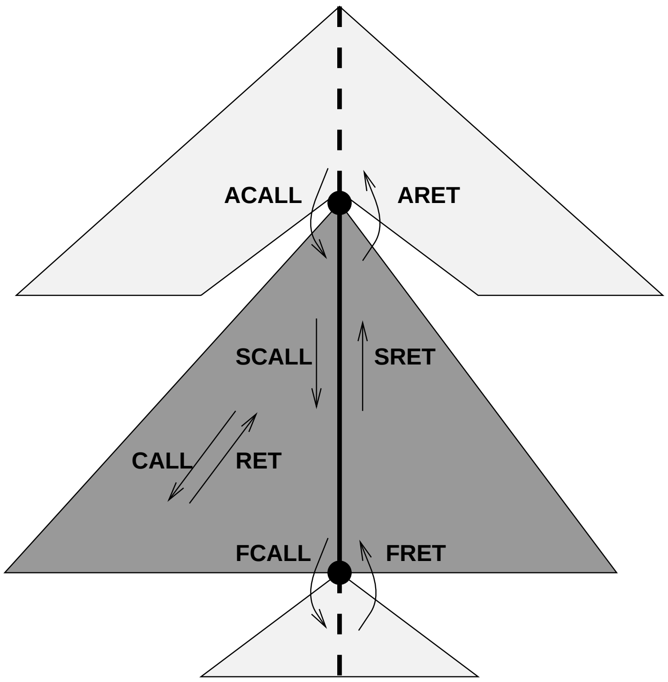

Figura 6.1: Reglas de compilación para TAG

Regla	Tarea
[INIT]	inicia los cálculos a partir de la pila inicial.
[FINAL]	finaliza los cálculos.
[CALL]	requiere el análisis de un determinado nodo que no forma parte de la espina de un árbol auxiliar.
[SCALL]	(de *spine call*) requiere el análisis de un determinado nodo que forma parte de la espina de un árbol auxiliar.
[SEL]	selecciona una producción de un árbol elemental.
[PUB]	determina que un nodo ha sido analizado.
[RET]	continua el proceso de análisis después del reconocimiento de un nodo que no forma parte de la espina de un árbol auxiliar.
[SRET]	(de *spine ret*) continua el proceso de análisis después del reconocimiento de un nodo que forma parte de la espina de un árbol auxiliar.
[SCAN]	lee los terminales que componen la cadena de entrada.
[ACALL]	inicia la operación de adjunción de un árbol en un nodo.
[ARET]	termina la operación de adjunción de un árbol en un nodo.
[FCALL]	comienza a reconocer la parte escindida de un árbol elemental que debe ser pegada al nodo pie de un árbol auxiliar.
[FRET]	termina de reconocer el nodo pie de un árbol auxiliar.

Tabla 6.5: Reglas para los esquemas de compilación de gramáticas de adjunción de árboles

Las estrategias utilizadas en el análisis de gramáticas de adjunción de árboles se definen mediante un par $\langle$*estrategia-CF*$\rangle - \langle$*estrategia-adjunción*$\rangle$ en el que *estrategia-CF* se refiere a la estrategia utilizada para el recorrido de los árboles elementales, que puede ser:

descendente si se predice el nodo que va a ser visitado pero no se propaga información acerca del nodo que acaba de ser visitado.

ascendente si no se predice el nodo que va a ser visitado pero se propaga información acerca del nodo que acaba de ser visitado.

Earley si se predice el nodo que va a ser visitado y se propaga información acerca del nodo que acaba de ser visitado.

mientras que *estrategia-adjunción* se refiere a la estrategia utilizada para el tratamiento de las adjunciones, que puede ser:

descendente si se almacena información del nodo de adjunción cuando se inicia la visita de la raíz del árbol auxiliar adjuntado.

ascendente si se almacena información del nodo de adjunción cuando se termina de visitar el nodo pie del árbol auxiliar adjuntado.

Earley si se almacena información del nodo de adjunción tanto cuando se inicia la visita de la raíz como cuando se termina de visitar el nodo pie del árbol auxiliar adjuntado.

Para referirnos a todas las estrategias con un determinado comportamiento con respecto al recorrido de los árboles elementales, independientemente del comportamiento con respecto al tratamiento de la adjunción, utilizaremos la notación $\langle$*estrategia-CF*$\rangle - *$. Para referirnos al conjunto de estrategias que presentan un determinado comportamiento respecto al tratamiento de las adjunciones, independientemente del comportamiento respecto al recorrido de los árboles elementales, utilizaremos la notación $* - \langle$*estrategia-adjunción*$\rangle$. Con ello logramos homogeneizar la nomenclatura de las estrategias de análisis.

En primer lugar definiremos una estrategia genérica basada en el paradigma llamada/retorno, parametrizada con respecto a la información que se predice y propaga en las fases de llamada y de retorno, respectivamente. Utilizaremos la siguiente notación:

- $\overrightarrow{N^\gamma_{r,s}}$ para referirnos a la predicción de información con respecto al nodo $N^\gamma_{r,s}$ del árbol elemental γ.
- $\overrightarrow{\infty}$ representa la información predicha de una pila completa de adjunciones pendientes.
- $\overleftarrow{N^\gamma_{r,s}}$ para representar la información propagada con respecto al nodo $N^\gamma_{r,s}$ del árbol elemental γ.
- $\overleftarrow{\infty}$ representa la información propagada de una pila completa de adjunciones pendientes.

La única condición que exigiremos sobre la forma de los árboles será que los terminales y la palabra vacía tan sólo deberán formar parte de producciones de la forma $N^\gamma_{r,0} \to a$, donde $a \in V_T \cup \{\epsilon\}$. Este hecho no restringe la capacidad expresiva de los gramáticas.

El esquema de compilación genérico de una gramática de adjunción de árboles en un autómata lineal de índices queda definido por el conjunto de reglas mostrado en la tabla 6.6 y los elementos inicial $\$_0[\,]$ y final $\overleftarrow{\top^\alpha}[\,]$, con $\alpha \in \boldsymbol{I}$.

[INIT]	$\$_0[\circ\circ] \longmapsto \$_0[\circ\circ]\ \overrightarrow{\top^\alpha}[\,]$	$\alpha \in \boldsymbol{I}$
[CALL]	$\nabla^\gamma_{r,s}[\circ\circ] \longmapsto \nabla^\gamma_{r,s}[\circ\circ]\ \overrightarrow{N^\gamma_{r,s+1}}[\,]$	$N^\gamma_{r,s+1} \notin \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SCALL]	$\nabla^\gamma_{r,s}[\circ\circ] \longmapsto \nabla^\gamma_{r,s}[\,]\ \overrightarrow{N^\gamma_{r,s+1}}[\overrightarrow{\circ\circ}]$	$N^\gamma_{r,s+1} \in \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SEL]	$\overrightarrow{N^\gamma_{r,0}}[\circ\circ] \longmapsto \overrightarrow{N^\gamma_{r,0}}[\,]\ \nabla^\gamma_{r,0}[\circ\circ]$	$r \neq 0$
[PUB]	$\overrightarrow{N^\gamma_{r,0}}[\,]\ \nabla^\gamma_{r,n_r}[\circ\circ] \longmapsto \overleftarrow{N^\gamma_{r,0}}[\circ\circ]$	
[RET]	$\nabla^\gamma_{r,s}[\circ\circ]\ \overleftarrow{N^\gamma_{r,s+1}}[\,] \longmapsto \nabla^\gamma_{r,s+1}[\circ\circ]$	$N^\gamma_{r,s+1} \notin \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SRET]	$\nabla^\gamma_{r,s}[\,]\ \overleftarrow{N^\gamma_{r,s+1}}[\overleftarrow{\circ\circ}] \longmapsto \nabla^\gamma_{r,s+1}[\overleftarrow{\circ\circ}]$	$N^\gamma_{r,s+1} \in \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SCAN]	$\nabla_{r,0}[\,] \overset{a}{\longmapsto} \nabla_{r,n_r}[\,]$	$N^\gamma_{r,0}[\,] \to a$
[ACALL]	$\nabla^\gamma_{r,s}[\circ\circ] \longmapsto \nabla^\gamma_{r,s}[\,]\ \overrightarrow{\top^\beta}[\overrightarrow{\circ\circ\, N^\gamma_{r,s+1}}]$	$\beta \in \text{adj}(N^\gamma_{r,s+1})$
[ARET]	$\nabla^\gamma_{r,s}[\,]\ \overleftarrow{\top^\beta}[\overleftarrow{\circ\circ\, N^\gamma_{r,s+1}}] \longmapsto \nabla^\gamma_{r,s+1}[\overleftarrow{\circ\circ}]$	$\beta \in \text{adj}(N^\gamma_{r,s+1})$
[FCALL]	$\nabla^\beta_{f,0}[\overrightarrow{\circ\circ\, N^\gamma_{r,s+1}}] \longmapsto \nabla^\beta_{f,0}[\,]\ \overrightarrow{N^\gamma_{r,s+1}}[\overrightarrow{\circ\circ}]$	$N^\beta_{f,0} = \mathbf{F}^\beta,\ \beta \in \text{adj}(N^\gamma_{r,s+1})$
[FRET]	$\nabla^\beta_{f,0}[\,]\ \overleftarrow{N^\gamma_{r,s+1}}[\overleftarrow{\circ\circ}] \longmapsto \nabla^\beta_{f,1}[\overleftarrow{\circ\circ\, N^\gamma_{r,s+1}}]$	$N^\beta_{f,0} = \mathbf{F}^\beta,\ \beta \in \text{adj}(N^\gamma_{r,s+1})$
[FINAL]	$\$_0[\circ\circ]\ \overleftarrow{\top^\alpha}[\,] \longmapsto \$_f[\circ\circ]$	$\alpha \in \boldsymbol{I}$

Tabla 6.6: Reglas del esquema de compilación genérico de TAG en LIA

El juego de instrucciones requerido por los esquemas de compilación de las diferentes familias de estrategias ($*$-ascendentes, $*$-descendentes y $*$-Earley) nos permitirán definir diferentes modelos de autómata especializados. En las secciones que siguen se describe la clase de los *autómatas lineales de índices orientados a la derecha*, que permiten definir estrategias en las cuales las pilas de índices se construyen de modo ascendente, de los *autómatas lineales de índices orientados a la izquierda*, en los cuales las pilas de índices se construyen de modo descendente, para finalizar con los *autómatas lineales de índices universales*, que permiten la descripción de estrategias mixtas.

6.2.2 Tabulación

La ejecución directa de un autómata lineal de índices puede tener complejidad de orden exponencial con respecto al tamaño de la cadena de entrada. Ello se debe a que en el caso de que varias transiciones sean aplicables en un configuración dada, el contenido de la pila del autómata deberá replicarse y cada transición deberá aplicarse sobre una copia diferente. Para conseguir una complejidad polinomial deberemos evitar replicar la pila del autómata. Un modo de conseguirlo consiste en diseñar una técnica de tabulación que permita trabajar con representaciones condensadas de las configuraciones en lugar de con las configuraciones completas. Dichas representaciones condensadas se denominan ítems y se almacenan en una tabla, por lo que pueden ser reutilizadas y compartidas.

Las gramáticas de adjunción de árboles poseen propiedades que utilizadas adecuadamente permiten asegurar que su análisis sintáctico se realiza en tiempo polinomial. Nos referimos a la independencia del contexto de la operación de adjunción, puesto que podemos determinar si un árbol auxiliar es adjuntable o no en un nodo a partir de la información proporcionada por dicho nodo. En el caso de los autómatas lineales de índices, esta propiedad se traduce en el hecho de que la aplicabilidad de una transición en un momento dado viene determinada por la coincidencia de uno o dos símbolos de la cima de la pila con los símbolos de la parte izquierda de la transición y por cierta información del contexto que viene dada por a lo sumo un elemento en la cima de la pila de índices asociada con uno de dichos símbolos.

Utilizando esta propiedad es posible diseñar una técnica de tabulación específica de los autómatas lineales de índices. Para ello, diferenciaremos los siguientes tipos de derivaciones:

Derivaciones de llamada. Son aquellas derivaciones producidas durante la fase de llamada o descendente de la estrategia.

Derivaciones de retorno. Son aquellas derivaciones producidas durante la fase de retorno o ascendente de la estrategia de análisis.

Derivaciones de puntos especiales. Son derivaciones que llevan a configuraciones que representan el inicio o el final de una pila de índices. En las estrategias $*$-ascendentes estas derivaciones son asimilables a las derivaciones de llamada. Sin embargo, en las estrategias $*$-Earley y $*$-descendentes ambos tipos de derivaciones deben ser diferenciados.

Para representar estos tipos de derivaciones definiremos las correspondientes clases de *ítems de llamada*, *ítems de retorno* e *ítems de puntos especiales*. En ellos se almacenará información suficiente para reconstruir la evolución de la pila del autómata. Para facilitar su manejo, consideraremos que cada ítem consta de dos partes diferenciadas:

- Una *cabeza*[1] que almacena aquella información del estado del autómata necesaria para poder determinar si una transición es aplicable. Habitualmente incluirá los no terminales de uno o dos elementos en la cima de la pila y el índice en la cima de la pila de índices.
- Una *cola*[2] que almacena la información que permitirá recuperar el resto de los componentes de las pilas de índices que se necesiten para verificar la congruencia de los ítems durante la aplicación de las transiciones.

Utilizaremos la notación [cabeza | cola] para representar un ítem. Dependiendo del modelo de autómata, será preciso incorporar más o menos información en la cabeza y/o en la cola.

6.3 Autómatas lineales de índices orientados a la derecha

Dada una gramática de adjunción de árboles, una estrategia que reconozca de modo ascendente las adjunciones ignorará la realización de posibles adjunciones durante su contacto con la parte *izquierda* de los nodos en la espina de un árbol en la fase descendente, mientras que durante la fase ascendente irá apilando las adjunciones pendientes cada vez que visite un nodo a la *derecha* de una espina, tal y como se muestra en la figura 6.2. Este tipo de estrategias son precisamente las que se pueden implementar en los autómatas lineales de índices orientados a la derecha (*Right-oriented Linear Indexed Automata*, R–LIA) [55, 57, 6], de ahí su nombre.

Los autómatas lineales de índices orientados a la derecha fueron definidos por Nederhof en [55] y [57] como una clase de los autómatas lineales de índices en la cual el juego de transiciones se restringe de tal modo que sólo los siguientes tipos de transiciones están disponibles:

[1] *head.*
[2] *rest.*

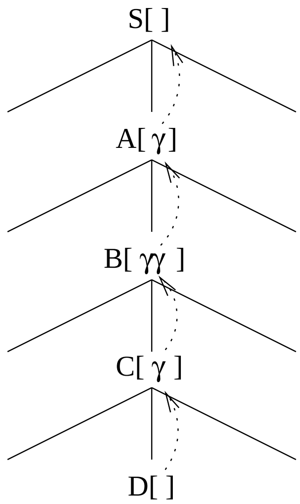

Figura 6.2: Construcción *orientada a la derecha* de las pilas de índices

- $C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ\gamma']$
- $C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,]$
- $C[\circ\circ\gamma]\ F[\,] \overset{a}{\longmapsto} G[\circ\circ\gamma']$
- $C[\,]\ F[\circ\circ\gamma] \overset{a}{\longmapsto} G[\circ\circ\gamma']$

6.3.1 Esquema de compilación

Las estrategias de análisis sintáctico que pueden ser implantadas en un autómata lineal de índices orientados a la derecha están limitadas por la forma de las transiciones permitidas, ya que sólo se puede apilar un elemento que tenga asociado una pila de índices vacía. Puesto que en un autómata a pila la información que se transmite en una operación de apilamiento representa la información propagada durante la fase descendente de un algoritmo de análisis, los autómatas lineales de índices orientados a la derecha no permiten definir estrategias de análisis descendentes con respecto a los contenidos de las pilas de índices. Esto es, las predicciones deben ser puramente independientes del contexto. En consecuencia, sólo permiten implementar estrategias de análisis sintáctico de gramáticas de adjunción de árboles en las cuales las adjunciones se tratan de manera ascendente. A continuación definimos un esquema de compilación genérico para estrategias $*$-ascendentes.

El esquema de compilación genérico de una gramática de adjunción de árboles en un autómata lineal de índices orientado a la derecha queda definido por el conjunto de reglas mostrado en la tabla 6.7 y por los elementos inicial $\$_0[\,]$ y final $\overleftarrow{\top}^{\alpha}[\,]$, con $\alpha \in \boldsymbol{I}$. Con respecto al esquema de compilación genérico, la regla de compilación **[SCAN]** ha sido modificada aunque dicha modificación no afecta a su comportamiento puesto que por construcción del esquema de compilación, la pila de índices asociada a cada nodo $\overrightarrow{\nabla^{\gamma}_{r,0}}$ es siempre $[\,]$. En la tabla 6.8 se muestran los tipos de transiciones de los R–LIA y su relación con las reglas de compilación.

En la tabla 6.9 se muestran los valores que deben tomar los parámetros de predicción y propagación de información para instanciar el esquema de compilación genérico en esquemas correspondientes a diferentes estrategias de análisis, donde $\Box$ representa un símbolo de pila nuevo. En el caso de la estrategia Earley-ascendente es necesario distinguir la llamada de un nodo $N^{\gamma}_{r,s+1}$ de su retorno, para lo cual utilizamos los símbolos $\overline{N^{\gamma}_{r,s+1}}$ y $\overline{\overline{N^{\gamma}_{r,s+1}}}$.

[INIT]	$\$_0[\circ\circ] \longmapsto \$_0[\circ\circ]\ \overrightarrow{\top^{\alpha}}[\,]$	$\alpha \in \boldsymbol{I}$
[CALL]	$\nabla^{\gamma}_{r,s}[\circ\circ] \longmapsto \nabla^{\gamma}_{r,s}[\circ\circ]\ \overrightarrow{N^{\gamma}_{r,s+1}}[\,]$	$N^{\gamma}_{r,s+1} \notin \text{espina}(\gamma),\ \mathbf{nil} \in \text{adj}(N_{r,s+1})$
[SCALL]	$\nabla^{\gamma}_{r,s}[\circ\circ] \longmapsto \nabla^{\gamma}_{r,s}[\circ\circ]\ \overrightarrow{N^{\gamma}_{r,s+1}}[\,]$	$N^{\gamma}_{r,s+1} \in \text{espina}(\gamma),\ \mathbf{nil} \in \text{adj}(N_{r,s+1})$
[SEL]	$\overrightarrow{N^{\gamma}_{r,0}}[\circ\circ] \longmapsto \overrightarrow{N^{\gamma}_{r,0}}[\,]\ \nabla^{\gamma}_{r,0}[\circ\circ]$	$r \neq 0$
[PUB]	$\overrightarrow{N^{\gamma}_{r,0}}[\,]\ \nabla^{\gamma}_{r,n_r}[\circ\circ] \longmapsto \overleftarrow{N^{\gamma}_{r,0}}[\circ\circ]$	
[RET]	$\nabla^{\gamma}_{r,s}[\circ\circ]\ \overleftarrow{N^{\gamma}_{r,s+1}}[\,] \longmapsto \nabla^{\gamma}_{r,s+1}[\circ\circ]$	$N^{\gamma}_{r,s+1} \notin \text{espina}(\gamma),\ \mathbf{nil} \in \text{adj}(N_{r,s+1})$
[SRET]	$\nabla^{\gamma}_{r,s}[\,]\ \overleftarrow{N^{\gamma}_{r,s+1}}[\circ\circ] \longmapsto \nabla^{\gamma}_{r,s+1}[\circ\circ]$	$N^{\gamma}_{r,s+1} \in \text{espina}(\gamma),\ \mathbf{nil} \in \text{adj}(N_{r,s+1})$
[SCAN]	$\nabla^{\gamma}_{r,0}[\circ\circ] \overset{a}{\longmapsto} \nabla^{\gamma}_{r,n_r}[\circ\circ]$	$N^{\gamma}_{r,0}[\,] \to a$
[ACALL]	$\nabla^{\gamma}_{r,s}[\circ\circ] \longmapsto \nabla^{\gamma}_{r,s}[\circ\circ]\ \overrightarrow{\top^{\beta}}[\,]$	$\beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[ARET]	$\nabla^{\gamma}_{r,s}[\,]\ \overleftarrow{\top^{\beta}}[\circ\circ N^{\gamma}_{r,s+1}] \longmapsto \nabla^{\gamma}_{r,s+1}[\circ\circ]$	$\beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FCALL]	$\nabla^{\beta}_{f,0}[\circ\circ] \longmapsto \nabla^{\beta}_{f,0}[\circ\circ]\ \overrightarrow{N^{\gamma}_{r,s+1}}[\,]$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta},\ \beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FRET]	$\nabla^{\beta}_{f,0}[\,]\ \overleftarrow{N^{\gamma}_{r,s+1}}[\circ\circ] \longmapsto \nabla^{\beta}_{f,1}[\circ\circ N^{\gamma}_{r,s+1}]$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta},\ \beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FINAL]	$\$_0[\circ\circ]\ \overleftarrow{\top^{\alpha}}[\,] \longmapsto \$_f[\circ\circ]$	$\alpha \in \boldsymbol{I}$

Tabla 6.7: Reglas del esquema de compilación genérico de TAG en R–LIA

Teorema 6.1 *Los autómatas lineales de índices orientados a la derecha aceptan la clase de los lenguajes de adjunción de árboles.*

Demostración:

Por el esquema de compilación de TAG en R–LIA sabemos que los lenguajes de adjunciones de árboles son aceptados por los autómatas lineales de índices orientados a la derecha.

