Capítulo 7

Autómatas a pila embebidos ascendentes

En este capítulo se presentan los autómatas a pila embebidos ascendentes o BEPDA, que constituye el modelo dual de los autómatas a pila embebidos estudiados en el capítulo precedente. Las aportaciones más importantes realizadas en este capítulo son: la definición formal consistente de los BEPDA con estados, la definición de los BEPDA sin estados, la demostración de que estos autómatas aceptan la clase de los lenguajes de adjunción de árboles, la definición de esquemas de compilación para gramáticas de adjunción de árboles y gramáticas lineales de índices y el diseño de técnicas de tabulación que permiten la ejecución de los BEPDA en tiempo polinomial. Este capítulo está basado en [17].

7.1 Introducción

Los autómatas a pila embebidos ascendentes (*Bottom-up Embedded Push-Down Automata*, BEPDA) fueron descritos por Schabes en [168] y por Schabes y Vijay-Shanker en [176] con el fin de proporcionar un modelo de autómata sobre el cual poder ejecutar un algoritmo de análisis sintáctico de tipo LR para gramáticas de adjunción de árboles. Rambow trata de definir formalmente las propiedades de este modelo de autómata en [152], aunque no lo consigue totalmente puesto que su trabajo presenta inconsistencias entre la definición de las configuraciones y la definición de las transiciones permitidas.

Los autómatas a pila embebidos ascendentes constituyen el modelo dual de los autómatas a pila embebidos, pues mientras estos últimos permiten simular derivaciones de una gramática de adjunción de árboles en las cuales las adjunciones se reconocen de modo descendente, los BEPDA sólo pueden tratar derivaciones en las cuales las adjunciones se reconocen de modo ascendente. Esta dualidad es una consecuencia del juego de transiciones que se permite utilizar en cada uno de los modelos de autómata citados.

7.2 Definición con estados

Al igual que los autómatas a pila embebidos, los autómatas a pila embebidos ascendentes trabajan con una pila, que denominamos *principal*, constituida a su vez por pilas. La pila principal, la cadena de entrada y el control finito constituyen los tres elementos básicos de un BEPDA. La *configuración* en un momento dado viene determinada por el estado del control finito en que se encuentra el autómata, el contenido de la pila y la parte de la cadena de entrada que resta por

leer, como se muestra en las figuras 7.1 y 7.3. Un conjunto de *transiciones* permite cambiar de configuración. Existen dos tipos diferentes de transiciones:

1. Transiciones que consultan:

   - el estado q del control finito;
   - el siguiente símbolo terminal a de la cadena de entrada.

   Como resultado, el autómata puede cambiar al estado q', avanzar una posición en la cadena de entrada y crear una nueva pila $[Z'$ y situarla en la cima de la pila principal. En la figura 7.2 se muestra el resultado de aplicar una transición

   $$(q', [Z) \in \delta(q, a, \epsilon, \epsilon, \epsilon)$$

   a un autómata a pila embebido ascendente con la configuración de la figura 7.1.

   El dual de este tipo de transición en EPDA con estados lo constituiría el borrado de las pilas vacías, con la salvedad de que en los EPDA con estados no es necesario especificar transiciones para realizar dicha acción, pues se lleva a cabo implícitamente siempre que una pila queda vacía.

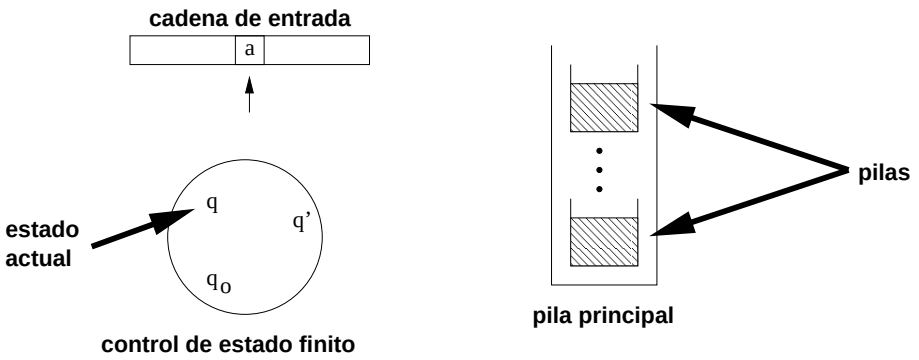

Figura 7.1: Una configuración de un BEPDA

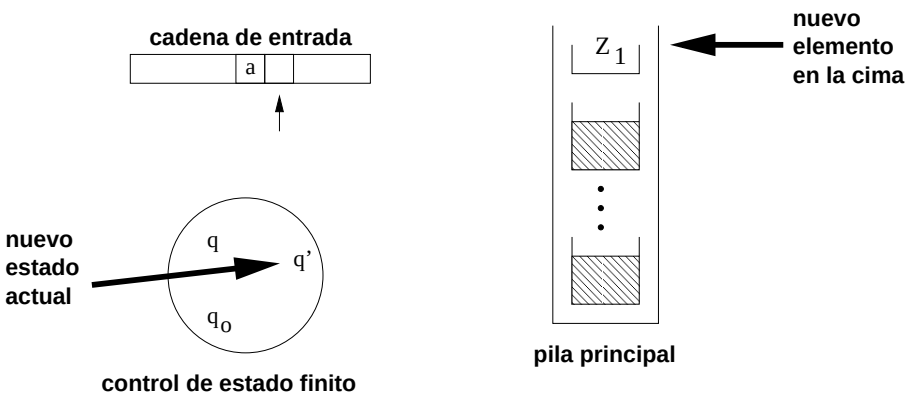

Figura 7.2: Configuración de un BEPDA tras aplicar una transición del primer tipo

2. Transiciones que consultan:

   - el estado q del control finito;
   - el siguiente símbolo terminal a de la cadena de entrada;

- el contenido de i pilas $[\alpha_j$, con $1 \leq j \leq i$, de tamaño acotado situadas en la cima de la pila principal;
- m elementos situados en la cima de la pila $[\alpha Z_m \ldots Z_1$ situada bajo las i anteriores;
- el contenido de las $k-i$ pilas $[\alpha_j$, con $i+1 \leq j \leq k$, de tamaño acotado situadas bajo la pila anterior.

Como resultado, el autómata puede:

- cambiar al estado q';
- avanzar una posición en la cadena de entrada;
- eliminar las i pilas de tamaño acotado de la cima;
- reemplazar los m elementos de la siguiente pila por el elemento X;
- eliminar las $k-i$ pilas de tamaño acotado situadas bajo dicha pila.

La figura 7.4 muestra el resultado de aplicar una transición

$$(q', Z) \in \delta(q,\ a,\ [\alpha_k \ldots [\alpha_{i+1},\ Z_m \ldots Z_1,\ [\alpha_i \ldots [\alpha_1)$$

a un autómata a pila embebido ascendente con la configuración de la figura 7.3.

Este tipo de transiciones se corresponde con el dual de las transiciones de un EPDA.

Definimos formalmente un autómata a pila embebido ascendente como una tupla $(Q, V_T, V_S, \delta, q_0, Q_F, \$_f)$ 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 es el estado inicial.
- $Q_F \subseteq Q$ es el conjunto de estados finales.
- $\$_f \in V_S$ es el símbolo final de la pila.
- δ es una relación de $Q \times V_T \cup \{\epsilon\} \times ([V_S^+)^* \times V_S^* \times ([V_S^+)^*$ en subconjuntos finitos de $Q \times V_S \cup \{[V_S\}$, donde $[\notin V_S$ es un símbolo utilizado para separar las diferentes pilas que componen la pila principal.

La *configuración* de un autómata a pila embebido ascendente en un momento dado viene definida por el triple (q, Υ, w), donde $q \in Q$ indica el estado en el que se encuentra, $\Upsilon \in ([V_S^+)^*$ el contenido de la pila principal y $w \in V_T^*$ la parte de la cadena de entrada que resta por leer. Es importante indicar que todas las pilas almacenadas en la pila principal deben contener al menos un elemento. La pila principal sólo estará vacía en la configuración inicial (q_0, ϵ, w).

El paso de una configuración (q, Υ, aw) a otra configuración (q', Υ', w) viene determinado por la aplicación de una transición y se denota mediante $\vdash$. Tenemos los dos casos siguientes:

1. Dada una configuración (q, Υ, aw) y una transición

   $$(q', [Z) \in \delta(q, a, \epsilon, \epsilon, \epsilon)$$

   tenemos que

   $$(q, \Upsilon, aw) \vdash (q', \Upsilon[Z, w)$$

   puesto que el autómata a pila embebido ascendente pasará a la nueva configuración $(q', \Upsilon[Z, w)$.

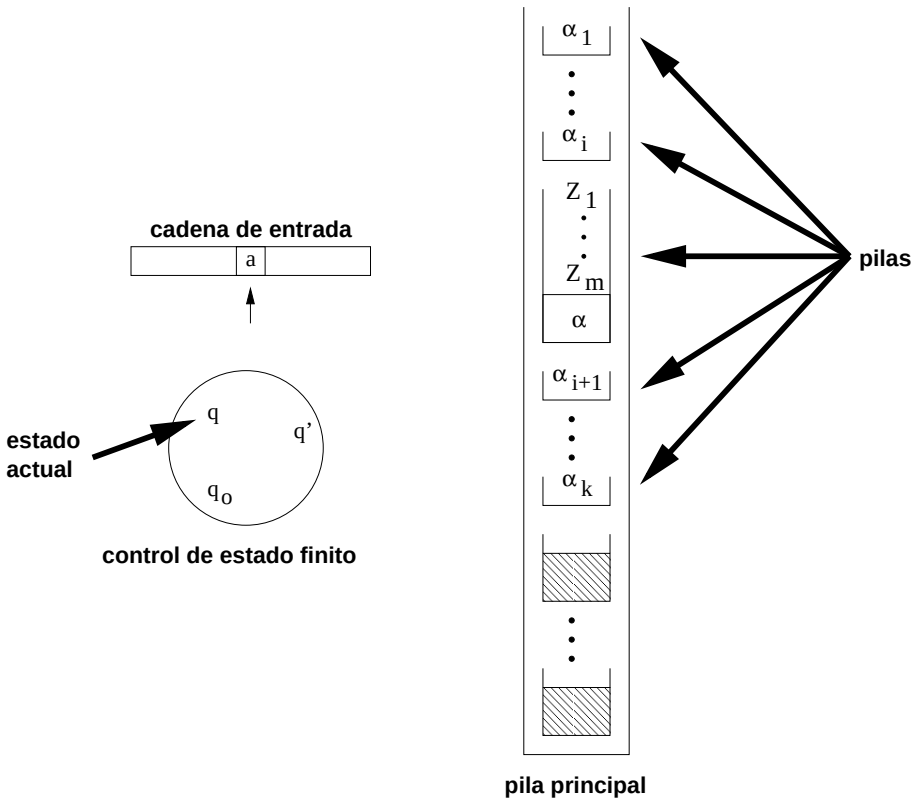

Figura 7.3: Otra configuración de un BEPDA

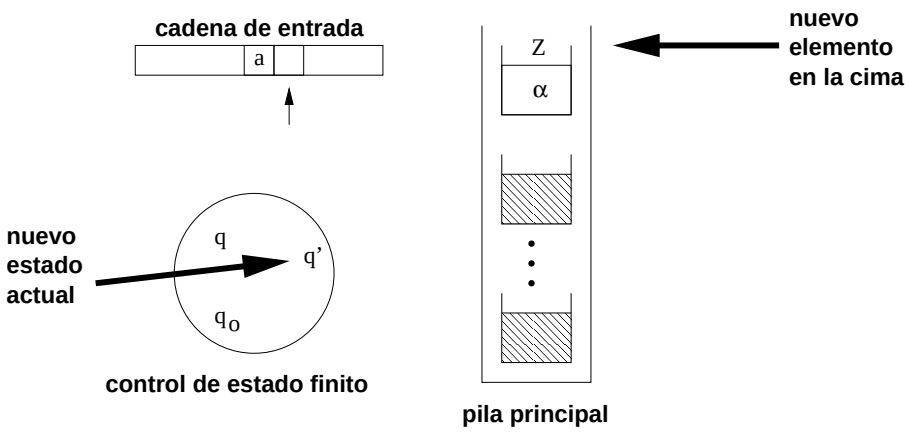

Figura 7.4: Configuración de un BEPDA tras aplicar una transición del segundo tipo

2. Dada una configuración $(q, \Upsilon[\alpha_k \ldots [\alpha_{i+1}[\alpha Z_m \ldots Z_1[\alpha_i \ldots [\alpha_1, aw)$ y una transición

$$(q', Z) \in \delta(q,\ a,\ [\alpha_k \ldots [\alpha_{i+1},\ Z_m \ldots Z_1,\ [\alpha_i \ldots [\alpha_1)$$

tenemos que

$$(q, \Upsilon[\alpha_k \ldots [\alpha_{i+1}[\alpha Z_m \ldots Z_1[\alpha_i \ldots [\alpha_1, aw) \vdash (q', \Upsilon[\alpha Z, w)$$

puesto que el autómata a pila embebido ascendente pasará a la nueva configuración $(q', \Upsilon[\alpha Z, w)$.

