

Autómatas con pila



Senén Barro Ameneiro, CiTIUS

@SenenBarro

Material elaborado fundamentalmente por
el profesor Manuel Mucientes Molina

Bibliografía

- J.E. Hopcroft, R. Motwani y J.D. Ullman, "Teoría de Autómatas, Lenguajes y Computación", Addison Wesley, 2008.
 - Capítulos 6 y 7
- P. Linz, "An Introduction to Formal Languages and Automata", Jones and Bartlett Publishers, Inc., 2001.
 - Capítulo 6-8

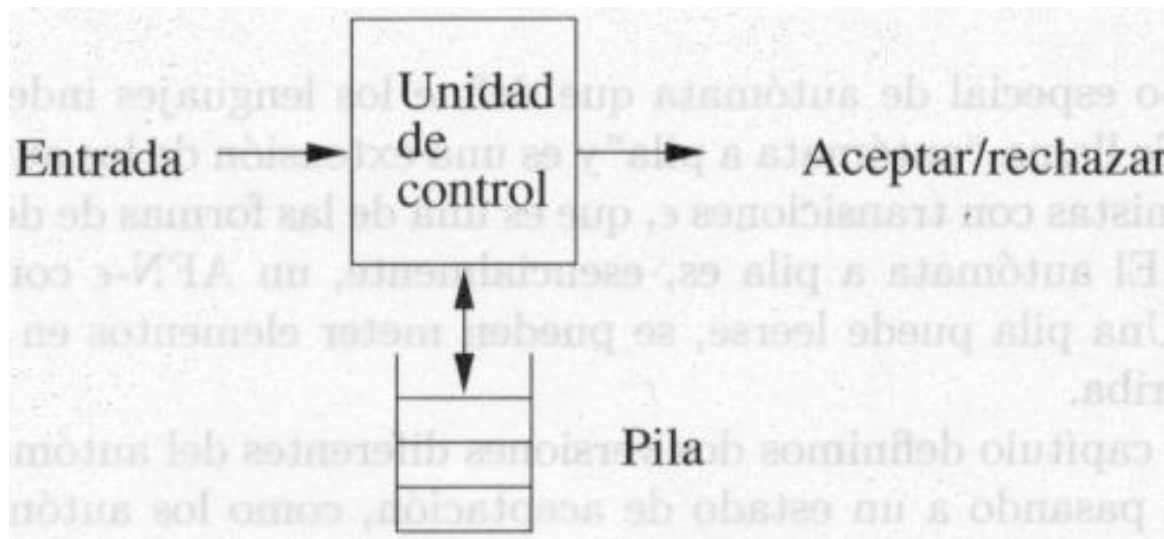
Introducción a los AP

- Un autómatata con pila (AP) es un AFN con transiciones ε y con una pila en la que se puede almacenar una cadena de “símbolos de pila”
- El AP puede recordar una cantidad **infinita** de información
 - LIFO
- Los AP reconocen todos los LIC y sólo estos
 - Existen lenguajes que no son LIC.
Ejemplo: $\{0^n 1^n 2^n \mid n \geq 1\}$

Introducción a los AP

Funcionamiento:

- Se consume de la entrada un símbolo o bien ε
- Se pasa a un nuevo estado
- Se reemplaza el símbolo en lo alto de la pila por una cadena (podría ser ε)



Introducción a los AP

Ejemplo: diseñar el AP para $L_{ww^R} = \{ww^R \mid w \text{ está en } (0+1)^*\}$

- se comienza en el estado q_0
 - suponemos que la cadena w aún no ha finalizado
 - se van almacenando los símbolos de entrada leídos en la pila
- en cualquier momento se puede suponer que w ha finalizado y se ha comenzado a leer w^R
 - el final de w estará en la cima de la pila
 - se transita al estado q_1
 - AP no determinista:
 - podemos suponer que hemos llegado al final de w
 - también podemos continuar en q_0 y seguir almacenando las entradas en la pila

Introducción a los AP

Ejemplo: diseñar el AP para el lenguaje $L_{ww^R} = \{ww^R \mid w \text{ está en } (0+1)^*\}$

Sigue de antes:

- en el estado q_1 se compara el símbolo de entrada con el símbolo en la cima de la pila
 - son iguales: se elimina el símbolo de la pila
 - son distintos: no habíamos llegado al final de w . Esa rama muere
- si se vacía la pila, hemos leído ww^R y se acepta la entrada

