Teoría de la Computación y Lenguajes Formales

Unidad II: Lenguajes Libres de Contexto

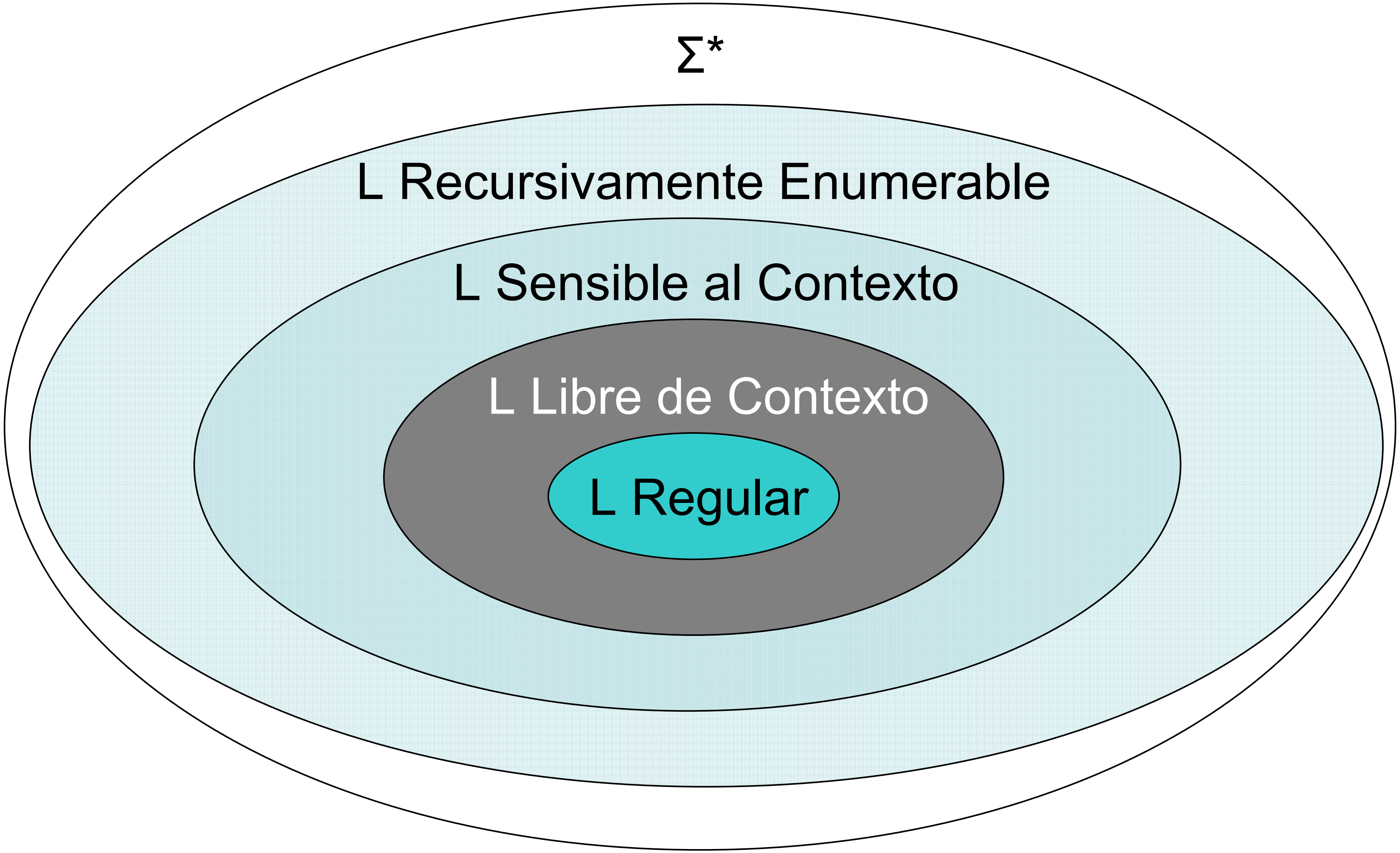
Autómatas de Pila

Prof. Hilda Y. Contreras
Departamento de Computación
hyelitza@ula.ve

Contenido

- Autómatas con Pila: definición
- Configuraciones y Computación
- Lenguajes independientes del contexto.
- Tipos de Automátas con Pila: Pila Vacía y Estado final.
- Equivalencia de autómatas con Pila y GLC (Gramáticas Libres de Contexto).
- Autómatas con pila deterministas.

