Unidad 4. Autómatas de Pila

- Una de las limitaciones de los AF es que no pueden reconocer el lenguaje {0n1n} debido a que no se puede registrar para todo n con un número finito de estados.
- Otro lenguaje es el de las expresiones con paréntesis balanceados. Que se pueden extender para describir expresiones anidadas usadas en los lenguajes de programación.
- Aún cuando los lenguajes regulares permiten describir identificadores, palabras claves y patrones de uso común en computación necesitamos un modelo más poderoso para describir las estructuras sintácticas de los lenguajes de programación.
- Las gramaticas Libres de Contexto (GLC) y los lenguajes que generan (Lenguajes Libres de Contexto) que permiten definir la sintaxis de los lenguajes de programación.

Ejemplo

$$
\begin{aligned}
\langle \mathit{stmt} \rangle &::= \langle \mathit{ifstmt} \rangle \mid \langle \mathit{whilestmt} \rangle \\
&\quad \mid \langle \mathit{beginstmt} \rangle \\
\langle \mathit{ifstmt} \rangle &::= \textbf{if } \langle \mathit{exprbool} \rangle \textbf{ then } \langle \mathit{stmt} \rangle \\
&\quad \textbf{else } \langle \mathit{stmt} \rangle \\
\langle \mathit{whilestmt} \rangle &::= \textbf{while } \langle \mathit{exprbool} \rangle \textbf{ do } \langle \mathit{stmt} \rangle \\
\langle \mathit{beginstmt} \rangle &::= \textbf{begin } \langle \mathit{stmtlist} \rangle \textbf{ end} \\
\langle \mathit{stmtlist} \rangle &::= \langle \mathit{stmt} \rangle \mid \langle \mathit{stmt} \rangle \textbf{ ; } \langle \mathit{stmtlist} \rangle
\end{aligned}
$$

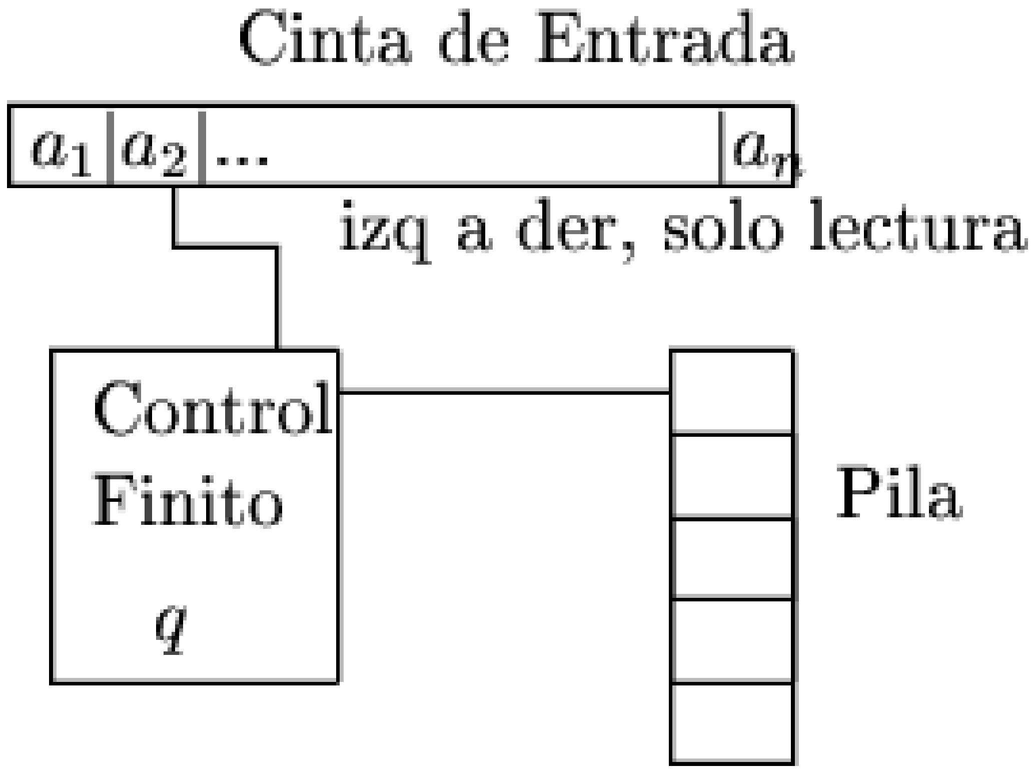
Autómatas de Pila (Push Down Automata)

- Los modelos mecánicos que corresponden a las GLC son los Autómatas de Pila que son como los AF pero tienen adicionalmente una pila para almacenamiento.
- Operaciones
  - Push (empilar un elemento en el tope)
  - Pop (desempilar el elemento situado en el tope)
- Registra información en forma LIFO (primero en entrar, último en salir).