Para mostrar que todo lenguaje aceptado por un R–LIA es un lenguaje de adjunción de árboles, definiremos un procedimiento para crear una gramática lineal de índices a partir de tales autómatas.

Sea $\mathcal{A} = (V_T, V_S, \$_0, \$_f, V_I, \mathcal{T})$ un autómata lineal de índices orientado a la derecha. Construiremos una gramática lineal de índices $\mathcal{L} = (V_T, V_N, V_I, S, P)$, donde el conjunto V_N de no terminales estará formado por pares $\langle E, B\rangle$ tal que $E, B \in V_S$. Para que $\mathcal{L}$ reconozca el lenguaje aceptado por $\mathcal{A}$ el conjunto de producciones en P ha de construirse a partir de las transiciones en $\mathcal{T}$ de la siguiente manera:

- Para toda transición $C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,]$ creamos una producción

$$\langle C, F\rangle[\,] \to a$$

- Para toda transición $C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ]$ y para todo $E \in V_S$ creamos una producción

$$\langle E, F\rangle[\circ\circ] \to \langle E, C\rangle[\circ\circ]\ a$$

Transición	Compilación de TAG
$C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,]$	**[INIT][CALL][SCALL][ACALL][FCALL]**
$C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ\gamma']$	**[SEL][PUB][SCAN]**
$C[\circ\circ\gamma]\ F[\,] \overset{a}{\longmapsto} G[\circ\circ\gamma']$	**[RET][FINAL]**
$C[\,]\ F[\circ\circ\gamma] \overset{a}{\longmapsto} G[\circ\circ\gamma']$	**[SRET][ARET][FRET]**

Tabla 6.8: Tipos de transiciones R–LIA

Estrategia-CF	**Estrategia-adjunción**	$\overrightarrow{N^{\gamma}_{r,s+1}}$	$\overrightarrow{\circ\circ\, N^{\gamma}_{r,s+1}}$	$\overleftarrow{N^{\gamma}_{r,s+1}}$	$\overleftarrow{\circ\circ\, N^{\gamma}_{r,s+1}}$
Ascendente	ascendente	$\square$	ϵ	$N^{\gamma}_{r,s+1}$	$\circ\circ N^{\gamma}_{r,s+1}$
Earley	ascendente	$\overline{N^{\gamma}_{r,s+1}}$	ϵ	$\overline{\overline{N^{\gamma}_{r,s+1}}}$	$\circ\circ N^{\gamma}_{r,s+1}$
Descendente	ascendente	$N^{\gamma}_{r,s+1}$	ϵ	$\square$	$\circ\circ N^{\gamma}_{r,s+1}$

Tabla 6.9: Parámetros del esquema de compilación genérico de TAG en R–LIA

- Para toda transición $C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ\gamma']$ y para todo $E \in V_S$ creamos una producción

$$\langle E, F\rangle[\circ\circ\gamma'] \to \langle E, C\rangle[\circ\circ]\ a$$

- Para toda transición $C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ]$ y para todo $E \in V_S$ creamos una producción

$$\langle E, F\rangle[\circ\circ] \to \langle E, C\rangle[\circ\circ\gamma]\ a$$

- Para toda transición $C[\circ\circ]\ F[\,] \overset{a}{\longmapsto} G[\circ\circ]$ y para todo $E \in V_S$ creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\circ\circ]\ \langle C, F\rangle[\,]\ a$$

- Para toda transición $C[\circ\circ]\ F[\,] \overset{a}{\longmapsto} G[\circ\circ\gamma']$ y para todo $E \in V_S$ creamos una producción

$$\langle E, G\rangle[\circ\circ\gamma'] \to \langle E, C\rangle[\circ\circ]\ \langle C, F\rangle[\,]\ a$$

- Para toda transición $C[\circ\circ\gamma]\ F[\,] \overset{a}{\longmapsto} G[\circ\circ]$ y para todo $E \in V_S$ creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\circ\circ\gamma]\ \langle C, F\rangle[\,]\ a$$

- Para toda transición $C[\,]\ F[\circ\circ] \overset{a}{\longmapsto} G[\circ\circ]$ y para todo $E \in V_S$ creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\,]\ \langle C, F\rangle[\circ\circ]\ a$$

- Para toda transición $C[\,]\ F[\circ\circ] \overset{a}{\longmapsto} G[\circ\circ\gamma']$ y para todo $E \in V_S$ creamos una producción

$$\langle E, G\rangle[\circ\circ\gamma'] \to \langle E, C\rangle[\,]\ \langle C, F\rangle[\circ\circ]\ a$$

- Para toda transición $C[\,]\ F[\circ\circ\gamma] \overset{a}{\longmapsto} G[\circ\circ]$ y para todo $E \in V_S$ creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\,]\ \langle C, F\rangle[\circ\circ\gamma]\ a$$

Con respecto al axioma de la gramática, tenemos que $S = \langle \$_0, \$_f\rangle$.

Mediante inducción en la longitud de las derivaciones, es posible mostrar que $\langle E, B\rangle[\alpha] \overset{*}{\Rightarrow} w$ si y sólo si $(E, w) \overset{*}{\vdash} (E\ B[\alpha], \epsilon)$. Esto es así puesto que:

- Si una derivación $(E, w) \overset{*}{\vdash} (E\ B[\alpha], \epsilon)$ es el resultado de aplicar la secuencia $t_1, \ldots, t_m$ de transiciones en $\mathcal{T}$, entonces existe una secuencia $p_1, \ldots, p_m$ de producciones en P tal que p_i es una producción creada a partir de t_i y la derivación derecha $\langle E, B\rangle[\alpha] \overset{*}{\Rightarrow} w$ resultado de aplicar $p_m, \ldots, p_1$ reconoce w.
- Si una derivación derecha $\langle E, B\rangle[\alpha] \overset{*}{\Rightarrow} w$ reconoce la cadena w como resultado de aplicar la secuencia $p_1, \ldots, p_m$ de producciones en P, entonces existe una secuencia de transiciones $t_1, \ldots, t_m$ tal que p_i es una producción creada a partir de t_i y la derivación $(E, w) \overset{*}{\vdash} (E\ B[\alpha], \epsilon)$ es el resultado de aplicar la secuencia de transiciones $t_m, \ldots, t_1$.

□

6.3.2 Tabulación

Extenderemos la técnica de Nederhof para la tabulación de autómatas a pila al caso de los R–LIA. Para ello nos basaremos en la observación de que toda configuración $(\Upsilon\ C[\beta\eta], a_{j+1} \ldots a_n)$ de un autómata lineal de índices orientado a la derecha puede ser clasificada en uno de los dos tipos de derivaciones que se definen a continuación:

Derivaciones de llamada. Corresponden a configuraciones en las que $\beta\eta = \epsilon$ e implican la existencia de la siguiente derivación:

$$(\Upsilon\ B[\,], a_{i+1} \ldots a_n) \overset{*}{\vdash} (\Upsilon\ C[\,], a_{j+1} \ldots a_n)$$

donde en toda la derivación no se ha modificado ni consultado Υ. En la figura 6.3 se muestra una representación gráfica de este tipo de transiciones.

Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ se cumple

$$(\Upsilon'\ B[\,], a_{i+1} \ldots a_n) \overset{*}{\vdash} (\Upsilon'\ C[\,], a_{j+1} \ldots a_n)$$

por lo que podemos representar este tipo de derivaciones mediante ítems de la forma

$$[B, i, C, j, - \mid -, -, -, -]$$

En los R–LIA, las derivaciones de puntos especiales no son distinguibles de las derivaciones de llamada.

Derivaciones de retorno. Corresponden a configuraciones en la que $\beta \in V_I^*$ y $\eta \in V_I$ e implican la existencia de la siguiente secuencia de subderivaciones:

$$\begin{aligned}(\Upsilon\ B[\,], a_{i+1} \ldots a_n)\ & \overset{*}{\vdash}_{d_1} (\Upsilon\ B[\,]\ \Upsilon_1\ D[\,], a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash}_{d_2} (\Upsilon\ B[\,]\ \Upsilon_1\ E[\beta], a_{q+1} \ldots a_n) \\ & \overset{*}{\vdash}_{d_3} (\Upsilon\ C[\beta\eta], a_{j+1} \ldots a_n)\end{aligned}$$

donde en toda la derivación no se ha modificado ni consultado Υ. En la subderivación d_2 no se ha modificado ni consultado $\Upsilon\ B[\,]\ \Upsilon_1$. Las dos ocurrencias de β son la misma pila en el sentido de que ha sido transmitida sin modificación alguna a través de d_3: es posible que se hayan apilado índices en β y que posteriormente se hayan extraído, pero no se permite la extracción de elementos de β, aunque posteriores operaciones de apilamiento den como resultado una copia de β. En la figura 6.4 se muestra una representación gráfica de este tipo de transiciones.

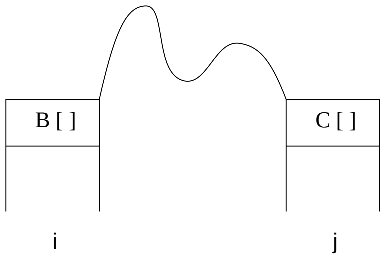

Figura 6.3: Derivaciones de llamada en R–LIA

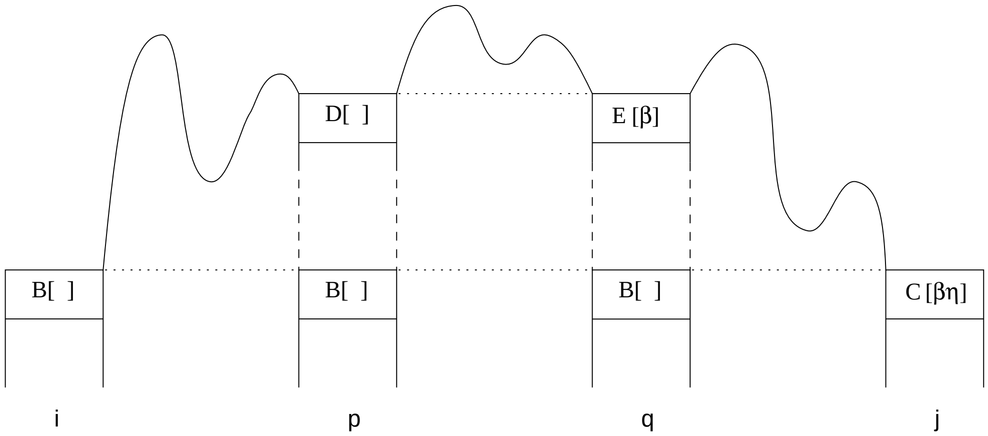

Figura 6.4: Derivaciones de retorno en R–LIA

Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ se cumple

$$\begin{aligned}(\Upsilon'\ B[\,], a_{i+1} \dots a_n) &\overset{*}{\vdash}_{d_1} (\Upsilon'\ B[\,]\ \Upsilon_1\ D[\,], a_{p+1} \dots a_n)\\ &\overset{*}{\vdash}_{d_2} (\Upsilon'\ B[\,]\ \Upsilon_1\ E[\beta], a_{q+1} \dots a_n)\\ &\overset{*}{\vdash}_{d_3} (\Upsilon'\ C[\beta\eta], a_{j+1} \dots a_n)\end{aligned}$$

En consecuencia, podemos utilizar ítems de la forma

$$[B, i, C, j, \gamma \mid D, p, E, q]$$

para representar este tipo de derivaciones.

Para completar la técnica de tabulación únicamente nos falta definir las reglas de inferencia de acuerdo con las transiciones permitidas en los R–LIA. Estas reglas se muestran en la tabla 6.10, donde:

- $k = j$ si $a = \epsilon$ y $k = j + 1$ si $a = a_{j+1}$.
- $l = k'$ si $b = \epsilon$ y $l = k' + 1$ si $b = a_{k'+1}$.

El papel de ítem inicial le corresponde a

$$[\$_0, 0, \$_0, 0, - \mid -, -, -, -]$$

mientras que los ítems finales son de la forma

$$[\$_0, 0, \$_f, n, - \mid -, -, -, -]$$

$$\frac{[B,i,C,j,\gamma \mid D,p,E,q]}{[F,k,F,k,- \mid -,-,-,-]} \quad C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,]$$

$$\frac{[B,i,C,j,\gamma \mid D,p,E,q]}{[B,i,F,k,\gamma \mid D,p,E,q]} \quad C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ]$$

$$\frac{[B,i,C,j,\gamma \mid D,p,E,q]}{[B,i,F,k,\gamma' \mid B,i,C,j]} \quad C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ\gamma']$$

$$\frac{\begin{array}{c}[B,i,C,j,\gamma \mid D,p,E,q]\\ [D,p,E,q,\gamma' \mid O,u,P,v]\end{array}}{[B,i,F,k,\gamma' \mid O,u,P,v]} \quad C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ]$$

$$\frac{\begin{array}{c}[F,k,F',k',- \mid -,-,-,-]\\ [B,i,C,j,\gamma \mid D,p,E,q]\end{array}}{[B,i,G,l,\gamma \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,] \\ C[\circ\circ]\ F'[\,] \overset{b}{\longmapsto} G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[F,k,F',k',- \mid -,-,-,-]\\ [B,i,C,j,\gamma \mid D,p,E,q]\end{array}}{[B,i,G,l,\gamma' \mid B,i,C,j]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,] \\ C[\circ\circ]\ F'[\,] \overset{b}{\longmapsto} G[\circ\circ\gamma']\end{array}$$

$$\frac{\begin{array}{c}[F,k,F',k',- \mid -,-,-,-]\\ [B,i,C,j,\gamma \mid D,p,E,q]\\ [D,p,E,q,\gamma' \mid O,u,P,v]\end{array}}{[B,i,G,l,\gamma' \mid O,u,P,v]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,] \\ C[\circ\circ\gamma]\ F'[\,] \overset{b}{\longmapsto} G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[F,k,F',k',\gamma \mid D,p,E,q]\\ [B,i,C,j,- \mid -,-,-,-]\end{array}}{[B,i,G,l,\gamma \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,] \\ C[\,]\ F'[\circ\circ] \overset{b}{\longmapsto} G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[F,k,F',k',\gamma \mid D,p,E,q]\\ [B,i,C,j,- \mid -,-,-,-]\end{array}}{[B,i,G,l,\gamma' \mid F,k,F',k']} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,] \\ C[\,]\ F'[\circ\circ] \overset{b}{\longmapsto} G[\circ\circ\gamma']\end{array}$$

$$\frac{\begin{array}{c}[F,k,F',k',\gamma \mid D,p,E,q]\\ [B,i,C,j,- \mid -,-,-,-]\\ [D,p,E,q,\gamma' \mid O,u,P,v]\end{array}}{[B,i,G,l,\gamma' \mid O,u,P,v]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,] \\ C[\,]\ F'[\circ\circ\gamma] \overset{b}{\longmapsto} G[\circ\circ]\end{array}$$

Tabla 6.10: Reglas de inferencia en R–LIA

Teorema 6.2 *La manipulación de configuraciones mediante la aplicación de transiciones en los autómatas lineales de índices orientados a la derecha es equivalente a la manipulación de ítems mediante las reglas de inferencia de la tabla 6.10.*

Demostración:

Puesto que un ítem representa una derivación y toda derivación debe ser representada por algún ítem, es suficiente con demostrar que la combinación de los ítems produce ítems que se corresponden con derivaciones válidas y que para toda derivación que se pueda producir como resultado de la aplicación de una transición, existe una regla de inferencia que produce un ítem que representa dicha derivación. Para ello debemos considerar una lista exhaustiva con todos los posibles casos de derivación que se pueden dar junto con la correspondiente regla de inferencia. Si aplicamos inducción en la longitud de una derivación, observaremos que para cualquier derivación obtenida mediante la aplicación de una transición, existe una regla de inferencia que busca los ítems correspondientes a las subderivaciones relevantes y que produce el ítem correspondiente a la derivación resultante. También podemos observar que dada una regla de inferencia, existen subderivaciones en el autómata que se corresponden con cada uno de los ítems antecedentes y que combinadas entre sí producen la derivación correspondiente al ítem consecuente. La demostración completa puede encontrarse en [3]. □

La complejidad temporal de la técnica de tabulación con respecto a longitud n de la cadena de entrada es de orden $\mathcal{O}(n^6)$ en el peor caso. Dicha complejidad viene dada por la regla de inferencia correspondiente a las transiciones del tipo $B[\ ]\ C[\circ\circ\gamma] \stackrel{a}{\longmapsto} F[\circ\circ]$, que involucra el manejo de 7 posiciones con respecto a la cadena de entrada, aunque sólo 6 de ellas de manera simultánea. La complejidad espacial es de orden $\mathcal{O}(n^4)$ puesto que cada ítem contiene cuatro posiciones con respeto a la cadena de entrada.

6.4 Autómatas lineales de índices orientados a la izquierda

Dada una gramática de adjunción de árboles, decimos que una estrategia reconoce las adjunciones de modo descendente si realiza el tratamiento de las mismas cuando visita la izquierda de los nodos de la espina de un árbol elemental, mientras que ignora la posibilidad de realizar adjunciones cuando visita la derecha de dichos nodos, tal y como se muestra en la figura 6.5. Este tipo de estrategias son precisamente las que se pueden implementar en los autómatas lineales de índices orientados a la izquierda (*Left-oriented Linear Indexed Automata, L–LIA*) [57], de ahí su nombre.

Los autómatas lineales de índices orientados a la izquierda fueron introducidos por Nederhof en [57]. La definición se basa nuevamente en limitar el tipo de transiciones permitidas en los autómatas lineales de índices. En este capítulo, sin embargo, presentamos un juego de transiciones ligeramente más restrictivo que el propuesto por el citado autor para facilitar la definición de la técnica de tabulación. De este modo, admitimos como válidas transiciones pertenecientes a los tipos siguientes:

- $C[\circ\circ] \stackrel{a}{\longmapsto} F[\circ\circ]$
- $C[\circ\circ] \longmapsto C[\circ\circ]\ F[\ ]$
- $C[\circ\circ\gamma] \longmapsto C[\ ]\ F[\circ\circ\gamma']$
- $C[\circ\circ]\ F[\ ] \longmapsto G[\circ\circ]$

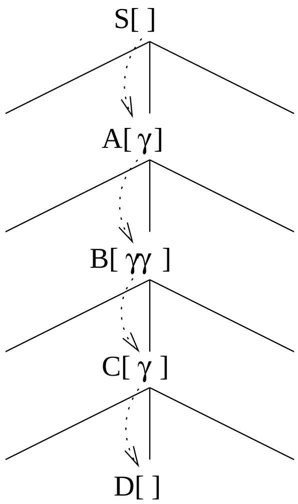

Figura 6.5: Construcción *orientada a la izquierda* de las pilas de índices

6.4.1 Esquema de compilación

Mediante los tipos de transiciones presentes en los L–LIA podemos implementar estrategias de análisis de gramáticas de adjunción de árboles $*$-descendentes, en las cuales las adjunciones se resuelven en la fase descendente. Para ello, cuando se visita la raíz de un árbol auxiliar β se almacenará en la pila de índices asociada el nodo en el cual se realizó la adjunción de β. Esa pila de índices es transportada a lo largo de la espina hasta alcanzar el nodo pie, momento en el cual su cima nos indicará el nodo en el cual debe continuar el análisis. En cambio, durante la fase ascendente la pila de índices asociada a los nodos visitados estará vacía.

En lo que respecta al recorrido de cada uno de los árboles elementales, no existe limitación alguna, lo cual nos permite definir un esquema de compilación genérico parametrizado precisamente en función de la información predicha y propagada con respecto a los nodos de los árboles elementales durante el recorrido de los mismos.

El esquema de compilación genérico de una gramática de adjunción de árboles en un autómata lineal de índices orientado a la izquierda queda definido por el conjunto de reglas mostrado en la tabla 6.11 y los elementos inicial $\$_0[\,]$ y final $\overleftarrow{\top^{\alpha}}[\,]$, con $\alpha \in \boldsymbol{I}$. Con respecto al esquema de compilación genérico, la regla de compilación **[SCAN]** ha sido modificada aunque dicha modificación no afecta a su comportamiento puesto que por construcción del esquema de compilación, la pila de índices asociada a cada nodo $\overrightarrow{\nabla^{\gamma}_{r,n_r}}$ es siempre $[\,]$. En la tabla 6.12 se muestra los tipos de transiciones y su relación con las reglas de compilación.

En la tabla 6.13 se muestran los valores que deben tomar los parámetros de predicción y propagación de información para instanciar el esquema de compilación genérico en esquemas correspondientes a diferentes estrategias de análisis.

Teorema 6.3 *Los autómatas lineales de índices orientados a la izquierda que utilizan el juego de transiciones de la tabla 6.12 aceptan la clase de los lenguajes de adjunción de árboles.*

<u>Demostración</u>:

> Por el esquema de compilación TAG en L–LIA sabemos que los lenguajes de adjunción de árboles son aceptados por los L–LIA que utilizan las transiciones de la tabla 6.12.
>
> Para demostrar que todo lenguaje aceptado por un L–LIA que utilice las transiciones de la tabla 6.12 es un lenguaje de adjunción de árboles, definiremos un procedimiento para crear una gramática lineal de índices a partir de tales autómatas.