Lenguaje Libre de Contexto



Lenguaje Libre de Contexto y GLC

- Jerarquía de Chomsky (Lenguaje Regular - Tipo 3)

Tipo	Lenguaje	Máquina	Gramática
0	Recursivamente enumerable	Máquina de Turing	Sin restricciones
1	Dependiente del Contexto	Autómata linealmente acotado	*Gramática dependiente del contexto* $\alpha A\beta \rightarrow \alpha\gamma\beta$
2	**Independiente del Contexto**	**Autómata de Pila**	***Gramática independiente de contexto*** $\boldsymbol{A \rightarrow \gamma, \gamma}$ **en (V U T)***
3	Lenguaje Regular	Autómata finito	*Gramática Regular* $A \rightarrow aB$ $A \rightarrow a$

Autómata de Pila

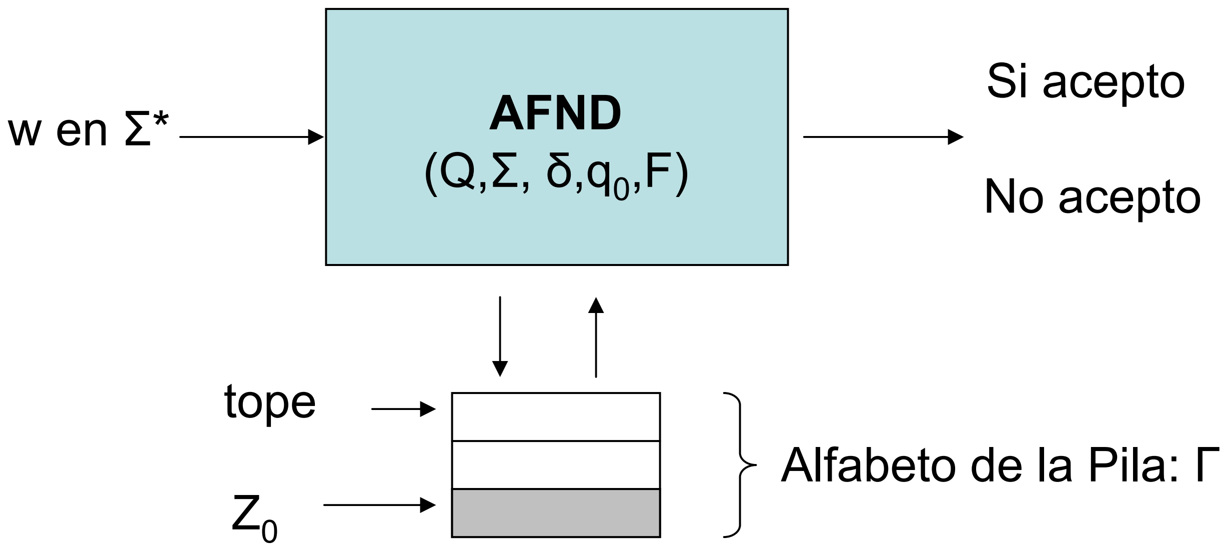
Extensión natural del Autómata Finito con:

Capacidad de memoria ilimitada, que va a determinar la potencia del autómata:

- En un solo sentido (Pila) → Autómata de Pila
- En ambos sentidos (Cinta) → Máquina de Turing

Autómata de Pila

Autómata con pila no determinista (AFND) + Pila: se apila y desapila en cada paso del autómata (cambio de estado)



Autómata de Pila

Autómata con pila no determinista (APND) es una septupla (Q, Σ, Γ, δ, q_0, Z_0, F):

Q es un conjunto finito de estados

Σ es un alfabeto de entrada

Γ es un alfabeto para la pila

δ es la función de transición

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow Q \times \Gamma^*$$

q_0 es el estado inicial

Z_0 es el símbolo inicial de la pila

F es el conjunto de estados finales

Ejemplo

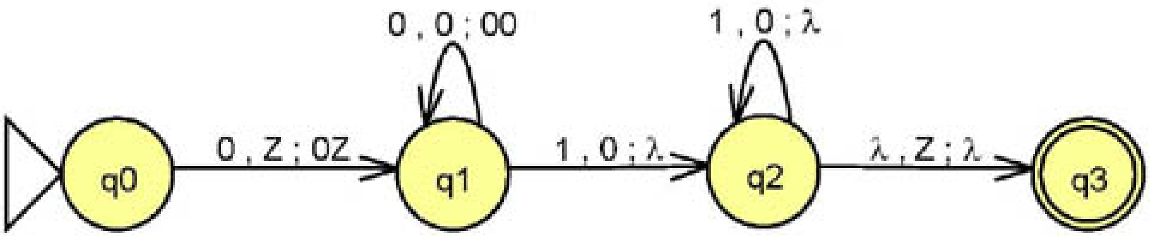
$$L_1 = \{ 0^n 1^n \mid n > 0 \}$$

La idea es que el autómata use la Pila para almacenar (apilar) los ceros de la cadena de entrada y luego el autómata va a leer de la cadena un uno por cada cero que obtenga del tope de la pila (desapilar). Si al terminar de procesar la cadena tampoco hay ceros entonces si esta en el lenguaje L_1

