Autómatas y Lenguajes Formales 2017-1
Facultad de Ciencias UNAM
Nota de Clase 11

Favio E. Miranda Perea   A. Liliana Reyes Cabello   Lourdes González Huesca

14 de octubre de 2016

1. Introducción

Los lenguajes regulares como lenguajes sencillos de reconocer por una máquina encuentran su mancuerna con los autómatas finitos, los cuales son dispositivos cuya memoria es limitada: sólo son útiles al recordar estado por estado alguna propiedad sin poder hacerlo a lo largo de todo una secuencia de estados.

Para poder reconocer lenguajes más expresivos, como los lenguajes libres de contexto podemos pensar en un alguna generalización de los autómatas finitos, la puede realizarse de diversos modos:

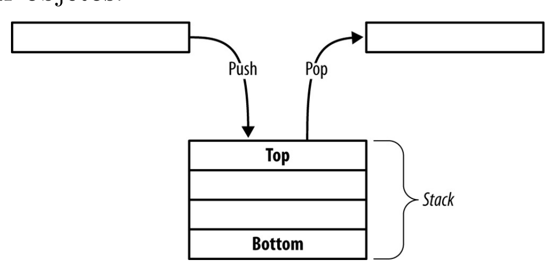
- permitir que el autómata *lea* una entrada un número arbitrario de veces;
- añadir un dispositivo de conteo (máquinas contadoras);
- añadir dispositivos de salida (traductores, autómatas de Mealy y Moore);
- agregar un dispositivo de memoria, por ejemplo mediante una pila para almacenar datos.

De las generalizaciones anteriores, la última permite abstraer características más complejas de un lenguaje al tener más memoria que almacene. Los autómatas que serán de utilidad para reconocer lenguajes libres de contexto son los que generalizan a los autómatas finitos al incluir una pila.

2. Autómatas de pila

Los autómatas de pila son una generalización de los autómatas finitos, el dispositivo de memoria es una pila que una estructura de datos para almacenar objetos.

Una pila es una estructura que puede almacenar elementos utilizando una función llamada `PUSH` y se puede acceder a los datos almacenados mediante dos funciones: obtener un elemento al "sacarlo" mediante la función `POP` o sólamente "ver" (`PEEK`) un elemento. Estas funciones sólo acceden al tope de la pila. El tope de la pila es exactamente el último elemento en ser almacenado y si no hay elementos almacenados decimos que la pila está vacía.



Esta característica le da otro nombre a las pilas **"Last In First Out" (LIFO)** es decir que el último elemento en ser almacenado será el primero en salir de la pila.
El manejo de pilas en los autómatas considera algunas variantes:

- cada pila tiene un alfabeto que contiene los símbolos permitidos para almacenarse en la pila. Este alfabeto debe incluir el símbolo inicial de la pila o "fondo" de la pila;
- la pila será tratada como una cadena de símbolos del alfabeto mencionado en donde el tope será el símbolo más a la izquierda de la cadena y el tope será el símbolo más a la derecha;
- diremos que una pila está vacía si está representada por la cadena vacía.

Una visión panorámica de los autómatas de pila distingue tres parámetros: el estado actual, el símbolo que se está procesando además del símbolo que se encuentra en el tope de la pila. También caracteriza su funcionamiento central: la máquina es capaz de leer el símbolo a procesar de la cadena de entrada, cambiar de estado y sustituir el símbolo en el tope de la pila por una secuencia de símbolos.

Al igual que los autómatas finitos, se puede incluir el no-determinismo, se hará énfasis en la diferencia entre determinismo y no-determinismo dentro de estas máquinas más adelante. Ahora se presentan formalmente los autómatas no deterministas así como los conceptos que detallan su funcionamiento y el procesamiento de cadenas.

Definición 1 *Un autómata de pila* ***no determinista*** *es una séptupla:*

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$$

donde:

- $Q \neq \varnothing$ *es un conjunto finito de estados.*
- Σ *es el alfabeto de entrada.*
- Γ *es el alfabeto de la pila.*
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^\star)$
- $q_0 \in Q$ *es el estado inicial.*
- $Z_0 \in \Gamma$ *es el símbolo inicial de la pila.*
- $F \subseteq Q$ *es el conjunto de estados finales.*

Analicemos con detalle la función de transición:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^\star)$$

esta función recibe tres argumentos: un estado, un símbolo o la cadena vacía y el símbolo al tope de la pila; devuelve un par con el nuevo estado y los símbolos que serán el nuevo tope de la pila.