[INIT]	$\$_0[\circ\circ] \longmapsto \$_0[\circ\circ]\ \overrightarrow{\top^\alpha}[\,]$	$\alpha \in \boldsymbol{I}$
[CALL]	$\nabla^\gamma_{r,s}[\circ\circ] \longmapsto \nabla^\gamma_{r,s}[\circ\circ]\ \overrightarrow{N^\gamma_{r,s+1}}[\,]$	$N^\gamma_{r,s+1} \not\in \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SCALL]	$\nabla^\gamma_{r,s}[\circ\circ] \longmapsto \nabla^\gamma_{r,s}[\,]\ \overrightarrow{N^\gamma_{r,s+1}}[\circ\circ]$	$N^\gamma_{r,s+1} \in \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SEL]	$\overrightarrow{N^\gamma_{r,0}}[\circ\circ] \longmapsto \overrightarrow{N^\gamma_{r,0}}[\,]\ \nabla^\gamma_{r,0}[\circ\circ]$	$r \neq 0$
[PUB]	$\overrightarrow{N^\gamma_{r,0}}[\,]\ \nabla^\gamma_{r,n_r}[\circ\circ] \longmapsto \overleftarrow{N^\gamma_{r,0}}[\circ\circ]$	
[RET]	$\nabla^\gamma_{r,s}[\circ\circ]\ \overleftarrow{N^\gamma_{r,s+1}}[\,] \longmapsto \nabla^\gamma_{r,s+1}[\circ\circ]$	$N^\gamma_{r,s+1} \not\in \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SRET]	$\nabla^\gamma_{r,s}[\circ\circ]\ \overleftarrow{N^\gamma_{r,s+1}}[\,] \longmapsto \nabla^\gamma_{r,s+1}[\circ\circ]$	$N^\gamma_{r,s+1} \in \text{espina}(\gamma),\ \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SCAN]	$\nabla^\gamma_{r,s}[\circ\circ] \overset{a}{\longmapsto} \nabla^\gamma_{r,n_r}[\circ\circ]$	$N^\gamma_{r,0}[\,] \to a$
[ACALL]	$\nabla^\gamma_{r,s}[\circ\circ] \longmapsto \nabla^\gamma_{r,s}[\,]\ \overrightarrow{\top^\beta}[\circ\circ N^\gamma_{r,s+1}]$	$\beta \in \text{adj}(N^\gamma_{r,s+1})$
[ARET]	$\nabla^\gamma_{r,s}[\circ\circ]\ \overleftarrow{\top^\beta}[\,] \longmapsto \nabla^\gamma_{r,s+1}[\circ\circ]$	$\beta \in \text{adj}(N^\gamma_{r,s+1})$
[FCALL]	$\nabla^\beta_{f,0}[\circ\circ N^\gamma_{r,s+1}] \longmapsto \nabla^\beta_{f,0}[\,]\ \overrightarrow{N^\gamma_{r,s+1}}[\circ\circ]$	$N^\beta_{f,0} = \mathbf{F}^\beta,\ \beta \in \text{adj}(N^\gamma_{r,s+1})$
[FRET]	$\nabla^\beta_{f,0}[\circ\circ]\ \overleftarrow{N^\gamma_{r,s+1}}[\,] \longmapsto \nabla^\beta_{f,1}[\circ\circ]$	$N^\beta_{f,0} = \mathbf{F}^\beta,\ \beta \in \text{adj}(N^\gamma_{r,s+1})$
[FINAL]	$\$_0[\circ\circ]\ \overleftarrow{\top^\alpha}[\,] \longmapsto \$_f[\circ\circ]$	$\alpha \in \boldsymbol{I}$

Tabla 6.11: Reglas del esquema de compilación genérico de TAG en L–LIA

Sea $\mathcal{A} = (V_T, V_S, \$_0, \$_f, V_I, \mathcal{T})$ un autómata lineal de índices orientado a la izquierda. Construiremos una gramática lineal de índices $\mathcal{L} = (V_T, V_N, V_I, S, P)$, donde el conjunto V_N de no terminales estará formado por pares $\langle E, B\rangle$ tal que $E, B \in V_S$. Para que $\mathcal{L}$ reconozca el lenguaje aceptado por $\mathcal{A}$, el conjunto de producciones en P ha de construirse a partir de las transiciones en $\mathcal{T}$ de la siguiente manera:

- Para toda transición $C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ]$ y para todo $E \in V_S$ creamos una producción
$$\langle C, E\rangle[\circ\circ] \to a\ \langle F, E\rangle[\circ\circ]$$
- Para todo par de transiciones $C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]$ y $C[\circ\circ] \longmapsto C[\circ\circ]\ F'[\,]$, y para todo $E \in V_S$ creamos una producción
$$\langle C, E\rangle[\circ\circ] \to \langle F', F\rangle[\,]\ \ \langle G, E\rangle[\circ\circ]$$
- Para todo par de transiciones $C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]$ y $C[\circ\circ\gamma] \longmapsto C[\,]\ F'[\circ\circ\gamma']$, y para todo $E \in V_S$ creamos una producción
$$\langle C, E\rangle[\circ\circ\gamma] \to \langle F', F\rangle[\circ\circ\gamma']\ \ \langle G, E\rangle[\,]$$
- Para todo $E \in V_S$ creamos una producción
$$\langle E, E\rangle[\,] \to \epsilon$$
- Para toda transición $\$_0[\circ\circ] \longmapsto \$_0[\circ\circ]\ F[\,]$ o $\$_0[\circ\circ] \longmapsto \$_0[\circ\circ]\ F[\circ\circ]$, donde $F \in V_S - \{\$_0\}$, creamos una producción
$$\langle \$_0, \$_0\rangle[\circ\circ] \to \langle F, \$_f\rangle[\circ\circ]$$

Transición	**Compilación de TAG**
$C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ]$	**[SEL][PUB][SCAN]**
$C[\circ\circ] \longmapsto C[\circ\circ]\ F[\,]$	**[INIT][CALL]**
$C[\circ\circ\gamma] \longmapsto C[\,]\ F[\circ\circ\gamma']$	**[SCALL][ACALL][FCALL]**
$C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]$	**[RET][SRET][ARET][FRET][FINAL]**

Tabla 6.12: Tipos de transiciones L–LIA

Estrategia-CF	**Estrategia-adjunción**	$\overrightarrow{N^{\gamma}_{r,s+1}}$	$\overrightarrow{\circ\circ\, N^{\gamma}_{r,s+1}}$	$\overleftarrow{N^{\gamma}_{r,s+1}}$	$\overleftarrow{\circ\circ\, N^{\gamma}_{r,s+1}}$
Ascendente	descendente	$\Box$	$\circ\circ N^{\gamma}_{r,s+1}$	$N^{\gamma}_{r,s+1}$	ϵ
Earley	descendente	$\overline{N^{\gamma}_{r,s+1}}$	$\circ\circ N^{\gamma}_{r,s+1}$	$\overline{\overline{N^{\gamma}_{r,s+1}}}$	ϵ
Descendente	descendente	$N^{\gamma}_{r,s+1}$	$\circ\circ N^{\gamma}_{r,s+1}$	$\Box$	ϵ

Tabla 6.13: Parámetros del esquema de compilación genérico de TAG en L–LIA

- Para toda transición $C[\circ\circ] \overset{a}{\longmapsto} \$_f[\circ\circ]$ creamos una transición

$$\langle C, \$_f \rangle[\,] \rightarrow a$$

Con respecto al axioma de la gramática, tenemos que $S = \langle \$_0, \$_0 \rangle$.

Mediante inducción en la longitud de las derivaciones, es posible mostrar que $\langle C, E\rangle[\alpha] \overset{*}{\Rightarrow} w$ si y sólo si $(C[\alpha], w) \overset{*}{\vdash} (E[\,], \epsilon)$, puesto que

- Si una derivación $(C[\alpha], w) \overset{*}{\vdash} (E[\,], \epsilon)$ es el resultado de aplicar la secuencia $t_1, \ldots, t_m$ de transiciones en $\mathcal{T}$, entonces existe una secuencia $p_1, \ldots, p_m$ de producciones en P tal que la derivación $\langle C, E\rangle[\alpha] \overset{*}{\Rightarrow} w$ resultado de aplicar $p_1, \ldots, p_m$ reconoce w. La demostración se realiza por inducción en la longitud de la derivación del autómata.

  El caso base lo constituye la derivación $(E[\,], \epsilon) \overset{0}{\vdash} (E[\,], \epsilon)$, para la que existe una producción $\langle E, E\rangle[\,] \rightarrow \epsilon$. Por hipótesis de inducción suponemos que la proposición se cumple para cualquier derivación del autómata de longitud m. En tal caso, durante el paso de inducción verificamos que se cumple para cualquier posible derivación de longitud mayor que m:

  - Si $(C[\alpha], aw) \vdash (F[\alpha], w) \overset{m}{\vdash} (E[\,], \epsilon)$, $\exists \langle C, E\rangle[\circ\circ] \rightarrow a\ \langle F, E\rangle[\circ\circ] \in P$, por hipótesis de inducción $\langle F, E\rangle[\alpha] \overset{*}{\Rightarrow} w$ y en consecuencia $\langle C, E\rangle[\alpha] \overset{*}{\Rightarrow} aw$.
  - Si $(C[\alpha], w_1 w_2) \vdash (C[\alpha]\ F'[\,], w_1 w_2) \overset{m_1}{\vdash} (C[\alpha]\ F[\,], w_2) \vdash (G[\alpha], w_2) \overset{m_2}{\vdash} (E[\,], \epsilon)$, existe una producción $\langle C, E\rangle[\circ\circ] \rightarrow \langle F', F\rangle[\,]\ \langle G, E\rangle[\circ\circ]$, por hipótesis de inducción $\langle F', F\rangle[\,] \overset{*}{\Rightarrow} w_1$ y $\langle G, E\rangle[\alpha] \overset{*}{\Rightarrow} w_2$ y en consecuencia $\langle B, E\rangle[\alpha] \overset{*}{\Rightarrow} w_1 w_2$.
  - Si $(C[\alpha\gamma], w_1 w_2) \vdash (C[\,]\ F'[\alpha\gamma'], w_1 w_2) \overset{m_1}{\vdash} (C[\,]\ F[\,], w_2) \vdash (G[\,], w_2) \overset{m_2}{\vdash} (E[\,], \epsilon)$, existe una producción $\langle C, E\rangle[\circ\circ\gamma] \rightarrow \langle F', F\rangle[\circ\circ\gamma']\ \langle G, E\rangle[\,]$, por hipótesis de inducción $\langle F', F\rangle[\alpha\gamma'] \overset{*}{\Rightarrow} w_1$ y $\langle G, E\rangle[\,] \overset{*}{\Rightarrow} w_2$ y en consecuencia $\langle C, E\rangle[\alpha\gamma] \overset{*}{\Rightarrow} w_1 w_2$.

- Si una derivación izquierda $\langle C, E\rangle[\alpha] \overset{*}{\Rightarrow} w$ reconoce la cadena w como resultado de aplicar la secuencia $p_1, \ldots, p_m$ de producciones en P, entonces existe una secuencia de transiciones $t_1, \ldots, t_m$ tal que la derivación $(C[\alpha], w) \overset{*}{\vdash} (E[\ ], \epsilon)$ es el resultado de aplicar la secuencia de transiciones $t_1, \ldots, t_m$. La demostración se realiza por inducción en la longitud de la derivación de la gramática. El caso base lo constituye la derivación $\langle E, E\rangle[\ ] \Rightarrow \epsilon$, para la que existe una derivación $(E[\ ], \epsilon) \overset{0}{\vdash} (E[\ ], \epsilon)$ en el autómata. Por hipótesis de inducción suponemos que la proposición se cumple para cualquier derivación de la gramática de longitud m. En tal caso, durante el paso de inducción verificamos que se cumple para cualquier posible derivación de longitud mayor que m:

- Si $\langle C, E\rangle[\alpha] \Rightarrow a\ \langle F, E\rangle[\alpha] \overset{m}{\Rightarrow} aw$, existe una transición $C[\circ\circ] \overset{a}{\longmapsto} F[\circ\circ]$, por hipótesis de inducción $(F[\alpha], w) \overset{*}{\vdash} (E[\ ], \epsilon)$ y en consecuencia $(C[\alpha], aw) \overset{*}{\vdash} (E[\ ], \epsilon)$.
- Si $\langle C, E\rangle[\alpha] \Rightarrow \langle F', F\rangle[\ ]\ \langle G, E\rangle[\alpha] \overset{m_1}{\Rightarrow} w_1\ \langle G, E\rangle[\alpha] \overset{m_2}{\Rightarrow} w_1 w_2$, existe un par de transiciones $C[\circ\circ] \longmapsto C[\circ\circ]\ F'[\ ]$ y $C[\circ\circ]\ F[\ ] \longmapsto G[\circ\circ]$, por hipótesis de inducción $(F'[\ ], w_1) \overset{*}{\vdash} (F[\ ], \epsilon)$ y $(G[\alpha], w_2) \overset{*}{\vdash} (E[\ ], \epsilon)$ y en consecuencia $(C[\alpha], w_1 w_2) \overset{*}{\vdash} (E[\ ], \epsilon)$.
- Si $\langle C, E\rangle[\alpha\gamma] \Rightarrow \langle F', F\rangle[\alpha\gamma']\ \langle G, E\rangle[\ ] \overset{m_1}{\Rightarrow} w_1\ \langle G, E\rangle[\ ] \overset{m_2}{\Rightarrow} w_1 w_2$, existe un par de transiciones $C[\circ\circ\gamma] \longmapsto C[\ ]\ F'[\circ\circ\gamma']$ y $C[\circ\circ]\ F[\ ] \longmapsto G[\circ\circ]$, por hipótesis de inducción $(F'[\alpha\gamma'], w_1) \overset{*}{\vdash} (F[\ ], \epsilon)$ y $(G[\ ], w_2) \overset{*}{\vdash} (E[\ ], \epsilon)$ y en consecuencia $(C[\alpha\gamma], w_1 w_2) \overset{*}{\vdash} (E[\ ], \epsilon)$.

$\square$

6.4.2 Tabulación

Presentamos en esta sección una técnica de tabulación alternativa a la propuesta por Nederhof en [57], la cual contiene errores e inconsistencias. Una diferencia fundamental es que el conjunto de transiciones que se considera es distinto, como se ha comentado anteriormente. Para proceder al diseño de la técnica de tabulación debemos primero definir los distintos tipos de derivaciones observables en un L–LIA:

Derivaciones de llamada. Corresponden a la transmisión de una pila de índices en la fase de llamada de la estrategia de análisis y son de la forma

$$\begin{aligned}(\Upsilon\ A[\delta], a_{h+1} \ldots a_n) &\overset{*}{\vdash} (\Upsilon\ A[\ ]\ \Upsilon_1\ B[\delta\gamma], a_{i+1} \ldots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\ ]\ \Upsilon_1\ C[\delta\gamma], a_{j+1} \ldots a_n)\end{aligned}$$

donde $\gamma \in V_I$, $\delta \in (V_S[V_I^*])^*$, tanto $B[\alpha\gamma]$ como $C[\alpha\gamma]$ son descendientes dependientes de $A[\alpha]$ y no existe un par $(F[\delta\gamma], f) \neq (B[\delta\gamma], j)$ tal que

$$\begin{aligned}(\Upsilon\ A[\delta], a_{h+1} \ldots a_n) &\overset{*}{\vdash} (\Upsilon\ A[\ ]\ \Upsilon_1\ F[\delta\gamma], a_{f+1} \ldots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\ ]\ \Upsilon_1\ B[\delta\gamma], a_{i+1} \ldots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\ ]\ \Upsilon_1\ C[\delta\gamma], a_{j+1} \ldots a_n)\end{aligned}$$

La figura 6.6 muestra una representación gráfica de las derivaciones de llamada.

Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ y $\alpha \in V_I*$ se cumple que:

$$\begin{aligned}(\Upsilon'\ A[\alpha], a_{h+1} \ldots a_n) &\overset{*}{\vdash} (\Upsilon'\ A[\ ]\ \Upsilon_1\ B[\alpha\gamma], a_{i+1} \ldots a_n)\\ &\overset{*}{\vdash} (\Upsilon'\ A[\ ]\ \Upsilon_1\ C[\alpha\gamma], a_{j+1} \ldots a_n)\end{aligned}$$

En consecuencia, este tipo de derivaciones se pueden representar mediante ítems de la forma

$$[A, h \mid B, i, \gamma, C, j, \gamma \mid -, -, -, -]$$

Derivaciones de retorno. Corresponden a la propagación de una pila durante la fase de retorno de la estrategia de análisis y son de la forma

$$\begin{aligned}(\Upsilon\ A[\delta], a_{h+1} \dots a_n) &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\delta\gamma], a_{i+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ D[\delta], a_{p+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ E[\,], a_{q+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ C[\,], a_{j+1} \dots a_n)\end{aligned}$$

donde $\gamma \in V_I$, $\delta \in V_I^*$, tanto $B[\alpha\gamma]$ como $D[\alpha]$ son descendientes dependientes de $A[\alpha]$ y no existen $(F[\delta\gamma], f) \neq (B[\delta\gamma], i)$ ni $(G[\delta], g) \neq (D[\delta], p)$ tal que

$$\begin{aligned}(\Upsilon\ A[\delta], a_{h+1} \dots a_n) &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ F[\delta\gamma], a_{f+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\delta\gamma], a_{i+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ G[\delta], a_{g+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ D[\delta], a_{p+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ E[\,], a_{q+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ C[\,], a_{j+1} \dots a_n)\end{aligned}$$

La figura 6.7 muestra una representación gráfica de las derivaciones de retorno.

Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ y pila de índices $\delta' \in V_I^*$ tal que existe una derivación $(D[\delta'], a_{p+1} \dots a_n) \overset{*}{\vdash} (E[\,], a_{q+1} \dots a_n)$ se cumple que

$$\begin{aligned}(\Upsilon'\ A[\delta'], a_{h+1} \dots a_n) &\overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\delta'\gamma], a_{i+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ D[\delta'], a_{p+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ E[\,], a_{q+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ C[\,], a_{j+1} \dots a_n)\end{aligned}$$

Ello permite representar este tipo de derivaciones mediante ítems de la forma

$$[A, h \mid B, i, \gamma, C, j, - \mid D, p, E, q]$$

donde los componentes (A, h) y (D, p, E, q) permiten asegurar que estamos trabajando con la misma pila de índices δ a lo largo de toda la derivación.

Derivaciones de puntos especiales. Son aquellas de la forma

$$(\Upsilon\ B[\,], a_{i+1} \dots a_n) \overset{*}{\vdash} (\Upsilon\ C[\,], a_{j+1} \dots a_n)$$

y no existe $(F[\,], f) \neq (B[\,], i)$ tal que

$$\begin{aligned}(\Upsilon\ F[\,], a_{f+1} \dots a_n) &\overset{*}{\vdash} (\Upsilon\ B[\,], a_{i+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ C[\,], a_{j+1} \dots a_n)\end{aligned}$$

La representación gráfica de las derivaciones de puntos especiales se muestra en la figura 6.8. Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ se cumple que

$$(\Upsilon' \; B[\,], a_{i+1} \ldots a_n) \;\; \overset{*}{\vdash} (\Upsilon' \; C[\,], a_{j+1} \ldots a_n)$$

por lo que este tipo de derivaciones puede ser representado mediante ítems de la forma

$$[-, - \mid B, i, -, C, j, - \mid -, -, -, -]$$

En la tablas 6.14 y 6.15 se muestran las reglas de inferencia para los diferentes tipos de transiciones. El ítem inicial es

$$[-, - \mid \$_0, 0, -, \$_0, 0, - \mid -, -, -, -]$$

y los ítems finales son de la forma

$$[-, - \mid B, 0, -, \$_f, n, - \mid -, -, -, -]$$

tal que existe una transición $\$_0[\circ\circ] \longmapsto \$_0[\,] \; B[\circ\circ]$ o bien una transición $\$_0[\circ\circ] \longmapsto \$_0[\circ\circ] \; B[\,]$.