Ejemplo

$L_1 = \{ 0^n 1^n \mid n > 0 \}$

JFLAP: [0n1n.jff]



$L_2 = \{ 0^n 1^n \mid n \geq 0 \}$, $L_1 \neq L_2$

Descripción Instantánea

Se llama descripción instantánea o configuración de un autómata con pila a una tripleta:

$$(q, u, \alpha) \text{ que está en } Q \times \Sigma^* \times \Gamma^*$$

Donde q es el estado en el se encuentra el autómata, u es la parte de la cadena de entrada que queda por leer (Σ^*) y α el contenido de la pila Γ^* (donde el primer símbolo es el tope de la pila).

Paso de Computación

Se dice que de la configuración $(q, \boldsymbol{a}u, Z\alpha)$ se puede llegar mediante un paso de cálculo a la configuración $(p, u, \beta\alpha)$ y se escribe

$$(\boldsymbol{q}, \boldsymbol{a}u, \boldsymbol{Z}\alpha) \vdash (p, u, \beta\alpha)$$

si y solo si $\delta(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{Z}) = (p, \beta)$ donde ***a*** puede ser cualquier símbolo de entrada o la cadena vacía

Configuración

Si C_1 y C_2 son dos configuraciones, se dice que se puede llegar de C_1 a C_2 mediante una sucesión de pasos de cálculo y se escribe

$$\mathbf{C_1 \vdash^* C_2}$$

si y solo si existe una sucesión de configuraciones $T_1, \ldots, T_n$ tales que

$$\mathbf{C_1 = T_1 \vdash T_2 \vdash \ldots \vdash T_{n-1} \vdash T_n = C_2}$$

Si M es un APND y w en Σ*, se llama configuración inicial a:

$$\mathbf{(q_0, w, Z_0)}$$

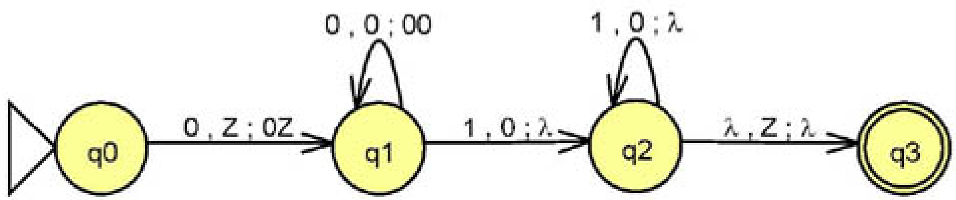
donde q_0 es el estado inicial y Z0 el símbolo inicial de la pila

Ejemplo

$$\mathbf{L_1 = \{ 0^n 1^n \mid n > 0\}}$$

$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow Q \times \Gamma^*$

$\delta(q_0,0,Z) = (q_1,0Z)$
$\delta(q_1,0,0) = (q_1,00)$
$\delta(q_1,1,0) = (q_2,\lambda)$
$\delta(q_2,1,0) = (q_2,\lambda)$
$\delta(q_2,\lambda,Z) = (q_3,\lambda)$



$(q_0,\mathbf{0}011,\mathbf{Z}) \vdash (q_1,\mathbf{0}11,\mathbf{0}Z) \vdash (q_1,\mathbf{1}1,\mathbf{0}0Z) \vdash$
$(q_2,\mathbf{1},\mathbf{0}Z) \vdash (q_2,\lambda,\mathbf{Z}) \vdash (q_3,\lambda,\lambda)$ Acepto.