Denotamos por $\overset{*}{\vdash}$ el cierre reflexivo y transitivo de $\vdash$.

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

El *lenguaje aceptado por pila vacía* por un autómata a pila embebido viene determinado por el conjunto de cadenas $w \in V_T^*$ tal que $(q_0, \epsilon, w) \overset{*}{\vdash} (p, [\$_f, \epsilon)$ para cualquier $q \in Q$. Observamos una vez más la dualidad existente entre EPDA y BEPDA, pues mientras el primero parte de una pila $[\$_0$ para finalizar con una pila vacía, el segundo parte de una pila vacía para finalizar con una pila $[\$_f$.

Teorema 7.1 *Dado un autómata a pila embebido ascendente $\mathcal{A}$ que acepta el lenguaje L por estado final, existe un autómata a pila embebido ascendente $\mathcal{A}'$ que acepta L por pila vacía.*

Demostración:

Sea $\mathcal{A} = (Q, V_T, V_S, \delta, q_0, Q_F, \$_f)$ el BEPDA que acepta el lenguaje L por estado final. El BEPDA $\mathcal{A}' = (Q', V_T, V_S, \delta', q_0, Q_F, \$_f)$ acepta L por pila vacía, con $Q' = Q \cup \{q'\}$ tal que $q' \notin Q$, y con δ' conteniendo las transiciones en δ junto con las siguientes transiciones encargadas de vaciar al pila cuando se alcanza un estado final:

$$(q', \$_f) \in \delta'(q_f, \epsilon, \epsilon, Z, \epsilon)$$

$$(q', \$_f) \in \delta'(q', \epsilon, \epsilon, Z\,\$_f, \epsilon)$$

$$(q', \$_f) \in \delta'(q', \epsilon, \epsilon, Z, [\$_f)$$

para todo $q_f \in Q_F$ y $Z \in V_S$. □

Teorema 7.2 *Dado un autómata a pila embebido ascendente $\mathcal{A}$ que acepta el lenguaje L por pila vacía, existe un autómata a pila embebido ascendente $\mathcal{A}'$ que acepta L por estado final.*

Demostración:

Sea $\mathcal{A} = (Q, V_T, V_S, \delta, q_0, Q_F, \$_f)$ el BEPDA que acepta el lenguaje L por pila vacía. El BEPDA $\mathcal{A}' = (Q, V_T, V_S', \delta, q_0, Q_F, \$_f)$ acepta L por estado final, con $V_S' = V_S \cup \{\$_0\}$ tal que $\$_0 \notin V_S$ y con δ' conteniendo las transiciones de δ junto con las siguientes transiciones encargadas de situar una pila $[\$_0$ al comienzo de al operación de $\mathcal{A}'$, y de detectar la pila $[\$_0[\$_f$, que correspondería a la pila $[\$_f$ de la configuración final de $\mathcal{A}$, para poder transitar a un estado final.

$$(q_0, [\$_0) \in \delta'(q_0, \epsilon, \epsilon, \epsilon, \epsilon)$$

$$(q_f, \$_f) \in \delta'(q, \epsilon, \epsilon, \$_0, [\$_f)$$

para todo $q \in Q$. □

Teorema 7.3 *Dado un BEPDA $\mathcal{A}$, existe un BEPDA $\mathcal{A}'$ tal que el lenguaje aceptado por $\mathcal{A}$ coincide con el lenguaje aceptado por $\mathcal{A}'$ y las transiciones en $\mathcal{A}'$ están normalizadas, esto es, tienen alguna de las dos formas siguientes:*

$$(q', [Z) \in \delta(q, a, \epsilon, \epsilon, \epsilon)$$

$$(q', Z) \in \delta(q,\ a,\ [Z'_k \dots [Z'_{i+1},\ Z_m \dots Z_1,\ [Z'_i \dots [Z'_1)$$

donde $q, q' \in Q$, $a \in V_T \cup \{\epsilon\}$, $Z, Z_1, \dots, Z_m \in V_S$, $Z'_1, \dots, Z'_k \in V_S$, $0 \leq i \leq k$ y $0 \leq m \leq 2$.

<u>Demostración</u>:

Las transiciones de $\mathcal{A}'$ se construyen a partir de las transiciones de $\mathcal{A}$. Las transiciones

$$(q', [Z) \in \delta(q, a, \epsilon, \epsilon, \epsilon)$$

permanecen sin cambios. Con respecto a las transiciones

$$(q', Z) \in \delta(q, a, [Z'_{k,l_k} \dots Z'_{k,1} [\dots [Z'_{i+1,l_{i+1}} \dots Z'_{i+1,1}, Z_p \dots Z_1, [Z'_{i,l_i} \dots Z'_{i,1} [\dots [Z'_{1,l_1} \dots Z'_{1,1})$$

precisaremos introducir un nuevo estado q'' que permita distinguir las configuraciones derivadas del proceso de normalización del resto de las transiciones. A continuación se muestra el nuevo conjunto de transiciones normalizadas que sustituyen a las transiciones de este último tipo:

- Una transición de inicio:

$$(q'', Z'_{1,1}) \in \delta(q, \epsilon, \epsilon, Z'_{1,1}, \epsilon) \qquad \text{si } [Z'_{i,l_i} \dots Z'_{i,1} [\dots [Z'_{1,l_1} \dots Z'_{1,1} \neq \epsilon$$

- Un conjunto de transiciones para el vaciado de cada una de las pilas en $[Z'_{i,l_i} \dots Z'_{i,1} [\dots [Z'_{1,l_1} \dots Z'_{1,1}$:

$$(q'', Z'_{r,s+1}) \in \delta(q'', \epsilon, \epsilon, Z'_{r,s+1}\, Z'_{r,s}, \epsilon) \qquad 1 \leq r \leq i,\ 1 \leq s < l_r$$
$$(q'', Z'_{r+1,1}) \in \delta(q'', \epsilon, \epsilon, Z'_{r+1,1}, [Z'_{r,l_r}, \epsilon) \quad 1 \leq r < i$$

  En el caso de $i = 0$, esto es $[Z'_{i,l_i} \dots Z'_{i,1} [\dots [Z'_{1,l_1} \dots Z'_{1,1} = \epsilon$, en lugar de las transiciones anteriores tendremos una sola transición

$$(q'', [Z'_{0,0}) \in \delta(q, \epsilon, \epsilon, \epsilon)$$

- Un conjunto de transiciones para pasar a la pila que tiene $Z_p \dots Z_1$ en su cima. En el caso de $p = 0$ estas transiciones también apilan Z:

$$(q'', Z_1) \in \delta(q'', \epsilon, \epsilon, Z_1, [Z'_{i,l_i}, \epsilon) \qquad \text{si } p \geq 1$$
$$(q'', Z) \in \delta(q'', \epsilon, \epsilon, Z', [Z'_{i,l_i}, \epsilon) \qquad \forall Z' \in V_S \text{ si } p = 0$$

- Un conjunto de transiciones para la sustitución de $Z_p \dots Z_1$ por Z cuando $p \geq 1$:

$$(q'', Z_{s+1}) \in \delta(q'', \epsilon, \epsilon, Z_{s+1}\, Z_s, \epsilon) \qquad 1 \leq s < p$$
$$(q'', Z) \in \delta(q'', \epsilon, \epsilon, Z_p, \epsilon)$$

- Un conjunto de transiciones que vacían las pilas en $[Z'_{i,l_i} \dots Z'_{i,1}[\dots [Z'_{1,l_1} \dots Z'_{1,1}$, dejándolas únicamente con el último elemento:

$$(q'', Z'_{r,1}) \in \delta(q''', \epsilon, \epsilon, Z'_{r,1}, \epsilon) \qquad \forall q''' \in Q$$
$$(q'', Z'_{r,s+1}) \in \delta(q'', \epsilon, \epsilon, Z'_{r,s+1}\, Z'_{r,s}, \epsilon) \quad i+1 \leq r \leq k,\ 1 \leq s < l_r$$

- Un conjunto de transiciones para ir eliminando las pilas $[Z'_{i,l_i}[\dots [Z'_{1,l_1}$ resultantes de la aplicación del anterior conjunto de transiciones:

$$(q'', Z) \in \delta(q'', \epsilon, Z'_{r,l_r}, Z, \epsilon) \quad i+1 \leq r \leq k$$

□

Ejemplo 7.1 El autómata embebido a pila ascendente definido por la tupla $(\{q_0, q_1, q_2, q_3\}, \{a, b, c, d\}, \{B, C, D\}, \delta, q_0, \emptyset, \$_f)$, donde δ contiene las transiciones mostradas en la parte izquierda de la tabla 7.1, acepta el lenguaje $\{a^n b^n c^n d^n \mid n \geq 0\}$ por pila vacía. En la parte derecha de la tabla 7.1 se muestra la secuencia de configuraciones que sigue el autómata para analizar correctamente la cadena de entrada *aabbccdd*. La primera columna muestra la transición aplicada, la segunda el estado, la tercera el contenido de la pila y la cuarta la parte que resta por leer de la cadena de entrada. Obsérvese como la secuencia de pilas coincide, en orden inverso, con la secuencia de pilas de las configuraciones del autómata a pila embebido de la tabla 6.1. ¶

7.3 Autómatas a pila embebidos ascendentes sin estados

Al igual que en el caso de los autómatas a pila y de los autómatas a pila embebidos, el control finito es un elemento prescindible de los autómatas a pila embebidos ascendentes puesto que el estado correspondiente a una configuración puede ser incluido en el elemento de la cima de la pila. Como resultado obtenemos una definición alternativa, que juzgamos más simple y homogénea, según la cual un autómata a pila embebido 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 pila.
- $\$_f \in V_S$ es el símbolo final de pila.
- Θ es un conjunto de transiciones, cada una de las cuales pertenece a uno de los siguientes tipos, donde $C, F, G \in V_S$, $\Upsilon \in ([V_S^*)^*$, $\alpha \in V_S^*$ y $a \in V_T \cup \{\epsilon\}$:

  SWAP: Transiciones de la forma $C \stackrel{a}{\longmapsto} F$ 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 $\Upsilon[\alpha C$ es una pila $\Upsilon[\alpha F$.