Ejemplo

- Para reconoce el lenguaje {0n1n} un AP puede empliar cada 0 que lea y cuando inicie a leer 1’s desempila los 0 que ya tiene en la pila.
- Si al final de la cadena de entrada se vació la pila entonces acepta la cadena, de lo contrario la rechaza.

Esquema de un Autómata de Pila (AP)



AP

- El AP lee símbolos de izquierda a derecha de su cinta de entrada.
- En cada paso la máquina desempila el símbolo en el tope de la pila lee el símbolo en la cinta bajo la cabeza lectora y en base al estado actual puede empilar una secuencia de símbolos en la pila, mover la cabeza lectora a la derecha y entrar en un nuevo estado de acuerdo a la regla de transición.
- También permite transiciones-ε en las que puede desempilar y empilar símbolos sin leer el siguiente símbolo de entrada de su cinta (o sin mover la cabeza lectora).
- Aunque puede almacenar una cantidad ilimitada de información en su pila, no puede leer símbolos de la misma en orden arbitrario, primero debe desempilar los símbolos situados en la parte superior (tope) de la misma, en cuyo caso se pierden.
- Por lo que el acceso a la información de la pila es limitada.

Definición AP no determinístico

- Un AP no determinístico es una tupla:
- M=< Q, ∑, Γ, δ, q0, ⊥ , F >
- Donde:
  - Q *es un conjunto finito de estados,*
  - Σ *es el alfabeto de entrada (cinta),*
  - Γ *es el alfabeto de la pila,*
  - $\delta \subseteq (Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times \Gamma^*)$ *es la relación de transición,*
  - $q_0 \in Q$ *es el estado inicial,*
  - $\perp \in \Gamma$ *es el símbolo inicial de la pila,*
  - $F \subseteq Q$ *es el conjunto de estados de aceptación.*

Transiciones

- La transición

$$((p, a, A), (q, B_1 B_2 \ldots B_k)) \in \delta$$

- Significa que estando la máquina en el estado p leyendo el símbolo de entrada a en la cinta y estando A en el tope de la pila, puede desempilar A y empilar a los símbolos B1B2…Bk (B1 en el tope y Bk al fondo), mover su cabeza lectora una celda a la derecha de a y entrar en el estado q.

Transición

La forma habitual de describir una *transición* es

$$\delta(p, a, A) = \{(q_1, \gamma_1), (q_2, \gamma_2), \ldots, (q_m, \gamma_m)\}$$

donde $q_i \in Q$ $(1 \leq i \leq m)$, $\gamma_i \in \Gamma^*$ (i.e., es una serie de símbolos del alfabeto de la pila). La transición indica que estando A en el tope de la pila y leyendo el símbolo a en la cinta, puede entrar en cualquier estado q_i, desempilar A y empilar en su lugar la secuencia γ_i y avanzar una celda a la derecha la cabeza lectora. Obviamente no se permite elegir un estado p_i y una secuencia γ_j para $i \neq j$ en un movimiento.

- Por otra parte, la descripción de transición.

$$((p, \epsilon, A), (q, B_1 B_2 \ldots B_k)) \in \delta$$

- Significa que estando la máquina en el estado p con A en el tope de la pila, a continuación desempila A y empila $B_1B_2\ldots B_k$ (B_1 primero y B_k al final) deja su cabeza lectora donde está (no la mueve) y entra al estado q. Una manera equivalente de denotar la transición anterior es

$$(p, \epsilon, A) \xrightarrow{\delta} (q, B_1 B_2 \ldots B_k)$$

- En general

$$\delta(p, \epsilon, A) = \{(q_1, \gamma_1), (q_2, \gamma_2), \ldots, (q_m, \gamma_m)\}$$

- Para todo 1 <= i <= m.