Las transiciones son los pares que devuelve la función δ: si $(p, \gamma) \in \delta(q, a, s)$ entonces estando en el estado q y leyendo a con s en el tope de la pila, el autómata hace lo siguiente:

- consume el símbolo a
- elimina s del tope de la pila, es decir `POP`(s)
- escribe γ en el tope de la pila, es decir `PUSH`(γ)
- cambia al estado p.

Es posible consumir, leer o escribir la cadena vacía ε, de ahí que estos autómatas sean no-deterministas.

Transiciones especiales Los siguientes tipos de transiciones son de especial interés:

- $(q', s) \in \delta(q, a, s)$
  en este caso el contenido de la pila no se ha alterado.
- $(q', \varepsilon) \in \delta(q, a, s)$
  el símbolo s se borra y el autómata tiene un nuevo tope en la pila, es decir, el símbolo colocado inmediatamente abajo de s.
- $(q', \gamma) \in \delta(q, \varepsilon, s)$
  esta es una ε-transición, el símbolo de entrada no se procesa, se cambió de estado y además el tope de la pila s se reemplaza por γ.

2.1. Aceptación de palabras por un autómata

Definición 2 *Una configuración o descripción instantánea de un autómata de pila es una terna*

$$\langle q, aw, s\beta \rangle$$

que representa lo siguiente:

- *el autómata está en el estado q*
- *aw es la parte aún **no** procesada de la cadena de entrada, siendo a el siguiente símbolo a leer*
- *$s\beta$ es el contenido total de la pila, siendo s el símbolo colocado en el tope.*

La noción informal de cómputo se formaliza mediante una relación $\vdash_M$ entre configuraciones, llamada **paso de computación** tal que

$$\langle q, aw, s\beta \rangle \vdash_M \langle p, w, \gamma\beta \rangle \quad \text{si y sólo si} \quad (p, \gamma) \in \delta(q, a, s)$$

- q es el estado actual de la máquina M
- aw es la parte de la cadena de entrada que falta por procesar con a el siguiente símbolo a procesar
- q es el nuevo estado de la máquina
- w es la nueva cadena a procesar
- $\gamma\beta$ es el contenido de la pila en donde γ está en el tope

Existen diferentes tipos de configuraciones que debemos enfatizar:

- Configuración inicial: $\langle q_0, w, Z_0 \rangle$ para cualquier $w \in \Sigma^\star$.
- Configuración de aceptación con estados finales: $\langle q_f, \varepsilon, \beta \rangle$ con $q_f \in F$.
- Configuración de aceptación por pila vacía: $\langle q_f, \varepsilon, \varepsilon \rangle$.

Las configuraciones descritas son importantes y las dos últimas distinguen dos formas de aceptar cadenas, una verificando el estado de la máquina y la otra considerando el contenido de la pila. Estas formas serán más claras después de introducir el concepto de procesamiento de cadenas.

El procesamiento de una cadena en un autómata de pila no-determinista puede hacerse mediante un árbol que analice los posibles cómputos. La raíz del árbol será la configuración inicial $\langle q_0, w, Z_0\rangle$ y sus descendientes los pasos de computación generados por la máquina. Las hojas del árbol pueden ser cómputos bloqueados o alguna de las configuraciones de aceptación descritas antes.

2.2. Lenguaje aceptado por un autómata de pila

Con la noción de paso de computación se formaliza la noción de lenguaje aceptado por un autómata por estados finales. La relación $\vdash^\star$ se define de la manera usual, es decir es la extensión a varios pasos de la noción de cómputo.

Se define el lenguaje de aceptación de un autómata de pila como se hizo con los autómatas finitos: al procesar una cadena, el autómata ha alcanzado un estado final

$$L(M) = \{w \in \Sigma^\star \mid \langle q_0, w, Z_0\rangle \vdash_M^\star \langle q_f, \varepsilon, \beta\rangle \text{ y } q_f \in F\}$$

Obsérvese que el contenido de la pila β es irrelevante y la cadena ha sido aceptada ya que se terminó de procesar llegando al estado final q_f.

