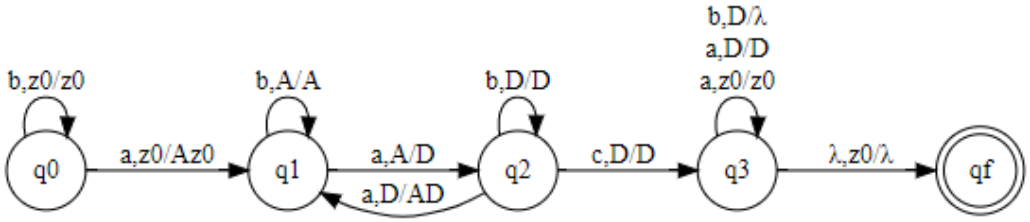
5. Autómatas de Pila.

Definición General.

Es una séptupla $AP =< \Sigma, \Gamma, Q, S, q_0, \delta, \ F >$ donde:
- Σ , alfabeto de entrada
$-\Gamma$, alfabeto de entrada
- Q , conjunto finito de estados
-$S \in \Gamma$, es el símbolo inicial de la pila (un único símbolo)
- $q_o \in Q$, estado inicial del autómata.
- $F \subseteq Q$, conjunto de estados finales o de aceptación.
- δ , función de transición $\delta: Q \times \{\Sigma \cup \{\lambda\}\} \times \Gamma \rightarrow P(Q \times \Gamma^*)$, donde $P(Q \times \Gamma^*)$denota todos los subconjuntos de $Q \times \Gamma^*$

Ejemplo: $A =< \{q_0, q_1, q_2, q_3, q_f\}, \{a, b, c\}, \{z_0, A, D\}, \delta, q_0, z_0, \{q_f\} >$



Las transiciones son:
$\delta(q_0, a, z_0) = \{(q_1, Az_0)\}$
$\delta(q_0, b, z_0) = \{(q_0, z_0)\}$

$\delta(q_1, a, A) = \{(q_2, D)\}$
$\delta(q_1, b, A) = \{(q_1, A)\}$

$\delta(q_2, a, D) = \{(q_1, AD)\}$
$\delta(q_2, b, D) = \{(q_2, D)\}$
$\delta(q_2, c, D) = \{(q_3, D)\}$

$\delta(q_3, a, D) = \{(q_3, D)\}$
$\delta(q_3, a, z_0) = \{(q_3, z_0)\},$
$\delta(q_3, b, D) = \{q_3, \lambda)\}$
$\delta(q_3, \lambda, z_0) = \{(q_f, \lambda))\}$

Configuración o descripción del autómata.

Configuración instantánea.

La ***configuración instantánea*** es una descripción del autómata de pila en un momento dado.

Se denota con una terna $(q, \omega, \alpha) \in Q \times \Sigma^* \times \Gamma^*$.

Es decir: q es el estado actual, ω es la cadena que falta consumir, y α es el contenido actual de la pila.

Si $\omega = \lambda$, no queda nada más para analizar.

Configuración inicial.

Un autómata de pila comienza su funcionamiento con la ***configuración inicial***:

$$(q_0, \omega, S)$$

Es decir, en el estado inicial, al inicio de toda la palabra completa y con el símbolo inicial de pila en la pila.

Secuencia de configuraciones.

El símbolo↦ indica el movimiento válido entre dos configuraciones.

El movimiento $(q, a\omega, Z\alpha) \mapsto (r, \omega, \beta\alpha)$ es un movimiento válido siempre y cuando: $(r, \beta) \in \delta(q, a, Z)$, donde $a \in \Sigma \wedge Z \in \Gamma \wedge \omega \in \Sigma^* \wedge \alpha \in \Gamma^* \wedge \beta \in \Gamma^*$

El movimiento $(q, \omega, Z\alpha) \mapsto (r, \omega, \beta\alpha)$ es un movimiento válido siempre y cuando: $(r, \beta) \in \delta(q, \lambda, Z)$

En cada transición sólo es posible consultar el tope de la pila (esto significa que se extrae ese elemento).
Luego, puede agregarse un elemento, muchos, o ninguno a la pila.