Lenguaje Aceptado

- Existen dos criterios para determinar el lenguaje aceptado por un APND:

1) Lenguaje aceptado por estados finales

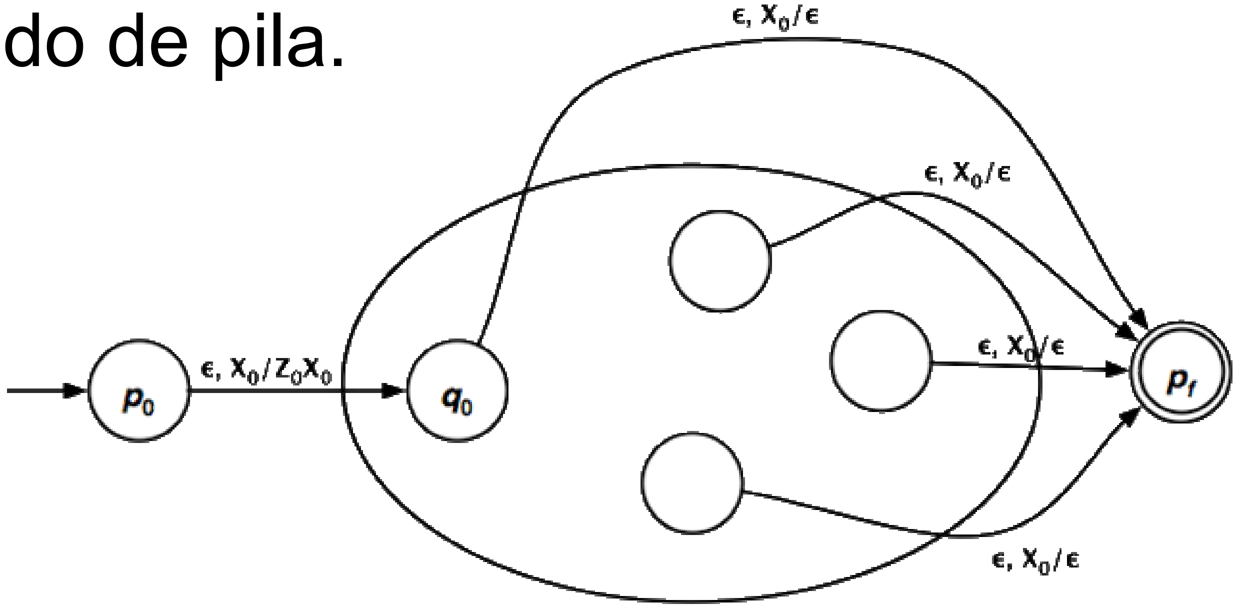
$\boldsymbol{L(M) = \{w \text{ en } \Sigma^* \mid (q_0,w,Z_0) \vdash^* (p,\lambda,\alpha),\ p \text{ en } F,\ \alpha \text{ en } \Gamma^* \}}$

2) Lenguaje aceptado por la pila vacía

$\boldsymbol{N(M) = \{w \text{ en } \Sigma^* \mid (q_0,w,Z_0) \vdash^* (p,\lambda,\lambda),\ p \text{ en } Q \}}$

De Pila Vacía a Estado Final

- El conjunto de los lenguajes que son aceptados por algún autómata a pila por estado final es el mismo que el conjunto de los lenguajes que son aceptados por algún autómata a pila por vaciado de pila.



De Pila Vacía a Estado Final

- La especificación del autómata a pila por estado final P_F es como sigue:

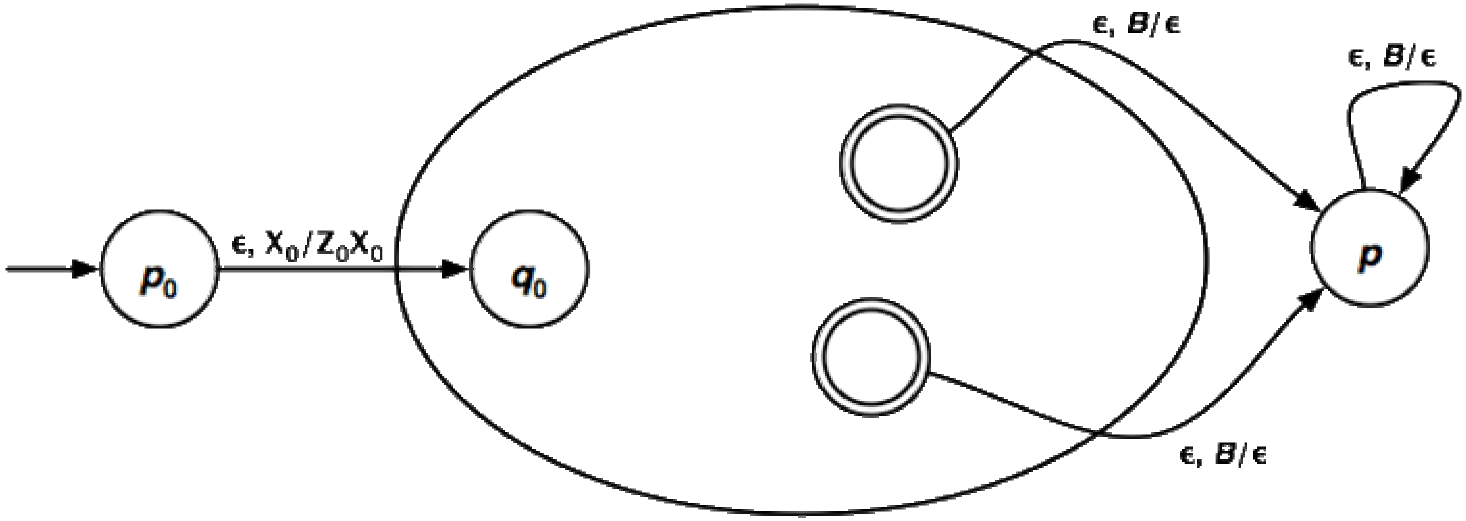
$$P_F = (Q \cup \{p_0, p_f\},\ \Sigma, \Gamma \cup \{X_0\}, \delta_f, p_0, X_0, \{p_f\})$$