Otra forma de aceptación para los autómatas de pila es mediante la pila vacía:

$$V(M) = \{w \in \Sigma^\star \mid \langle q_0, w, Z_0\rangle \vdash_M^\star \langle p, \varepsilon, \varepsilon\rangle\}$$

En este caso, obsérvese que el estado p es irrelevante pero lo importante es procesar por completo la cadena y que la pila esté vacía.

Estos dos tipos de aceptación son equivalentes como veremos más adelante, ahora veamos ejemplos de estas máquinas.

Ejemplo: Considere el lenguaje $L_1 = \{a^i b^i \mid i \geq 0\}$, el siguente autómata de pila reconoce a L_1:

$$M_1 = \langle \{q_0, q_1, q_2\}, \{a, b\}, \{X, Z_0\}, \delta, q_0, Z_0, \{q_2\}\rangle$$

donde:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\delta(q_0, a, Z_0) & = & \{(q_0, XZ_0)\} & \delta(q_0, a, X) & = & \{(q_0, XX)\} \\
\delta(q_0, \varepsilon, Z_0) & = & \{(q_2, \varepsilon)\} & \delta(q_0, b, X) & = & \{(q_1, \varepsilon)\} \\
\delta(q_1, b, X) & = & \{(q_1, \varepsilon)\} & \delta(q_1, \varepsilon, Z_0) & = & \{(q_2, \varepsilon)\}
\end{array}$$

La aceptación de cadenas es por medio de estados finales.

Ejemplo: Ahora veamos un autómata que reconoce a L_1 por medio de la pila vacía:

$$M_1' = \langle \{q_0, q_1\}, \{a, b\}, \{X, Z_0\}, \delta, q_0, Z_0, \varnothing\rangle$$

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\delta(q_0, a, Z_0) & = & \{(q_0, XZ_0)\} & \delta(q_0, a, X) & = & \{(q_0, XX)\} \\
\delta(q_0, \varepsilon, Z_0) & = & \{(q_0, \varepsilon)\} & \delta(q_0, b, X) & = & \{(q_1, \varepsilon)\} \\
\delta(q_1, b, X) & = & \{(q_1, \varepsilon)\} & \delta(q_1, \varepsilon, Z_0) & = & \{(q_1, \varepsilon)\}
\end{array}$$

Ejemplo: Sea $L_2 = \{a^i b^j \mid i \geq j \geq 0\}$, el siguiente autómata acepta por pila vacía el lenguaje:

$$M_2 = \langle \{q_0, q_1\}, \{a, b\}, \{X, Z_0\}, \delta, q_0, Z_0, \varnothing \rangle$$

$$\begin{array}{rclcrcl}
\delta(q_0, a, Z_0) & = & \{(q_0, XZ_0)\} & \qquad & \delta(q_0, a, X) & = & \{(q_0, XX)\} \\
\delta(q_0, \varepsilon, Z_0) & = & \{(q_0, \varepsilon)\} & & \delta(q_0, b, X) & = & \{(q_1, \varepsilon)\} \\
\delta(q_1, b, X) & = & \{(q_1, \varepsilon)\} & & \delta(q_1, \varepsilon, Z_0) & = & \{(q_1, \varepsilon)\} \\
\delta(q_1, \varepsilon, X) & = & \{(q_1, \varepsilon)\} & & & &
\end{array}$$

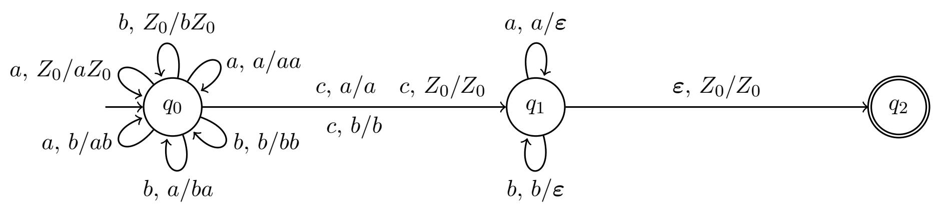
Al igual que los autómatas finitos, los autómatas de pila tienen una representación gráfica usando vértices o nodos y transiciones entre ellos. Las transiciones estarán etiquetadas por los pares que representan el símbolo a procesar y el *antes/después* del tope en la pila. Veamos un ejemplo:

Ejemplo: El lenguaje $L_3 = \{wcw^R \mid w \in \{a, b\}^\star\}$ es aceptado por el autómata

$$M_3 = \langle \{q_0, q_1\}, \{a, b, c\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \varnothing \rangle$$

$$\begin{array}{rclcrcl}
\delta(q_0, a, Z_0) & = & \{(q_0, AZ_0)\} & \qquad & \delta(q_0, b, Z_0) & = & \{(q_0, BZ_0)\} \\
\delta(q_0, a, A) & = & \{(q_0, AA)\} & & \delta(q_0, b, A) & = & \{(q_0, BA)\} \\
\delta(q_0, a, B) & = & \{(q_0, AB)\} & & \delta(q_0, b, B) & = & \{(q_0, BB)\} \\
\delta(q_0, c, Z_0) & = & \{(q_1, Z_0)\} & & \delta(q_0, c, A) & = & \{(q_1, A)\} \\
\delta(q_0, c, B) & = & \{(q_1, B)\} & & \delta(q_1, a, A) & = & \{(q_1, \varepsilon)\} \\
\delta(q_1, b, B) & = & \{(q_1, \varepsilon)\} & & \delta(q_1, \varepsilon, Z_0) & = & \{(q_1, \varepsilon)\}
\end{array}$$

También es aceptado por el siguiente autómata:



2.3. Equivalencia de la aceptación

Veamos las demostraciones para justificar que los dos tipos de aceptación en un autómata de pila, por estados finales o por pila vacía, son equivalentes.

De estados finales a pila vacía

Lema 1 *Dado un lenguaje L tal que $L = L(M)$ para algún autómata de pila M, existe un autómata de pila M' tal que $L = V(M')$. Es decir, todo lenguaje aceptado por estados finales es aceptado por pila vacía.*

La prueba consiste en construir un nuevo autómata que acepta el mismo lenguaje con la pila vacía. Sea $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ entonces definimos a M' como sigue:

$$M' = \langle Q \cup \{p_0, p\}, \Sigma, \Gamma \cup \{N_0\}, \delta, p_0, N_0, F \rangle$$

en donde

- se agregan dos estados p_0, p siendo p_0 el nuevo estado inicial;
- se agrega un símbolo N_0 a Γ, el cual será el nuevo símbolo inicial de la pila;
- la función de transición conserva todas las transiciones de M además de las siguientes:

$$\delta(p_0, \varepsilon, N_0) = \{(q_0, Z_0 N_0)\}$$
$$(p, s) \in \delta(q_f, \varepsilon, s), \text{ para todo } q_f \in F,\ s \in \Gamma \cup \{N_0\}$$
$$\delta(p, \varepsilon, s) = \{(p, \varepsilon)\}$$

De pila vacía a estados finales

Lema 2 *Dado un lenguaje L tal que $L = V(M)$ para un autómata de pila M, existe un autómata de pila M' tal que $L = L(M')$. Es decir, todo lenguaje aceptado por pila vacía es aceptado por estados finales.*

Esta prueba consiste en construir los estados finales del autómata.
Sea $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ entonces definimos a M' como sigue:

$$M' = \langle Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{N_0\}, \delta, p_0, N_0, \{p_f\}\rangle$$

es decir

- se agregan dos estados, p_0, p_f siendo p_0 el nuevo estado inicial y p_f el **único** estado final;
- se agrega un símbolo N_0 a Γ, el cual será el nuevo símbolo inicial de la pila;
- la función de transición conserva todas las transiciones de M y se agregan:

$$\delta(p_0, \varepsilon, N_0) = \{(q_0, Z_0 N_0)\}$$
$$(p_f, s) \in \delta(q, \varepsilon, N_0), \text{ para todo } q \in Q$$

2.4. Autómatas de pila deterministas

Los autómatas presentados hasta ahora son no-deterministas. La capacidad de ser determinista está dada por que sólo hay un estado en la transición de cualquier estado leyendo un símbolo del alfabeto.