Definición formal del AP

$$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

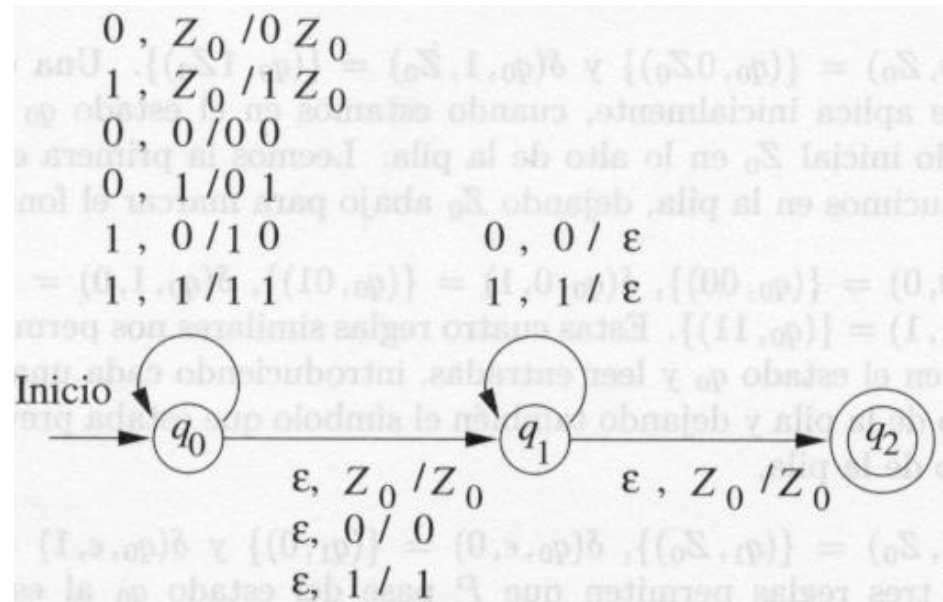
- Q : conjunto finito de estados
- Σ : conjunto finito de símbolos de entrada
- Γ : alfabeto de pila finito
- δ : función de transición, $\delta(q, a, X) = (p, \gamma)$
- q_0 : estado inicial
- Z_0 : símbolo inicial de la pila
- F : conjunto de estados de aceptación

Definición formal del AP

Ejemplo: diseñar un AP que acepte el lenguaje L_{WWR} :