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

(a)	$(q_0, [D) \in \delta(q_0, a, \epsilon, \epsilon, \epsilon)$
(b)	$(q_1, [C) \in \delta(q_0, b, \epsilon, \epsilon, \epsilon)$
(c)	$(q_1, [C) \in \delta(q_1, b, \epsilon, \epsilon, \epsilon)$
(d)	$(q_2, B) \in \delta(q_1, c, \epsilon, C, \epsilon)$
(e)	$(q_2, B) \in \delta(q_2, c, [C, \epsilon, \epsilon)$
(f)	$(q_3, B) \in \delta(q_2, d, [D, BB, \epsilon)$
(g)	$(q_3, B) \in \delta(q_3, d, [D, BB, \epsilon)$
(h)	$(q_3, \$_f) \in \delta(q_3, d, [D, B, \epsilon)$
(i)	$(q_0, [\$_f) \in \delta(q_0, a, \epsilon, \epsilon, \epsilon)$

	q_0		$aabbccdd$
(a)	q_0	$[D$	$abbccdd$
(a)	q_0	$[D[D$	$bbccdd$
(b)	q_1	$[D[D[C$	$bccdd$
(c)	q_1	$[D[D[C[C$	$ccdd$
(d)	q_2	$[D[D[C[B$	cdd
(e)	q_2	$[D[D[BB$	dd
(f)	q_3	$[D[B$	d
(h)	q_3	$[\$_f$	

Tabla 7.1: Transiciones del BEPDA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ (izquierda) y configuraciones de dicho autómata durante el análisis de *aabbccdd* (derecha)

POP: Transiciones de la forma $CF \stackrel{a}{\longmapsto} G$ 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 $\Upsilon[\alpha CF$ es una pila $\Upsilon[\alpha G$.

UNWRAP-A: Transiciones de la forma $C, [F \stackrel{a}{\longmapsto} G$ que eliminan la pila $[F$ situada en la cima de la pila principal, mientras la cima C de la pila situada inmediatamente debajo es reemplazada por G. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C[F$ es una pila $\Upsilon[\alpha G$.

UNWRAP-B: Transiciones de la forma $[C, F \stackrel{a}{\longmapsto} G$ que reemplazan el elemento F situado en la cima de la pila ubicada en la cima de la pila principal por G, mientras que eliminan la pila $[C$ situada bajo dicha pila. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[C[\alpha F$ es una pila $\Upsilon[\alpha G$.

WRAP: Transiciones de la forma $C \stackrel{a}{\longmapsto} C, [F$ que sitúan una nueva pila $[F$ en la cima de la pila principal. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C$ es una pila $\Upsilon[\alpha C[F$.

Los tres primeros tipos coinciden con las transiciones del mismo nombre presentes en los autómatas a pila embebidos. Las transiciones de tipo UNWRAP-A y UNWRAP-B permiten eliminar una pila situado encima y debajo, respectivamente, de la pila que ocupará la cima de la pila principal una vez aplicada la transición, como se puede observar en la figura 7.5. Las transiciones de tipo WRAP permiten crear una nueva pila en la cima de la pila principal, como muestra la figura 7.6.

Comparando la figura 7.5 con la 6.4 se observa que UNWRAP-A y UNWRAP-B son las transiciones duales de WRAP-A y WRAP-B, respectivamente. Comparando la figura 7.6

con la 6.5 se observa que WRAP es la transición dual de UNWRAP.

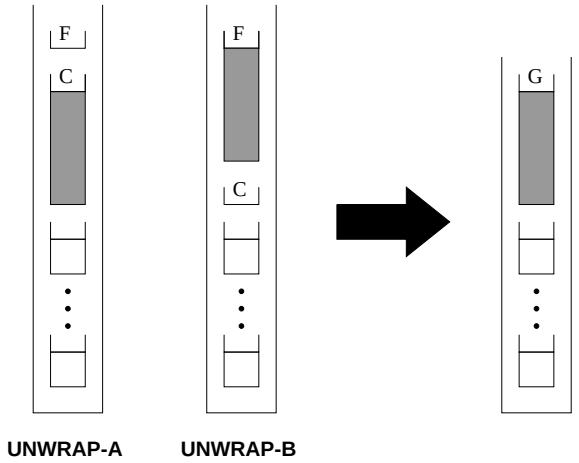

Figura 7.5: Transiciones UNWRAP-A y UNWRAP-B

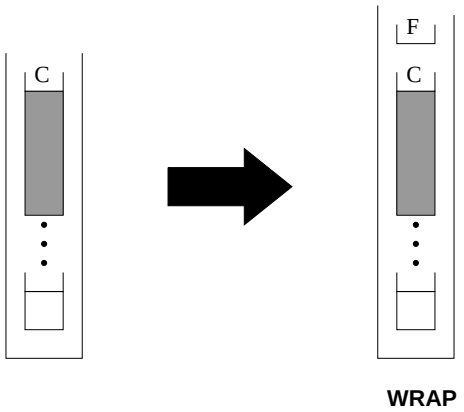

Figura 7.6: Transición WRAP

La *configuración* en un momento dado de un autómata a pila embebido ascendente sin estados viene determinada por el par (Υ, w), donde Υ es el contenido de la pila y w la parte de la cadena de entrada que resta por leer. Una configuración (Υ, aw) deriva una configuración (Υ', w), hecho que denotamos mediante $(\Upsilon, aw) \vdash (\Upsilon', w)$, si y sólo si existe una transición que aplicada a Υ devuelve Υ' y consume a de la cadena de entrada. En caso de ser necesario identificar una derivación d concreta, utilizaremos la notación $\overset{*}{\vdash}_d$. Denotamos por $\overset{*}{\vdash}$ el cierre reflexivo y transitivo de $\vdash$.

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

En lo que resta, cuando nos refiramos a BEPDA sin especificar si se trata de la versión con estados o sin estados, debe entenderse que nos estamos refiriendo a la versión sin estados que acabamos de definir.

(a)	$\$_0 \overset{a}{\longmapsto} \$_0, [D$
(b)	$D \overset{a}{\longmapsto} D, [D$
(c)	$D \longmapsto D, [C$
(d)	$C \overset{b}{\longmapsto} C, [C$
(e)	$C \longmapsto B$
(f)	$B \longmapsto BE$
(g)	$[C, E \overset{c}{\longmapsto} C$
(h)	$BC \longmapsto B$
(i)	$[D, B \overset{d}{\longmapsto} D$
(j)	$BD \longmapsto B$
(k)	$D \longmapsto \$_f$

	$[\$_0$	$aabbccdd$
(a)	$[\$_0[D$	$aabbccdd$
(b)	$[\$_0[D[D$	$abbccdd$
(c)	$[\$_0[D[D[C$	$bbccdd$
(d)	$[\$_0[D[D[C[C$	$bccdd$
(d)	$[\$_0[D[D[C[C[C$	$ccdd$
(e)	$[\$_0[D[D[C[C[B$	$ccdd$
(f)	$[\$_0[D[D[C[C[BE$	$ccdd$
(g)	$[\$_0[D[D[C[BC$	cdd
(e)	$[\$_0[D[D[C[BB$	cdd
(f)	$[\$_0[D[D[C[BBE$	cdd
(g)	$[\$_0[D[D[BBC$	dd
(h)	$[\$_0[D[D[BB$	dd
(i)	$[\$_0[D[BD$	d
(j)	$[\$_0[D[B$	d
(i)	$[\$_0[D$	
(k)	$[\$_0[\$_f$	

Tabla 7.2: Transiciones del BEPDA sin estados que acepta $\{a^n b^n c^n d^n\}$ (izquierda) y configuraciones de dicho autómata durante el análisis de *aabbccdd* (derecha)

Ejemplo 7.2 El autómata a pila embebido ascendente sin estados definido por la tupla $(\{a, b, c, d\}, \{B, C, D, E, F, \$_0, \$_f\}, \Theta, \$_0, \$_f)$, donde Θ contiene las transiciones mostradas en la tabla 7.2, acepta el lenguaje $\{a^n b^n c^n d^n \mid n \geq 0\}$. En la misma tabla se muestra la secuencia de configuraciones que sigue el autómata para analizar correctamente la cadena de entrada *aabbccdd*. La primera columna muestra la transición aplicada, la segunda el contenido de la pila y la tercera la parte que resta por leer de la cadena de entrada. Obsérvese como la secuencia de pilas coincide, en orden inverso, con la secuencia de pilas de las configuraciones del autómata a pila embebido sin estados de la tabla 6.2. ¶

Teorema 7.4 *Mediante la utilización de un conjunto de transiciones SWAP, PUSH, POP, UNWRAP-A, UNWRAP-B y WRAP es posible emular transiciones complejas del tipo*

$$[F_k \ldots [F_{i+1}, DC_1 \ldots C_m, [F_i \ldots [F_1 \overset{a}{\longmapsto} \ DB$$

tal que su aplicación da lugar a un paso de derivación

$$\Upsilon[F_k \ldots [F_{i+1}[\alpha DC_1 \ldots C_m[F_i \ldots [F_1 \ \vdash \ \Upsilon[\alpha DB$$

donde

- $a \in V_T \cup \{\epsilon\}$.

- $0 \leq m \leq 2$.
- $B, C_1, \ldots, C_2, F_1, \ldots, F_k, \in V_S$.
- *Si* $m = 0$ *entonces* $D \in V_S$.
- *Si* $m > 0$ *entonces* $D = \epsilon$.

Demostración:

Realizaremos una prueba constructiva creando un procedimiento que permita emular el tipo de transiciones propuesto en el teorema 7.4 a partir de las transiciones de los autómatas a pila embebidos ascendentes sin estados. Para ello precisamos nuevos símbolos de pila ∇_j y X', donde $i+1 \leq j \leq k$ y $X \in V_S$. La emulación se realizará en tres fases: la primera se encargará de eliminar las pilas $[F_i \ldots [F_1$, la segunda de obtener la pila $[\alpha DB$ y la tercera se encargará de eliminar las pilas $[F_k \ldots [F_{i+1}$.

Fase 1 Partimos de una configuración

$$(\Upsilon [F_k \ldots [F_{i+1} [\alpha DC_1 \ldots C_m, [F_i \ldots [F_1, aw)$$

donde $D \in V_S$ si $m = 0$ y $D = \epsilon$ en otro caso. Comenzamos la emulación mediante la creación de una transición que sustituye F_1 por ∇_1

$$F_1 \longmapsto \nabla_1$$

Para eliminar las $i-1$ pilas de la cima crearemos las siguientes $i-1$ transiciones

$$[F_{j+1}, \nabla_j \longmapsto \nabla_{j+1}$$

donde $1 \leq j \leq i-1$.
En el caso de que $i = 0$, esto es $[F_i \ldots [F_1 = \epsilon$, esta fase constaría únicamente de la transición

$$X \longmapsto X, [\nabla_0$$

donde $X = D$ si $m = 0$, $X = C_1$ si $m = 1$ y $X = C_2$ si $m = 2$.
Como resultado de la aplicación de esta fase obtendremos una configuración

$$(\Upsilon [F_k \ldots [F_{i+1} [\alpha DC_1 \ldots C_m [\nabla_i, aw)$$

Fase 2 Las transiciones de esta fase dependen del valor de m, por lo que tenemos tres posibilidades:

- Si $m = 2$ se trata de reemplazar C_1C_2 por B:

$$C_2 [\nabla_i \longmapsto \nabla'_i$$

$$C_1 \nabla'_i \longmapsto B'$$

- Si $m = 1$ nos encontramos ante el cambio de C_1 por B:

$$C_1 [\nabla_i \longmapsto \nabla'_i$$

$$\nabla'_i \longmapsto B'$$

- Si $m = 0$ se trata de apilar B:

$$D [\nabla_i \longmapsto D'$$

$$D' \longmapsto D'B'$$

Después de esta fase obtenemos una configuración

$$(\Upsilon[F_k \ldots [F_{i+1}[\alpha D'B'aw)$$

donde $D' = \epsilon$ si $m > 0$.