En los autómatas de pila se logra al modificar la función de transición:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to Q \times \Gamma^\star$$

es decir hay a lo más una transición en un estado y símbolo dados:

- $\delta(q, a, s) = (p, \gamma)$ para p, γ únicos
- o bien $\delta(q, a, s)$ no está definida, es decir, δ es una función parcial

Al permitirse las ε-transiciones necesitamos la siguiente convención para mantener el determinismo cuando son retiradas:

> Si $\delta(q, a, s)$ está definida entonces $\delta(q, \varepsilon, s)$ está indefinida y viceversa. Es decir, ambas transiciones no pueden estar definidas al mismo tiempo.

2.5. Determinismo vs. No-determinismo

El poder expresivo que proporciona el no-determinismo en una máquina puede ser valioso pero también puede agregar complejidades que no son deseadas al momento de una implementación seria.

Obviamente un autómata de pila determinista es un caso particular de un autómata de pila no-determinista. A diferencia de los autómatas finitos, en el caso de los autómatas de pila no hay equivalencia entre los autómatas deterministas y los no-deterministas.

Existen lenguajes que sólo pueden ser reconocidos mediante un autómata de pila no-determinista. Por ejemplo, $L = \{w \in \{a,b\}^\star \mid w = w^R\}$ el lenguaje de los palíndromos. Esto implica que hay lenguajes libres de contexto que no pueden ser reconocidos por un autómata de pila determinista.

En lo siguiente estudiaremos a fondo la equivalencia entre estas máquinas con memoria y los lenguajes libres de contexto.

3. Equivalencia con Lenguajes Libres de Contexto

Teorema 1 *Un lenguaje es libre de contexto si y sólo si es aceptado por un autómata de pila (no-determinista).*

Demostración. La prueba es en dos partes:

I Síntesis: Dado un lenguaje libre de contexto L existe un autómata de pila M tal que $L = L(M)$.

II Análisis: Dado un autómata de pila M existe una gramática libre de contexto G tal que $L(M) = L(G)$. Es decir, $L(M)$ es libre de contexto.

3.1. De una Gramática Libre de Contexto a un Autómata de Pila

Como vimos anteriormente, un lenguaje libre de contexto está determinado por una gramática libre de contexto que lo genere. Sea $G = \langle V, T, S, P\rangle$ una gramática libre de contexto, definimos un autómata de pila M_G como sigue:

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = T,\ \Gamma = V \cup \{Z_0\}$
- $F = \{q_2\}$
- δ se define mediante:

$$\begin{aligned}
\delta(q_0, \varepsilon, Z_0) &= \{(q_1, SZ_0)\} \\
\delta(q_1, \varepsilon, A) &= \{(q_1, \alpha) \mid A \to \alpha \in P\} \\
\delta(q_1, a, a) &= \{(q_1, \varepsilon)\}\ \ \forall a \in \Sigma \\
\delta(q_1, \varepsilon, Z_0) &= \{(q_2, Z_0)\}
\end{aligned}$$

Se cumple $L(G) = L(M_G)$.

Ejemplo: Dada G mediante

$$S \to aSb \mid cSbS \mid a$$

el autómata de pila correspondiente es:

$$\begin{aligned} \delta(q_0, \varepsilon, Z_0) &= \{(q_1, SZ_0)\} & \delta(q_1, \varepsilon, S) &= \{(q_1, aSb),\ (q_1, cSb),\ (q_1, a)\} \\ \delta(q_1, a, a) &= \{(q_1, \varepsilon)\} & \delta(q_1, b, b) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, c, c) &= \{(q_1, \varepsilon)\} & \delta(q_1, \varepsilon, Z_0) &= \{(q_2, Z_0)\} \end{aligned}$$

3.2. De un Autómata de Pila a una Gramática Libre de Contexto

Dado un autómata de pila M que acepta por pila vacía, vamos a construir una gramática libre de contexto G_M tal que $L(M) = L(G_M)$. Este método genera gramáticas bastante complejas, con un gran número de variables y producciones. Además de que pueden generarse variables inútiles, pero ya se han estudiado los métodos de transformación de gramáticas para simplificarlas o normalizarlas.

Las variables de la gramática serán S como símbolo inicial y las expresiones de la forma:

$$[p, X, q] \quad \text{donde } p, q \in Q,\ X \in \Gamma$$

La idea básica es simular con derivaciones los cómputos de M: la variable $[p, X, q]$ debe generar todas las cadenas que llevan al autómata de p a q al eliminar X de la pila.