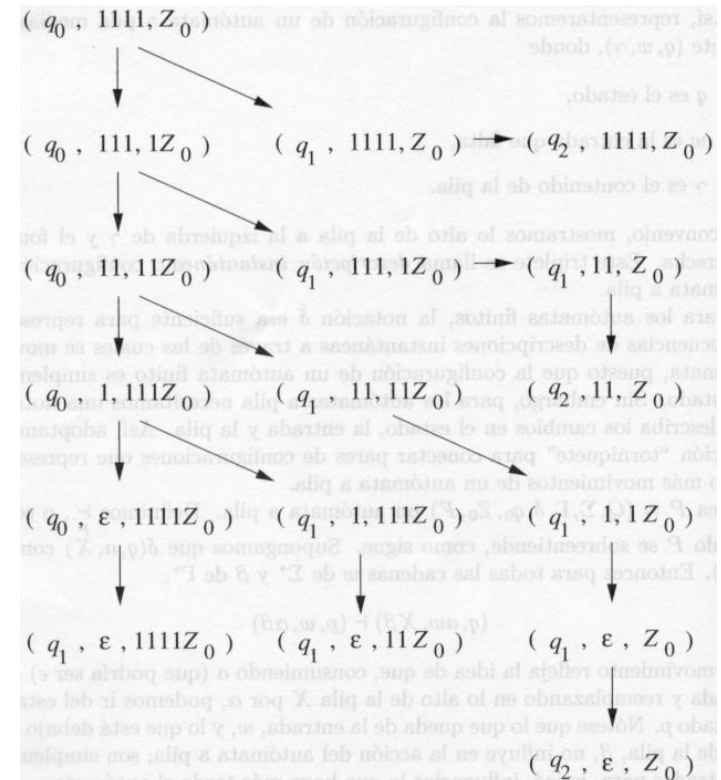
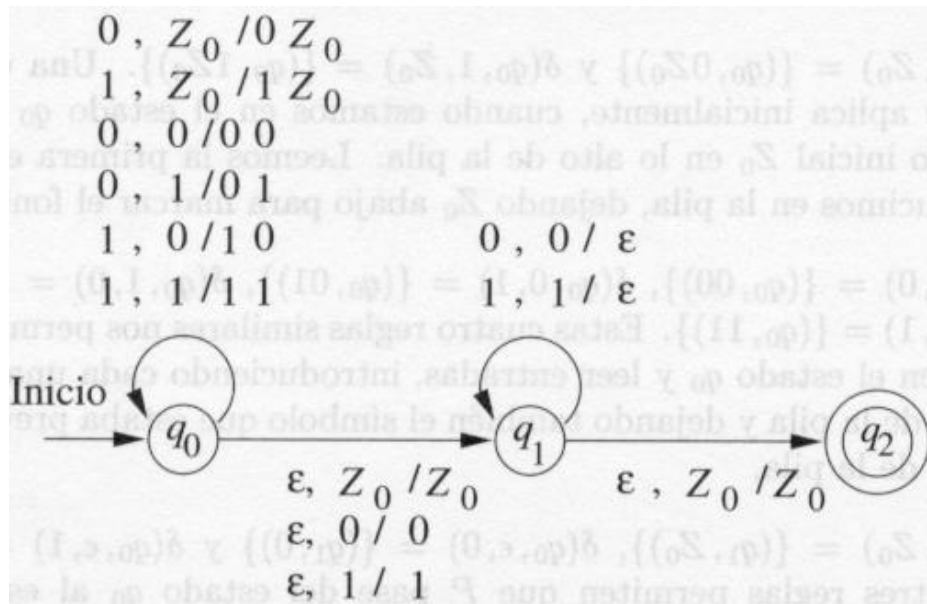
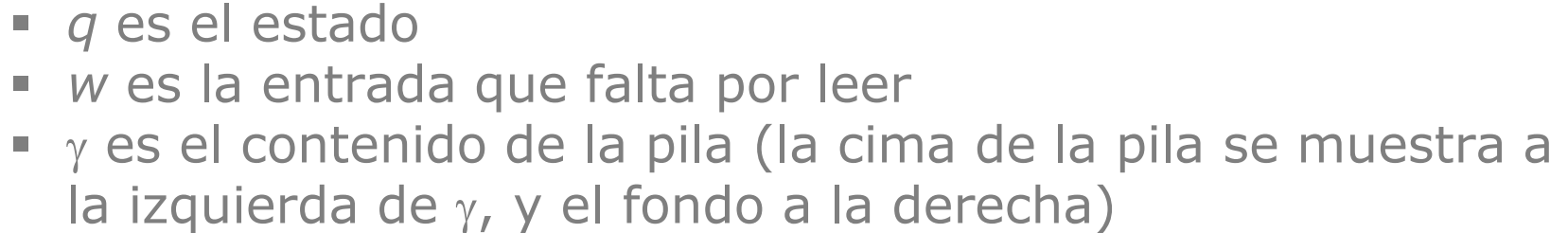
- $P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$, donde δ se define:

Q	S	G	Movimiento
q_0	0	Z_0	$(q_0, 0Z_0)$
q_0	1	Z_0	$(q_0, 1Z_0)$
q_0	0	0	$(q_0, 00)$
q_0	0	1	$(q_0, 01)$
q_0	1	0	$(q_0, 10)$
q_0	1	1	$(q_0, 11)$
q_0	e	Z_0	(q_1, Z_0)
q_0	e	0	$(q_1, 0)$
q_0	e	1	$(q_1, 1)$
q_1	0	0	(q_1, e)
q_1	1	1	(q_1, e)
q_1	e	Z_0	(q_2, Z_0)