Configuración

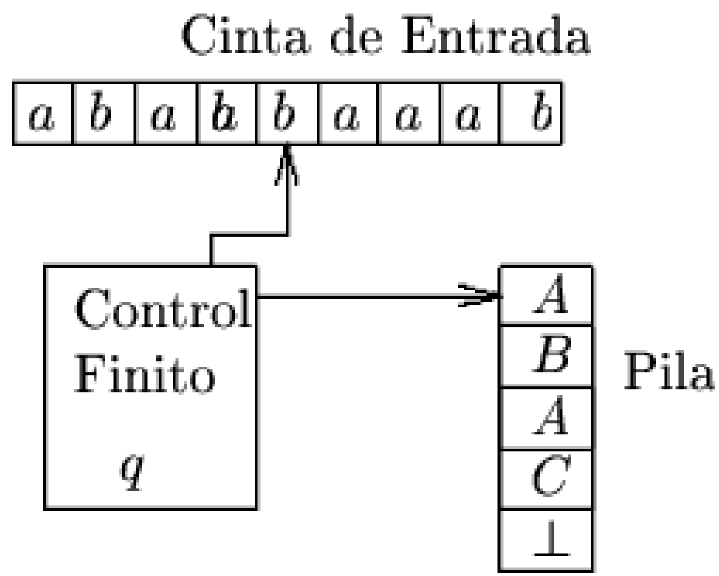
- Una configuración ó descripción instantánea de un AP registra el estado actual (un elemento de $Q \times \Sigma^* \times \Gamma^*$), así como la porción de la cinta de entrada aún no leída (todo lo situado a la derecha de la cabeza lectora) y el contenido actual de la pila.
- Una configuración ofrece información completa del estado global de la máquina en un instante durante un cómputo.

Ejemplo

- La configuración

$$(q, baaab, ABAC\perp)$$

corresponde a la que se muestra en la Figura



- Configuración inicial

$$(q_0, w, \perp)$$

- La máquina siempre inicia en el estado inicial q0 con la cabeza lectora apuntando al símbolo de w situado más a la izquierda y la pila vacía.

Definición Siguiente configuración

- La relación Siguiente Configuración indicada por →M describe cómo el AP M cambia de una configuración a otra en un solo paso, solamente hay dos casos:

1. *Si se tiene registrada la transición*

$$\big((p, a, A), (q_i, \gamma_i)\big) \in \delta$$

entonces para cualquier $w \in \Sigma^*$ *y* $\beta \in \Gamma^*$,

$$(p, aw, A\beta) \longrightarrow_M (q_i, w, \gamma_i\beta)$$

2. *y si es el caso que*

$$\big((p, \epsilon, A), (q_i, \gamma_i)\big) \in \delta$$

entonces para cualquier $w \in \Sigma^*$ *y* $\beta \in \Gamma^*$,

$$(p, w, A\beta) \longrightarrow_M (q_i, w, \gamma_i\beta)$$

Definición 7.1.3 ($\xrightarrow{*}_M$) *Se define la Cerradura Reflexiva y Transitiva de* $\longrightarrow_M$, *denotada por* $\xrightarrow{*}_M$ *como sigue: para* C, D, E *configuraciones*

$$
\begin{aligned}
C \xrightarrow{0}_M D &\quad \overset{def}{\Longleftrightarrow} \quad C = D,\\
C \xrightarrow{n+1}_M D &\quad \overset{def}{\Longleftrightarrow} \quad \exists E, C \xrightarrow{n}_M E \text{ y } E \xrightarrow{1}_M D,\\
C \xrightarrow{*}_M D &\quad \overset{def}{\Longleftrightarrow} \quad \exists n \geq 0, C \xrightarrow{n}_M D.
\end{aligned}
$$

Aceptación

Definición 7.1.4 (Por Estado Final) *Sea un AP* $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \perp, F \rangle$, *se define al lenguaje* $L(M)$ *aceptado por* M *por estado final como*

$$L(M) \stackrel{def}{=} \{w \mid (q_o, w, \perp) \xrightarrow{*}_M (p, \epsilon, \gamma)\}$$

es decir, M *acepta la entrada* w *si entra en un estado de aceptación* $p \in F$ *después de haber examinado completamente a* w $(\gamma \in \Gamma^*)$.

Aceptación

Definición 7.1.5 (Por Vaciado de Pila) *Se define $N(M)$ el lenguaje aceptado por M por vaciado de pila (ó pila nula) como*

$$N(M) \stackrel{def}{=} \{w \mid (q_o, w, \perp) \xrightarrow{*}_M (p, \epsilon, \epsilon)\}$$

es decir, M vacía la pila después de examinar la cadena de entrada completamente, sin importar en qué estado quede (i.e., $p \in Q$).

Ejemplo

- AP que reconoce el lenguaje de paréntesis balanceados '(', ')' por vaciado de pila:

$$M = \langle \{q_o\}, \{`(', `)'\}, \{A, \perp\}, \delta, q_o, \perp, \{\}, \rangle$$

$$\begin{array}{ll}
i) & \delta(q_o, `(', \perp) = \{(q_o, A\perp)\} \\
ii) & \delta(q_o, `(', A) = \{(q_o, AA)\} \\
iii) & \delta(q_o, `)', A) = \{(q_o, \epsilon)\} \\
iv) & \delta(q_o, \epsilon, \perp) = \{(q_o, \epsilon)\}
\end{array}$$