Teorema 6.4 *La manipulación de configuraciones mediante la aplicación de transiciones en los autómatas lineales de índices orientados a la izquierda es equivalente a la manipulación de ítems mediante las reglas de inferencia de las tablas 6.14 y 6.15.*

Demostración:

La demostración en análoga a la realizada para el caso de los autómatas lineales de índices orientados a la derecha. Se realiza mediante inducción en la longitud de las derivaciones tras considerar la lista exhaustiva de todas las posibles derivaciones que se pueden obtener tras la aplicación de una transición del autómata. La demostración completa puede encontrase en [3]. □

La complejidad espacial de la técnica de tabulación con respecto a la longitud n de la cadena de entrada es $\mathcal{O}(n^5)$ puesto que cada ítem almacena 5 posiciones de la cadena de entrada. La complejidad temporal en el peor caso es $\mathcal{O}(n^7)$ y viene dada por la regla de inferencia

$$\frac{\begin{array}{l}[C, j \mid F', j, \gamma', F, k, - \mid D, p, E, q] \\ [A, h \mid B, i, \gamma, C, j, \gamma \mid -, -, -, -] \\ [A, h \mid D, p, \gamma, E, q, - \mid O, u, P, v]\end{array}}{[A, h \mid B, i, \gamma, G, k, - \mid O, u, P, v]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,] \; F'[\circ\circ\gamma'] \\ C[\circ\circ] \; F[\,] \longmapsto G[\circ\circ] \end{array}$$

puesto que aunque se combinan ocho posiciones de la cadena de entrada, mediante aplicación parcial sólo es preciso utilizar simultáneamente siete de dichas posiciones. La complejidad temporal puede reducirse utilizando la técnica propuesta en [27, 56], consistente en dividir la regla mencionada en dos reglas de menor complejidad, de tal modo que la primera genere un pseudo-ítem intermedio que proporcione la información relevante para la segunda. En el caso que nos ocupa, se trata de repartir la información proporcionada por el ítem $[C, j \mid F', j, \gamma', F, k, - \mid D, p, E, q]$ entre las dos nuevas reglas

$$\frac{\begin{array}{l}[C, j \mid F', j, \gamma', F, k, - \mid D, p, E, q] \\ [A, h \mid D, p, \gamma, E, q, - \mid O, u, P, v]\end{array}}{[[C, j \mid F', j, \gamma', F, k, - \mid O, u, P, v]]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,] \; F'[\circ\circ\gamma'] \\ C[\circ\circ] \; F[\,] \longmapsto G[\circ\circ] \end{array}$$

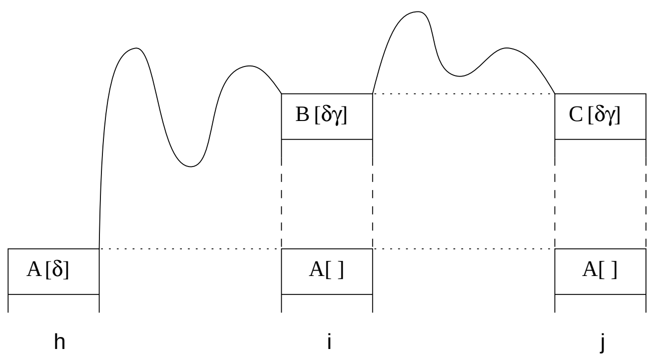

Figura 6.6: Derivaciones de llamada en L–LIA

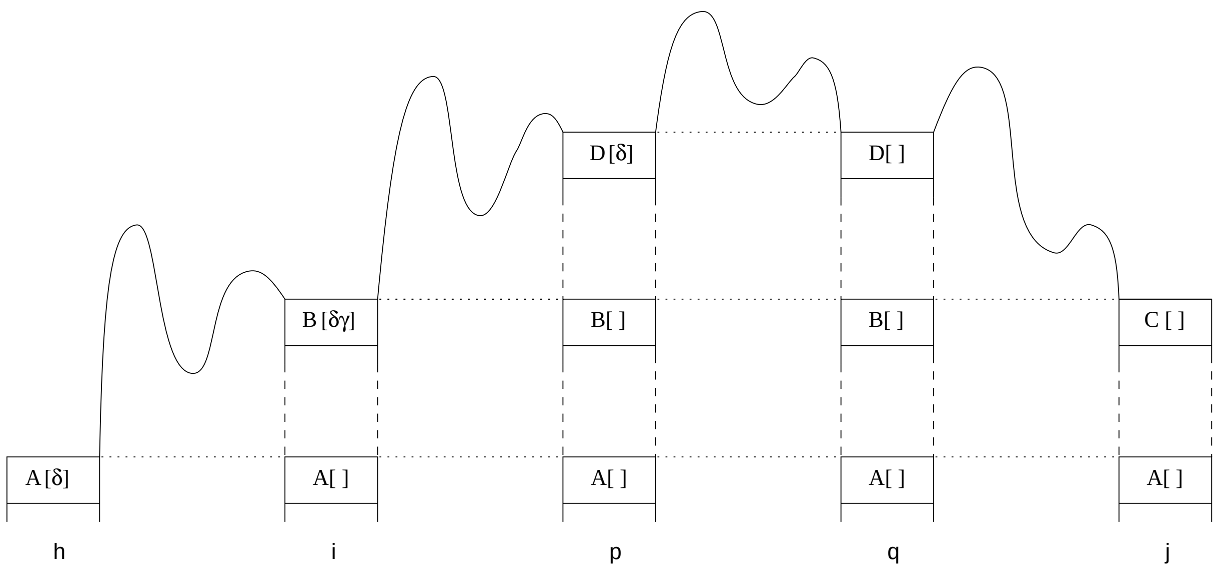

Figura 6.7: Derivaciones de retorno en L–LIA

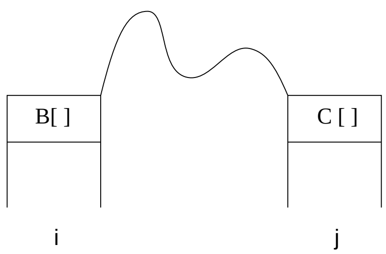

Figura 6.8: Derivaciones de puntos especiales en L–LIA

$$\frac{[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]}{[A,h \mid B,i,\gamma,F,k,\gamma \mid -,-,-,-]} \quad C[\circ\circ] \stackrel{a}{\longmapsto} F[\circ\circ],\ k=j \text{ si } a=\epsilon,\ k=j+1 \text{ si } a \in V_T$$

$$\frac{[A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]}{[A,h \mid B,i,\gamma,F,k,- \mid D,p,E,q]} \quad C[\circ\circ] \stackrel{a}{\longmapsto} F[\circ\circ],\ k=j \text{ si } a=\epsilon,\ k=j+1 \text{ si } a \in V_T$$

$$\frac{[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]}{[-,- \mid F,j,-,F,j,- \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\circ\circ]\ F[\,]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]}{[-,- \mid F,j,-,F,j,- \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\circ\circ]\ F[\,]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]}{[A,h \mid F,j,\gamma,F,j,\gamma \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\,]\ F[\circ\circ]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]}{[-,- \mid F,j,-,F,j,- \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\,]\ F[\circ\circ]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]}{[C,j \mid F,j,\gamma',F,j,\gamma' \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\,]\ F[\circ\circ\gamma']$$

$$\frac{[A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]}{[C,j \mid F,j,\gamma',F,j,\gamma' \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\,]\ F[\circ\circ\gamma']$$

$$\frac{\begin{array}{c}[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [M,m \mid N,t,\gamma',A,h,\gamma' \mid -,-,-,-]\end{array}}{[M,m \mid F,j,\gamma',F,j,\gamma' \mid -,-,-,-]} \quad C[\circ\circ\gamma] \longmapsto C[\,]\ F[\circ\circ]$$

$$\frac{\begin{array}{c}[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [M,m \mid N,t,\gamma',A,h,- \mid D,p,E,q]\end{array}}{[-,- \mid F,j,-,F,j,- \mid -,-,-,-]} \quad C[\circ\circ\gamma] \longmapsto C[\,]\ F[\circ\circ]$$

Tabla 6.14: Reglas de inferencia en L–LIA (fase de llamada)

$$\frac{\begin{array}{c}[-,- \mid F',j,-,F,k,- \mid -,-,-,-]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\end{array}}{[A,h \mid B,i,\gamma,G,k,\gamma \mid -,-,-,-]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\circ\circ]\ F'[\,] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[-,- \mid F',j,-,F,k,- \mid -,-,-,-]\\ [A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\circ\circ]\ F'[\,] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[A,h \mid F',j,\gamma,F,k,- \mid D,p,E,q]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[-,- \mid F',j,-,F,k,- \mid -,-,-,-]\\ [A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[C,j \mid F',j,\gamma',F,k,- \mid D,p,E,q]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [A,h \mid D,p,\gamma,E,q,- \mid O,u,P,v]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid O,u,P,v]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma'] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[C,j \mid F',j,\gamma',F,k,- \mid O,u,P,v]\\ [A,h \mid B,i,\gamma,C,j,- \mid D,p,E,q]\\ [-,- \mid O,u,-,P,v,- \mid -,-,-,-]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma'] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[M,m \mid F',j,\gamma',F,k,- \mid D,p,E,q]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [M,m \mid N,t,\gamma',A,h,\gamma' \mid -,-,-,-]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid F',j,F,k]} \quad \begin{array}{l} C[\circ\circ\gamma] \longmapsto C[\,]\ F'[\circ\circ] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

$$\frac{\begin{array}{c}[-,- \mid F',j,-,F,k,- \mid -,-,-,-]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [M,m \mid N,t,\gamma',A,h,- \mid D,p,E,q]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid F',j,F,k]} \quad \begin{array}{l} C[\circ\circ\gamma] \longmapsto C[\,]\ F'[\circ\circ] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ] \end{array}$$

Tabla 6.15: Reglas de inferencia en L–LIA (fase de retorno)

$$\frac{\begin{array}{l}[[C,j \mid F',j,\gamma',F,k,- \mid O,u,P,v]] \\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-] \\ [A,h \mid D,p,\gamma,E,q,- \mid O,u,P,v]\end{array}}{[A,h \mid B,i,\gamma,G,k,- \mid O,u,P,v]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma'] \\ C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]\end{array}$$

donde $[[C,j \mid F',j,\gamma',F,k,- \mid O,u,P,v]]$ es un pseudo-ítem intermedio que relaciona las dos reglas. La primera regla ignora la posición h, que es posteriormente recuperada del segundo y tercer ítem que intervienen en la segunda regla, que junto con el pseudo-ítem son suficientes para garantizar la existencia del ítem $[C,j \mid F',j,\gamma',F,k,- \mid D,p,E,q]$ por la definición de derivaciones de llamada y retorno. La primera regla presenta una complejidad temporal $\mathcal{O}(n^6)$ (la posición h no interviene) y la segunda presenta también una complejidad $\mathcal{O}(n^6)$ (las posiciones p y q no intervienen) por lo que hemos logrado rebajar la complejidad temporal en el peor caso de la técnica de tabulación a $\mathcal{O}(n^6)$.

6.5 Autómatas lineales de índices universales

Un autómata lineal de índices no orientado (*Non-oriented Lineal Indexed Automata*, N-LIA) es aquel capaz de describir estrategias de análisis para gramáticas lineales de índices en las cuales las pilas de índices pueden ser construidas durante la fase descendente, durante la fase ascendente o en ambas fases. En el caso de las gramáticas de adjunción de árboles, dichos autómatas permitirían definir algoritmos de análisis en los cuales las adjunciones se reconociesen de forma descendente, ascendente o mixta.

Debido a su carencia de orientación, los N-LIA deben poder utilizar un juego de transiciones que sea el resultado de unir las transiciones permitidas en los R–LIA con aquellas permitidas en los L–LIA:

- $C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ\gamma']$
- $C[\circ\circ] \overset{a}{\longmapsto} C[\circ\circ]\ F[\,]$
- $C[\circ\circ\gamma] \overset{a}{\longmapsto} C[\,]\ F[\circ\circ\gamma']$
- $C[\circ\circ\gamma]\ F[\,] \overset{a}{\longmapsto} G[\circ\circ\gamma']$
- $C[\,]\ F[\circ\circ\gamma] \overset{a}{\longmapsto} G[\circ\circ\gamma']$

Sin embargo, la utilización de este juego de transiciones permite definir autómatas que aceptan lenguajes que no son lenguajes de adjunción de árboles, tal y como se muestra en el siguiente ejemplo.

Ejemplo 6.2 El siguiente conjunto de transiciones define un N-LIA que acepta el lenguaje $\{a_1^n \ldots a_{2k}^n \mid k > 0, n \geq 0\}$. En dichas transiciones, el subíndice i denota números enteros impares mientras que el subíndice j denota números enteros pares.

$$
\begin{array}{ll}
(a) & \$_0[\circ\circ] \longmapsto \$_0[\circ\circ]\ A_1[\,] \\
(b) & A_1[\circ\circ] \overset{a_1}{\longmapsto} A_1[\,]\ A_1[\circ\circ\gamma] \\
(c) & A_i[\circ\circ] \longmapsto A_{i+1}[\circ\circ] \\
(d) & A_j[\circ\circ\gamma] \overset{a_j}{\longmapsto} A_j[\,]\ A_j[\circ\circ] \\
(e) & A_j[\circ\circ] \longmapsto A_{j+1}[\circ\circ] \\
(f) & A_j[\circ\circ]\ A_{j+1}[\,] \overset{a_{j+1}}{\longmapsto} A_{j+1}[\circ\circ\gamma] \\
(g) & A_j[\,]\ A_{j+1}[\circ\circ] \overset{a_{j+1}}{\longmapsto} A_{j+1}[\circ\circ\gamma] \\
(h) & A_1[\,]\ A_{2k}[\circ\circ\gamma] \overset{a_{2k}}{\longmapsto} A_{2k}[\circ\circ]
\end{array}
$$

En la tabla 6.16 se muestra la derivación correspondiente a la cadena $a_1^2 \dots a_{2k}^2$. ¶

$$
\begin{array}{lll}
 & \$_0[\,] & a_1a_1a_2a_2a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(a) & \$_0[\,]\ A_1[\,] & a_1a_1a_2a_2a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(b) & \$_0[\,]\ A_1[\,]\ A_1[\gamma] & a_1a_2a_2a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(b) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_1[\gamma\gamma] & a_2a_2a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(c) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_2[\gamma\gamma] & a_2a_2a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(d) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_2[\,]\ A_2[\gamma] & a_2a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(d) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_2[\,]\ A_2[\,]\ A_2[\,] & a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(e) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_2[\,]\ A_2[\,]\ A_3[\,] & a_3a_3a_4a_4\dots a_{2k}a_{2k} \\
(f) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_2[\,]\ A_3[\gamma] & a_3a_4a_4\dots a_{2k}a_{2k} \\
(g) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_3[\gamma\gamma] & a_4a_4\dots a_{2k}a_{2k} \\
(c) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_4[\gamma\gamma] & a_4a_4\dots a_{2k}a_{2k} \\
(d) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_4[\,]\ A_4[\gamma] & a_4\dots a_{2k}a_{2k} \\
(d) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_4[\,]\ A_4[\,]\ A_4[\,] & \dots a_{2k}a_{2k} \\
 & \vdots & \\
(g) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_{2k-1}[\gamma\gamma] & a_{2k}a_{2k} \\
(c) & \$_0[\,]\ A_1[\,]\ A_1[\,]\ A_{2k}[\gamma\gamma] & a_{2k}a_{2k} \\
(h) & \$_0[\,]\ A_1[\,]\ A_{2k}[\gamma] & a_{2k} \\
(h) & \$_0[\,]\ A_{2k}[\,] &
\end{array}
$$

Tabla 6.16: Derivación de la cadena $a_1^2 \dots a_{2k}^2$ en N-LIA

Con el fin de limitar el conjunto de lenguajes aceptados a la clase de los lenguajes de adjunción de árboles, estableceremos una serie de restricciones en la combinación de transiciones, dando lugar a la definición de un nuevo modelo de autómata, los *autómatas lineales de índices universales* (U–LIA). Un autómata de este tipo es una 7-tupla $(V_T, V_S^L, V_S^R, \$_0, \$_f, V_I, \mathcal{T})$, donde

V_T es un conjunto finito de símbolos terminales, V_S^L y V_S^R son dos conjuntos finitos disjuntos de símbolos de pila, $\$_0 \in V_S^L$ es el símbolo inicial de pila, $\$_f \in V_S^R$ es el símbolo final de pila, V_I es un conjunto finito de índices, y $\mathcal{T}$ es un conjunto finito de transiciones. Denotaremos el conjunto de todos los símbolos de pila como $V_S = V_S^L \cup V_S^R$. Las transiciones en $\mathcal{T}$ pueden ser de los siguientes tipos:

1. $C[\,] \overset{a}{\longmapsto} F[\,]$, donde $C \in V_S^L$, $F \in V_S^R$ y $a \in V_T$.
2. $C[\circ\circ\gamma] \longmapsto C[\,]\ F[\circ\circ\gamma']$, donde $C, F \in V_S^L$ y $\gamma, \gamma' \in V_I$.
3. $C[\,]\ F[\circ\circ\gamma] \longmapsto G[\circ\circ\gamma']$, donde $F, G \in V_S^R$, $C \in V_S$ y $\gamma, \gamma' \in V_I$.
4. $C[\circ\circ] \longmapsto C[\circ\circ]\ F[\,]$, donde $C \in V_S^d$, para algún $d \in \{L, R\}$, y $F \in V_S^L$.
5. $C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]$, donde $C, G \in V_S^d$, para algún $d \in \{L, R\}$, y $F \in V_S^R$.

Para las transiciones del segundo y tercer tipo, bien γ, bien γ', o bien ambas γ y γ' han de ser ϵ. Adicionalmente impondremos la siguiente restricción: para cada par de transiciones de la forma $C[X_1] \longmapsto C[Y_1]\ F[Z_1]$ y $C[Y_2]\ F'[Z_2] \longmapsto G[X_2]$ presentes en $\mathcal{T}$, debe satisfacerse una de estas dos condiciones:

1. $Y_1 = Y_2 = \epsilon$, $|X_1| = |X_2|$, y $|Z_1| = |Z_2|$, o
2. $X_1 = Y_1 = Y_2 = X_2 = \circ\circ$, $Z_1 = Z_2 = \epsilon$.

Esta restricción asegura que una transición PUSH de tipo 2 únicamente puede ser "acoplada" a una transición POP de tipo 3 (caso 1 de la restricción), y de modo similar, que una transición de tipo 4 solo puede ser acoplada a una transición de tipo 5 (caso 2 de la restricción). En consecuencia, si la altura de la pila se ha incrementado en una unidad mediante la aplicación de un cierto tipo de transición PUSH, conocemos qué tipo de transición POP será aplicada posteriormente para restaurar la altura de la pila.

El particular, el primer caso de la restricción asegura que la longitud de la pila de índices asociada al símbolo C es idéntica a la altura de la pila de índices asociada a G, en virtud de la aplicación recursiva de la misma restricción a F y F' y al hecho de que el cambio en la longitud de la pila de índices producido por X_1 y Z_1 es el inverso que el cambio producido por Z_2 y X_2, dado que $|X_1| = |X_2|$ y $|Z_1| = |Z_2|$.

Es interesante remarcar que los símbolos en V_S^L permiten que las pilas de índices sean propagadas a regiones superiores de la pila, mientras que los símbolos en V_S^R llevan las listas de índices a regiones inferiores. Típicamente, en la construcción de un U–LIA para TAG, los símbolos en V_S^L se usarán para propagar información predictiva mientras que los símbolos en V_S^R se usarán para propagar información en la fase ascendente del algoritmo de análisis.

6.5.1 Esquema de compilación

El esquema de compilación genérico mostrado en la sección 6.2.1 puede ser aplicado sin cambios a los autómatas lineales de índices universales para implementar estrategias de análisis descendentes, ascendentes y mixtas, tanto en lo que concierne al recorrido de los árboles elementales como al reconocimiento de las adjunciones. En la tabla 6.17 se muestran los valores que deben tomar los parámetros de predicción y propagación de información para instanciar el esquema de compilación genérico en esquemas correspondientes a diferentes estrategias de análisis, donde $\Box$ representa un símbolo de pila nuevo mientras que $\diamond$ representa un índice nuevo.

Estrategia-CF	**Estrategia-adjunción**	$\overrightarrow{N^{\gamma}_{r,s+1}}$	$\overrightarrow{\circ\circ\, N^{\gamma}_{r,s+1}}$	$\overleftarrow{N^{\gamma}_{r,s+1}}$	$\overleftarrow{\circ\circ\, N^{\gamma}_{r,s+1}}$
Ascendente		$\Box$	$\circ\circ\Diamond$	$N^{\gamma}_{r,s+1}$	$\circ\circ N^{\gamma}_{r,s+1}$
Earley	ascendente	$\overline{N^{\gamma}_{r,s+1}}$	$\circ\circ\Diamond$	$\overline{\overline{N^{\gamma}_{r,s+1}}}$	$\circ\circ N^{\gamma}_{r,s+1}$
Descendente		$N^{\gamma}_{r,s+1}$	$\circ\circ\Diamond$	$\Box$	$\circ\circ N^{\gamma}_{r,s+1}$
Ascendente		$\Box$	$\circ\circ N^{\gamma}_{r,s+1}$	$N^{\gamma}_{r,s+1}$	$\circ\circ N^{\gamma}_{r,s+1}$
Earley	Earley	$\overline{N^{\gamma}_{r,s+1}}$	$\circ\circ N^{\gamma}_{r,s+1}$	$\overline{\overline{N^{\gamma}_{r,s+1}}}$	$\circ\circ N^{\gamma}_{r,s+1}$
Descendente		$N^{\gamma}_{r,s+1}$	$\circ\circ N^{\gamma}_{r,s+1}$	$\Box$	$\circ\circ N^{\gamma}_{r,s+1}$
Ascendente		$\Box$	$\circ\circ N^{\gamma}_{r,s+1}$	$N^{\gamma}_{r,s+1}$	$\circ\circ\Diamond$
Earley	descendente	$\overline{N^{\gamma}_{r,s+1}}$	$\circ\circ N^{\gamma}_{r,s+1}$	$\overline{\overline{N^{\gamma}_{r,s+1}}}$	$\circ\circ\Diamond$
Descendente		$N^{\gamma}_{r,s+1}$	$\circ\circ N^{\gamma}_{r,s+1}$	$\Box$	$\circ\circ\Diamond$

Tabla 6.17: Parámetros del esquema de compilación genérico de TAG en U–LIA

Teorema 6.5 *Los autómatas lineales de índices universales aceptan la clase de los lenguajes de adjunción de árboles.*

Demostración:

Por el esquema de compilación de TAG en autómatas lineales de índices universales sabemos que a partir de cualquier gramática de adjunción de árboles es posible construir un U–LIA que acepta el lenguaje reconocido por dicha gramática.