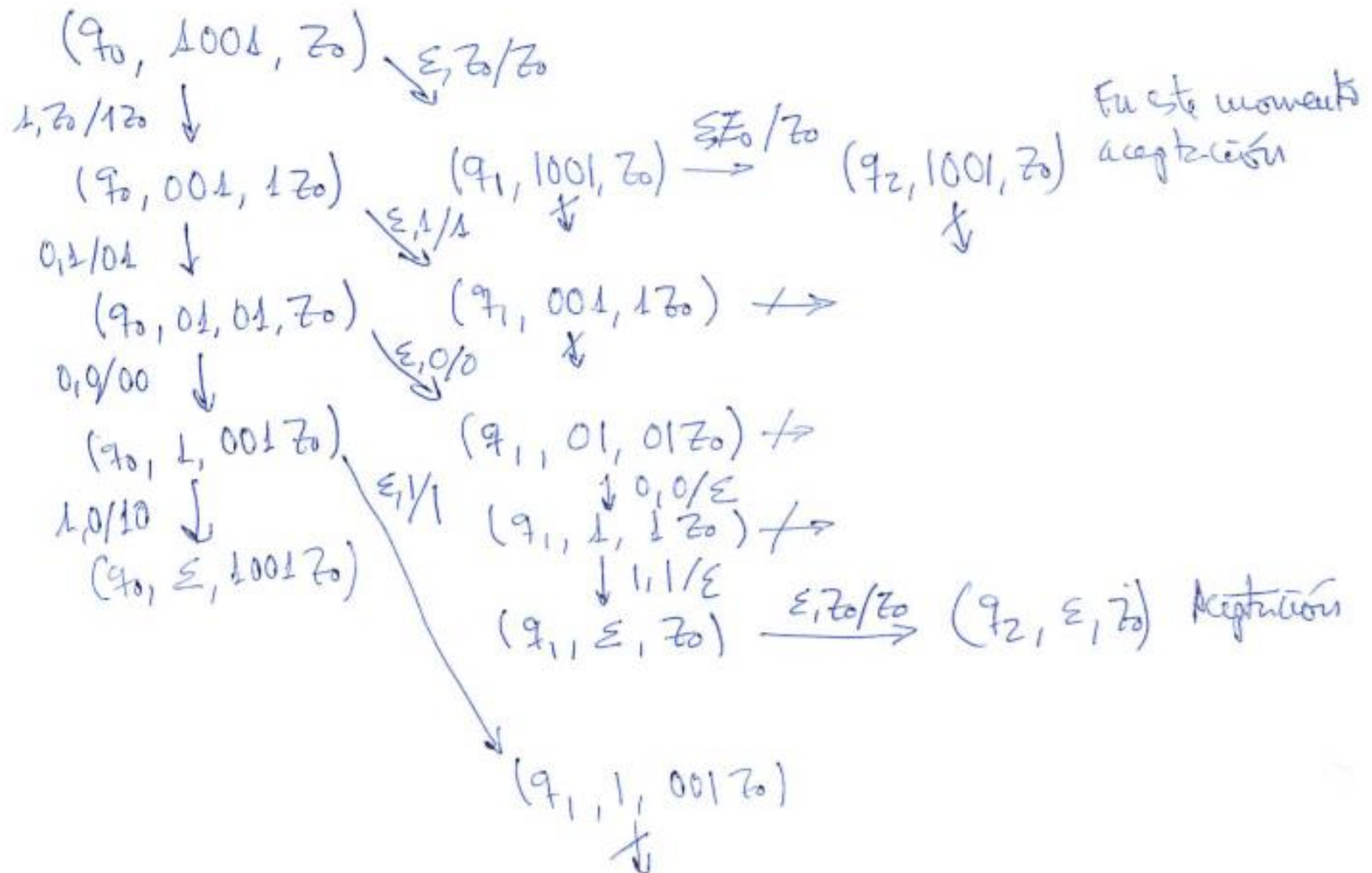


9

Ejemplo para la entrada 1111



Ejemplo para la entrada 1001



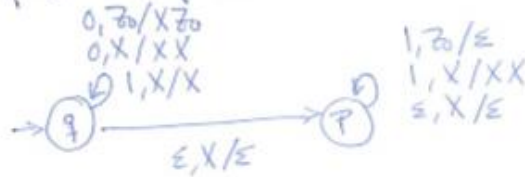
Problemas

Dado el AP $P = (\{q, p\}, \{0, 1\}, \{Z_0, X\}, \delta, q, Z_0, \{p\})$, donde δ se define en la siguiente tabla, mostrar las configuraciones alcanzables a partir de la inicial (q, w, Z_0) , para w igual a 01 y 010

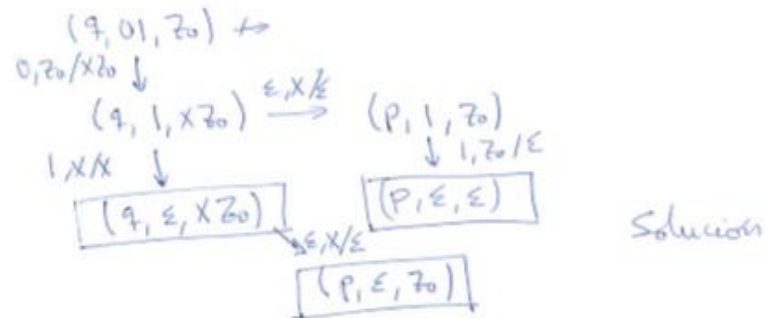
Q	Σ	Γ	Movimiento
q	0	Z_0	(q, XZ_0)
q	0	X	(q, XX)
q	1	X	(q, X)
q	ε	X	(p, ε)
p	ε	X	(p, ε)
p	1	X	(p, XX)
p	1	Z_0	(p, ε)