- Donde δ_f es igual a δ mas:

1) $\delta_f(p_0, \lambda, X_0) = \{(q_0, Z_0X_0)\}$
2) $\delta_f(q, \lambda, X_0)$ contiene (p_f, λ) para todo estado q de Q

De Estado Final a Pila Vacía

- Dado un autómata a pila P_F que acepta un lenguaje L por estado final, se construye otro autómata a pila P_N que acepta L por pila vacía.



De Estado Final a Pila Vacía

- La especificación del autómata a pila por pila vacia P_N es como sigue:

$$P_N = (Q \cup \{p_0, p\},\ \Sigma, \Gamma \cup \{X_0\}, \delta, p_0, X_0, \Phi)$$

- Donde δ_n es igual a δ mas:

  1) $\delta_f(p_0, \lambda, X_0) = \{(q_0, Z_0X_0)\}$

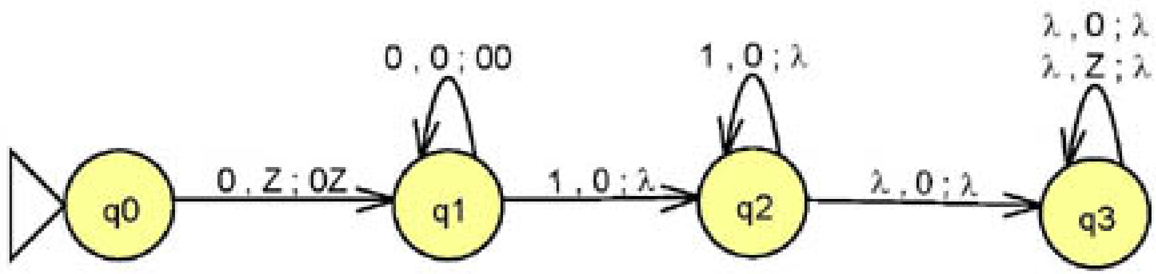
  2) Para todos los estados de aceptación q en F y símbolos de pila Y en Γ o $Y = X_0$, $\delta_n(q, \lambda, Y)$ contiene (p, λ)

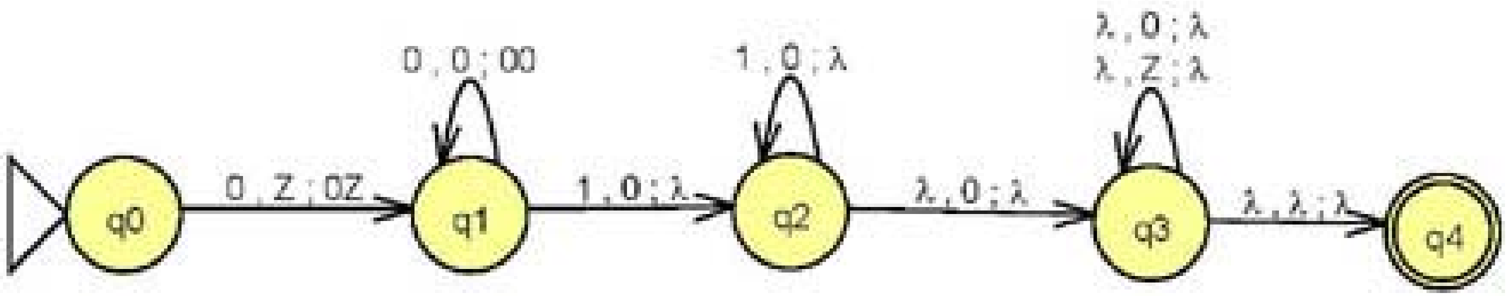
  3) Para todos los símbolos de pila Y en Γ o $Y = X_0$,