Computo

$$M = \langle \{q_o\}, \{`(', `)'\}, \{A, \perp\}, \delta, q_o, \perp, \{\}, \rangle$$

$$\begin{array}{ll}
i) & \delta(q_o, `(', \perp) = \{(q_o, A\perp)\} \\
ii) & \delta(q_o, `(', A) = \{(q_o, AA)\} \\
iii) & \delta(q_o, `)', A) = \{(q_o, \epsilon)\} \\
iv) & \delta(q_o, \epsilon, \perp) = \{(q_o, \epsilon)\}
\end{array}$$

- Cadena de entrada "(())()":

	Configuración			transición aplicada
	$(q_o,$	$(())(),$	$\perp)$	inicio
$\longrightarrow_M$	$(q_o,$	$())(),$	$A\perp)$	$i)$
$\longrightarrow_M$	$(q_o,$	$))(),$	$AA\perp)$	$ii)$
$\longrightarrow_M$	$(q_o,$	$)(),$	$A\perp)$	$iii)$
$\longrightarrow_M$	$(q_o,$	$(),$	$\perp)$	$iii)$
$\longrightarrow_M$	$(q_o,$	$),$	$A\perp)$	$i)$
$\longrightarrow_M$	$(q_o,$	$\epsilon,$	$\perp)$	$iii)$
$\longrightarrow_M$	$(q_o,$	$\epsilon,$	$\epsilon)$	$iv)$

Ejercicio

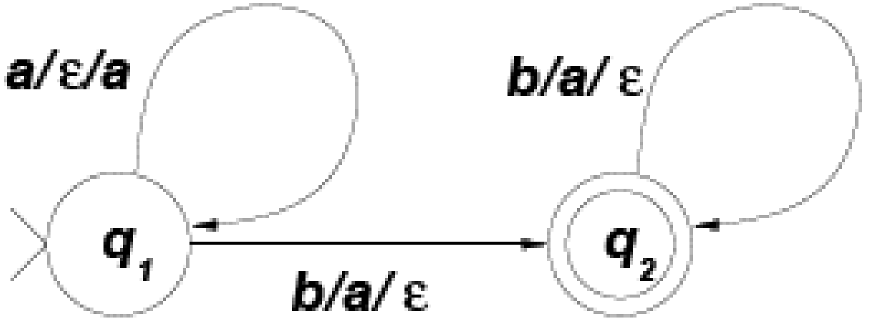
- Construir el AP para $a^n b^n$

Solución

- Construir el AP para $a^n b^n$

Ejercicio

- Construir el AP para $a^n b^n$
- Nota: mejorar propuesta de solución



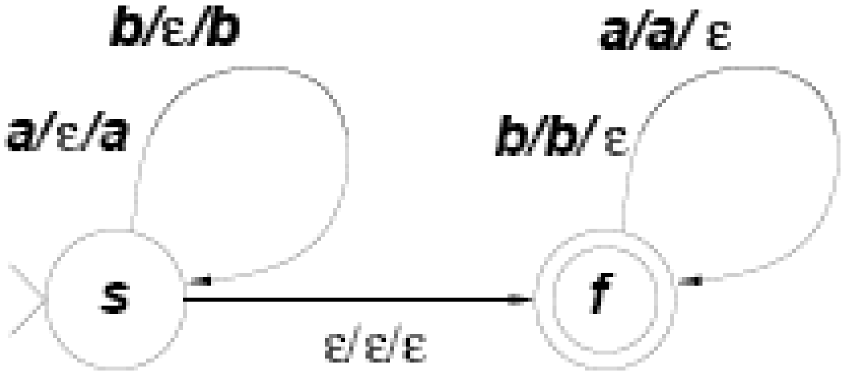
Estado	Por leer	Pila
q	$aabb$	ε
q	abb	a
q	bb	aa
q	b	a
q	ε	ε

Ejercicio

- Construye un AP para el conjunto $\{ww^R\}$

Solución

- Construye un AP para el conjunto $\{ww^R\}$
- Nota: mejorar propuesta de solución



Estado	Falta leer	Pila	Transición
s	$abba$	ε	
s	bba	a	1
s	ba	ba	2
f	ba	ba	3
f	a	a	5
f	ε	ε	4

Ejercicios

- Encuentra un AP para cada uno de los siguientes lenguajes:
  - $\{ab^ncd^n \mid n >= 0\}$
  - Todas las cadenas de {a,b} con el mismo número de a's y b's.
  - $\{wcw^R \mid w \{a,b\}^* \}$
  - $\{a^nb^{n+2} \mid n>=0\}$
- Encuentra un AF que por estado final obtenga:
  - $\{a^nb^n \mid n>=0\}$
  - $\{a^nb^{2n} \mid n >=0\}$