Para demostrar que todo lenguaje aceptado por un autómata lineal de índices universal es un lenguaje de adjunción de árboles, mostraremos que para todo U–LIA existe una gramática lineal de índices tal que el lenguaje reconocido por la gramática coincide con el lenguaje aceptado por el autómata.

Sea $\mathcal{A} = (V_T, V_S^R, V_S^L, \$_0, \$_f, V_I, \mathcal{T})$ un autómata lineal de índices universal. Construiremos una gramática lineal de índices $\mathcal{L} = (V_T, V_N, V_I', S, P)$. El conjunto V_N de no terminales estará formado por pares $\langle E, B\rangle$ tal que $E, B \in V_S$. El conjunto V_I' estará formado por pares $\langle\gamma, \eta\rangle$ tal que $\gamma, \eta \in V_I$. Para que $\mathcal{L}$ reconozca el lenguaje aceptado por $\mathcal{A}$ el conjunto de producciones en P ha de construirse a partir de las transiciones en $\mathcal{T}$ de la siguiente manera:

- Para toda transición $C[\,] \overset{a}{\longmapsto} F[\,]$ y para todo $E \in V_S$ creamos la producción

$$\langle E, F\rangle[\,] \to \langle E, C\rangle[\,]\ a$$

- Para todo par de transiciones $C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]$ y $C[\circ\circ] \longmapsto C[\circ\circ]\ F'[\,]$, y para todo $E \in V_S$ tal que o bien $E, G \in V_S^d$ o bien $E \in V_S^L$ y $G \in V_S^R$, creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\circ\circ]\ \ \langle F', F\rangle[\,]$$

- Para todo par de transiciones $C[\,]\ F[\circ\circ] \longmapsto G[\circ\circ]$ y $C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ]$, y para todo $E \in V_S^L$ creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\,]\ \ \langle F', F\rangle[\circ\circ]$$

- Para todo par de transiciones $C[\,]\ F[\circ\circ\eta'] \longmapsto G[\circ\circ]$ y $C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma']$, y para todo $E \in V_S^L$ creamos una producción

$$\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\,]\ \langle F', F\rangle[\circ\circ\langle\gamma', \eta'\rangle]$$

- Para todo par de transiciones $C[\,]\ F[\circ\circ] \longmapsto G[\circ\circ\eta]$ y $C[\circ\circ\gamma] \longmapsto C[\,]\ F'[\circ\circ]$, y para todo $E \in V_S^L$ creamos una producción

$$\langle E, G\rangle[\circ\circ\langle\gamma, \eta\rangle] \to \langle E, C\rangle[\,]\ \langle F', F\rangle[\circ\circ]$$

- Para todo $E \in V_S$ creamos una producción

$$\langle E, E\rangle[\,] \to \epsilon$$

Con respecto al axioma de la gramática, tenemos que $S = \langle \$_0, \$_0\rangle$.

Para realizar la demostración, consideraremos dos casos diferentes de derivación, tratando cada uno de los mismos por separado.

Caso 1. El caso principal de esta demostración establece que $\langle E, C\rangle[\alpha] \overset{*}{\Rightarrow} w$, con $E \in V_S^L$ y $C \in V_S^R$, si y sólo si $(E[\delta], w) \overset{*}{\vdash} (C[\beta], \epsilon)$ y se cumple que $\delta = \gamma_1\gamma_2 \ldots \gamma_p$, $\beta = \eta_1\eta_2 \ldots \eta_p$ y $\alpha = \langle\gamma_1, \eta_1\rangle\langle\gamma_2, \eta_2\rangle \ldots \langle\gamma_p, \eta_p\rangle$, esto es, δ es la pila obtenida como resultado de la proyección del primer componente de los elementos almacenados en α mientras que β es la pila obtenida como resultado de la proyección del segundo componente de los elementos almacenados en la pila de índices α.

A continuación demostraremos cada una de las direcciones de la implicación:

- Si una derivación $(E[\delta], w) \overset{*}{\vdash} (C[\beta], \epsilon)$ es el resultado de aplicar la secuencia $t_1, \ldots, t_s$ de transiciones en $\mathcal{T}$, entonces existe una secuencia $p_1, \ldots, p_s$ de producciones en P tal que la derivación $\langle E, C\rangle[\alpha] \overset{*}{\Rightarrow} w$ resultado de aplicar $p_1, \ldots, p_s$ reconoce w. La demostración se realiza por inducción en la longitud de la derivación del autómata.

  El caso base lo constituye la derivación $(E[\ ], \epsilon) \vdash (C[\ ], \epsilon)$, para la que existe la producción $\langle E, C\rangle[\,] \to \langle E, E\rangle[\,]\ a$ y la producción $\langle E, E\rangle[\,] \to \epsilon$, por lo que $\langle E, C\rangle[\,] \overset{*}{\Rightarrow} a$.

  Por hipótesis de inducción suponemos que la proposición se cumple para cualquier derivación del autómata de longitud s. En tal caso, durante el paso de inducción verificamos que se cumple para cualquier posible derivación de longitud mayor que s:

  - Si $(E[\delta], w_1w_2) \overset{s_1}{\vdash} (C[\beta], w_2) \vdash (C[\beta]\ F'[\ ], w_2) \overset{s_2}{\vdash} (C[\beta]\ F[\ ], \epsilon) \vdash (G[\beta], \epsilon)$, con $F' \in V_S^L$ y $F, G \in V_S^R$, existe una producción $\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\circ\circ]\ \langle F', F\rangle[\,]$, por hipótesis de inducción se cumple que $\langle E, C\rangle[\alpha] \overset{*}{\Rightarrow} w_1$ y $\langle F', F\rangle[\,] \overset{*}{\Rightarrow} w_2$, y en consecuencia $\langle E, G\rangle[\alpha] \overset{*}{\Rightarrow} w_1w_2$.
  - Si $(E[\delta], w_1w_2) \overset{s_1}{\vdash} (C[\delta], w_2) \vdash (C[\,]\ F'[\beta], w_2) \overset{s_2}{\vdash} (C[\,]\ F[\beta], \epsilon) \vdash (G[\beta], \epsilon)$, con $F' \in V_S^L$ y $F, G \in V_S^R$, existe una producción $\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\,]\ \langle F', F\rangle[\circ\circ]$, por los casos 1 y 2 se cumple que $\langle E, C\rangle[\,] \overset{*}{\Rightarrow} w_1$ y $\langle F', F\rangle[\,] \overset{*}{\Rightarrow} w_2$, y en consecuencia $\langle E, G\rangle[\alpha] \overset{*}{\Rightarrow} w_1w_2$.
  - Si $(E[\delta], w_1w_2) \overset{s_1}{\vdash} (C[\delta], w_2) \vdash (C[\ ]\ F'[\beta\gamma'], w_2) \overset{s_2}{\vdash} (C[\ ]\ F[\beta\eta'], \epsilon) \vdash (G[\beta], \epsilon)$, con $F' \in V_S^L$ y $F, G \in V_S^R$, existe una producción $\langle E, G\rangle[\circ\circ] \to \langle E, C\rangle[\,]\ \langle F', F\rangle[\circ\circ\langle\gamma', \eta'\rangle]$, por los casos 1 y 2 se cumple que $\langle E, C\rangle[\,] \overset{*}{\Rightarrow} w_1$ y $\langle F', F\rangle[\alpha\langle\gamma', \eta\rangle] \overset{*}{\Rightarrow} w_2$, y en consecuencia $\langle E, G\rangle[\alpha] \overset{*}{\Rightarrow} w_1w_2$.
  - Si $(E[\delta\gamma], w_1w_2) \overset{s_1}{\vdash} (C[\delta], w_2) \vdash (C[\ ]\ F'[\beta], w_2) \overset{s_2}{\vdash} (C[\ ]\ F[\beta], \epsilon) \vdash (G[\beta\eta], \epsilon)$, con $F' \in V_S^L$ y $F, G \in V_S^R$, existe una producción $\langle E, G\rangle[\circ\circ\langle\gamma, \eta\rangle] \to \langle E, C\rangle[\,]\ \langle F', F\rangle[\circ\circ]$, por los casos 1 y 2 se cumple que $\langle E, C\rangle[\,] \overset{*}{\Rightarrow} w_1$ y $\langle F', F\rangle[\alpha] \overset{*}{\Rightarrow} w_2$, y en consecuencia $\langle E, G\rangle[\alpha\langle\gamma', \eta\rangle] \overset{*}{\Rightarrow} w_1w_2$.

- Si una derivación izquierda $\langle E,C\rangle[\alpha] \overset{*}{\Rightarrow} w$, con $E \in V_S^L$ y $C \in V_S^R$, reconoce la cadena w como resultado de aplicar la secuencia $p_1,\ldots,p_s$ de producciones en P, entonces existe una secuencia de transiciones $t_1,\ldots,t_s$ en $\mathcal{T}$ tal que la derivación $(E[\delta],w) \overset{*}{\vdash} (C[\beta],\epsilon)$ es el resultado de aplicar la secuencia de transiciones $t_1,\ldots,t_s$. La demostración se realiza por inducción en la longitud de la derivación de la gramática.
  El caso base lo constituye la derivación $\langle E,C\rangle[\,] \Rightarrow \langle E,E\rangle[\,]\ a \Rightarrow a$, para la que existe una derivación $(E[\,],a) \vdash (C[\,],\epsilon)$ en el autómata.
  Por hipótesis de inducción suponemos que la proposición se cumple para cualquier derivación de la gramática de longitud s. En tal caso, durante el paso de inducción verificamos que se cumple para cualquier posible derivación de longitud mayor que s:
  - Si $\langle E,G\rangle[\alpha] \Rightarrow \langle E,C\rangle[\alpha]\ \langle F',F\rangle[\,] \overset{s_1}{\Rightarrow} w_1\ \langle F',F\rangle[\,] \overset{s_2}{\Rightarrow} w_1w_2$, con $F' \in V_S^L$ y $F,G \in V_S^R$, existe una transición $C[\circ\circ]\longmapsto C[\circ\circ]\ F'[\,]$ y una transición $C[\,]\ F[\,] \longmapsto G[\circ\circ]$, por hipótesis de inducción se cumple que $(E[\delta],w_1) \overset{*}{\vdash} (C[\beta],\epsilon)$ y que $(F'[\,],w_2) \overset{*}{\vdash} (F[\ \ ],\epsilon)$ y en consecuencia se cumple que $(E[\delta],w_1w_2) \overset{*}{\vdash} (C[\beta],w_2) \vdash (C[\beta]\ F'[\,],w_2) \overset{*}{\vdash} (C[\beta]\ F[\,],\epsilon) \vdash (G[\beta],\epsilon)$.
  - Si $\langle E,G\rangle[\alpha] \Rightarrow \langle E,C\rangle[\,]\ \langle F',F\rangle[\alpha] \overset{s_1}{\Rightarrow} w_1\ \langle F',F\rangle[\alpha] \overset{s_2}{\Rightarrow} w_1w_2$, con $F' \in V_S^L$ y $F,G \in V_S^R$, existe una transición $C[\circ\circ]\longmapsto C[\,]\ F'[\circ\circ]$ y una transición $C[\,]\ F[\circ\circ] \longmapsto G[\circ\circ]$, por los casos 1 y 2 se cumple que $(E[\delta],w_1) \overset{*}{\vdash} (C[\delta],\epsilon)$ y por hipótesis de inducción $(F'[\delta],w_2) \overset{*}{\vdash} (F[\beta],\epsilon)$ y en consecuencia se cumple que $(E[\delta],w_1w_2) \overset{*}{\vdash} (C[\delta],w_2) \vdash (C[\,] \to F'[\delta],w_2) \overset{*}{\vdash} (C[\,]\ F[\beta],\epsilon) \vdash (G[\beta],\epsilon)$.
  - Si $\langle E,G\rangle[\alpha] \Rightarrow \langle E,C\rangle[\ ]\ \langle F',F\rangle[\alpha\langle\gamma',\eta'\rangle] \overset{s_1}{\Rightarrow} w_1\ \langle F',F\rangle[\alpha\langle\gamma',\eta'\rangle] \overset{s_2}{\Rightarrow} w_1w_2$, con $F' \in V_S^L$ y $F,G \in V_S^R$, existe una transición $C[\circ\circ]\longmapsto C[\,]\ F'[\circ\circ\gamma']$ y una transición $C[\circ\circ]\ F[\circ\circ\eta'] \longmapsto G[\circ\circ]$, por los casos 1 y 2 se cumple que $(E[\delta],w_1) \overset{*}{\vdash} (C[\delta],\epsilon)$ y por hipótesis de inducción $(F'[\delta\gamma'],w_2) \overset{*}{\vdash} (F[\beta\eta'],\epsilon)$ y en consecuencia se cumple que $(E[\delta],w_1w_2) \overset{*}{\vdash} (C[\delta],w_2) \vdash (C[\,]\ F'[\delta\gamma'],w_2) \overset{*}{\vdash} (C[\,]\ F[\beta\eta'],\epsilon) \vdash (G[\beta],\epsilon)$.
  - Si $\langle E,G\rangle[\alpha\langle\gamma,\eta\rangle] \Rightarrow \langle E,C\rangle[\,]\ \langle F',F\rangle[\alpha] \overset{s_1}{\Rightarrow} w_1\ \langle F',F\rangle[\alpha] \overset{s_2}{\Rightarrow} w_1w_2$, con $F' \in V_S^L$ y $F,G \in V_S^R$, existe una transición $C[\circ\circ\gamma]\longmapsto C[\,]\ F'[\circ\circ]$ y una transición $C[\,]\ F[\circ\circ] \longmapsto G[\circ\circ\eta]$, por los casos 1 y 2 se cumple que $(E[\delta\gamma],w_1) \overset{*}{\vdash} (C[\delta\gamma],\epsilon)$ y por hipótesis de inducción $(F'[\delta],w_2) \overset{*}{\vdash} (F[\beta],\epsilon)$ y en consecuencia se cumple que $(E[\delta\gamma],w_1w_2) \overset{*}{\vdash} (C[\delta\gamma],w_2) \vdash (C[\,]\ F'[\delta],w_2) \overset{*}{\vdash} (C[\,]\ F[\beta],\epsilon) \vdash (G[\beta\eta],\epsilon)$.

Caso 2. Existe una derivación del autómata

$$\begin{aligned}(\Upsilon E[\alpha_1],w_1w_2) &\overset{*}{\vdash} (\Upsilon C[\alpha_1],w_2)\\ &\vdash (\Upsilon C[\alpha_1]\ F'[\,],w_2)\\ &\overset{*}{\vdash} (\Upsilon C[\alpha_1]\ F[\,],\epsilon)\\ &\vdash (\Upsilon G[\alpha_1],\epsilon)\end{aligned}$$

con $E,C,F',G \in V_S^L$ y $F \in V_S^R$, si y sólo si $\langle E,G\rangle[\,] \overset{*}{\Rightarrow} w_1w_2$. La demostración se obtiene por inducción a partir del caso anterior y de la existencia de la producción $\langle E,G\rangle[\circ\circ] \to \langle E,C\rangle[\circ\circ]\ \langle F',F\rangle[\,]$.

□

6.5.2 Tabulación

Para diseñar una técnica de tabulación para los autómatas lineales de índices universales debemos considerar los diferentes tipos de derivaciones que se pueden dar en la evolución de un autómata de este tipo. En concreto, podemos distinguir los siguientes tres tipos:

Derivaciones de llamada. Son aquellas que recorren la parte izquierda de la espina durante la fase descendente de un algoritmo de análisis, por lo que presentan la forma

$$\begin{array}{rl}(\Upsilon\ A[\delta], a_{h+1}\dots a_n) & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\delta\gamma], a_{i+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ C[\delta\gamma], a_{j+1}\dots a_n)\end{array}$$

donde $\Upsilon, \Upsilon_1 \in (V_S[V_I^*])^*$, $\gamma \in V_I$, $\delta \in (V_S[V_I^*])^*$, $A, B, C \in V_S^L$ y tanto $B[\alpha\gamma]$ como $C[\alpha\gamma]$ son descendientes dependientes de $A[\alpha]$ y no existe $(F[\delta\gamma], f) \neq (B[\delta\gamma], i)$ tal que

$$\begin{array}{rl}(\Upsilon\ A[\delta], a_{h+1}\dots a_n) & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ F[\delta\gamma], a_{f+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\delta\gamma], a_{i+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ C[\delta\gamma], a_{j+1}\dots a_n)\end{array}$$

En la figura 6.9 se muestra una representación gráfica de las derivaciones de llamada.

Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ y $\alpha \in V_I^*$ se cumple que

$$\begin{array}{rl}(\Upsilon'\ A[\alpha], a_{h+1}\dots a_n) & \overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\alpha\gamma], a_{i+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ C[\alpha\gamma], a_{j+1}\dots a_n)\end{array}$$

Es por ello que este tipo de derivaciones puede ser representado mediante ítems de la forma

$$[A, h \mid B, i, \gamma, C, j, \gamma \mid -, -, -, -]$$

Derivaciones de retorno. Son aquellas que se inician en la parte izquierda de una espina pero terminan en la parte derecha de la misma, por lo que tienen la siguiente forma:

$$\begin{array}{rl}(\Upsilon\ A[\delta], a_{h+1}\dots a_n) & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\delta\gamma], a_{i+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ D[\delta], a_{p+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ E[\beta], a_{q+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ C[\beta\eta], a_{j+1}\dots a_n)\end{array}$$

donde $\Upsilon, \Upsilon_1, \Upsilon_2 \in (V_S[V_I^*])^*$ $\gamma, \eta \in V_I$, $\delta, \beta \in V_I^*$, $A, B, D \in V_S^L$, $E, C \in V_S^R$ y $B[\alpha\gamma]$ y $D[\alpha]$ son descendientes dependientes de $A[\alpha]$, $C[\alpha\eta]$ es un descendiente dependiente de $D[\alpha]$ y no existen $(F[\delta\gamma], f) \neq (B[\delta\gamma], i)$ ni $(G[\delta], g) \neq (D[\delta], p)$ tal que

$$\begin{array}{rl}(\Upsilon\ A[\delta], a_{h+1}\dots a_n) & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ F[\delta\gamma], a_{f+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\delta\gamma], a_{i+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ G[\delta], a_{g+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ D[\delta], a_{p+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ E[\beta], a_{q+1}\dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ A[\,]\ \Upsilon_1\ C[\beta\eta], a_{j+1}\dots a_n)\end{array}$$

Una representación gráfica de las derivaciones de retorno se muestra en la figura 6.10.

Para cualquier $\Upsilon' \in (V_S[V_I^*])^*$ y pilas de índices $\delta', \beta' \in V_I^*$ tal que existe una derivación $(D[\delta'], a_{p+1} \dots a_n) \overset{*}{\vdash} (E[\beta'], a_{q+1} \dots a_n)$ se cumple que

$$\begin{array}{ll}(\Upsilon'\ A[\delta'], a_{h+1} \dots a_n) & \overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\delta'\gamma], a_{i+1} \dots a_n) \\ & \overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ D[\delta'], a_{p+1} \dots a_n) \\ & \overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ B[\,]\ \Upsilon_2\ \ E[\beta'], a_{q+1} \dots a_n) \\ & \overset{*}{\vdash} (\Upsilon'\ A[\,]\ \Upsilon_1\ C[\beta'\eta], a_{j+1} \dots a_n)\end{array}$$

por lo que este tipo de derivaciones puede ser representado de modo compacto por ítems de la forma

$$[A, h \mid B, i, \gamma, C, j, \eta \mid D, p, E, q]$$

Derivaciones de puntos especiales. Son aquellas derivaciones que involucran la propagación de una pila de índices vacía. La importancia de este tipo de derivaciones radica en que son las únicas en las cuales es posible realizar el cambio de un símbolo de pila de V_S^L a otro de V_S^R. Las derivaciones de puntos especiales presentan la siguiente forma:

$$(\Upsilon\ B[\,], a_{i+1} \dots a_n) \ \ \overset{*}{\vdash} (\Upsilon\ C[\,], a_{j+1} \dots a_n)$$

donde $\Upsilon \in (V_S[V_I^*])^*$, bien se cumple $B, C \in V_S^d$ o bien $B \in V_S^L$ y $C \in V_S^R$, y no existe $(F[\,], f) \neq (B[\,], i)$ tal que

$$\begin{array}{ll}(\Upsilon\ F[\,], a_{f+1} \dots a_n) & \overset{*}{\vdash} (\Upsilon\ B[\,], a_{i+1} \dots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ C[\,], a_{j+1} \dots a_n)\end{array}$$

La figura 6.11 muestra las derivaciones de puntos especiales de un modo más gráfico.