Lenguaje aceptado por un Autómata de Pila.

Aceptación por estado final.

Un AP acepta una cadena de entrada por ***estado final*** si y sólo si partiendo de su configuración inicial, se llega a un estado final al momento de consumir la entrada.
Es decir:

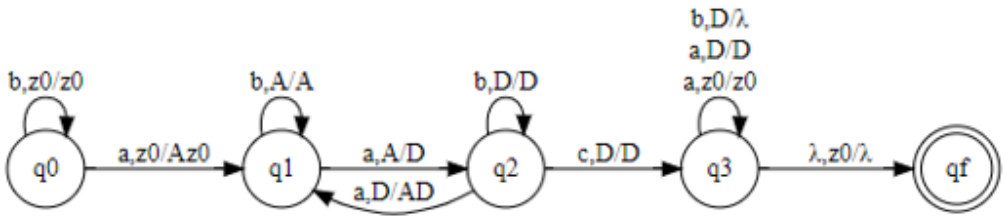
$$L_{pf}(A) = \{\omega \in \Sigma^* | (q_0, \omega, S) \mapsto^* (q_f, \lambda, \alpha)\}$$

Aceptación por pila vacía.

Un AP acepta una cadena de entrada por ***pila vacía*** si y sólo si partiendo de su configuración inicial, después de leerse toda la cadena se llega a un estado con la pila vacía (independientemente del tipo de estado en el que se encuentre el AP).
Es decir:

$$L_{pv}(A) = \{\omega \in \Sigma^* | (q_0, \omega, S) \mapsto^* (p, \lambda, \lambda)\}$$

Ejemplo: $A = < \{q_0, q_1, q_2, q_3, q_f\}, \{a, b, c\}, \{z_0, A, D\}, \delta, q_0, z_0, \{q_f\} >$



$$(q_0, aacb, z_0) \mapsto (q_1, acb, Az_0) \mapsto (q_2, cb, Dz_0) \mapsto (q_3, b, Dz_0) \mapsto (q_3, \lambda, z_0) \mapsto (q_f, \lambda, \lambda)$$

En este caso el autómata reconoce **aacb**, tanto por pila vacía como por estado final.

Equivalencia entre los lenguajes aceptados por pila vacía y los aceptados por estado final.

Un lenguaje L tiene un autómata a pila que lo acepta ***por estado final*** si y sólo si L tiene un autómata a pila que lo acepta ***por pila vacía***.

Importante: El autómata no necesariamente es el mismo.

Es decir, dado un autómata de pila, el lenguaje que ese autómata reconoce por pila vacía en general no es el mismo que el lenguaje que reconoce ese autómata de pila por estado final.

De pila vacía a estado final.

Teorema: Si L = $L_{pv}(P_v)$ para algún autómata de pila $P_v =< Q, \Sigma, \Gamma, \delta_v, q_0, S >$, entonces existe un autómata P_f, tal que L = $L_{pf}(P_f)$

Demostración:

Se construye $P_f =< \mathrm{Q} \cup \{\mathrm{p}_0, \mathrm{p}_\mathrm{f}\}, \Sigma, \Gamma \cup \{T\}, \delta_f, p_0, T, \{p_f\} >$

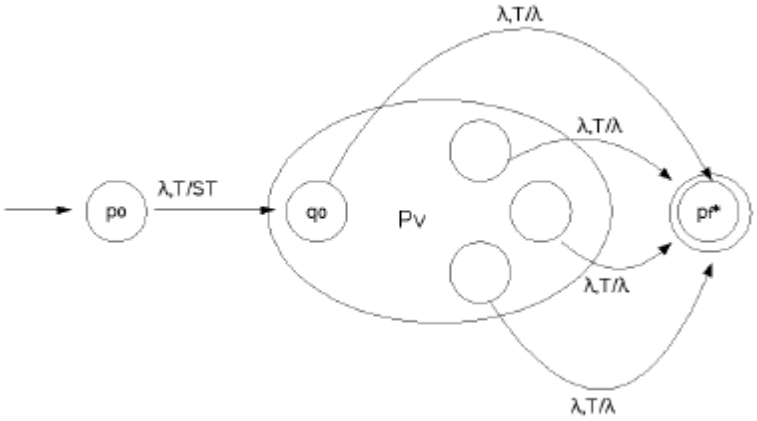
Donde la función de transición se define como:

1. $\delta_f(p_0, \lambda, T) = \{(q_0, ST)\}$
2. $\forall q \in Q, \forall a \in \Sigma \cup \{\lambda\}, \forall Y \in \Gamma: \delta_f(q, a, Y) = \delta_v(q, a, Y)$
3. $\forall q \in Q, \delta_f(q, \lambda, T) = (p_f, \lambda)$

Es decir: como primera transición, se coloca en la pila el estado inicial del autómata que acepta por pila vacía.

Luego, P_f simula P_v

Por último, se efectúan las transiciones necesarias para llegar al estado final, una vez vaciada la pila.



Ahora se demuestra que $\omega \in L_{pv}(P_v) \Leftrightarrow \omega \in L_{pf}(P_f)$.

1) $\omega \in L_{pv}(P_v) \Rightarrow \omega \in L_{pf}(P_f)$
   Como $\omega \in L_{pv}(P_v)$, eso significa que $(q_0, \omega, S) \mapsto^* (q, \lambda, \lambda)$ para algún q.
   Vamos a ver si $\omega \in L_{pf}(P_f)$.
   $$(p_0, \omega, T) \mapsto (q_0, \omega, ST) \mapsto^* (q, \lambda, T) \mapsto (p_f, \lambda, \lambda)$$
2) $\omega \in L_{pf}(P_f) \Rightarrow \omega \in L_{pv}(P_v)$
   Las reglas 1 y 3 nos proporcionan formas muy limitadas de aceptar a ω por estado final.
   La transición $\delta_f(p_0, \lambda, T) = \{(q_0, ST)\}$ sólo puede ser usada como primer paso en la secuencia de configuraciones, y no consume ningún símbolo de la cadena ω.
   Por otra parte, la transición $\forall q \in Q, \delta_f(q, \lambda, T) = (p_f, \lambda)$ exige que se haya consumido ya la cadena ω y que se tenga T en el tope de la pila, y eso sólo puede ocurrir si ya se sacó el símbolo S que es el que vacía la pila P_v, al tiempo que ya se consumió la entrada.
   Por lo tanto en $L(P_f)$ sólo tenemos cadenas que ya pertenecían a $L_{pv}(P_v)$ (porque las demás transiciones, son las mismas).

De estado final a pila vacía.

Teorema: Si L = $L_{pf}(P_f)$ para algún autómata de pila $P_f =< Q, \Sigma, \Gamma, \delta_f, q_0, S, F >$, entonces existe un autómata P_v, tal que L = $L_{pv}(P_v)$

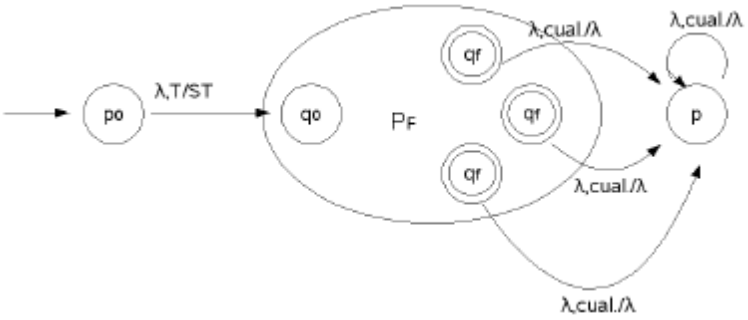
Demostración:

Se construye $P_v =< \mathrm{Q} \cup \{\mathrm{p}_0, \mathrm{p}\}, \Sigma, \Gamma \cup \{T\}, \delta_v, p_0, T >$

Donde la función de transición se define como:

1. $\delta_v(p_0, \lambda, T) = \{(q_0, ST)\}$
2. $\forall q \in Q, \forall a \in \Sigma \cup \{\lambda\}, \forall Y \in \Gamma, \delta_v(q, a, Y) = \delta_f(q, a, Y)$
3. $\forall q_f \in F, \forall Y \in \Gamma \cup \{T\}, \delta_v(q_f, \lambda, Y) = (p, \lambda)$