  $\delta(p, \lambda, Y) = \{(p, \lambda)\}$

De Pila Vacía a Estado Final

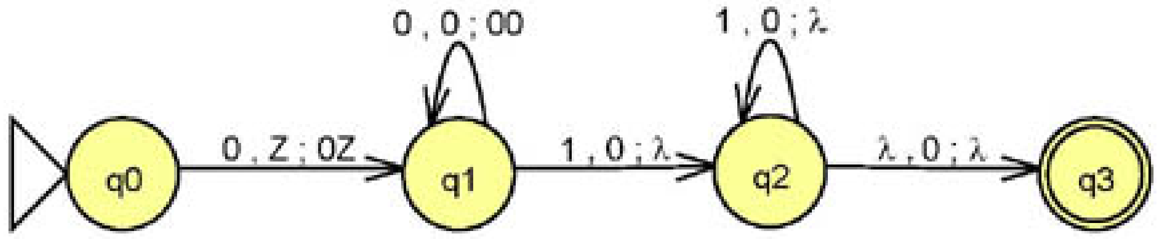
- Ejemplo: $\mathbf{L_3 = \{ 0^n 1^m \mid n > m \}}$

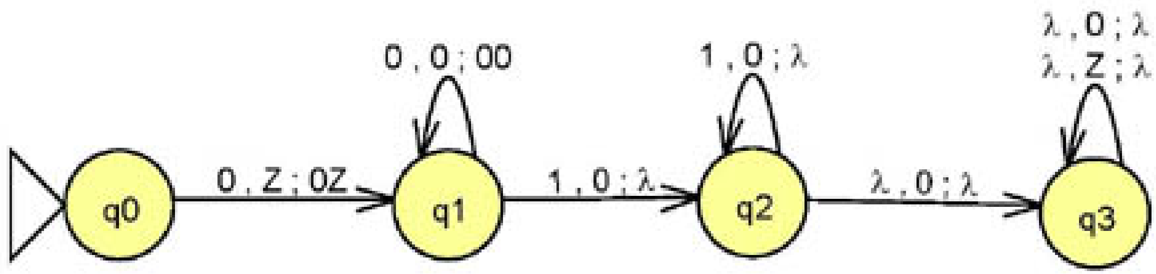




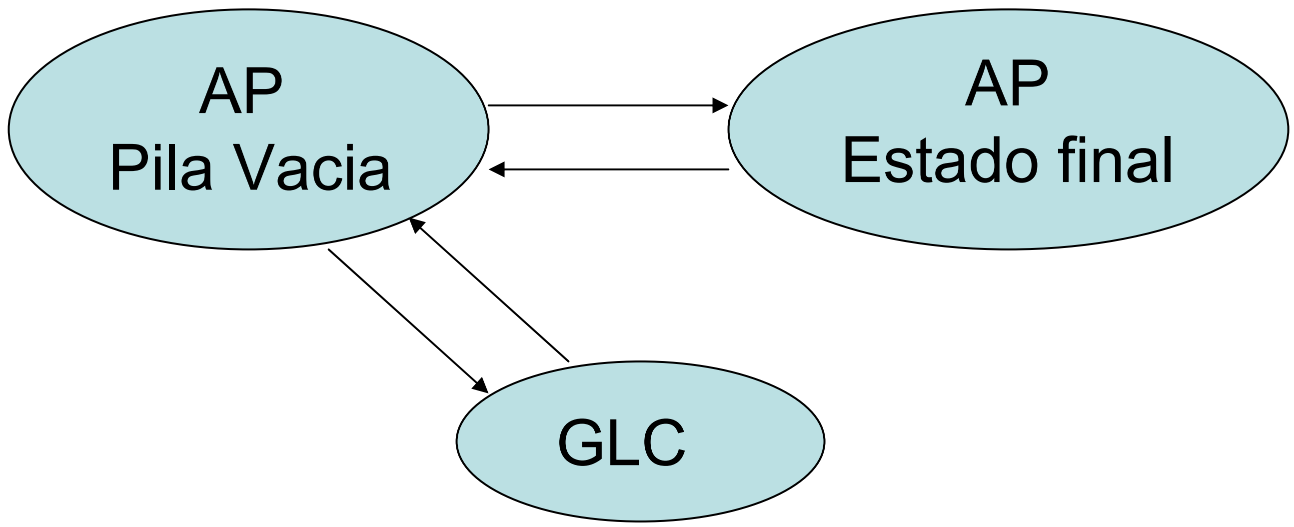
De Estado Final a Pila Vacía

- Ejemplo: $\mathbf{L_3 = \{ 0^n1^m \mid n > m\}}$





Relación GLC – AP



Autómata Pila	**Gramática Libre de Contexto**
Reconocer, Verificar	Generar

GLC - APND

Dada una gramática G = (V, T, P, S), se puede construir el autómata a pila P que acepta L(G) por pila vacía como sigue:

P = ({q}, T, V ∪ T, δ, q, S)

- Donde la función de transición δ se define mediante:

1) Para cada variable A,

δ(q, λ, A) = {(q, β)} si A → β es una producción de P}

2) Para cada símbolo terminal a, δ(q, a, a) = {(q, λ)}

GLC - APND

Dada una gramática G = (*V*, *T*, *P*, *S*), en la **FNG** se puede construir el autómata a pila P más simple que acepta L(G) por pila vacía como sigue:

P = ({*q*}, *T*, *V* U *T*, δ, *q*, *S*)

- Donde la función de transición δ se define:

δ(*q*, a, *A*) = {(*q*, *β*)} si A → aβ esta en P}

GLC - APND

$\delta(q, \text{a}, A) = \{(q, \beta)\}$ si A → aβ es una producción de P}

p.e. $\mathbf{L_1 = \{ 0^n1^n \mid n > 0\}}$, $L_1 = L(G)$,

$G = (\{A,S\},\{0,1\}, P, S)$, $P =\{S \rightarrow 0A \mid 0SA,\ A \rightarrow 1\}$

$AP = (\{q\},\{0,1\},\{S,A\},\delta,S,\Phi)$

1. $\delta(q, 0, S) = \{(q, A), (q, SA)\}$
2. $\delta(q, 1, A) = \{(q, \lambda)\}$

$(q,\mathbf{0}011,\mathbf{S}) \vdash_{1.b} (q,\mathbf{0}11,\mathbf{S}A) \vdash_{1.a} (q,\mathbf{1}1,\mathbf{A}A) \vdash_2$
$(q,\mathbf{1},\mathbf{A}) \vdash_2 (q, \lambda, \lambda)$ Acepto

APND - GLC

- Dado un AP $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi)$ cuyo lenguaje aceptado por pila vacía es $L(M)$ existe una GIC $G = (V, T, P, S)$ que genera $L(M)$:
  - $V = \{s\} \cup \{[pXq] \rightarrow p, q \text{ en } Q$ y X en $\Gamma\}$ donde S en $(T \cup \Gamma)$. El símbolo-tupla $[pXq]$ representa a las cadena w que cumplen $(q, w, X) \vdash (\text{p}, \lambda, \lambda)$
  - $S \rightarrow [q_0Z_0p]$ en P para todo p en Q
  - Si $\delta(q, a, X) = (r, Y_1Y_2...Y_k)$, entonces para todo $r_1, r_2, ..., r_{k+1}$ en Q, con a en $\Sigma \cup \{\lambda\}$ y $k \geq 0$, se tienen que $[qXr_{k+1}] \rightarrow a[r_1Y_1r_2][r_2Y_2r_3] ... [r_kY_kr_{k+1}]$ en P

Ejemplo: APND → GLC

$L_1 = \{ 0^n1^n \mid n > 0\}$, JFLAP: 0n1n.jff

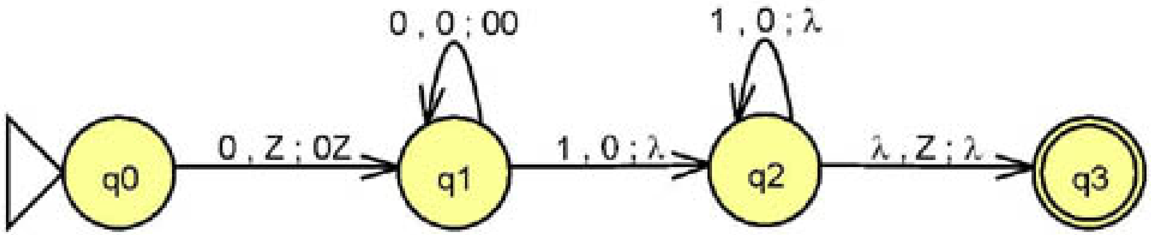
GLC = ({S,R,O,E}, {0,1},P,S)