Para todo $\Upsilon' \in (V_S[V_I^*])^*$ se cumple

$$(\Upsilon'\ B[\,], a_{i+1} \dots a_n) \ \ \overset{*}{\vdash} (\Upsilon'\ C[\,], a_{j+1} \dots a_n)$$

por lo que podemos utilizar los siguientes ítems para representar este tipo de derivaciones:

$$[-, - \mid B, i, -, C, j, - \mid -, -, -, -]$$

Los ítems se combinan mediante las reglas de inferencia descritas en la tabla 6.18, a partir del ítem inicial $[-, - \mid \$_0, 0, -, \$_0, 0, - \mid -, -, -, -]$. La aceptación de una cadena de entrada $a_1 \dots a_n$ se indica mediante la presencia del ítem final $[-, - \mid \$_0, 0, -, \$_f, 0, - \mid -, -, -, -]$

Teorema 6.6 *La manipulación de configuraciones mediante la aplicación de transiciones en los autómatas lineales de índices universales es equivalente a la manipulación de ítems mediante las reglas de inferencia de la tabla 6.18.*

Demostración:

De forma análoga al caso de los R–LIA y L–LIA, la demostración se realiza por inducción en la longitud de las derivaciones tras considerar la lista exhaustiva de todas las posibles derivaciones que se pueden obtener tras la aplicación de una transición del autómata. La demostración completa puede encontrarse en [3]. □

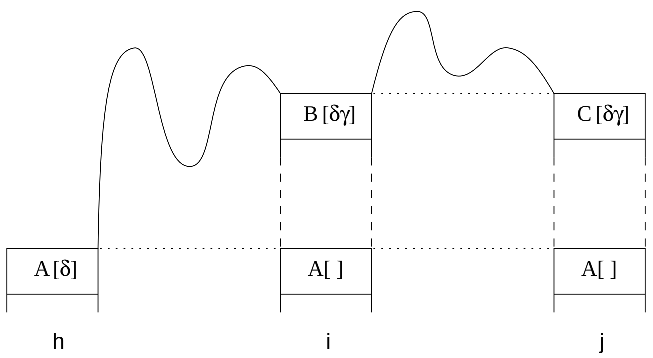

Figura 6.9: Derivaciones de llamada en U–LIA

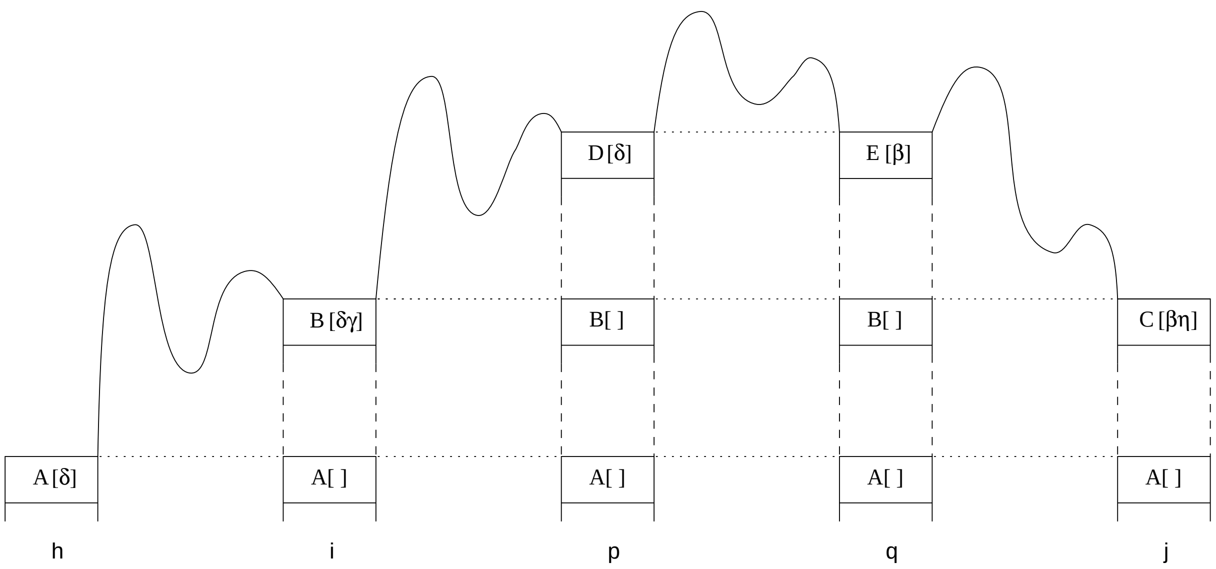

Figura 6.10: Derivaciones de retorno en U–LIA

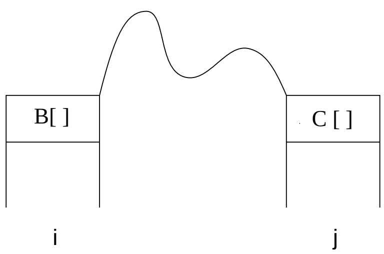

Figura 6.11: Derivaciones de puntos especiales en U–LIA

$$\frac{[-,- \mid B,i,-,C,j,- \mid -,-,-,-]}{[-,- \mid B,i,-,F,k,- \mid -,-,-,-]} \quad C[\,] \overset{a}{\longmapsto} F[\,],\;\; k=j \text{ si } a=\epsilon,\;\; k=j+1 \text{ si } a \in V_T$$

$$\frac{[-,- \mid B,i,\gamma,C,j,\eta \mid -,-,-,-]}{[-,- \mid F,j,-,F,j,- \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\circ\circ]\; F[\,]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,\eta \mid D,p,E,q]}{[-,- \mid F,j,-,F,j,- \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\circ\circ]\; F[\,]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]}{[A,h \mid F,j,\gamma,F,j,\gamma \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\,]\; F[\circ\circ]$$

$$\frac{[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]}{[C,j \mid F,j,\gamma',F,j,\gamma' \mid -,-,-,-]} \quad C[\circ\circ] \longmapsto C[\,]\; F[\circ\circ\gamma']$$

$$\frac{\begin{array}{c}[A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [M,m \mid N,t,\gamma',A,h,\gamma' \mid -,-,-,-]\end{array}}{[M,m \mid F,j,\gamma',F,j,\gamma' \mid -,-,-,-]} \quad C[\circ\circ\gamma] \longmapsto C[\,]\; F[\circ\circ]$$

$$\frac{\begin{array}{c}[-,- \mid F',j,-,F,k,- \mid -,-,-,-]\\ [A,h \mid B,i,\gamma,C,j,\eta \mid D,p,E,q]\end{array}}{[A,h \mid B,i,\gamma,G,k,\eta \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\circ\circ]\; F'[\,] \\ C[\circ\circ]\; F[\,] \longmapsto G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[A,h \mid F',j,\gamma,F,k,\eta \mid D,p,E,q]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\end{array}}{[A,h \mid B,i,\gamma,G,k,\eta \mid D,p,E,q]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\; F'[\circ\circ] \\ C[\,]\; F[\circ\circ] \longmapsto G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[C,j \mid F',j,\gamma',F,k,\eta' \mid D,p,E,q]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [A,h \mid D,p,\gamma,E,q,\eta \mid O,u,P,v]\end{array}}{[A,h \mid B,i,\gamma,G,k,\eta \mid O,u,P,v]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\; F'[\circ\circ\gamma'] \\ C[\,]\; F[\circ\circ\eta'] \longmapsto G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[M,m \mid F',j,\gamma',F,k,\eta' \mid D,p,E,q]\\ [A,h \mid B,i,\gamma,C,j,\gamma \mid -,-,-,-]\\ [M,m \mid N,t,\gamma',A,h,\gamma' \mid -,-,-,-]\end{array}}{[A,h \mid B,i,\gamma,G,k,\eta \mid F',j,F,k]} \quad \begin{array}{l} C[\circ\circ\gamma] \longmapsto C[\,]\; F'[\circ\circ] \\ C[\,]\; F[\circ\circ] \longmapsto G[\circ\circ\eta]\end{array}$$

Tabla 6.18: Reglas de inferencia en U–LIA

Como resultado colateral del teorema anterior obtenemos una nueva propiedad de este modelo de autómatas, asociada a la técnica de tabulación aplicada:

Teorema 6.7 *Si un autómata lineal de índices universal satisface la propiedad del prefijo válido, entonces dicha propiedad es preservada por la técnica de tabulación propuesta.*

Demostración:

En efecto, el j-ésimo símbolo de la cadena de entrada es leído en una derivación $(\$_0[\,], a_1 \cdots a_n) \overset{*}{\vdash} (\Upsilon, a_j a_{j+1} \cdots a_n) \overset{*}{\vdash} (\Upsilon', a_{j+1} \cdots a_n)$ si y sólo si un ítem $[A, h \mid B, i, \gamma, C, j, \eta \mid D, p, E, q]$ (donde algunos de sus elementos pueden ser $-$) ha sido generado, lo cual implica que si la cadena de entrada no pertenece al lenguaje reconocido por el autómata, tanto el autómata como los ítems resultantes de su ejecución tabular leen una parte idéntica del prefijo de dicha entrada o, dicho más informalmente, ambos casos finalizan en la misma posición de la cadena de entrada. Como consecuencia, si el autómata satisface la propiedad del prefijo válido, entonces dicha propiedad es preservada por la técnica de tabulación propuesta [9].

Esta propiedad también se satisface en el caso de los R–LIA y de los L–LIA. Sin embargo, en el caso de los R–LIA el que la técnica de tabulación mantenga la propiedad del prefijo válido es irrelevante puesto que los autómatas de este tipo construidos a partir de gramáticas de adjunción de árboles (por ejemplo siguiendo el modelo de compilación propuesto en la sección 6.3.1) habitualmente no satisfacen esta propiedad. No es el caso de los L-LIA ni de los U–LIA, que satisfacen dicha propiedad si son construidos mediante estrategias descendentes o Earley. □

La complejidad espacial de la técnica de tabulación propuesta es $\mathcal{O}(n^5)$ puesto que cada ítem almacena 5 posiciones de la cadena de entrada. La complejidad temporal en el peor caso es $\mathcal{O}(n^7)$ y viene dada por la siguiente regla:

$$\frac{\begin{array}{c}[C, j \mid F', j, \gamma', F, k, \eta' \mid D, p, E, q] \\ [A, h \mid B, i, \gamma, C, j, \gamma \mid -, -, -, -] \\ [A, h \mid D, p, \gamma, E, q, \eta \mid O, u, P, v]\end{array}}{[A, h \mid B, i, \gamma, G, k, \eta \mid O, u, P, v]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma'] \\ C[\,]\ F[\circ\circ\eta'] \longmapsto G[\circ\circ]\end{array}$$

Esta regla involucra la manipulación de 8 posiciones de la cadena de entrada, aunque mediante aplicación parcial sólo se necesita manipular simultáneamente 7 de dichas posiciones. La complejidad temporal puede reducirse utilizando la técnica propuesta en [27, 56]. Siguiendo dicha técnica, podemos descomponer la regla anterior en las dos reglas siguientes:

$$\frac{\begin{array}{c}[C, j \mid F', j, \gamma', F, k, \eta' \mid D, p, E, q] \\ [A, h \mid D, p, \gamma, E, q, \eta \mid O, u, P, v]\end{array}}{[[C, j \mid F', j, \gamma', F, k, \eta' \mid O, u, P, v]]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma'] \\ C[\,]\ F[\circ\circ\eta'] \longmapsto G[\circ\circ]\end{array}$$

$$\frac{\begin{array}{c}[[C, j \mid F', j, \gamma', F, k, \eta' \mid O, u, P, v]] \\ [A, h \mid B, i, \gamma, C, j, \gamma \mid -, -, -, -] \\ [A, h \mid D, p, \gamma, E, q, \eta \mid O, u, P, v]\end{array}}{[A, h \mid B, i, \gamma, G, k, \eta \mid O, u, P, v]} \quad \begin{array}{l} C[\circ\circ] \longmapsto C[\,]\ F'[\circ\circ\gamma'] \\ C[\,]\ F[\circ\circ\eta'] \longmapsto G[\circ\circ]\end{array}$$

donde $[[C, j \mid F', j, \gamma', F, k, \eta' \mid O, u, P, v]]$ es un pseudo-ítem que reparte equilibradamente la información entre ambas reglas y garantiza que la ejecución consecutiva de estas últimas sea equivalente a la ejecución de la regla original. La primera regla tiene complejidad $\mathcal{O}(n^6)$, puesto que la posición h no interviene en la combinación de posiciones de la cadena de entrada, al igual que la segunda, puesto que en esta última las posiciones p y q no intervienen.

6.6 Autómatas lineales de índices bidireccionales

Los autómatas que hemos tratado hasta ahora se caracterizan por efectuar la lectura de la entrada en una sola dirección. Pasaremos ahora a ver otros modelos donde se admiten lecturas bidireccionales. Primero veremos como extender los modelos anteriores para el caso de las gramáticas independientes del contexto. Finalmente, mostraremos la aplicación de dicha técnica en el análisis de gramáticas de adjunción de árboles.

6.6.1 Autómatas a pila bidireccionales

Recordamos que un autómata a pila o PDA no es más que una tupla $(V_T, V_S, \Theta, \$_0, \$_f)$ donde V_T representa un conjunto finito de símbolos terminales, V_S es un conjunto finitos de símbolos de pila, $\$_0 \in V_S$ es el símbolo inicial de pila, $\$_f \in V_S$ es el símbolo final de pila y Θ es un conjunto finito de transiciones SWAP, PUSH y POP. Las configuraciones de los autómatas a pila se describen habitualmente como un par $(\xi, a_l \ldots a_n)$, donde ξ indica el contenido de la pila y $a_l \ldots a_n$ la parte de la cadena de entrada que resta por leer. En esta sección consideraremos una definición equivalente de configuración en la cual la posición l se almacena conjuntamente con el elemento situado en la cima de ξ. En consecuencia, una configuración vendrá dada por el contenido de $\hat{\xi}$, una pila de pares de la forma (C, l), donde $C \in V_S$ y $l \in \mathbb{N}$. Estos pares reciben el nombre de *modos* en [15]. La configuración inicial viene definida por la pila que contiene únicamente el par $(\$_0, 0)$. A continuación definimos los diferentes tipos de transiciones en el contexto de este nueva representación de las derivaciones:

- La aplicación de una transición SWAP de la forma $C \stackrel{a}{\longmapsto} F$ a una configuración $\hat{\xi}(C, l)$ permite obtener una nueva configuración $\hat{\xi}(F, l + |a|)$ como resultado del reemplazamiento de C por F y de la lectura del terminal $a = a_{l+1}$.

- La aplicación de una transición PUSH de la forma $C \longmapsto C\,F$ a una configuración $\hat{\xi}(C, l)$ da como resultado una configuración $\hat{\xi}(C, l)(F, l)$ al apilar F sobre C.

- La aplicación de una transición POP de la forma $C\,F \longmapsto G$ a una configuración $\hat{\xi}(C, l)(F, m)$ da como resultado una configuración $\hat{\xi}(G, m)$ donde se extrae de la pila del autómata C y F, los cuales pasan a ser reemplazados por G.

donde $C, F, G \in V_S$ y $a \in V_T \cup \{\epsilon\}$. Un autómata a pila finaliza satisfactoriamente cuando alcanza una configuración que contiene únicamente el par $(\$_f, n)$, donde n es la longitud de la cadena de entrada. Hemos considerado que tan solo las transiciones SWAP pueden leer elementos de la cadena de entrada. Esta consideración no supone ninguna limitación, puesto que una transición $C \stackrel{a}{\longmapsto} C\,F$ puede ser emulada mediante la aplicación consecutiva de dos transiciones $C \longmapsto C\,F'$ y $F' \stackrel{a}{\longmapsto} F$. Análogamente, una transición $C\ F \stackrel{a}{\longmapsto} G$ puede ser emulada mediante la aplicación consecutiva de dos transiciones $C\ F \longmapsto G'$ y $G' \stackrel{a}{\longmapsto} G$. En ambos casos F' y G' representan símbolos de pila nuevos que no son utilizados en ninguna otra transición.

Denominaremos *transiciones-r* a los tres tipos de transiciones SWAP, PUSH y POP ya que sólo pueden leer la cadena de entrada de izquierda a derecha. Es por ello que los autómatas a pila sólo pueden ser aplicados para la implementación de estrategias de análisis sintáctico unidireccionales que leen la cadena de entrada de izquierda a derecha.

Las estrategias bidireccionales pueden comenzar el análisis en cualquier posición de la cadena de entrada y a partir de ahí expandirse a derecha e izquierda para incluir subcadenas que han sido analizadas siguiendo esta misma estrategia bidireccional. Como primer paso hacia la definición de los Autómatas a Pila Bidireccionales (*Bidirectional Push-Down Automata*, BPDA), deberemos extender la definción de configuración con el fin de poder representar el reconocmiento discontinuo de la cadena de entrada. Es por ello que definiremos una configuración de un BPDA como una pila Ξ formada por triples (C,k,l), donde $C \in V_S$, y $k,l \in \mathbb{N}$. La configuración inicial vendrá dada por la pila que contiene únicamente el triple $(\$_0,0,0)$. La configuración final vendrá dada por una pila que contendrá como único elemento el par $(\$_f,0,n)$, donde n es la longitud de la cadena de entrada. Las configuraciones intermedias se obtendrán tras aplicar alguna de las transiciones siguientes:

- La aplicación de una transición $\mathrm{SWAP_R}$ de la forma $C \overset{a}{\longmapsto}_R F$ a una configuración $\Xi(C,k,l)$ da como resultado una configuración $\Xi(F,k,l+|a|)$ al reemplazar C por F y leer $a = a_{l+1}$ ó $a = \epsilon$ a la derecha de la subcadena reconocida por C.

- La aplicación de una transición $\mathrm{SWAP_L}$ de la forma $C \overset{a}{\longmapsto}_L F$ a una configuración $\Xi(C,k,l)$ resulta en una configuración $\Xi(F,k-|a|,l)$ al reemplazar C por F y leer $a = a_k$ ó $a = \epsilon$ a la izquierda de la subcadena reconocida por C.

- La aplicación de una transición $\mathrm{PUSH_R}$ de la forma $C \longmapsto_R C\,F$ a una configuración $\Xi(C,k,j)$ da como resultado una configuración $\Xi(C,k,l)(F,l,l)$ al apilar F sobre C. Se espera que F reconozca una subcadena situada inmediatamente a la derecha de la subcadena reconocida por C.

- La aplicación de una transición $\mathrm{PUSH_L}$ de la forma $C \longmapsto_L C\,F$ a una configuración $\Xi(C,k,j)$ permite obtener una configuración $\Xi(C,k,l)(F,k,k)$ como resultado de apilar F sobre C. Se espera que F reconozca una subcadena situada inmediatamente a la izquierda de la subcadena reconocida por C.

- La aplicación de una transición PUSH_U de la forma $C \overset{a}{\longmapsto}_U C\,F$ a una configuración $\Xi(C,k,j)$ resulta en una configuración $\Xi(C,k,l)(F,m,m+|a|)$ al apilar F sobre C y leer $a = a_{m+1}$ ó $a = \epsilon$. Las transiciones PUSH_U son *no dirigidas* en el sentido de que la subcadena reconocida por F no tiene porqué ser adyacente a la subcadena reconocida por C.

- La aplicación de una transición $\mathrm{POP_R}$ de la forma $C\ F \longmapsto_R G$ a una configuración $\Xi(C,k,l)(F,l,m)$ resulta en una configuración $\Xi(G,k,m)$ al reemplazar los elementos C y F de la cima de la pila del autómata por G. La subcadena reconocida por F debe estar situada inmediatamente a la derecha de la subcadena reconocida por C.

- La aplicación de una transición $\mathrm{POP_L}$ de la forma $C\ F \longmapsto_L G$ a una configuración $\Xi(C,k,l)(F,m,k)$ da como resultado una configuración $\Xi(G,m,l)$ al reemplazar los elementos C y F de la cima de la pila del autómata por G. La subcadena reconocida por F debe estar situada inmediatamente a la izquierda de la subcadena reconocida por C.

Las transiciones $\mathrm{SWAP_R}$, $\mathrm{PUSH_R}$ y $\mathrm{POP_R}$ son las transiciones-r ya presentes en los autómats a pila unidireccionales. Las transiciones $\mathrm{SWAP_L}$, $\mathrm{PUSH_L}$ y $\mathrm{POP_L}$ son las *transiciones-l* duales de las anteriores ya que avanzan hacia la izquierda en la lectura de la cadena de entrada. Las transiciones PUSH_U de la forma $C \overset{a}{\longmapsto}_U C\,F$ son las que nos permitirán comenzar el análisis en cualquier punto de la cadena de entrada. Las transiciones PUSH_U son transiciones de apilamiento, puesto que el comienzo del análisis de un nuevo componente requiere apilar un nuevo símbolo, y son transiciones que pueden leer de la cadena de entrada.

Es importante resaltar que la definición de las transiciones $\mathrm{SWAP_R}$ y $\mathrm{SWAP_L}$ — que no pueden releer elementos que pertenecen a la subcadena reconocida por el elemento en la cima de la pila del autómata — y de las transiciones $\mathrm{POP_R}$ y $\mathrm{POP_L}$ — que no pueden extraer elementos

[INIT]	$\$_0 \longmapsto \$_0 \; (S \rightarrow \bullet \bullet \delta)$
[SCAN$_a$]	$(A \rightarrow \delta_1 \bullet \delta_2 \bullet \delta_3) \overset{a}{\longmapsto}_U (A \rightarrow \delta_1 \bullet \delta_2 \bullet \delta_3) \; (B \rightarrow \nu \bullet a \bullet \omega)$
[SCAN$_\epsilon$]	$(A \rightarrow \delta_1 \bullet \delta_2 \bullet \delta_3) \overset{\epsilon}{\longmapsto}_U (A \rightarrow \delta_1 \bullet \delta_2 \bullet \delta_3) \; (B \rightarrow \bullet\bullet)$
[CONC-R]	$(A \rightarrow \nu \bullet \delta_1 \bullet \delta_2\omega) \; (A \rightarrow \nu\delta_1 \bullet \delta_2 \bullet \omega) \longmapsto_R (A \rightarrow \nu \bullet \delta_1\delta_2 \bullet \omega)$
[CONC-L]	$(A \rightarrow \nu\delta_1 \bullet \delta_2 \bullet \omega) \; (A \rightarrow \nu \bullet \delta_1 \bullet \delta_2\omega) \longmapsto_L (A \rightarrow \nu \bullet \delta_1\delta_2 \bullet \omega)$
[INC]	$(B \rightarrow \bullet\delta\bullet) \overset{\epsilon}{\longmapsto}_R (A \rightarrow \nu \bullet B \bullet \omega)$
[FINAL]	$\$_0 \; (S \rightarrow \bullet\delta\bullet) \longmapsto_R \$_f$

Tabla 6.19: Compilación de una CFG aplicando una estrategia bidireccional ascendente

cuyas subcadenas respectivas se solapen — nos permite garantizar que todo elemento de la cadena de entrada solamente es leído una vez en una computación dada.