Fase 3 Para iniciar esta fase precisamos de una transición

$$[F_{i+1,}B' \longmapsto \nabla_{i+1}$$

Para crear las las restantes $k-i-1$ pilas unitarias utilizaremos $k-i-1$ transiciones de la forma

$$[F_{j+1}, \nabla_j \longmapsto \nabla_{j+1}$$

donde $i+1 \leq j < k$, para finalizar con la transición

$$\nabla_k \overset{a}{\longmapsto} B$$

En el caso de que $k = i$, esto es, $[F_k \ldots [F_{i+1} = \epsilon$, esta fase constaría únicamente de la transición

$$B' \longmapsto B$$

Tras esta fase obtendremos la configuración

$$(\Upsilon[\alpha D'B, w)$$

donde $D' = \epsilon$ si $m > 0$. En el caso de que $m = 0$, para emular completamente la transición deseada deberemos crear una transición

$$D'X \longmapsto Y$$

por cada transición $DX \longmapsto Y$ presente en el BEPDA.

□

Ejemplo 7.3 La transición $[E[F, BC, [G[H \overset{a}{\longmapsto} B$ se emula mediante el conjunto de transiciones mostrado en la tabla 7.3. La misma tabla muestra el resultado de aplicar las transiciones de la tabla 7.3 a una configuración $(\Upsilon[\alpha B, aw)$. La primera columna muestra la transición aplicada, la segunda muestra el contenido de la pila del BEPDA y la tercera la parte de la cadena de entrada que falta por leer. ¶

7.4 Equivalencia entre autómatas a pila embebidos sin estados y con estados

Para establecer la equivalencia entre las versiones con estados y sin estados de los autómatas a pila embebidos ascendentes, haremos uso del teorema 7.3, que posibilita la utilización de transiciones complejas en los BEPDA con estados, y del teorema 7.4, que establece una forma normal para las transiciones de los BEPDA sin estados.

(a)	$H \longmapsto \nabla_1$
(b)	$[G, \nabla_1 \longmapsto \nabla_2$
(c)	$C, [\nabla_2 \longmapsto \nabla_2'$
(d)	$B\nabla_2' \longmapsto B'$
(e)	$[F, B' \longmapsto \nabla_3$
(f)	$[E, \nabla_3 \longmapsto \nabla_4$
(g)	$\nabla_4 \overset{a}{\longmapsto} B$

	$\Upsilon[E[F[\alpha BC[G[H$	aw
(a)	$\Upsilon[E[F[\alpha BC[G[\nabla_1$	aw
(b)	$\Upsilon[E[F[\alpha BC[\nabla_2$	aw
(c)	$\Upsilon[E[F[\alpha B\nabla_2'$	aw
(d)	$\Upsilon[E[F[\alpha B'$	aw
(e)	$\Upsilon[E[\alpha\nabla_3$	aw
(f)	$\Upsilon\alpha\nabla_4$	aw
(g)	$\Upsilon\alpha B$	w

Tabla 7.3: Normalización de una transición compleja en un BEPDA (izquierda) y emulación de la misma (derecha)

Teorema 7.5 *Para todo BEPDA sin estados $\mathcal{A}$, existe un BEPDA con estados $\mathcal{A}'$ tal que el lenguaje aceptado por $\mathcal{A}$ es igual al lenguaje aceptado por $\mathcal{A}'$.*

Demostración:

Sea $\mathcal{A} = (V_T, V_S, \Theta, \$_0, \$_f)$ un BEPDA sin estados. El BEPDA con estados $\mathcal{A}' = (Q, V_T, V_S, \delta, q_0, Q_F, \$_f)$ acepta el mismo lenguaje (por pila vacía) que $\mathcal{A}$ si las transiciones en δ se obtienen mediante una traducción adecuada de las transiciones en Θ. Consideremos los posibles casos:

SWAP: Una transición $C \overset{a}{\longmapsto} F$ se traduce por una transición $(q, F) \in \delta(q, a, \epsilon, C, \epsilon)$

PUSH: Una transición $C \overset{a}{\longmapsto} CF$ se traduce por una transición $(q', C) \in \delta(q, a, \epsilon, C, \epsilon)$ más una transición $(q, F) \in \delta(q', \epsilon, \epsilon, \epsilon, \epsilon)$, donde q' es un estado que sólo se utiliza en estas dos transiciones.

POP: Una transición $CF \overset{a}{\longmapsto} G$ se traduce por una transición $(q, G) \in \delta(q, a, \epsilon, CF, \epsilon)$

UNWRAP-A: Una transición $C, [F \overset{a}{\longmapsto} G$ se traduce por una transición $(q, G) \in \delta(q, a, \epsilon, C, [F)$.

UNWRAP-B: Una transición $[C, F \overset{a}{\longmapsto} G$ se traduce por una transición $(q, G) \in \delta(q, a, [C, F, \epsilon)$.

WRAP: Una transición $C \overset{a}{\longmapsto} C, [F$ se traduce por una transición $(q'', C) \in \delta(q, a, \epsilon, C, \epsilon)$ más una transición $(q, [F) \in \delta(q'', a, \epsilon, \epsilon, \epsilon)$, donde q'' es un estado que sólo se utiliza en estas dos transiciones.

Adicionalmente, deberemos considerar la transición

$$(q', \$_f) \in \delta(q, \epsilon, \epsilon, \$_0, [\$_f)$$

que obtiene una pila $[\$_f$ cuando se alcanza la configuración $(q, [\$_0[\$_f, \epsilon)$, equivalente a la configuración final del BEPDA sin estados.

El conjunto Q estará formado por el estado q y todos los estados q' y q'' utilizados en la traducción de transiciones POP y WRAP. $\square$

Teorema 7.6 *Para todo BEPDA con estados $\mathcal{A}$, existe un BEPDA sin estados $\mathcal{A}'$ tal que el lenguaje aceptado por $\mathcal{A}$ es igual al lenguaje aceptado por $\mathcal{A}'$.*

Demostración:

Dado un un BEPDA con estados $\mathcal{A} = (Q, V_T, V_S, \delta, q_0, Q_F, \$_f)$ construiremos un BEPDA sin estados $\mathcal{A}' = (V_T, V'_S, \Theta, \$'_0, \$'_f)$ que acepte el mismo lenguaje es aceptado por $\mathcal{A}$ por pila vacía. El conjunto V'_S estará formado por pares $\langle Z, q\rangle$, donde $Z \in V_S \cup \{-\}$ y $q \in Q \cup \{-\}$, y por los elementos inicial $\$'_0 = \langle -, q_0\rangle$ y final $\$'_f = \langle \$_f, -\rangle$.

Las transiciones en Θ tendrán el formato de las transiciones descritas en el teorema 7.4 y serán el resultado de traducir las transiciones en δ. Supondremos que las transiciones de $\mathcal{A}$ están en la forma normal definida en el teorema 7.3. Consideremos cada uno de los posibles casos:

- Una transición
$$(q', [Z) \in \delta(q, a, \epsilon, \epsilon, \epsilon)$$
se traduce por una transición
$$\langle B, q\rangle \longmapsto \langle B, q\rangle, [\langle Z, q'\rangle$$
para todo $B \in V_S \cup \{-\}$.
- Una transición
$$(q', Z) \in \delta(q, a, [Z'_k \ldots [Z'_{i+1}, Z_1, [Z'_i \ldots [Z'_1)$$
se traduce por transiciones
$$[\langle F_k, q_k\rangle \ldots [\langle F_{i+1}, q_{i+1}\rangle, \langle Z_1, q''\rangle, [\langle F_i, q_i\rangle \ldots [\langle F_1, q_1\rangle \stackrel{a}{\longmapsto} \langle Z, q'\rangle$$
para todo $q_1, \ldots, q_k, q'' \in Q$.
- Una transición
$$(q', Z) \in \delta(q, a, [Z'_k \ldots [Z'_{i+1}, \epsilon, [Z'_i \ldots [Z'_1)$$
se traduce por transiciones
$$[\langle F_k, q_k\rangle \ldots [\langle F_{i+1}, q_{i+1}\rangle, \langle D, q''\rangle, [\langle F_i, q_i\rangle \ldots [\langle F_1, q_1\rangle \stackrel{a}{\longmapsto} \langle D, q''\rangle\langle Z, q'\rangle$$
para todo $D \in V_S$ y $q_1, \ldots, q_k, q'' \in Q$. Es importante resaltar que esta traducción es correcta porque en toda configuración válida de un BEPDA con estados cada pila debe contener al menos un elemento, lo cual garantiza la existencia de D.
- Una transición
$$(q', Z) \in \delta(q, a, [Z'_k \ldots [Z'_{i+1}, Z_2 Z_1, [Z'_i \ldots [Z'_1)$$
se traduce por transiciones
$$[\langle F_k, q_k\rangle \ldots [\langle F_{i+1}, q_{i+1}\rangle, \langle Z_2, q'''\rangle\langle Z_1, q''\rangle, [\langle F_i, q_i\rangle \ldots [\langle F_1, q_1\rangle \stackrel{a}{\longmapsto} \langle Z, q'\rangle$$
para todo $q_1, \ldots, q_k, q'', q''' \in Q$.

La transición inicial
$$\langle -, -\rangle \longmapsto \langle -, -\rangle[\langle -, q_0\rangle$$
configura la pila para que puedan comenzar a ser aplicadas las transiciones definidas anteriormente.

La configuración final de $\mathcal{A}'$ es $(\langle -, -\rangle[\langle \$_f, q\rangle, \epsilon)$, que se corresponde con la configuración final $(q, [\$_f, \epsilon)$ de $\mathcal{A}$. □

Corolario 7.7 *Los BEPDA sin estados son equivalentes a los BEPDA con estados.*

Demostración:

La veracidad del enunciado se deriva directamente de los teoremas 7.5 y 7.6. □

7.5 Esquemas de compilación de gramáticas independientes del contexto

De forma análoga al caso de los EPDA (sección 6.5), tomamos una configuración $(\$_0 B_1 B_2 \ldots B_n, w)$ de un autómata a pila como equivalente a una configuración $([\$_0 [B_1 [B_1 \ldots [B_1, w)$ de un BEPDA, con lo cual las pilas unitarias pasan a representar el papel de los símbolos de pila. Con respecto a las transiciones, tenemos que:

- Las transiciones SWAP $C \longmapsto F$ permanecen sin cambios.
- Las transiciones PUSH $C \longmapsto CF$ del autómata a pila son reemplazadas por transiciones WRAP $C \longmapsto C, [F$ en el autómata a pila embebido ascendente.
- Las transiciones POP $CF \longmapsto G$ del autómata a pila son reemplazadas por transiciones UNWRAP-A $C, [F \longmapsto G$ en el autómata a pila embebido ascendente.

Un autómata a pila embebido ascendente cuyas configuraciones sólo contengan pilas unitarias y que sólo utilice transiciones SWAP, WRAP y UNWRAP-A, será equivalente a un autómata a pila cuyas configuraciones se obtendrán a partir de las configuraciones del EPDA eliminando los símbolos $[$. El conjunto de transiciones contendrá las transiciones SWAP y el resultado de convertir las transiciones WRAP en transiciones PUSH y las transiciones UNWRAP-A en transiciones POP.