S → 0ER

R → λ

O → 1

E → 0EO | 1



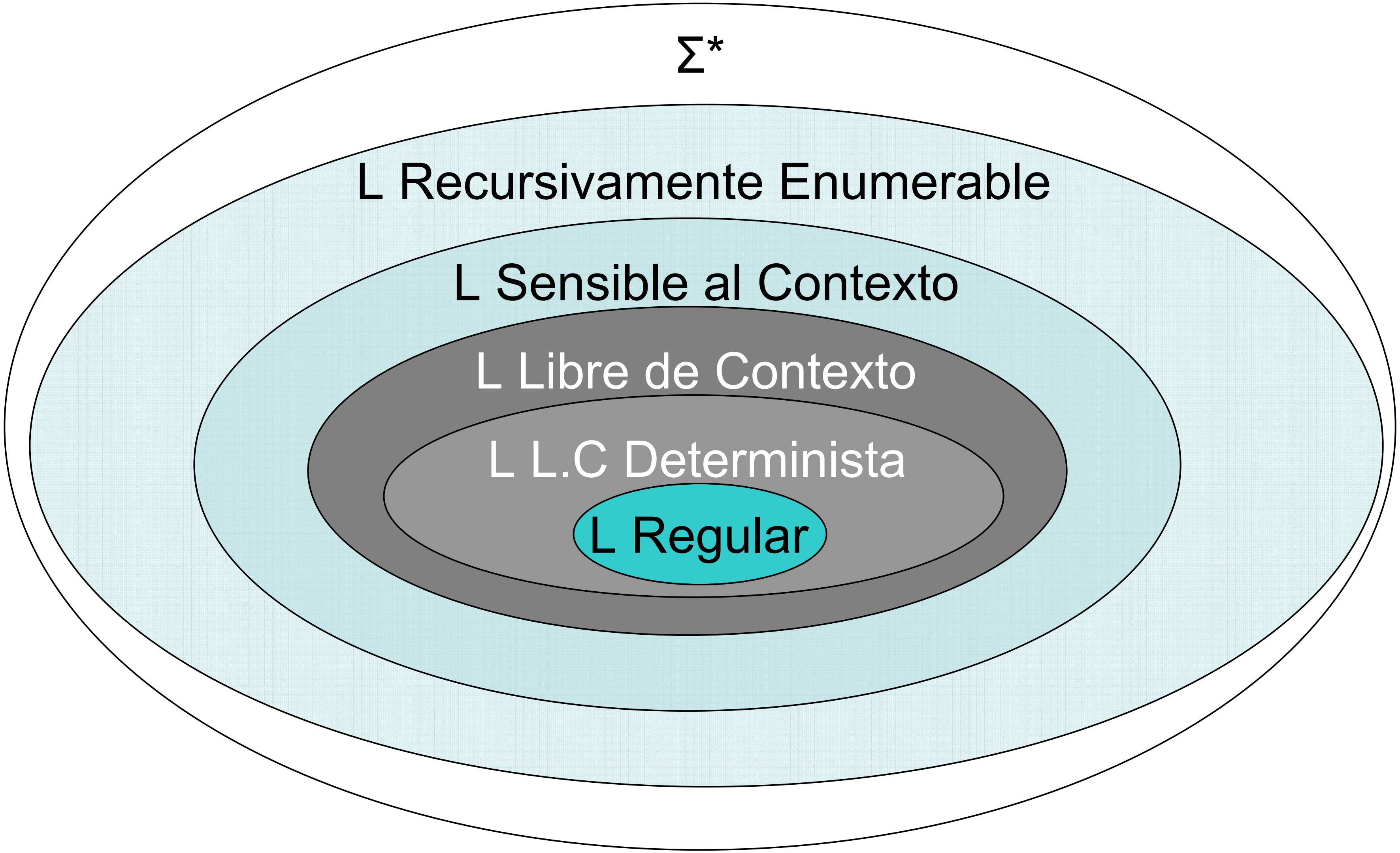
Autómatas de Pila Determinista

- Para todo q en Q, y Z en Γ, y a en Σ U {λ} entonces δ(q,a,Z) tiene un solo elemento
- LR ≤ L aceptados por APD ≤ LLC

  p.e. { ww^r | w en $(0+1)^*$ } es LLC

  p.e. { wcw^r | w en $(0+1)^*$ } es LLCD

Lenguaje Libre de Contexto



APD y Analizadores Sintácticos

Análisis sintáctico (buena formación de una cadena) se realiza con un AP en compiladores e intérpretes

Importancia: Normalmente de usan los APD porque tienen gramáticas No ambiguas. lenguajes formales (programación)

Dos tipos:

- Análisis ascendentes (top-down)
- Análisis descendentes (bottom-up)