En consecuencia, definiremos un Autómata a Pila Bidireccional como un autómata a pila $(V_T, V_S, \Theta_B, \$_0, \$_f)$ en el cual Θ_B puede contener transiciones de los siete tipos SWAP_R, SWAP_L, PUSH_R, PUSH_L, POP_R, POP_L y PUSH_U. Como ejemplo ilustrativo del tipo de estrategias que se pueden implementar en esta clase de autómatas, consideraremos un esquema de compilación que a partir de una gramática independiente del contexto nos permite obtener un BPDA que incorpora una estrategia de análisis bidireccional ascendente. En el autómata resultante, V_T es obtenido a partir del conjunto de símbolos terminales de la gramática considerada, V_S contiene la unión de $\{\$_0, \$_f\}$ y un conjunto de producciones con dos puntos obtenidas a partir de las producciones originales de la gramática[3], $\$_0$ es el símbolo inicial de la pila, $\$_f$ es el símbolo final de la pila y Θ_B contiene el conjunto de transiciones obtenido a partir de las reglas de compilación mostradas en la tabla 6.19. Las transiciones producidas por las reglas **[INIT]** y **[FINAL]** se encargan de iniciar y finalizar el análisis, respectivamente. Las transiciones **[SCAN$_a$]** se encargan de leer los terminales, mientras que las producciones vacías son tratadas por las transiciones creadas por la regla **[SCAN$_\epsilon$]**. Las transiciones **[CONC-R]** y **[CONC-L]** se encargan de concatenar una nueva parte de la producción a la derecha y a la izquierda, respectivamente, de la parte que había sido previamente reconocida. Una vez que una producción con el terminal B en su lado izquierdo ha sido completamente reconocida, una transición **[INC]** se encargar de continuar el proceso de análisis en una producción que contenga B en su parte derecha.

Al igual que en el caso de los autómatas a pila convencionales, la ejecución directa de los autómatas a pila bidireccionales puede tener una complejidad exponencial con respecto al tamaño de la cadena de entrada, e incluso puede no terminar. Para obtener una complejidad polinomial es preciso evitar la realización de cálculos redundantes. Esto se consigue mediante la utilización de técnicas tabulares que almacenan en forma de ítems las trazas de las configuraciones alcanzadas. El aspecto más importante para lograr una ejecución eficiente consiste en determinar la información que debe almacenarse en cada uno de dichos ítems. Por los trabajos de De la Clergerie y Lang [30] sabemos que ítems S^2 de la forma $[B, i, j; C, k, l]$ permiten diseñar técnicas de tabulación correctas y completas para cualquier estrategia de análisis sintáctico. Un ítem de este tipo indica que hemos alcanzado una configuración con C en la cima tras leer la parte $a_{k+1} \ldots a_l$ de la cadena de entrada y que B, el cual se encuentra situado inmediatamente

[3]Una producción con dos puntos $A \rightarrow \delta_1 \bullet \delta_2 \bullet \delta_3$ nos indica que la parte δ_2 de la producción $A \rightarrow \delta_1\delta_2\delta_3$ ha sido reconocida.

debajo de C, ha sido apilado tras leer la parte $a_{i+1} \dots a_j$ de la cadena de entrada[4].

Un conjunto de reglas de inferencia de la forma $\frac{antecedentes}{consecuente}$ *condiciones* nos permite derivar nuevos ítems a partir de los ítems existentes: si todos los ítems antecedentes existen y se satisfacen las condiciones, el ítem consecuente es creado. Las condiciones se refieren habitualmente a la existencia de determinadas transiciones en el autómata y a la presencia de determinados terminales en la cadena de entrada. En el caso que nos ocupa, el conjunto de reglas de inferencia es el siguiente:

$$\frac{[B,i,j;C,k,l]}{[B,i,j;F,k,l+|a|]} \quad \begin{array}{l} C \overset{a}{\longmapsto}_R F \\ a = a_{l+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{[B,i,j;C,k,l]}{[B,i,j;F,k-|a|,l]} \quad \begin{array}{l} C \overset{a}{\longmapsto}_L F \\ a = a_k \text{ or } a = \epsilon \end{array}$$

$$\frac{[B,i,j;C,k,l]}{[C,k,l;F,l,l]} \quad C \longmapsto_R C\, F$$

$$\frac{[B,i,j;C,k,l]}{[C,k,l;F,k,k]} \quad C \longmapsto_L C\, F$$

$$\frac{[B,i,j;C,k,l]}{[C,k,l;F,m,m+|a|]} \quad \begin{array}{l} C \overset{a}{\longmapsto}_U C\, F \\ a = a_{m+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{\begin{array}{c}[C,k,l;F,l,m] \\ [B,i,j;C,k,l]\end{array}}{[B,i,j;G,k,m]} \quad C\, F \longmapsto_R G$$

$$\frac{\begin{array}{c}[C,k,l;F,m,k] \\ [B,i,j;C,k,l]\end{array}}{[B,i,j;G,m,l]} \quad C\, F \longmapsto_L G$$

La complejidad en el peor caso con respecto a la longitud n de la cadena de entrada es $\mathcal{O}(n^5)$. Este hecho no es sorprendente, pues se corresponde con la complejidad de los algoritmos bidireccionales predictivos[5]. Esta complejidad puede ser reducida utilizando ítems más compactos. Del trabajo realizado por De la Clergerie y Lang [30] sabemos que si los resultados de una computación no determinista están restringidos únicamente por la propagación ascendente de los hechos calculados (lo que es aplicable a estrategias ascendentes y de tipo Earley pero no a estrategias descendentes) podemos utilizar ítems S^1 de la forma $[C,k,l]$ que sólo almacenan el símbolo C que se encuentra en la cima de la pila y las posiciones k y l que indican la parte de la cadena de entrada leída durante el análisis de C. El conjunto de las reglas de inferencia que se aplica para la generación de este tipo de ítems se obtiene a partir de las reglas de inferencia definidas para los ítems S^2, resultando ser el siguiente:

$$\frac{[C,k,l]}{[F,k,l+|a|]} \quad \begin{array}{l} C \overset{a}{\longmapsto}_R F \\ a = a_{l+1} \text{ or } a = \epsilon \end{array}$$

[4]En el caso de un autómata unidireccional que trabaje únicamente con transiciones-r, tan sólo se precisan las posiciones j y l. En el caso de que dicho autómata trabajase únicamente con transiciones-l, tan sólo se precisarían las posiciones i y k. En el caso de los autómatas bidireccionales, precisamos de las cuatro posiciones puesto que mezclamos transiciones-r y transiciones-l.

[5]Véase por ejemplo el algoritmo *head-corner* predictivo presentado por Sikkel en [83, chapter 11].

$$\frac{[C,k,l]}{[F,k-|a|,l]} \quad \begin{array}{l} C \overset{a}{\longmapsto}_L F \\ a = a_k \text{ or } a = \epsilon \end{array}$$

$$\frac{[C,k,l]}{[F,l,l]} \quad C \longmapsto_R C\,F$$

$$\frac{[C,k,l]}{[F,k,k]} \quad C \longmapsto_L C\,F$$

$$\frac{[C,k,l]}{[F,m,m+|a|]} \quad \begin{array}{l} C \overset{a}{\longmapsto}_U C\,F \\ a = a_{m+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{\begin{array}{c}[F,l,m]\\ [C,k,l]\end{array}}{[G,k,m]} \quad C\,F \longmapsto_R G$$

$$\frac{\begin{array}{c}[F,m,k]\\ [C,k,l]\end{array}}{[G,m,l]} \quad C\,F \longmapsto_L G$$

Este conjunto de reglas de inferencia, cuando se aplica a un autómata obtenido a partir de una gramática independiente del contexto mediante las reglas de compilación de la tabla 6.19, resulta ser equivalente al conjunto de pasos deductivos del esquema de análisis asociado al analizador bidireccional ascendente diseñado por De Vreught y Honig [31] descrito en [83]. Del esquema de compilación obtenemos directamente los siguientes pasos:

$$[\textbf{SCAN}_a] \quad \frac{[A \to \delta_1 \bullet \delta_2 \bullet \delta_3, k, l]}{[B \to \nu \bullet a \bullet \omega, m, m+1]} \quad a = a_{m+1}$$

$$[\textbf{SCAN}_\epsilon] \quad \frac{[A \to \delta_1 \bullet \delta_2 \bullet \delta_3, k, l]}{[B \to \bullet\bullet, m, m]}$$

$$[\textbf{CONC-R}] \quad \frac{\begin{array}{c}[A \to \nu\delta_1 \bullet \delta_2 \bullet \omega, l, m]\\ [A \to \nu \bullet \delta_1 \bullet \delta_2\omega, k, l]\end{array}}{[A \to \nu \bullet \delta_1\delta_2 \bullet \omega, k, m]}$$

$$[\textbf{CONC-L}] \quad \frac{\begin{array}{c}[A \to \nu \bullet \delta_1 \bullet \delta_2\omega, m, k]\\ [A \to \nu\delta_1 \bullet \delta_2 \bullet \omega, k, l]\end{array}}{[A \to \nu \bullet \delta_1\delta_2 \bullet \omega, m, l]}$$

$$[\textbf{INC}] \quad \frac{[B \to \bullet\delta\bullet, k, l]}{[A \to \nu \bullet B \bullet \omega, k, l]}$$

Teniendo en cuenta que en los pasos deductivos las condiciones $a = a_{m+1}$ referidas a la presencia de terminales en la cadena de entrada se introducen entre los antecedentes bajo la forma de unos ítems especiales $[a, m, m+1]$ llamados hipótesis, que los antecedentes que no intervienen en la aplicación de un determinado paso pueden ser eliminados y que el orden de los antecedentes es irrelevante, obtenemos el siguiente conjunto de pasos deductivos, ahora sí idéntico al conjunto de pasos deductivos del esquema descrito por Sikkel en [83]:

$$\textbf{[SCAN}_a\textbf{]} \quad \frac{[a, m, m+1]}{[B \to \nu \bullet a \bullet \omega, m, m+1]}$$

$$\textbf{[SCAN}_\epsilon\textbf{]} \quad \frac{}{[B \to \bullet\bullet, m, m]}$$

$$\textbf{[CONC]} \quad \frac{\begin{array}{c}[A \to \nu \bullet \delta_1 \bullet \delta_2\omega, k, l] \\ [A \to \nu\delta_1 \bullet \delta_2 \bullet \omega, l, m]\end{array}}{[A \to \nu \bullet \delta_1\delta_2 \bullet \omega, k, m]}$$

$$\textbf{[INC]} \quad \frac{[B \to \bullet\delta\bullet, k, l]}{[A \to \nu \bullet B \bullet \omega, k, l}$$

6.6.2 Autómatas lineales de índices bidireccionales orientados a la derecha

Los autómatas lineales de índices orientados a la derecha son una subclase de los autómatas lineales de índices apta para implementar estrategias de análisis sintáctico de gramáticas de adjunción de árboles en las cuales las adjunciones se reconocen de modo ascendente. Si tenemos en cuenta que todas las estrategias bidireccionales para TAG descritas en la literatura reconocen las adjunciones precisamente de modo ascendente [52, 87, 35], los R–LIA surgen de modo natural como punto de inicio para la definición de un modelo de autómata bidireccional para lenguajes de adjunción de árboles.

Recordamos que un R–LIA se puede definir como una tupla $(V_T, V_S, V_I, \Theta, \$_0, \$_f)$ en la cual V_T representa un conjunto finito de símbolos terminales, V_S un conjunto también finito de símbolos de pila, V_I un conjunto finito de índices, $\$_0 \in V_S$ es el símbolo inicial de pila, $\$_f \in V_S$ es el símbolo final de pila y Θ es un conjunto finito de transiciones, que pueden ser de cualquiera de los cuatro tipos siguientes:

SWAP$[\circ\circ]$**:** Transiciones de la forma $C[\circ\circ\gamma] \overset{a}{\longmapsto} F[\circ\circ\gamma']$, donde $\circ\circ$ representa una lista de índices η cualquiera, $\gamma, \gamma' \in V_I \cup \{\epsilon\}$ y bien γ ó γ' (o ambos) debe ser la cadena vacía, que sustituyen el elemento $C[\delta\,\gamma]$ situado en la cima de la pila por $F[\delta\,\gamma']$ al tiempo que leen a en la cadena de entrada.

PUSH$[\,]$**:** Transiciones de la forma $C[\circ\circ] \longmapsto C[\circ\circ]\ F[\,]$ que apilan $F[\,]$ sobre $C[\delta]$.

POP$[\,]$**:** Transiciones de la forma $C[\circ\circ]\ F[\,] \longmapsto G[\circ\circ]$ que reemplazan $C[\delta]$ y $F[\,]$ por $G[\delta]$.

POP$[\circ\circ]$**:** Transiciones de la forma $C[\,]\ F[\circ\circ] \longmapsto G[\circ\circ]$ que reemplazan $C[\,]$ y $F[\delta]$ por $G[\delta]$.

Todas estas transiciones son transiciones-r puesto que leen la cadena de entrada de izquierda a derecha.

Con el fin de definir los Autómatas Lineales de Índices Bidireccionales orientados a la Derecha (*Bidirectional Right-oriented Linear Indexed Automata*, BR–LIA), extenderemos el concepto de BPDA para que sea capaz de trabajar con estas transiciones y sus correspondientes versiones duales. En primer lugar definiremos el concepto de configuración. Una configuración de un BR–LIA viene definida por la pila Υ, que contiene triples $(C[\delta], k, l)$, donde $C \in V_S$, $\delta \in V_I^*$ y $k, l \in \mathbb{N}$. La configuración inicial vendrá dada por la pila que contiene únicamente el triple $(\$_0[\,], 0, 0)$. La configuración final vendrá dada por una pila que contendrá como único elemento el triple $(\$_f[\,], 0, n)$, donde n es la longitud de la cadena de entrada. Las configuraciones intermedias se obtendrán tras aplicar alguna de las transiciones siguientes:

SWAP$_{\rm R}[\circ\circ]$**:** Transiciones de la forma $C[\circ\circ p] \overset{a}{\longmapsto}_R F[\circ\circ q]$, donde $\circ\circ$ representa una lista de índices δ cualquiera, $\gamma, \gamma' \in V_I \cup \{\epsilon\}$ y bien γ ó γ' (o ambos) debe ser la cadena vacía, que al ser aplicadas a una configuración $\Upsilon\,(C[\delta\,\gamma], k, l)$ producen una nueva configuración $\Upsilon\,(F[\delta\,\gamma'], k, l + |a|)$.

SWAP$_{\rm L}[\circ\circ]$**:** Transiciones de la forma $C[\circ\circ\gamma] \overset{a}{\longmapsto}_L F[\circ\circ\gamma']$, donde $\circ\circ$ representa una lista de índices δ cualquiera, $\gamma, \gamma' \in V_I \cup \{\epsilon\}$ y bien γ ó γ' (o ambos) debe ser la cadena vacía, que al ser aplicadas a una configuración $\Upsilon\,(C[\delta\,\gamma], k, l)$ producen una nueva configuración $\Upsilon\,(F[\delta\,\gamma'], k - |a|, l)$.

PUSH$_{\rm R}[\,]$**:** Transiciones de la forma $C[\circ\circ] \longmapsto_R C[\circ\circ]\ F[\,]$ que al ser aplicadas a una configuración $\Upsilon\,(C[\delta], k, l)$ resulta en una configuración $\Upsilon\,(C[\delta], k, l)\,(F[\,], l, l)$.

PUSH$_{\rm L}[\,]$**:** Transiciones de la forma $C[\circ\circ] \longmapsto_L C[\circ\circ]\ F[\,]$ que al ser aplicadas a una configuración $\Upsilon\,(C[\delta], k, l)$ resulta en una configuración $\Upsilon\,(C[\delta], k, l)\,(F[\,], k, k)$.

PUSH$_U[\,]$**:** Transiciones de la forma $C[\circ\circ] \overset{a}{\longmapsto}_U C[\circ\circ]\ F[\,]$, donde $a = a_{m+1}$ o $a = \epsilon$, que al ser aplicadas a una configuración $\Upsilon\,(C[\delta], k, l)$ permiten obtener una configuración $\Upsilon\,(C[\delta], k, l)\,(F[\,], m, m + |a|)$

POP$_{\rm R}[\,]$**:** Transiciones de la forma $C[\circ\circ]\ F[\,] \longmapsto_R G[\circ\circ]$, que al ser aplicadas a una configuración $\Upsilon\,(C[\delta], k, l)\,(F[\,], l, m)$ permiten obtener una configuración $\Upsilon\,(G[\delta], k, m)$.

POP$_{\rm L}[\,]$**:** Transiciones de la forma $C[\circ\circ]\ F[\,] \longmapsto_L G[\circ\circ]$ que al ser aplicadas a una configuración $\Upsilon\,(C[\delta], k, l)\,(F[\,], m, k)$ permiten obtener una configuración $\Upsilon\,(G[\delta], m, l)$.

POP$_{\rm R}[\circ\circ]$**:** Transiciones de la forma $C[\,]\ F[\circ\circ] \longmapsto_R G[\circ\circ]$ que al ser aplicadas a una configuración $\Upsilon\,(C[\,], k, l)\,(F[\delta], l, m)$ permiten obtener una configuración $\Upsilon\,(G[\delta], k, m)$.

POP$_{\rm L}[\circ\circ]$**:** Transiciones de la forma $C[\,]\ F[\circ\circ] \longmapsto_L G[\circ\circ]$ que al ser aplicadas a una configuración $\Upsilon\,(C[\,], k, l)\,(F[\delta], m, k)$ permiten obtener una configuración $\Upsilon\,(G[\delta], m, l)$.

Definimos por tanto un Autómata Lineal de Índices Bidireccional orientado a la Derecha como un R–LIA $(V_T, V_S, V_I, \Theta_B, \$_0, \$_f)$ donde Θ_B contiene transiciones de los nueve tipos descritos anteriormente.

Como un ejemplo de la clase de analizadores sintácticos que pueden ser implementados en BR–LIA, mostraremos un esquema de compilación de una gramática de adjunción de árboles en un BR–LIA que aplica una estrategia bidireccional ascendente. En el autómata resultante, V_T es el conjunto de terminales de la gramática de partida, V_S es la unicón de $\{\$_0, \$_f\}$ con un conjunto de producciones con dos puntos, V_I es el conjunto de nodos de adjunción, el elemento inicial es $\$_0$ mientras que el elemento final es $\$_f$, y Θ_B contiene el conjunto de transiciones producido por las reglas de compilación mostradas en la tabla 6.20.

A continuación se explica la función de cada una de las transiciones. Las transiciones producidas por las reglas **[INIT]** y **[FINAL]** se encargan de iniciar y finalizar el proceso de análisis, respectivamente. Las transiciones **[SCAN**$_a$**]** y **[SCAN**$_\epsilon$**]** se encargan de reconocer los terminales y las producciones vacías, respectivamente. Las transiciones **[CONC-R]** y **[CONC-L]** concatenan a la derecha e izquierda de la parte de la producción previamente analizada una nueva porción que acaba de ser reconocida y que no incluye nodos de la espina de un árbol elemental. Las transiciones **[SCONC-R]** y **[SCONC-L]** realizan una tarea similar pero en el caso de haya nodos de la espina involucrados. Las transiciones **[INC]** continúan el recorrido ascendente de un árbol elemental cuando un subárbol ha sido completamente analizado. Las transiciones **[FOOT]** inician el recorrido ascendente de un árbol auxiliar a partir de su nodo pie.