Puesto que las transiciones WRAP de los BEPDA coinciden con las transiciones WRAP-A de los EPDA y que las transiciones UNWRAP-A de los BEPDA coinciden con las transiciones UNWRAP de los EPDA, el resultado de traducir un autómata a pila a un EPDA coincide con el resultado de traducir dicho autómata a pila a un BEPDA. En consecuencia, el esquema de compilación genérico 7.1 de gramáticas independientes del contexto en autómatas a pila embebidos ascendentes coincide con el esquema de compilación genérico 6.1 de gramáticas independientes del contexto en autómatas a pila embebidos.

Esquema de compilación 7.1 El esquema de compilación genérico de una gramática independiente del contexto en un autómata a pila embebido ascendente queda definido por el siguiente conjunto de reglas y los elementos inicial $\$_0$ y final $\overleftarrow{S}$.

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

§

7.6 Esquemas de compilación de gramáticas de adjunción de árboles

Los autómatas a pila embebidos ascendentes sólo pueden ser utilizados para definir esquemas de compilación de gramáticas de adjunción de árboles en los que las las derivaciones se reconocen de modo ascendente, esto es, el reconocimiento de una adjunción se inicia en el nodo pie y se termina en el nodo raíz de un árbol auxiliar. En consecuencia, un esquema de compilación de una TAG en un BEPDA debe realizar las tareas siguientes:

- Recorrer los nodos de los árboles elementales.
- En el caso de la adjunción del árbol auxiliar β en un nodo N^γ:
  - Una vez recorrido el subárbol de γ que cuelga del nodo N^γ, suspender el recorrido de γ y comenzar el recorrido de β a partir de su nodo pie.
  - Al alcanzar el nodo raíz de β, suspender el recorrido de este último para continuar el recorrido del superárbol de γ a partir del nodo N^γ.

El recorrido de un árbol elemental es equivalente al recorrido del conjunto de producciones independientes del contexto que lo componen, por lo que podemos utilizar para esta tarea las reglas de compilación **[CALL]**, **[SEL]**, **[PUB]**y **[RET]** definidas en el esquema de compilación 6.1. Al igual que en el capítulo 3, consideraremos una producción adicional $\top^\beta \to \mathbf{R}^\alpha$ para cada árbol inicial α y dos producciones adicionales para cada árbol auxiliar β: $\top^\beta \to \mathbf{R}^\beta$ y $\mathbf{F}^\beta \to \bot^\beta$, donde $\mathbf{R}^\beta$ y $\mathbf{F}^\beta$ se refieren a los nodos raíz y pie de β, respectivamente.

Para poder tratar las adjunciones deberemos ser capaces de trasmitir, desde el nodo pie a la raíz de un árbol auxiliar, el nodo sobre el cual se ha adjuntado dicho árbol auxiliar. En el caso de que se realicen otras adjunciones en nodos de la espina, deberemos ir apilando los nodos de adjunción en los cuales se aplican. Utilizaremos las propias pilas del BEPDA para almacenar las pilas de adjunciones pendientes. Para ello, dotaremos de la siguiente semántica a las pilas del BEPDA: dada una pila $[\alpha C$, el elemento C de la cima nos informa del punto en el que se encuentra el recorrido de un árbol elemental mientras que la parte restante α contiene la pila de adjunciones pendientes en dicho punto. La figura 6.6 muestra de modo intuitivo las diferentes reglas de compilación, que en el caso de un BEPDA realizan las funciones siguientes:

- La regla de compilación **[SRET]** será la encargada de crear las transiciones que propaguen la pila de adjunciones pendientes a través de la espina de los árboles auxiliares, desde el pie hasta la raíz.
- La regla **[SCALL]** es una regla **[CALL]** que trata con elementos de la espina de un árbol auxiliar. Se define con el fin de formar un par **[SCALL]**–**[SRET]** análogo al par **[CALL]**–**[RET]**.
- Al llegar al nodo de adjunción N^γ, deberemos apilar dicho nodo en la pila de adjunciones pendientes y pasar al pie del árbol auxiliar β. La regla de compilación **[FRET]** se encargará de crear las transiciones adecuadas para tal fin.
- Al llegar al nodo raíz de β, deberemos eliminar el nodo N^γ de la pila de adjunciones pendientes y continuar el recorrido en el superárbol de γ a partir de dicho nodo. De crear las transiciones necesarias para este fin se encargará la regla de compilación **[FRET]**.
- La regla **[FCALL]** es un tipo especial de regla **[CALL]** que permite pasar del pie del árbol auxiliar β a un nodo de un árbol elemental que pueda servir como nodo de adjunción.

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

Tabla 7.4: Reglas del esquema de compilación genérico de TAG en BEPDA

- La regla **[ACALL]** es un tipo especial de regla **[CALL]** que permite pasar de un nodo de un árbol elemental a la raíz de un árbol auxiliar que pueda ser adjuntado en dicho nodo.

A continuación definimos un esquema de compilación genérico, derivado del esquema de compilación 6.1, en el cual se ha parametrizado la información concerniente al recorrido de los árboles elementales.

Esquema de compilación 7.2 El esquema de compilación genérico de una gramática de adjunción de árboles en un autómata a pila embebido ascendente queda definido por el conjunto de reglas mostrado en la figura 6.4 y los elementos inicial $\$_0$ y final $\overleftarrow{S}$. Es interesante señalar que las pilas de adjunciones pendientes no almacenan directamente los nodos $N^{\gamma}_{r,s+1}$ en los que se realizaron las adjunciones sino el elemento $\nabla^{\gamma}_{r,s}$ que indica el punto en el que se lanzó la adjunción, almacenado en las pilas bajo la forma $\Delta^{\gamma}_{r,s}$ para evitar confusiones. Podemos ver un símbolo Δ como un símbolo ∇ en espera de la finalización de una adjunción.

Las reglas de compilación **[ARET]** y **[FRET]** podrían escribirse alternativamente:

$$\textbf{[ARET]} \quad [\nabla^{\gamma}_{r,s},\ \Delta^{\gamma}_{r,s}\ \overleftarrow{\top}^{\beta} \longmapsto \nabla^{\gamma}_{r,s+1} \qquad \beta \in \mathrm{adj}(N^{\gamma}_{r,s+1})$$

$$\textbf{[FRET]} \quad [\nabla^{\beta_1}_{f_1,0},\ \overleftarrow{N^{\gamma}_{r,s+1}} \longmapsto \Delta^{\gamma}_{r,s}\ \nabla^{\beta_2}_{f_2,1} \quad N^{\beta_1}_{f_1,0} = \mathbf{F}^{\beta_1},\ \ N^{\beta_2}_{f_2,0} = \mathbf{F}^{\beta_2},\ \ \beta_1, \beta_2 \in \mathrm{adj}(N^{\gamma}_{r,s+1})$$

pero como las transiciones resultantes no son transiciones elementales, hemos tenido que descomponer dichas reglas de compilación en dos pares de reglas: **[ARET-a]** más **[ARET-b]** y **[FRET-a]** más **[FRET-b]**. §

El esquema de compilación genérico establece que las adjunciones se tienen que reconocer de modo ascendente, puesto que al pasar al pie de un árbol auxiliar se apila el nodo de adjunción en la pila de adjunciones pendientes y al llegar al nodo raíz dicho nodo se saca de la pila de adjunciones pendientes. Es interesante remarcar que la regla de compilación **[RET]** utiliza una transición UNWRAP-A con el significado de "elimina una pila unitaria que indica un nodo del árbol γ que no pertenece a la espina y por tanto debe tener una una pila vacía de adjunciones pendientes", mientras que la regla de compilación **[SRET]** utiliza una transición UNWRAP-B a la que dota del significado "elimina una pila que indica el reconocimiento de un nodo perteneciente a la espina de un árbol auxiliar β pero preserva la lista de adjunciones pendientes". En la figura 7.5 se observa cómo la parte sombreada (en nuestro caso, la pila de adjunciones pendientes) permanece en su posición original en la operación UNWRAP-A mientras que en la operación UNWRAP-B la parte sombreada es pasada a la nueva pila en la cima.

El esquema de compilación genérico no establece ninguna restricción sobre la estrategia utilizada para recorrer de los árboles elementales. A continuación se definen, de acuerdo con la tabla 5.2, los esquemas de compilación correspondientes a tres estrategias particulares aplicadas al recorrido de los árboles elementales: ascendente, Earley y descendente.

7.6.1 Estrategia descendente

Esquema de compilación 7.3 El esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido ascendente que incorpora una estrategia descendente para el recorrido de los árboles elementales queda definido por el conjunto de reglas mostrado en la tabla 7.5 y los elementos inicial $\$_0$ y final $\Box$. §

7.6.2 Estrategia Earley

Esquema de compilación 7.4 El esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido ascendente que incorpora una estrategia descendente para el recorrido de los árboles elementales queda definido por el conjunto de reglas mostrado en la tabla 7.6 y los elementos inicial $\$_0$ y final $\overline{\overline{S}}$. §

7.6.3 Estrategia ascendente

Esquema de compilación 7.5 El esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido ascendente que incorpora una estrategia ascendente

[INIT]	$\$_0 \longmapsto \$_0 \; [\nabla^{\alpha}_{0,0}$	$\alpha \in \boldsymbol{I}$
[CALL]	$\nabla^{\gamma}_{r,s} \longmapsto \nabla^{\gamma}_{r,s}, \; [N^{\gamma}_{r,s+1}$	$N^{\gamma}_{r,s+1} \notin \text{espina}(\gamma), \; \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SCALL]	$\nabla^{\beta}_{r,s} \longmapsto \nabla^{\beta}_{r,s} \; [N^{\beta}_{r,s+1}$	$N^{\beta}_{r,s+1} \in \text{espina}(\beta), \; \textbf{nil} \in \text{adj}(N^{\beta}_{r,s+1})$
[SEL]	$N^{\gamma}_{r,0} \longmapsto \nabla^{\gamma}_{r,0}$	
[PUB]	$\nabla^{\gamma}_{r,n_r} \longmapsto \Box$	
[RET]	$\nabla^{\gamma}_{r,s}, \; [\Box \longmapsto \nabla^{\gamma}_{r,s+1}$	$N^{\gamma}_{r,s+1} \notin \text{espina}(\gamma), \; \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SRET]	$[\nabla^{\beta}_{r,s}, \; \Box \longmapsto \nabla^{\beta}_{r,s+1}$	$N^{\beta}_{r,s+1} \in \text{espina}(\beta), \; \textbf{nil} \in \text{adj}(N^{\beta}_{r,s+1})$
[SCAN]	$N^{\gamma}_{r,0} \overset{a}{\longmapsto} \Box$	$N^{\gamma}_{r,0} \rightarrow a$
[ACALL]	$\nabla^{\gamma}_{r,s} \longmapsto \nabla^{\gamma}_{r,s}, \; [\top^{\beta}$	$\beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[ARET-a]	$[\nabla^{\gamma}_{r,s}, \; \Box \longmapsto \top$	$\text{adj}(N^{\gamma}_{r,s+1}) \neq \{\textbf{nil}\}$
[ARET-b]	$\Delta^{\gamma}_{r,s} \; \top \longmapsto \nabla^{\gamma}_{r,s+1}$	$\text{adj}(N^{\gamma}_{r,s+1}) \neq \{\textbf{nil}\}$
[FCALL]	$\nabla^{\beta}_{f,0} \longmapsto \nabla^{\beta}_{f,0}, \; [N^{\gamma}_{r,s+1}$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta}, \quad \beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FRET-a]	$[\nabla^{\beta}_{f,0}, \; \Box \longmapsto \Delta^{\gamma}_{r,s}$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta}, \quad \beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FRET-b]	$\Delta^{\gamma}_{r,s} \longmapsto \Delta^{\gamma}_{r,s} \; \nabla^{\beta}_{f,1}$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta}, \quad \beta \in \text{adj}(N^{\gamma}_{r,s+1})$