Solución

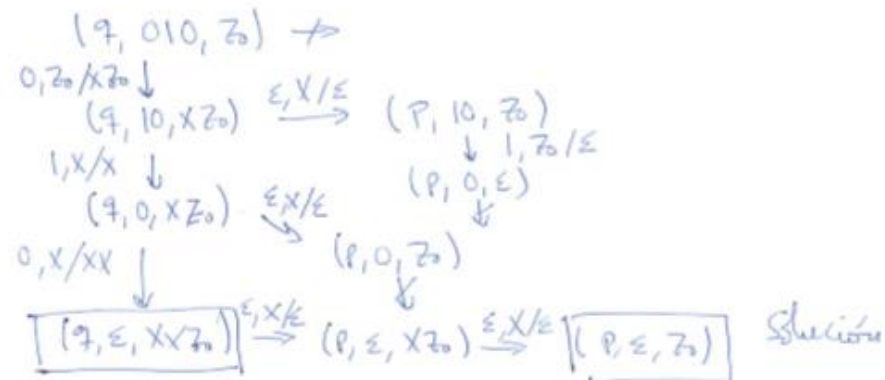
Problema pág. 11 capítulo 5-TALF



$w = 01$



$w = 010$



Problemas

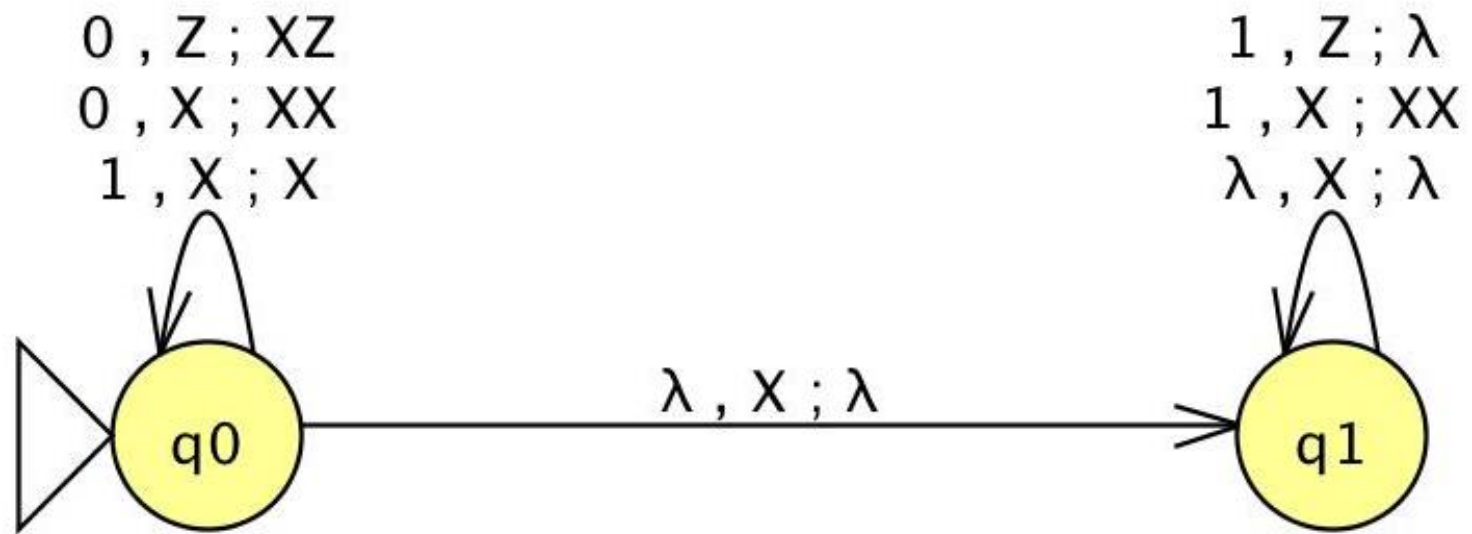
Dado el AP $P = (\{q, p\}, \{0, 1\}, \{Z_0, X\}, \delta, q, Z_0, \{p\})$, donde δ se define en la siguiente tabla, mostrar las configuraciones alcanzables a partir de la inicial (q, w, Z_0) , para w igual a 01 y 010