4. $\forall Y \in \Gamma \cup \{T\}, \delta_v(p, \lambda, Y) = (p, \lambda)$



Es decir: como primera transición, se coloca en la pila el estado inicial del autómata que acepta por estado final.
Luego, $P_v$simula P_f
Por último, se efectúan las transiciones necesarias para vaciar la pila en el estado p, no pudiendo consumirse ningún otro símbolo de la entrada.
Luego se demuestra que $\omega \in L_{pf}(P_f) \Leftrightarrow \omega \in L_{pv}(P_v)$.
(La demostración es similar a la anterior).

Autómata de Pila Determinístico.

Si bien los autómatas de pila pueden ser no determinísticos, el caso de los deterministas es bastante importante ya que los analizadores sintácticos se comportan como autómatas de pila deterministas.
Los autómatas de pila deterministas definen los tipos de construcciones que se pueden utilizar en lenguajes de programación.

Definición de APD.

Un AP $< Q, \Sigma, \Gamma, \delta, q_0, z_0, F >$ es ***determinista*** si se cumplen las siguientes condiciones:

1.$\forall q \in Q, a \in \Sigma \cup \{\lambda\}, X \in \Gamma$: $\delta(q, a, X)$ tiene como máximo un elemento.

2.Si $\delta(q, a, X)$ no está vacío para algún $a \in \Sigma$, entonces $\delta(q, \lambda, X) = \emptyset$

Lenguajes regulares y APD.

Todos los lenguajes regulares pueden ser aceptados por un APD.

Construcción:

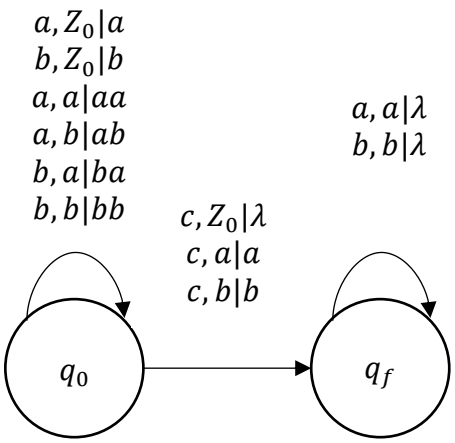
Sea $A =< Q, \Sigma, \delta_A, q_0, F >$ un AFD.
Construimos un APD $P =< Q, \Sigma, \{Z_0\}, \delta_P, q_0, Z_0, F >$ donde:
$\delta_P(q, a, Z_0) = \{(p, Z_0)\}$, $\forall p \in Q, q \in Q | \delta_A(q, a) = p$
Establecemos que $(q_0, \omega, Z_0) \mapsto^* (p, \lambda, Z_0) \Leftrightarrow \widehat{\delta_A}(q_0, \omega) = p$.
Este autómata de pila acepta por estado final, y coincidiendo los estados finales de P con los de A.
Luego, se demuestra por inducción sobre $|\omega|$ que $L(A) = L(P)$.

Lenguajes no regulares y APD.

Un APD puede reconocer lenguajes que no son regulares.

Ejemplo:

Por ejemplo: para el lenguaje no regular $L = \{\alpha \boldsymbol{c} \alpha^r | \alpha \in \{a, \mathrm{b}\}^*\}$
El APD es: $< \{q_0, q_f\}, \{a, b, c\}, \{a, b, Z_0\}, \delta, q_0, Z_0, \{q_f\} >$



Sin embargo, para el lenguaje no regular $L = \{\alpha\alpha^r | \alpha \in \{a, \mathrm{b}\}^*\}$ no se puede encontrar un Autómata de pila determinístico (aunque sí un existe un autómata de pila que lo reconoce)

Existen lenguajes independientes del contexto que no pueden ser reconocidos por un Autómata de Pila Determinístico (APD), pero sí por un Autómata de Pila.

Por lo anterior, podemos describir la siguiente relación:

$$L_R \subset L_{APD} \subset L_{AP}$$