Las producciones de la gramática son:

- Las producciones iniciales: $S \to [q_0, Z_0, p]$ para todo $p \in Q$.
- Si $(p, \varepsilon) \in \delta(q, a, X)$ entonces leyendo a el autómata pasa de q a p y elimina a X de la pila. Se agrega $[q, X, p] \to a$, esto incluye el caso $a = \varepsilon$.
- Si $(p, Y_1 Y_2 \dots Y_m) \in \delta(q, a, X)$ entonces leyendo a el autómata pasa de q a p y sustituye a X por $Y_1 Y_2 \dots Y_m$ en la pila. Se agrega $[q, X, p_m] \to a[p, Y_1, p_1][p_1, Y_2, p_2] \dots [p_{m-1} Y_m p_m]$ para todas las elecciones posibles de $p_1, \dots, p_m \in Q$. Esto incluye el caso $a = \varepsilon$.

Ejemplo: Considere el lenguaje $L = \{a^i b^i \mid i \geq 0\}$ cuyo autómata de pila es:

$$\begin{aligned} \delta(q_0, a, Z_0) &= \{(q_0, XZ_0)\} & \delta(q_0, a, X) &= \{(q_0, XX)\} \\ \delta(q_0, \varepsilon, Z_0) &= \{(q_0, \varepsilon)\} & \delta(q_0, b, X) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, b, X) &= \{(q_1, \varepsilon)\} & \delta(q_1, \varepsilon, Z_0) &= \{(q_1, \varepsilon)\} \end{aligned}$$

La gramática correspondiente tiene las siguientes producciones iniciales:

$$S \to [q_0, Z_0, q_0] \qquad\qquad S \to [q_0, Z_0, q_1]$$

Además de las producciones correspondientes a las transiciones

$$\begin{aligned} \delta(q_0, \varepsilon, Z_0) &= \{(q_0, \varepsilon)\} & [q_0, Z_0, q_0] &\to \varepsilon \\ \delta(q_0, b, X) &= \{(q_1, \varepsilon)\} & [q_0, X, q_1] &\to b \\ \delta(q_1, b, X) &= \{(q_1, \varepsilon)\} & [q_1, X, q_1] &\to b \\ \delta(q_1, \varepsilon, Z_0) &= \{(q_1, \varepsilon)\} & [q_1, Z_0, q_1] &\to \varepsilon \end{aligned}$$

Para la transición $\delta(q_0, a, Z_0) = \{(q_0, XZ_0)\}$ se tienen las siguientes producciones:

$$\begin{array}{rcl} [q_0, Z_0, q_1] & \rightarrow & a[q_0, X, q_0][q_0, Z_0, q_1] \\ [q_0, Z_0, q_0] & \rightarrow & a[q_0, X, q_1][q_1, Z_0, q_0] \\ [q_0, Z_0, q_0] & \rightarrow & a[q_0, X, q_0][q_0, Z_0, q_0] \\ [q_0, Z_0, q_1] & \rightarrow & a[q_0, X, q_1][q_1, Z_0, q_1] \end{array}$$

Y finalmente para $\delta(q_0, a, X) = \{(q_0, XX)\}$ las producciones:

$$\begin{array}{rcl} [q_0, X, q_0] & \rightarrow & a[q_0, X, q_0][q_0, X, q_0] \\ [q_0, X, q_0] & \rightarrow & a[q_0, X, q_1][q_1, X, q_0] \\ [q_0, X, q_1] & \rightarrow & a[q_0, X, q_0][q_0, X, q_1] \\ [q_0, X, q_1] & \rightarrow & a[q_0, X, q_1][q_1, X, q_1] \end{array}$$

Veamos la derivación de la cadena *aabb*:

$$\begin{array}{rcl} S & \rightarrow & [q_0, Z_0, q_1] \\ & \rightarrow & a[q_0, X, q_1][q_1, Z_0, q_1] \\ & \rightarrow & aa[q_0, X, q_1][q_1, X, q_1][q_1, Z_0, q_1] \\ & \rightarrow & aab[q_1, X, q_1][q_1, Z_0, q_1] \\ & \rightarrow & aabb[q_1, Z_0, q_1] \\ & \rightarrow & aabb \end{array}$$