Q	Σ	Γ	Movimiento
q	0	Z_0	(q, XZ_0)
q	0	X	(q, XX)
q	1	X	(q, X)
q	ε	X	(p, ε)
p	ε	X	(p, ε)
p	1	X	(p, XX)
p	1	Z_0	(p, ε)

Soluciones:

- $(q, 01, Z_0) \vdash^* (p, \varepsilon, Z_0)$
 $\vdash^* (p, \varepsilon, \varepsilon)$
 $\vdash^* (q, \varepsilon, XZ_0)$
- $(q, 010, Z_0) \vdash^* (p, \varepsilon, Z_0)$
 $\vdash^* (q, \varepsilon, XXZ_0)$

Autómata de Pila



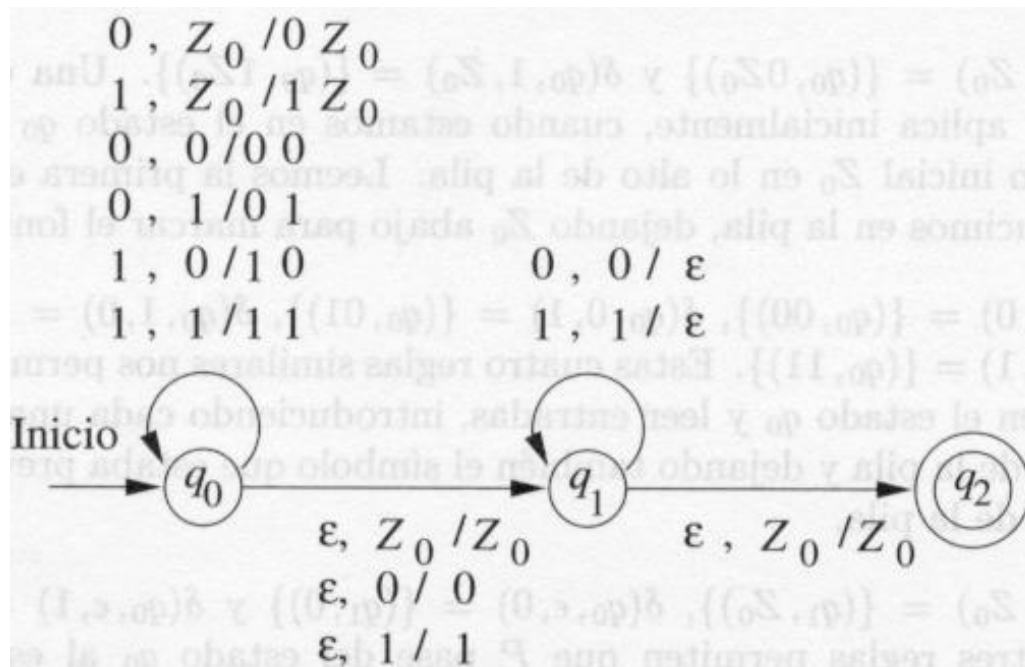
Correspondiente al AP de la página anterior

Lenguajes aceptados por un AP

- Dos tipos de aceptación
 - por estado final
 - por vaciado de pila
- Ambos métodos son equivalentes
- **Aceptación por estado final**
 - sea $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ un AP
 - $L(P) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \alpha)\}$
para algún estado q de F y cualquier cadena de pila α

Lenguajes aceptados por un AP

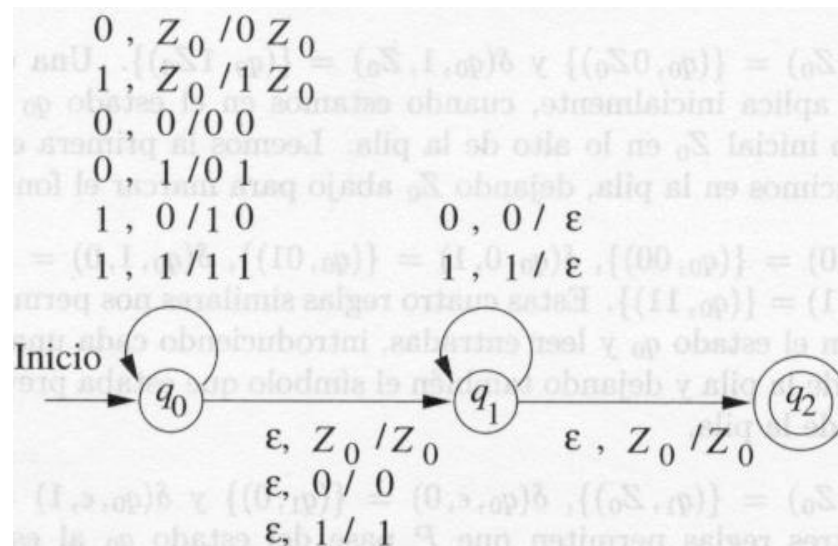
Ejemplo: el AP de la figura acepta cadenas x por estado final si y sólo si x tiene la forma ww^R