[INIT]	$\$_0[\circ\circ] \longmapsto_R \$_0[\circ\circ]\ (\top \to \bullet\bullet\, \mathbf{R}^\alpha)[\,]$ $\alpha \in \boldsymbol{I}$
[SCAN$_a$]	$(N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3)[\circ\circ] \overset{a}{\longmapsto}_U (N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3)[\circ\circ]\ (M^{\gamma'} \to \nu \bullet a \bullet \omega)[\,]$
[SCAN$_\epsilon$]	$(N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3)[\circ\circ] \overset{\epsilon}{\longmapsto}_U (N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3)[\circ\circ]\ (M^{\gamma'} \to \bullet\bullet)[\,]$
[CONC-R]	$(N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega)[\circ\circ]\ (N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega)[\,] \longmapsto_R (N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega)[\circ\circ]$ $\forall M^\gamma \in \delta_2,\ M^\gamma \notin \text{espina}(\gamma)$
[CONC-L]	$(N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega)[\circ\circ]\ (N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega)[\,] \longmapsto_L (N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega)[\circ\circ]$ $\forall M^\gamma \in \delta_1,\ M^\gamma \notin \text{espina}(\gamma)$
[SCONC-R]	$(N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega)[\,]\ (N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega)[\circ\circ] \longmapsto_R (N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega)[\circ\circ]$ $\exists M^\gamma \in \delta_2,\ M^\gamma \in \text{espina}(\gamma)$
[SCONC-L]	$(N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega)[\,]\ (N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega)[\circ\circ] \longmapsto_L (N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega)[\circ\circ]$ $\exists M^\gamma \in \delta_1,\ M^\gamma \in \text{espina}(\gamma)$
[INC]	$(M^\gamma \to \bullet\delta\bullet)[\circ\circ] \overset{\epsilon}{\longmapsto}_R (N^\gamma \to \nu \bullet M^\gamma \bullet \omega)[\circ\circ]$ $\mathbf{nil} \in \text{adj}(M^\gamma)$
[FOOT]	$(M^\gamma \to \bullet\delta\bullet)[\circ\circ] \overset{\epsilon}{\longmapsto}_R (\mathbf{F}^\beta \to \bullet\bot\bullet)[\circ\circ M^\gamma]$ $\beta \in \text{adj}(M^\gamma)$
[ADJ]	$(\top \to \bullet\mathbf{R}^\beta\bullet)[\circ\circ M^\gamma] \overset{\epsilon}{\longmapsto}_R (N^\gamma \to \nu \bullet M^\gamma \bullet \omega)[\circ\circ]$ $\beta \in \text{adj}(M^\gamma)$
[FINAL]	$\$_0[\circ\circ]\ (\top \to \bullet\bullet\, \mathbf{R}^\alpha)[\,] \longmapsto_R \$_f$

Tabla 6.20: Compilación de una TAG aplicando una estrategia bidireccional ascendente

Las transiciones **[ADJ]** dan por finalizada una adjunción en el momento en que se ha finalizado el recorrido de un árbol auxiliar.

Tal y como ocurría en el caso de los autómatas a pila bidireccionales, la ejecución directa de los BR–LIA puede presentar complejidad exponencial o incluso no terminar. Podemos obtener una complejidad polinomial mediante la utilización de una técnica de tabulación que haga uso de los ítems S^2 o S^1 definidos para el caso de los BPDA. Para llevar a cabo esta idea, es preciso extender la forma de estos ítems para almacenar información de la pila de índices que lleva asociado cada uno de los símbolos de pila. En lugar de almacenar la pila de índices completa, sólo almacenaremos el elemento de la cima y un puntero lógico al resto de dicha pila [55]. En caso de necesitar acceder al resto de elementos almacenados en una pila de índices, basta con seguir la cadena de enlaces. En consecuencia, podemos extender los ítems S^2 para obtener ítems de la forma $[B, i, j; C, k, l \mid \gamma \mid D, r, s; E, t, u]$, donde γ es el índice en la cima de la pila de índices asociada a C, mientras que $(D, r, s; E, t, u)$ es un puntero lógico. En el caso de que C tenga asociado una pila de índices vacía, en lugar de γ, D, E, r, s, t y u almacenaremos un indicador especial denotado por $-$.

Todas los algoritmos de análisis sintáctico para gramáticas de adjunción de árboles bidireccionales descritos en la literatura realizan un recorrido ascendente de los árboles elementales [87, 35, 53], combinado a veces con un recorrido mixto de tipo Earley [52] en algunas partes de los árboles. En consecuencia, una interpretación tabular basada en ítems S^1 de la forma $[C,k,l \mid \gamma \mid E,t,u]$ será correcta y completa cuando se aplique a un autómata que implemente dichas estrategias. El conjunto de reglas de inferencia que permite generar nuevos ítems a partir de ítems existentes es el que se muestra a continuación:

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,k,l+|a| \mid \gamma \mid E,t,u]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto}_R F[\circ\circ] \\ a = a_{l+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,k-|a|,l \mid \gamma \mid E,t,u]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto}_L F[\circ\circ] \\ a = a_{k} \text{ or } a = \epsilon \end{array}$$

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,k,l+|a| \mid \gamma' \mid C,k,l]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto}_R F[\circ\circ\gamma'] \\ a = a_{l+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,k-|a|,l \mid \gamma' \mid C,k,l]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto}_L F[\circ\circ\gamma'] \\ a = a_{k} \text{ or } a = \epsilon \end{array}$$

$$\frac{\begin{array}{c}[C,k,l \mid \gamma \mid E,t,u] \\ [E,t,u \mid \gamma' \mid E',t',u']\end{array}}{[F,k,l+|a| \mid \gamma' \mid E',t',u']} \quad \begin{array}{l} C[\circ\circ\gamma] \overset{a}{\longmapsto}_R F[\circ\circ] \\ a = a_{l+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{\begin{array}{c}[C,k,l \mid \gamma \mid E,t,u] \\ [E,t,u \mid \gamma' \mid E',t',u']\end{array}}{[F,k-|a|,l \mid \gamma' \mid E',t',u']} \quad \begin{array}{l} C[\circ\circ\gamma] \overset{a}{\longmapsto}_L F[\circ\circ] \\ a = a_{k} \text{ or } a = \epsilon \end{array}$$

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,l,l \mid - \mid -,-,-]} \quad C[\circ\circ] \longmapsto_R C[\circ\circ]\ F[\,]$$

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,k,k \mid - \mid -,-,-]} \quad C[\circ\circ] \longmapsto_L C[\circ\circ]\ F[\,]$$

$$\frac{[C,k,l \mid \gamma \mid E,t,u]}{[F,m,m+|a| \mid - \mid -,-,-]} \quad \begin{array}{l} C[\circ\circ] \overset{a}{\longmapsto}_U C[\circ\circ]\ F[\,] \\ a = a_{m+1} \text{ or } a = \epsilon \end{array}$$

$$\frac{\begin{array}{c}[F,l,m \mid - \mid -,-,-] \\ [C,k,l \mid \gamma \mid E,t,u]\end{array}}{[G,k,m \mid \gamma \mid E,t,u]} \quad C[\circ\circ]\ F[\,] \longmapsto_R G[\circ\circ]$$

$$\frac{\begin{array}{c}[F,m,k \mid - \mid -,-,-] \\ [C,k,l \mid \gamma \mid E,t,u]\end{array}}{[G,m,l \mid \gamma \mid E,t,u]} \quad C[\circ\circ]\ F[\,] \longmapsto_L G[\circ\circ]$$

$$\frac{\begin{array}{c}[F,l,m \mid \gamma \mid E,t,u] \\ [C,k,l \mid - \mid -,-,-]\end{array}}{[G,k,m \mid \gamma \mid E,t,u]} \quad C[\,]\ F[\circ\circ] \longmapsto_R G[\circ\circ]$$

$$\frac{\begin{array}{c}[F, m, k \mid \gamma \mid E, t, u] \\ [C, k, l \mid - \mid -, -, -]\end{array}}{[G, m, l \mid \gamma \mid E, t, u]} \quad C[\,]\ F[\circ\circ] \longmapsto_L G[\circ\circ]$$

La complejidad en el peor caso con respecto a la longitud n de la cadena de entrada es $\mathcal{O}(n^6)$, la complejidad estándar de los algoritmos tabulares de análisis sintáctico de gramáticas de adjunción de árboles. De haber aplicado ítems S^2, la complejidad habría aumentado a $\mathcal{O}(n^{12})$, que se correspondería con la complejidad de un hipotético algoritmo de tipo *head-corner* predictivo, extensión para TAG del definido por Sikkel para las gramáticas independientes del contexto [83, chapter 11].

Este conjunto de reglas de inferencia, cuando se aplica a un autómata resultante del esquema de compilación cuyas reglas se muestran en la tabla 6.20 da lugar al conjunto de pasos deductivos del esquema de análisis sintáctico **dVH** ya presentado al ver los esquemas de análisis ascendentes. El conjunto de pasos deductivos obtenidos directamente de las reglas de compilación es el siguiente:

$$[\mathbf{SCAN}_a] \quad \frac{[N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3, k, l \mid \eta \mid E, t, u]}{[M^{\gamma'} \to \nu \bullet a \bullet \omega, m, m+1 \mid - \mid -, -, -]} \quad a = a_{m+1}$$

$$[\mathbf{SCAN}_\epsilon] \quad \frac{[N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3, k, l \mid \eta \mid E, t, u]}{[M^{\gamma'} \to \bullet\bullet, m, m \mid - \mid -, -, -]}$$

$$[\textbf{CONC-R}] \quad \frac{\begin{array}{c}[N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega, l, m \mid - \mid -, -, -] \\ [N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega, k, l \mid \eta \mid E, t, u]\end{array}}{[N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega, k, m \mid \eta \mid E, t, u]}$$

$$[\textbf{CONC-L}] \quad \frac{\begin{array}{c}[N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega, m, k \mid - \mid -, -, -] \\ [N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega, k, l \mid \eta \mid E, t, u]\end{array}}{[N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega, m, l \mid \eta \mid E, t, u]}$$

$$[\textbf{SCONC-R}] \quad \frac{\begin{array}{c}[N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega, l, m \mid \eta \mid E, t, u] \\ [N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega, k, l \mid - \mid -, -, -]\end{array}}{[N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega, k, m \mid \eta \mid E, t, u]}$$

$$[\textbf{SCONC-L}] \quad \frac{\begin{array}{c}[N^\gamma \to \nu \bullet \delta_1 \bullet \delta_2\omega, m, k \mid \eta \mid E, t, u] \\ [N^\gamma \to \nu\delta_1 \bullet \delta_2 \bullet \omega, k, l \mid - \mid -, -, -]\end{array}}{[N^\gamma \to \nu \bullet \delta_1\delta_2 \bullet \omega, m, l \mid \eta \mid E, t, u]}$$

$$[\textbf{INC}] \quad \frac{[M^\gamma \to \bullet\delta\bullet, k, l \mid \eta \mid E, t, u]}{[N^\gamma \to \nu \bullet M^\gamma \bullet \omega, k, l \mid \eta \mid E, t, u]}$$

$$[\textbf{FOOT}] \quad \frac{[M^\gamma \to \bullet\delta\bullet, k, l \mid \eta \mid E, t, u]}{[\mathbf{F}^\beta \to \bullet\bot\bullet, k, l \mid M^\gamma \mid M^\gamma \to \bullet\delta\bullet, k, l]}$$

$$[\textbf{ADJ}] \quad \frac{\begin{array}{c}[\top \to \bullet\mathbf{R}^\beta\bullet, k, l \mid M^\gamma \mid M^\gamma \to \bullet\delta\bullet, t, u] \\ [M^\gamma \to \bullet\delta\bullet, t, u \mid \eta' \mid E', t', u']\end{array}}{[N^\gamma \to \nu \bullet M^\gamma \bullet \omega, k, l \mid \eta' \mid E', t', u']}$$

Observamos que el antecedente $[N^\gamma \to \delta_1 \bullet \delta_2 \bullet \delta_3, k, l \mid \eta \mid E, t, u]$ en los pasos $[\mathbf{SCAN}_a]$ y $[\mathbf{SCAN}_\epsilon]$ puede ser eliminado puesto que no influye en la aplicación de dichos pasos deductivos. Por otra parte, los pasos **[CONC-R]** y **[SCONC-L]** son realmente un mismo paso, puesto que el orden de los antecedentes no es relevante. Lo mismo ocurre en el caso de los pasos **[CONC-L]** y **[SCONC-R]**. Una vez aplicadas estas modificaciones obtenemos el siguiente conjunto de pasos deductivos:

$$\textbf{[SCAN}_a\textbf{]} \quad \frac{[a, m, m+1]}{[M^{\gamma'} \rightarrow \nu \bullet a \bullet \omega, m, m+1 \mid - \mid -, -, -]}$$

$$\textbf{[SCAN}_\epsilon\textbf{]} \quad \frac{}{[M^{\gamma'} \rightarrow \bullet\bullet, m, m \mid - \mid -, -, -]}$$

$$\textbf{[CONC-R]} \quad \frac{\begin{array}{c}[N^\gamma \rightarrow \nu \bullet \delta_1 \bullet \delta_2\omega, k, l \mid \eta \mid E, t, u] \\ [N^\gamma \rightarrow \nu\delta_1 \bullet \delta_2 \bullet \omega, l, m \mid - \mid -, -, -]\end{array}}{[N^\gamma \rightarrow \nu \bullet \delta_1\delta_2 \bullet \omega, k, m \mid \eta \mid E, t, u]}$$

$$\textbf{[CONC-L]} \quad \frac{\begin{array}{c}[N^\gamma \rightarrow \nu \bullet \delta_1 \bullet \delta_2\omega, m, k \mid - \mid -, -, -] \\ [N^\gamma \rightarrow \nu\delta_1 \bullet \delta_2 \bullet \omega, k, l \mid \eta \mid E, t, u]\end{array}}{[N^\gamma \rightarrow \nu \bullet \delta_1\delta_2 \bullet \omega, m, l \mid \eta \mid E, t, u]}$$

$$\textbf{[INC]} \quad \frac{[M^\gamma \rightarrow \bullet\delta\bullet, k, l \mid \eta \mid E, t, u]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, k, l \mid \eta \mid E, t, u]}$$

$$\textbf{[FOOT]} \quad \frac{[M^\gamma \rightarrow \bullet\delta\bullet, k, l \mid \eta \mid E, t, u]}{[\mathbf{F}^\beta \rightarrow \bullet\bot\bullet, k, l \mid M^\gamma \mid M^\gamma \rightarrow \bullet\delta\bullet, k, l]}$$

$$\textbf{[ADJ]} \quad \frac{\begin{array}{c}[\top \rightarrow \bullet\mathbf{R}^\beta\bullet, k, l \mid M^\gamma \mid M^\gamma \rightarrow \bullet\delta\bullet, t, u] \\ [M^\gamma \rightarrow \bullet\delta\bullet, t, u \mid \eta' \mid E', t', u']\end{array}}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, k, l \mid \eta' \mid E', t', u']}$$

La única diferencia entre estos pasos deductivos y los definidos para el esquema **dVH** radica en el hecho de que cierta información presente en los ítems es redundante. Este hecho se ve claramente en el caso del paso deductivo **[ADJ]**. En este paso, el elemento $M^\gamma \rightarrow \bullet\delta\bullet$ del ítem $[\top \rightarrow \bullet\mathbf{R}^\beta\bullet, k, l \mid M^\gamma \mid M^\gamma \rightarrow \bullet\delta\bullet, t, u]$ es redundante puesto que la producción asociada con M^γ ya está presente en el ítem. El propio elemento M^γ es en sí mismo redundante, puesto que el ítem es válido para todo nodo M^γ tal que β pueda ser adjuntado en M^γ. Aplicando este refinamiento a los ítems, obtenemos exactamente el esquema de análisis **dVH**.

6.7 Estudio empírico

Los experimentos con los autómatas ha sido realizados utilizando el sistema DyALog [26], un entorno para la ejecución de programas lógicos basado en la interpretación mediante programación dinámica (tabulación) de autómatas lógicos a pila (*Logic Push-Down Automata*, LPDA) [30]. Posteriormente, DyALog se ha ido enriqueciendo con la incorporación de otros modelos de autómata más restrictivos, especializándose poco a poco en el desarrollo y prueba de analizadores sintácticos para lenguajes naturales. Es en este contexto en el que se han incorporado diversos modelos de autómata para el análisis sintáctico de gramáticas de adjunción de árboles junto con sus técnicas de tabulación específicas [10].

Para realizar los experimentos se ha tomado una pequeña gramática del francés obtenida de la distribución de XTAG [36]. Dicha gramática consta de 50 esquemas de árboles, 117 lemas y 345 realizaciones léxicas. En XTAG, los árboles elementales se derivan de esquemas de árboles en los

	TD	TD + BU-F
Jean aime Marie (Jean ama a Marie)	1.22	0.91
Yves montre des fleurs à Sharon (Yves muestra las flores a Sharon)	1.85	1.20
Kathy regarde un livre avec Stéphane (Kathy mira un libro con Stéphane)	1.43	1.23
un ami américain de Yves qui visite la France boit du bon vin (un amigo americano de Yves que visita Francia bebe buen vino)	2.91	2.03
la jolie fille pense que Andrew est le grand ami de son chien (la chica bonita piensa que Andrew es el gran amigo de su perro)	2.18	1.64
Marie comprend que Kathy dort bien chez Andrew (Marie cree que Kathy duerme bien en casa de Andrew)	1.86	1.41
Anne voit les fleurs que Yves cherche sur la table (Anne ve las flores que Yves busca sobre la mesa)	2.25	1.76
Jean pense que Béatrice perd les pédales (Jean piensa que Béatrice pierde los estribos)	1.80	1.45
le petit chat de Sabine court vite (el pequeño gato de Sabine corre rápido)	1.38	1.17
la fille de Marie aime bien les fleurs (a la hija de Marie le gustan mucho las flores)	1.60	1.32
qui est Bonnie (quién es Bonnie)	1.28	0.95
qui dit que Yves a un chien (quién dice que Yves tiene un perro)	1.66	1.35

Tabla 6.21: Resultados con la técnica de tabulación genérica

cuales un nodo ancla distinguido (y puede que también algún nodo co-ancla) puede ser asociado a una determinada palabra. Los nodos de los árboles elementales pueden estar decorados por estructuras de rasgos que permiten expresar de forma concisa concordancias y otras restricciones semánticas. Aunque en este capítulo no se ha tratado el tema de la evaluación de gramáticas de adjunción con estructuras de rasgos, estas no presentan mayor problema. Concretamente, podemos utilizar las características que DyALog posee como entorno de programación lógica para evaluar dichas estructuras.

Los experimentos han consistido en medir el tiempo de ejecución (en segundos) de diferentes analizadores sobre un conjunto de 14 frases en un ordenador Pentium II a 450 Mhz con sistema operativo Linux. La tabla 6.21 muestra los resultados para una estrategia predictiva implementada mediante U–LIA. Bajo la columna TD se indican los resultados para una estrategia descendente tanto en el tratamiento de las adjunciones como en la evaluación de las estructuras de rasgos. Bajo la columna TD + BU-F se indican los resultados para un estrategia descendente en lo que respecta al reconocimiento de las adjunciones pero ascendente en lo que respecta a la evaluación de las estructuras de rasgos. Podemos observar que la evaluación ascendente de las

n	1	2	3	4	5	6	7	8	9	10
Tiempo	1,67	2,16	2,89	3,41	3,97	5,09	5,72	6,58	7,87	8,86
Análisis	2	5	14	42	132	429	1.430	4.862	16.796	58.786

n	11	12	13	14	15
Tiempo	9,87	11,09	13,12	14,59	16,16
Análisis	208.012	742.900	2.674.440	9.694.845	35.357.670

Tabla 6.22: Resultados para *Yves aime (la fille de)*^{n} *Sabine*

	TD	BU
Jean aime Marie	0.35	0.31
Yves montre des fleurs à Sharon	0.39	0.31
Kathy regarde un livre avec Stéphane	0.36	0.31
un ami américain de Yves qui visite la France boit du bon vin	0.65	0.46
la jolie fille pense que Andrew est le grand ami de son chien	0.48	0.39
Marie comprend que Kathy dort bien chez Andrew	0.43	0.35
Anne voit les fleurs que Yves cherche sur la table	0.49	0.36
Jean pense que Béatrice perd les pédales	0.42	0.32
le petit chat de Sabine court vite	0.42	0.33
la fille de Marie aime bien les fleurs	0.40	0.35
qui est Bonnie	0.34	0.30
qui dit que Yves a un chien	0.38	0.33

Tabla 6.23: Resultados tras aplicar optimizaciones específicas para TAG

estructuras de rasgos aligera la carga computacional del analizador.

Para comprobar el rendimiento de los autómatas en entornos muy desfavorables, hemos probado a analizar diversas sentencias a las que hemos ido añadiendo artificiosamente conjuntos de palabras que provocaban un aumento exponencial en el número de posibles análisis de la frase. En la tabla 6.22 se muestran los resultados para la frase *Yves aime (la fille de)*^{n} *Sabine*[6], donde n representa el número de veces que *la fille de* aparece en la frase, con una analizador descendente. Como podemos observar, el rendimiento del analizador no se resiente a pesar del aumento desmesurado del número de posibles análisis. Ello se debe al alto grado de compartición de subestructuras comunes alcanzado por las técnicas de tabulación descritas en este capítulo.

Los modelos de autómata que hemos definido son genéricos en el sentido de que no presuponen la utilización de ningún formalismo gramatical ni de ningún esquema de compilación predeterminado. Esto hace que sea posible aplicarlos tanto al análisis de TAG como de otros formalismos gramaticales que también generan la clase de los lenguajes de adjunción de árboles, tales como las gramáticas lineales de índices [9]. El precio a pagar por esta generalidad es una sobrecarga en el tiempo de ejecución, sobrecarga que podría ser evitada si se conociese de antemano el formalismo gramatical y algunas características de los esquemas de compilación utilizados. En particular, se pueden definir optimizaciones muy eficaces cuando se trata de analizar

[6]Yves ama a (la hija de)^n Sabine.

gramáticas de adjunción de árboles compiladas mediante esquemas de tipo llamada/retorno [10]. La tabla 6.23 muestra los resultados obtenidos tras aplicar estas optimizaciones. Bajo la columna TD se encuentran los resultados para una estrategia descendente en lo que respecta al tratamiento de las adjunciones. Bajo la columna BU se muestran los resultados para una evaluación ascendente de las adjunciones, implementada en un R–LIA. En ambos casos las estructuras de rasgos se han evaluado ascendentemente.