Tabla 7.5: Reglas del esquema de compilación descendente de TAG en BEPDA

[INIT]	$\$_0 \longmapsto \$_0 \; [\nabla^{\alpha}_{0,0}$	$\alpha \in \boldsymbol{I}$
[CALL]	$\nabla^{\gamma}_{r,s} \longmapsto \nabla^{\gamma}_{r,s}, \; [\overline{N^{\gamma}_{r,s+1}}$	$N^{\gamma}_{r,s+1} \notin \text{espina}(\gamma), \; \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SCALL]	$\nabla^{\beta}_{r,s} \longmapsto \nabla^{\beta}_{r,s} \; [\overline{N^{\beta}_{r,s+1}}$	$N^{\beta}_{r,s+1} \in \text{espina}(\beta), \; \textbf{nil} \in \text{adj}(N^{\beta}_{r,s+1})$
[SEL]	$\overline{N_{r,0}}^{\gamma} \longmapsto \nabla^{\gamma}_{r,0}$	
[PUB]	$\nabla^{\gamma}_{r,n_r} \longmapsto \overline{\overline{N^{\gamma}_{r,0}}}$	
[RET]	$\nabla^{\gamma}_{r,s}, \; [\overline{\overline{N^{\gamma}_{r,s+1}}} \longmapsto \nabla^{\gamma}_{r,s+1}$	$N^{\gamma}_{r,s+1} \notin \text{espina}(\gamma), \; \textbf{nil} \in \text{adj}(N_{r,s+1})$
[SRET]	$[\nabla^{\beta}_{r,s}, \; \overline{\overline{N^{\beta}_{r,s+1}}} \longmapsto \nabla^{\beta}_{r,s+1}$	$N^{\beta}_{r,s+1} \in \text{espina}(\beta), \; \textbf{nil} \in \text{adj}(N^{\beta}_{r,s+1})$
[SCAN]	$\overline{N^{\gamma}_{r,0}} \overset{a}{\longmapsto} \overline{\overline{N^{\gamma}_{r,0}}}$	$N^{\gamma}_{r,0} \rightarrow a$
[ACALL]	$\nabla^{\gamma}_{r,s} \longmapsto \nabla^{\gamma}_{r,s}, \; [\overline{\top^{\beta}}$	$\beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[ARET-a]	$[\nabla^{\gamma}_{r,s}, \; \overline{\overline{\top^{\beta}}} \longmapsto \top$	$\text{adj}(N^{\gamma}_{r,s+1}) \neq \{\textbf{nil}\}$
[ARET-b]	$\Delta^{\gamma}_{r,s} \; \top \longmapsto \nabla^{\gamma}_{r,s+1}$	$\text{adj}(N^{\gamma}_{r,s+1}) \neq \{\textbf{nil}\}$
[FCALL]	$\nabla^{\beta}_{f,0} \longmapsto \nabla^{\beta}_{f,0}, \; [\overline{N^{\gamma}_{r,s+1}}$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta}, \quad \beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FRET-a]	$[\nabla^{\beta}_{f,0}, \; \overline{\overline{N^{\gamma}_{r,s+1}}} \longmapsto \Delta^{\gamma}_{r,s}$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta}, \quad \beta \in \text{adj}(N^{\gamma}_{r,s+1})$
[FRET-b]	$\Delta^{\gamma}_{r,s} \longmapsto \Delta^{\gamma}_{r,s} \; \nabla^{\beta}_{f,1}$	$N^{\beta}_{f,0} = \mathbf{F}^{\beta}, \quad \beta \in \text{adj}(N^{\gamma}_{r,s+1})$

Tabla 7.6: Reglas del esquema de compilación Earley de TAG en BEPDA

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

Tabla 7.7: Reglas del esquema de compilación ascendente de TAG en BEPDA

para el recorrido de los árboles elementales queda definido por el conjunto de reglas mostrado en la tabla 7.7 y los elementos inicial $\$_0$ y final S. §

7.7 Esquemas de compilación de gramáticas lineales de índices

Para el análisis de gramáticas lineales de índices mediante autómatas a pila embebidos utilizaremos la misma semántica utilizada en el caso de los EPDA: una pila $[\alpha B$ se corresponde con el símbolo $B[\alpha]$ de una gramática lineal de índices. En consecuencia, para emular una derivación de una gramática lineal de índices en un BEPDA, es preciso ir modificando las pilas del autómata de acuerdo con los cambios en las pilas de índices indicados por la gramática. De ello se encargan las reglas de compilación descritas en la tabla 7.8.

La novedad con respecto al esquema de compilación 7.1 para gramáticas independientes del contexto estriba en la presencia de las reglas de compilación **[SRET]**, que son las encargadas de transmitir las pilas de índices durante la fase ascendente del algoritmo de análisis. Una vez más, sale a relucir la dualidad existente entre los EPDA y los BEPDA, pues mientras en los primeros sólo es posible trasmitir las pilas de índices durante la fase predictiva o descendente, en los segundos las forma de las transiciones sólo permite que sean transmitidas en la fase ascendente.

Con respecto al tratamiento de los no-terminales, la forma de las transiciones de los BEPDA no imponen ninguna restricción sobre su tratamiento. Este hecho nos permite, de forma análoga al caso de los EPDA, definir un esquema de compilación genérico que será posteriormente adaptado para crear los esquemas de compilación correspondientes a las estrategias descendente, Earley y ascendente aplicadas a los no-terminales de la gramática lineal de índices.

Regla	Tarea
[INIT]	Inicia los cálculos a partir de la pila inicial.
[CALL]	Requiere el análisis de un determinado elemento gramatical $B[\,]$, lo cual implica situar en la cima de la pila principal una nueva pila $[B$.
[SEL]	Selecciona una producción.
[PUB]	Determina que una producción de la gramática ha sido completamente analizada.
[RET]	Continúa el proceso de análisis después de que se haya completado una producción que tiene el no-terminal B en su lado izquierdo, lo cual implica eliminar una pila $[B$ de la cima de la pila principal.
[SRET]	Continúa el proceso de análisis después de que se haya reconocido un elemento gramatical que es un hijo dependiente, lo que implica tener que pasarle la pila de índices con los cambios correspondientes.
[SCAN]	Reconoce los terminales que componen la cadena de entrada.

Tabla 7.8: Tipos de reglas de los esquemas de compilación de LIG en BEPDA

Esquema de compilación 7.6 El esquema de compilación genérico de una gramática lineal de índices en un autómata a pila embebido ascendente queda definido por el conjunto de reglas mostrado en la tabla 7.9 y los elementos inicial $\$_0$ y final $\overleftarrow{S}$.
Las reglas de compilación **[SRET-2]** y **[SRET-3]** podrían escribirse alternativamente de la forma:

$$\textbf{[SRET-2]} \quad [\nabla_{r,s},\ \gamma'\ \overrightarrow{A_{r,s+1}} \longmapsto \nabla_{r,s+1} \qquad A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$$

$$\textbf{[SRET-3]} \quad [\nabla_{r,s},\ \overrightarrow{A_{r,s+1}} \longmapsto \gamma\ \nabla_{r,s+1} \qquad A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$$

pero en tal caso las producciones involucradas no forman parte de la familia de transiciones elementales de los BEPDA. Esta circunstancia nos ha llevado a descomponer la regla de compilación **[SRET-2]** en dos reglas **[SRET-2a]** y **[SRET-2b]** y a descomponer la regla **[SRET-3]** en dos reglas **[SRET-3a]** y **[SRET-3b]**. Al igual que ocurrió en el caso de los EPDA, esta descomposición de las reglas de compilación conlleva como efecto colateral un cambio en las pilas del autómata, que pasarán a tener la forma $[\alpha B$, donde B será un no-terminal de la gramática lineal de índices y α estará formado por una sucesión de triples $\langle\gamma, r, s\rangle$, donde γ es un índices mientras que r y s señalan una posición s en una producción r. La proyección del primer componente de dicho triples proporciona la pila de índices asociada a B. Los componentes r y s sólo se utilizan en la regla de compilación **[SRET-3b]** mientras que son ignorados en **[SRET-2b]**.
§

[INIT]	$\$_0 \longmapsto \$_0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \longmapsto \nabla_{r,s}, [\overrightarrow{A_{r,s+1}}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 \boldsymbol{A_{r,s+1}} \Upsilon_2$
[SEL]	$\overrightarrow{A_{r,0}} \longmapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \longmapsto \overleftarrow{A_{r,0}}$	
[RET]	$\nabla_{r,s}, [\overleftarrow{A_{r,s+1}} \longmapsto \nabla_{r,s+1}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 A_{r,s+1}[\,]\Upsilon_2$
[SRET-1]	$[\nabla_{r,s}, \overleftarrow{A_{r,s+1}} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-2a]	$[\nabla_{r,s}, \overleftarrow{A_{r,s+1}} \longmapsto \nabla'_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-2b]	$\langle \gamma', t, u \rangle\, \nabla'_{r,s+1} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-3a]	$[\nabla_{r,s}, \overleftarrow{A_{r,s+1}} \longmapsto \langle \gamma, r, s+1 \rangle$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-3b]	$\langle \gamma, r, s+1 \rangle \longmapsto \langle \gamma, r, s+1 \rangle\, \nabla_{r,s+1}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SCAN]	$\overrightarrow{A_{r,0}} \overset{a}{\longmapsto} \overleftarrow{A_{r,0}}$	$A_{r,0}[\,] \rightarrow a$

Tabla 7.9: Reglas del esquema de compilación genérico de LIG en BEPDA

7.7.1 Estrategia descendente

Esquema de compilación 7.7 El esquema de compilación con estrategia descendente de una gramática lineal de índices en un autómata a pila embebido ascendente queda definido por el conjunto de reglas de la tabla 6.10 y los elementos inicial $\$_0$ y final $\Box$. §

7.7.2 Estrategia Earley

Esquema de compilación 7.8 El esquema de compilación con estrategia Earley de una gramática lineal de índices en un autómata a pila embebido ascendente queda definido por el conjunto de reglas de la tabla 6.11 y los elementos inicial $\$_0$ y final $\overline{\overline{S}}$. §

7.7.3 Estrategia ascendente

Esquema de compilación 7.9 El esquema de compilación con estrategia ascendente de una gramática lineal de índices en un autómata a pila embebido ascendente queda definido por el conjunto de reglas de la tabla 6.12 y los elementos inicial $\$_0$ y final S. §

[INIT]	$\$_0 \longmapsto \$_0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \longmapsto \nabla_{r,s}, [A_{r,s+1}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 \boldsymbol{A_{r,s+1}} \Upsilon_2$
[SEL]	$A_{r,0} \longmapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \longmapsto \Box$	
[RET]	$\nabla_{r,s}, [\Box \longmapsto \nabla_{r,s+1}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 A_{r,s+1}[\,]\Upsilon_2$
[SRET-1]	$[\nabla_{r,s}, \Box \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-2a]	$[\nabla_{r,s}, \Box \longmapsto \nabla'_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-2b]	$\langle\gamma', t, u\rangle\, \nabla'_{r,s+1} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-3a]	$[\nabla_{r,s}, \Box \longmapsto \langle\gamma, r, s+1\rangle$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-3b]	$\langle\gamma, r, s+1\rangle \longmapsto \langle\gamma, r, s+1\rangle\, \nabla_{r,s+1}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SCAN]	$A_{r,0} \overset{a}{\longmapsto} \Box$	$A_{r,0}[\,] \rightarrow a$