Lenguajes aceptados por un AP

Aceptación por pila vacía

- Para todo AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ se define el lenguaje que acepta como:
 $N(P) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)\}$ para cualquier estado q
- Ejemplo: modificar el AP de la figura para que reconozca por vaciado de pila
 - se cambia $\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$ por $\delta(q_1, \varepsilon, Z_0) = \{(q_2, \varepsilon)\}$
 - se transforma q_2 en no final

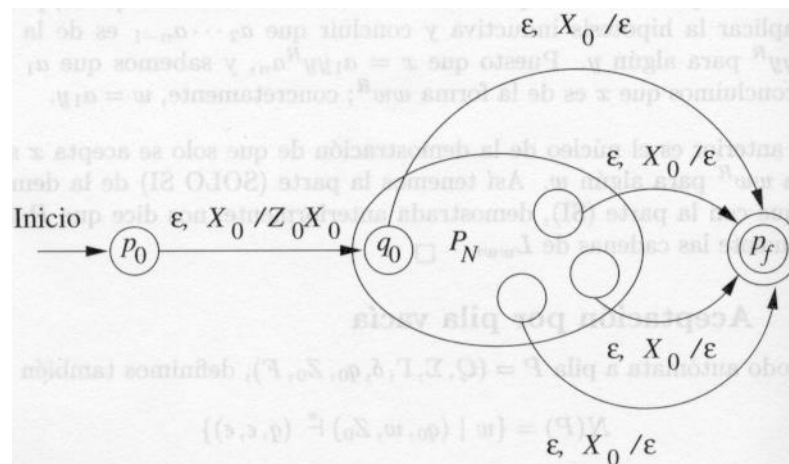


Conversión vaciado de pila a estado final

Teorema: si $L = N(P_N)$ para algún AP $P_N = (Q, \Sigma, \Gamma, \delta_N, q_0, Z_0)$, existe un AP P_F tal que $L = L(P_F)$

Prueba: $P_F = (Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, p_0, X_0, \{p_f\})$, donde δ_F se define:

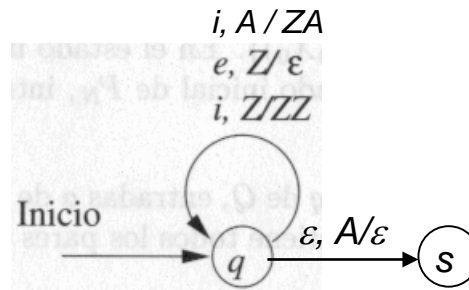
1. $\delta_F(p_0, \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$
2. para todo estado q de Q , entrada a de Σ o $a = \varepsilon$, y símbolos de pila γ de Γ , $\delta_F(q, a, \gamma)$ contiene todos los pares de $\delta_N(q, a, \gamma)$
3. además, $\delta_F(q, \varepsilon, X_0)$ contiene (p_f, ε) para todo estado q de Q



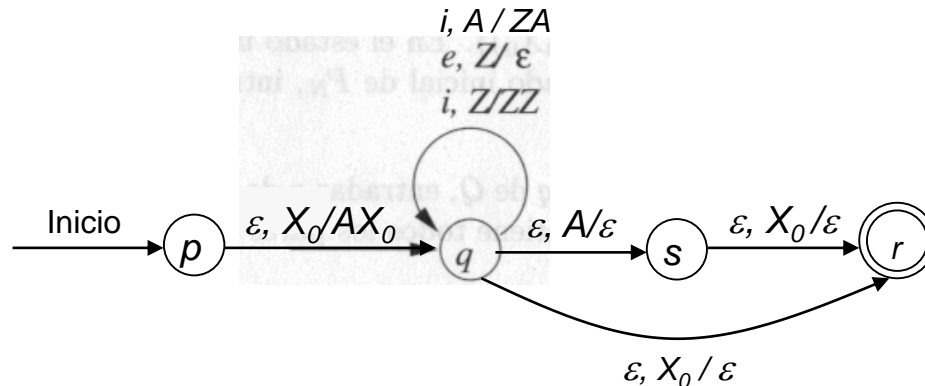
Conversión vaciado de pila a estado final