Tabla 7.10: Esquema de compilación descendente de LIG en BEPDA

[INIT]	$\$_0 \longmapsto \$_0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \longmapsto \nabla_{r,s}, [\overline{A_{r,s+1}}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 \boldsymbol{A_{r,s+1}} \Upsilon_2$
[SEL]	$\overline{A_{r,0}} \longmapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \longmapsto \overline{\overline{A_{r,0}}}$	
[RET]	$\nabla_{r,s}, [\overline{\overline{A_{r,s+1}}} \longmapsto \nabla_{r,s+1}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 A_{r,s+1}[\,]\Upsilon_2$
[SRET-1]	$[\nabla_{r,s}, \overline{\overline{A_{r,s+1}}} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-2a]	$[\nabla_{r,s}, \overline{\overline{A_{r,s+1}}} \longmapsto \nabla'_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-2b]	$\langle\gamma', t, u\rangle\, \nabla'_{r,s+1} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-3a]	$[\nabla_{r,s}, \overline{\overline{A_{r,s+1}}} \longmapsto \langle\gamma, r, s+1\rangle$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-3b]	$\langle\gamma, r, s+1\rangle \longmapsto \langle\gamma, r, s+1\rangle\, \nabla_{r,s+1}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SCAN]	$\overline{A_{r,0}} \overset{a}{\longmapsto} \overline{\overline{A_{r,0}}}$	$A_{r,0}[\,] \rightarrow a$

Tabla 7.11: Reglas del esquema de compilación Earley de LIG en BEPDA

[INIT]	$\$_0 \longmapsto \$_0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \longmapsto \nabla_{r,s}, [\Box$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 \boldsymbol{A_{r,s+1}} \Upsilon_2$
[SEL]	$\Box \longmapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \longmapsto A_{r,0}$	
[RET]	$\nabla_{r,s}, [A_{r,s+1} \longmapsto \nabla_{r,s+1}$	$\boldsymbol{A_{r,0}} \rightarrow \Upsilon_1 A_{r,s+1}[\,]\Upsilon_2$
[SRET-1]	$[\nabla_{r,s}, A_{r,s+1} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-2a]	$[\nabla_{r,s}, A_{r,s+1} \longmapsto \nabla'_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-2b]	$\langle\gamma', t, u\rangle\, \nabla'_{r,s+1} \longmapsto \nabla_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$
[SRET-3a]	$[\nabla_{r,s}, A_{r,s+1} \longmapsto \langle\gamma, r, s+1\rangle$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SRET-3b]	$\langle\gamma, r, s+1\rangle \longmapsto \langle\gamma, r, s+1\rangle\, \nabla_{r,s+1}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$
[SCAN]	$\Box \overset{a}{\longmapsto} A_{r,0}$	$A_{r,0}[\,] \rightarrow a$

Tabla 7.12: Reglas del esquema de compilación ascendente de LIG en BEPDA

7.8 Lenguajes de adjunción de árboles y BEPDA

Durante todo este capítulo hemos estado suponiendo que la clase de los lenguajes aceptados por los BEPDA coincide con la clase de los lenguajes de adjunción de árboles, principalmente debido a su dualidad con los EPDA. En esta sección proporcionamos una justificación formal a tal suposición. Para ello haremos uso de los esquemas de compilación definidos previamente.

Teorema 7.8 *Los lenguajes adjunción de árboles son un subconjunto de los lenguajes aceptados por la clase de los autómatas a pila embebidos ascendentes.*

Demostración:

Por el esquema de compilación 7.2, a partir de cualquier gramática de adjunción de árboles es posible construir un BEPDA que acepta el lenguaje reconocido por dicha gramática. Análogamente, por el esquema de compilación 7.6, a partir de cualquier gramática lineal de índices es posible construir un BEPDA que acepta el lenguaje reconocido por dicha gramática. □

Teorema 7.9 *La clase de los lenguajes aceptados por los BEPDA es un subconjunto de los lenguajes de adjunción de árboles.*

Demostración:

Mostraremos que para todo BEPDA existe una gramática lineal de índices tal que el lenguaje reconocido por la gramática coincide con el lenguaje aceptado por el autómata.

Sea $\mathcal{A} = (V_T, V_S, \Theta, \$_0, \$_f)$ un autómata a pila embebido ascendente. Construiremos una gramática lineal de índices $\mathcal{L} = (V_T, V_N, V_I, S, P)$, donde $V_I = V_S$ y el conjunto V_N de no-terminales estará formado por pares $\langle E, B\rangle$ tal que $A, B \in V_S$. 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 Θ de la siguiente manera:

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

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 $\langle E, C\rangle[\alpha] \stackrel{*}{\Rightarrow} w$ si y sólo si $(E, w) \stackrel{*}{\vdash} (E[\alpha C, \epsilon)$. Esto es así puesto que

- Si una derivación $(E, w) \stackrel{*}{\vdash} (E[\alpha C, \epsilon)$ es el resultado de aplicar la secuencia $t_1, \ldots, t_m$ de transiciones en Θ, 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 $\langle E, C\rangle[\alpha] \stackrel{*}{\Rightarrow} w$ resultado de aplicar $p_m, \ldots, p_1$ reconoce w.
- Si una derivación derecha $\langle E, C\rangle[\alpha] \stackrel{*}{\Rightarrow} w$ 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 p_i es una producción creada a partir de t_i y la derivación $(E, w) \stackrel{*}{\vdash} (E[\alpha C, \epsilon)$ es el resultado de aplicar la secuencia de transiciones $t_m, \ldots, t_1$.

□

Ejemplo 7.4 El autómata a pila embebido ascendente $(V_T, V_S, \Theta, \$_0, \$_f)$, donde $V_T = \{a, b, c, d\}$ y $V_S = \{B, C, D, E, F, \$_0, \$_f\}$, del ejemplo 7.2 acepta el lenguaje $\{a^n b^n c^n d^n \mid n \geq 0\}$. A partir de dicho autómata construiremos una gramática lineal de índices $(V_T, V_S \times V_S, V_S, \langle \$_0, \$_f\rangle, P)$. La tabla 7.13 muestra el conjunto de transiciones P, que ha sido obtenido a partir de las transiciones del autómata mostradas en la tabla 7.2. Para facilitar la lectura, hemos utilizado Γ para denotar cualquier posible elemento de V_S. La tabla 7.14 muestra la derivación de la cadena *aabbccdd* en esta gramática. La primera columna muestra la producción aplicada para obtener la forma sentencial de la segunda columna. Es interesante resaltar que la secuencia de producciones aplicada en la derivación coincide en orden inverso con la secuencia de transiciones aplicada por el autómata para aceptar al misma entrada (tabla 7.2). ¶

Corolario 7.10 *La clase de los lenguajes aceptados por los autómatas a pila embebidos ascendentes coincide con la clase de los lenguajes de adjunción de árboles.*

Demostración:

Inmediata a partir de los teoremas 7.8 y 7.8. □

$$
\begin{array}{ll}
(a) & \langle \$_0, D\rangle[\,] \rightarrow a \\
(b) & \langle D, D, [\,]\rangle \rightarrow a \\
\\
(c) & \langle D, C\rangle[\,] \rightarrow \epsilon \\
(d) & \langle C, C\rangle[\,] \rightarrow b \\
\\
(e) & \langle \Gamma, B\rangle[\circ\circ] \rightarrow \langle \Gamma, C\rangle[\circ\circ] \\
(f) & \langle \Gamma, E\rangle[\circ\circ B] \rightarrow \langle \Gamma, B\rangle[\circ\circ] \\
(g) & \langle \Gamma, C\rangle[\circ\circ] \rightarrow \langle \Gamma, C\rangle[\,]\ \langle C, E\rangle[\circ\circ]\ c \\
\\
(h) & \langle \Gamma, B\rangle[\circ\circ] \rightarrow \langle \Gamma, C\rangle[\circ\circ B] \\
(i) & \langle \Gamma, D\rangle[\circ\circ] \rightarrow \langle \Gamma, D\rangle[\,]\ \langle D, B\rangle[\circ\circ]\ d \\
(j) & \langle \Gamma, B\rangle[\circ\circ] \rightarrow \langle \Gamma, D\rangle[\circ\circ B] \\
(k) & \langle \Gamma, \$_f\rangle[\circ\circ] \rightarrow \langle \Gamma, D\rangle[\circ\circ]
\end{array}
$$

Tabla 7.13: Producciones de la LIG derivada del BEPDA que acepta $\{a^n b^n c^n d^n\}$

$$
\begin{array}{lll}
(k) & \langle \$_0, \$_f\rangle[\,] & \Rightarrow \langle \$_0, D\rangle[\,] \\
(i) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, B\rangle[\,]\ \ d \\
(j) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ d \\
(i) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, B\rangle[B]\ \ dd \\
(i) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, B\rangle[B]\ \ dd \\
(h) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[BB]\ \ dd \\
(g) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, E\rangle[BB]\ \ cdd \\
(f) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, B\rangle[B]\ \ cdd \\
(f) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, C\rangle[B]\ \ cdd \\
(g) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, C\rangle[\,]\ \ \langle C, E\rangle[B]\ \ ccdd \\
(f) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, C\rangle[\,]\ \ \langle C, B\rangle[\,]\ \ ccdd \\
(e) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, C\rangle[\,]\ \ \langle C, C\rangle[\,]\ \ ccdd \\
(d) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ \langle C, C\rangle[\,]\ \ bccdd \\
(d) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ \langle D, C\rangle[\,]\ \ bbccdd \\
(c) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ \langle D, D\rangle[\,]\ \ bbccdd \\
(b) & & \Rightarrow \langle \$_0, D\rangle[\,]\ \ abbccdd \\
(a) & & \Rightarrow aabbccdd
\end{array}
$$

Tabla 7.14: Derivación de la cadena *aabbccdd*

7.9 Tabulación

Al igual que en el caso de los EPDA, la ejecución directa de los autómatas a pila embebidos ascendentes puede tener complejidad exponencial puesto que será preciso duplicar el contenido de la pila del autómata en aquellos casos en que sea posible aplicar más de una transición a una configuración dada. Para obtener una complejidad polinomial, recurriremos a una técnica de tabulación que nos permita representar las configuraciones en forma compacta mediante ítems que a su vez se almacenarán en una tabla. Para ello debemos distinguir los diferentes tipos de derivaciones que se pueden dar, que en el caso de los BEPDA son los dos tipos siguientes:

Derivaciones de llamada. Son derivaciones que sitúan una pila con un único elemento en la cima de la pila principal:

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

La pila Υ no debe haber sido tocada en toda la derivación y no debe existir $([F, f) \neq ([B, i)$ tal que

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

Las derivaciones de este tipo son independientes del valor concreto de Υ, puesto que para cualquier $\Upsilon' \in ([V_S^*)^*$ se cumple que

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

tal y como se observa en la figura 7.7. En consecuencia, las derivaciones de llamada pueden ser representadas de modo compacto por ítems de la forma

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

Derivaciones de retorno. Son aquellas derivaciones que sitúan en la cima de la pila principal una pila con más de un símbolo de pila:

$$\begin{aligned}(\Upsilon\ [B, a_{i+1} \dots a_n) &\overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D, a_{p+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [\alpha E, a_{q+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon\ [\alpha X C, a_{j+1} \dots a_n)\end{aligned}$$

No deben existir $([F, f) \neq ([B, i)$ ni $([G, g) \neq ([D, p)$ tal que

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

Las derivaciones de este tipo son independientes de Υ y α, puesto que para cualquier $\Upsilon' \in ([V_S^*)^*$ y $\alpha' \in V_S^*$ se cumple que

$$\begin{aligned}(\Upsilon'\ [B, a_{i+1} \dots a_n) &\overset{*}{\vdash} (\Upsilon'\ [B\ \ \Upsilon_1\ [D, a_{p+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon'\ [B\ \ \Upsilon_1\ [\alpha' E, a_{q+1} \dots a_n)\\ &\overset{*}{\vdash} (\Upsilon'\ [\alpha' X C, a_{j+1} \dots a_n)\end{aligned}$$

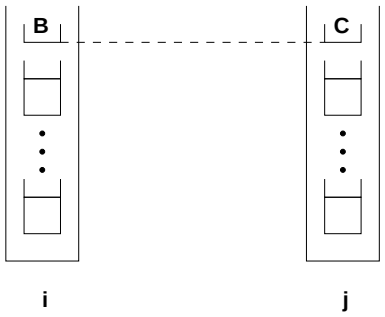

Figura 7.7: Derivaciones de llamada en BEPDA

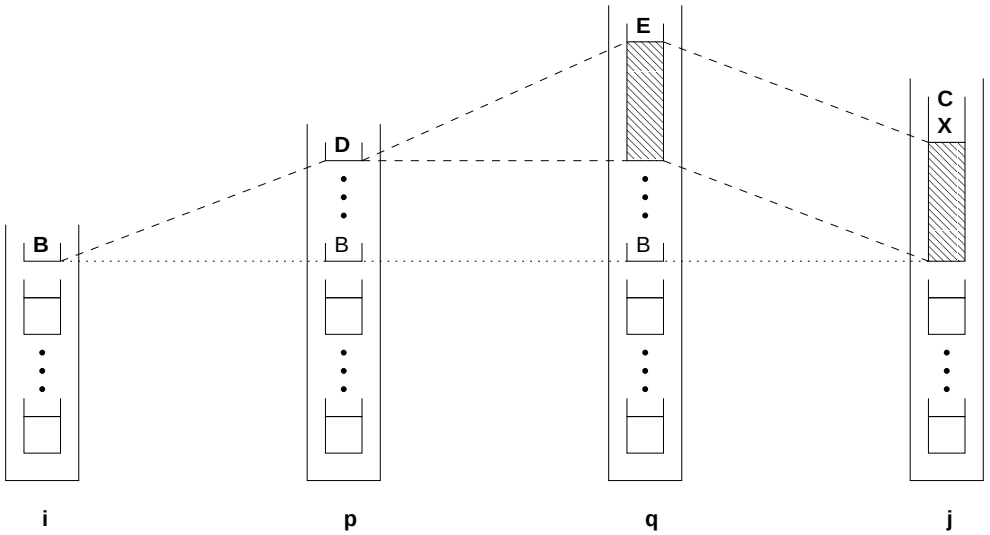

Figura 7.8: Derivaciones de retorno en BEPDA

tal y como se observa en la figura 7.8. En consecuencia, las derivaciones de retorno se pueden representar de forma compacta mediante ítems de la forma

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

Los ítems se combinan mediante las reglas de combinación mostradas en la tabla 7.15, en las cuales se cumple que

- $k = j$ 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 ítem inicial a partir del cual se aplican las reglas de la tabla 7.15 es de la forma

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

Si la cadena de entrada $a_1 \ldots a_n$ pertenece al lenguaje aceptado por el autómata, se producirán ítems finales de la forma

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

tal que existe una transición $\$_0 \longmapsto \$_0, [B$.

$$\frac{[B,i,C,j,X \mid D,p,E,q]}{[B,i,F,k,X \mid D,p,E,q]} \quad C \xmapsto{a} F$$

$$\frac{[B,i,C,j,X \mid D,p,E,q]}{[B,i,F,k,C \mid B,i,C,j]} \quad C \xmapsto{a} CF$$

$$\frac{\begin{array}{c}[B,i,F,j,C \mid D,p,E,q]\\ [D,p,E,q,X' \mid O,u,P,v]\end{array}}{[B,i,G,k,X' \mid O,u,P,v]} \quad CF \xmapsto{a} G$$

$$\frac{[B,i,C,j,X \mid D,p,E,q]}{[F,k,F,k,- \mid -,-,-,-]} \quad C \xmapsto{a} C, [F$$

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

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

Tabla 7.15: Combinación de ítems en BEPDA

Teorema 7.11 *La manipulación de configuraciones mediante la aplicación de transiciones en los autómatas a pila embebidos ascendentes es equivalente a la manipulación de ítems mediante las reglas de combinación de la tabla 7.15.*

Demostración:

Debemos demostrar que para toda derivación del autómata se producirá un ítem que la representa, y que para todo ítem producido por las reglas de combinación de la tabla 7.15 existe una configuración en el autómata a la que dicho ítem representa. Para ello daremos una lista exhaustiva de los diferentes tipos de derivaciones que se pueden producir, junto con la reglas de combinación de ítem que produce al ítem que representa dicha configuración. Veremos que toda reglas de combinación de ítem aparece en dicha lista, por lo que habremos demostrado los puntos. Las posibles derivaciones que se pueden producir son:

- Derivaciones que son el resultado de aplicar una transición $C \xmapsto{a} F$
  - a una derivación de llamada:

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

$$\frac{[B,i,C,j,- \mid -,-,-,-]}{[B,i,F,k,- \mid -,-,-,-]} \quad C \xmapsto{a} F$$

– a una derivación de retorno:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D, a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [\alpha E, a_{q+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [\alpha XC, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [\alpha XF, a_{k+1} \ldots a_n)\end{array}$$

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

- Derivaciones que son el resultado de aplicar una transición $C \overset{a}{\longmapsto} CF$
  – a una derivación de llamada:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [C, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [CF, a_{k+1} \ldots a_n)\end{array}$$

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

  – a una derivación de retorno:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D, a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [\alpha E, a_{q+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [\alpha XC, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [\alpha XCF, a_{k+1} \ldots a_n)\end{array}$$

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

- Derivaciones que son el resultado de aplicar una transición $CF \overset{a}{\longmapsto} G$ a una derivación de retorno[1] que a su vez a sido obtenida:
  – a partir de una derivación de llamada:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [B\ \Upsilon_1\ [D, a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [B\ \Upsilon_1\ [E, a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [CF, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [G, a_{k+1} \ldots a_n)\end{array}$$

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

  – a partir de una derivación de retorno:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D, a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D\ \Upsilon_2\ \ [O, a_{u+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D\ \Upsilon_2\ \ [\alpha P, a_{v+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [\alpha X'E, a_{q+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [\alpha X'CF, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [\alpha X'G, a_{k+1} \ldots a_n)\end{array}$$

[1] Este tipo de transiciones no son aplicables a derivaciones de llamada, pues estas últimas contienen en su cima una pila de la forma $[F$.

$$\frac{\begin{array}{l}[B,i,F,j,C \mid D,p,E,q]\\ [D,p,E,q,X' \mid O,u,P,v]\end{array}}{[B,i,G,k,X' \mid O,u,P,v]} \quad CF \stackrel{a}{\longmapsto} G$$

- Derivaciones que son el resultado de aplicar una transición $C \stackrel{a}{\longmapsto} C, [F$
  - a una derivación de llamada:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \stackrel{*}{\vdash} (\Upsilon\ [C, a_{j+1} \ldots a_n)\\ & \vdash (\Upsilon\ [C\ \ [F, a_{k+1} \ldots a_n)\end{array}$$

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

  - a una derivación de retorno:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \stackrel{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D, a_{p+1} \ldots a_n)\\ & \stackrel{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [\alpha E, a_{q+1} \ldots a_n)\\ & \stackrel{*}{\vdash} (\Upsilon\ [\alpha XC, a_{j+1} \ldots a_n)\\ & \vdash (\Upsilon\ [\alpha XC\ \ [F, a_{k+1} \ldots a_n)\end{array}$$

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

- Derivaciones que son el resultado de aplicar una transición $C, [F' \stackrel{b}{\longmapsto} G$ a una derivación obtenida a partir de la aplicación de una transición $C \stackrel{a}{\longmapsto} C, [F$:
  - a una derivación de llamada:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \stackrel{*}{\vdash} (\Upsilon\ [C, a_{j+1} \ldots a_n)\\ & \vdash (\Upsilon\ [C\ \ [F, a_{k+1} \ldots a_n)\\ & \stackrel{*}{\vdash} (\Upsilon\ [C\ \ [F', a_{k'+1} \ldots a_n)\\ & \vdash (\Upsilon\ [G, a_{l+1} \ldots a_n)\end{array}$$

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

  - a una derivación de retorno:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \stackrel{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [D, a_{p+1} \ldots a_n)\\ & \stackrel{*}{\vdash} (\Upsilon\ [B\ \ \Upsilon_1\ [\alpha E, a_{q+1} \ldots a_n)\\ & \stackrel{*}{\vdash} (\Upsilon\ [\alpha XC, a_{j+1} \ldots a_n)\\ & \vdash (\Upsilon\ [\alpha XC\ \ [F, a_{k+1} \ldots a_n)\\ & \stackrel{*}{\vdash} (\Upsilon\ [\alpha XC\ \ [F', a_{k'+1} \ldots a_n)\\ & \vdash (\Upsilon\ [\alpha XG, a_{l+1} \ldots a_n)\end{array}$$

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

- Derivaciones que son el resultado de aplicar una transición $C, [F' \xmapsto{b} G$ a una derivación obtenida a partir de la aplicación de una transición $C \xmapsto{a} C, [F$ a una derivación de llamada[2], con los dos casos siguientes:
  - la derivación obtenida es una derivación de llamada:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [C, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [C\ \ [F, a_{k+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [C\ \ [F', a_{k'+1} \ldots a_n) \\ & \vdash (\Upsilon\ [G, a_{l+1} \ldots a_n)\end{array}$$

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

  - la derivación obtenida es una derivación de retorno:

$$\begin{array}{ll}(\Upsilon\ [B, a_{i+1} \ldots a_n) & \overset{*}{\vdash} (\Upsilon\ [C, a_{j+1} \ldots a_n) \\ & \vdash (\Upsilon\ [C\ \ [F, a_{k+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [C\ \ [F\ \ \Upsilon_1\ [D, a_{p+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [C\ \ [F\ \ \Upsilon_1\ [\alpha E, a_{q+1} \ldots a_n) \\ & \overset{*}{\vdash} (\Upsilon\ [C\ \ [\alpha X F', a_{k'+1} \ldots a_n) \\ & \vdash (\Upsilon\ [\alpha X G, a_{l+1} \ldots a_n)\end{array}$$

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

□

La complejidad espacial de la técnica de tabulación propuesta con respecto a la longitud n de la cadena de entrada es $\mathcal{O}(n^4)$, puesto que cada ítem almacena 4 posiciones de la cadena de entrada. La complejidad temporal es $\mathcal{O}(n^6)$ y es debida a las reglas de combinación correspondientes a las transición $CF \xmapsto{a} G$, a la transición $C, [F' \xmapsto{b} G$ y a la transición $[C, F' \xmapsto{b} G$, pues no debemos olvidar que la posición k está ligada a j y que la posición l está ligada a k'.

[2] Si se hubiese aplicado esta transición a una derivación de retorno, no sería posible tener una pila $[C$ bajo la pila [F'.