Ejemplo: diseñar un AP que procese secuencias "if else", detectando cuándo se introducen igual número de *else* que *if*

- $P_N = (\{q, s\}, \{i, e\}, \{Z, A\}, \delta_N, q, A)$



- $P_F = (\{p, q, r, s\}, \{i, e\}, \{Z, A, X_0\}, \delta_F, p, X_0, \{r\})$



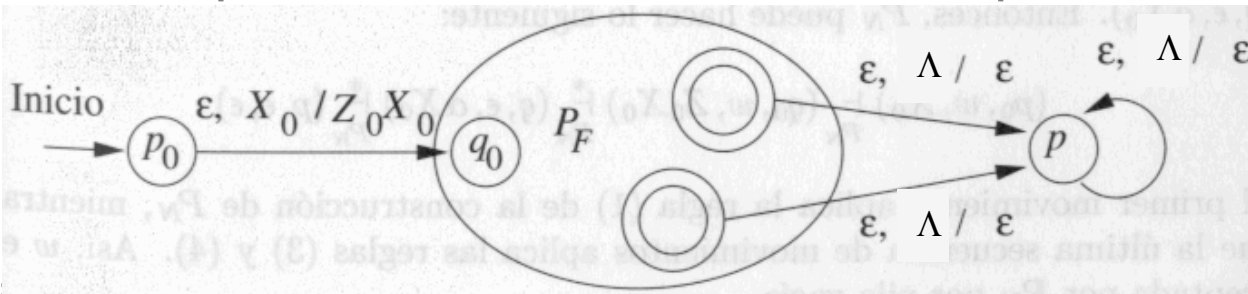
Conversión estado final a vaciado de pila

Teorema: sea L el lenguaje $L(P_F)$ de algún autómata de pila

$P_F = (Q, \Sigma, \Gamma, \delta_F, q_0, Z_0, F)$. Entonces existe un AP P_N tal que $L = N(P_N)$

Prueba: $P_N = (Q \cup \{p_0, p\}, \Sigma, \Lambda = \Gamma \cup \{X_0\}, \delta_N, p_0, X_0)$, donde δ_N se define:

1. $\delta_N(p_0, \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$
2. para todo estado q de Q , entrada a de Σ o $a = \varepsilon$, y símbolos de pila Y en Γ , $\delta_N(q, a, Y)$ contiene todos los pares de $\delta_F(q, a, Y)$
3. para todo estado de aceptación q en F y símbolos de pila Y en Λ , $\delta_N(q, \varepsilon, Y)$ contiene (p, ε)
4. para todos los símbolos de la pila Y en Λ , $\delta_N(p, \varepsilon, Y) = \{(p, \varepsilon)\}$



Necesidad del nuevo símbolo inicial de pila: si el APF vacía su pila en un estado no final no debería reconocer la secuencia. Si no se añadiese el nuevo símbolo inicial de pila, el APN pasaría a reconocer en esas situaciones.

Equivalencia entre AP y GIC

El objetivo es demostrar que los tres siguientes lenguajes son todos de la misma clase:

1. lenguajes independientes del contexto
 2. lenguajes aceptados por estado final por algún AP
 3. lenguajes aceptados por vaciado de pila por algún AP
- La equivalencia de (2) y (3) ya se ha demostrado
 - Demostraremos que de (1) se sigue (3), aunque no que de (3) se sigue (1)

Conversión de gramáticas a AP

- Sea la GIC $G = (V, T, Q, S)$. El AP que acepta $L(G)$ por pila vacía será:
 - $P = (\{q\}, T, V \cup T, \delta, q, S)$
 - δ se define por:
 1. para cada variable A , $\delta(q, \varepsilon, A) = \{(q, \beta) \mid A \rightarrow \beta \text{ es una producción de } P\}$
 2. para cada símbolo terminal a , $\delta(q, a, a) = (q, \varepsilon)$
- Ejemplo: obtener el AP que reconozca por vaciado de pila la siguiente gramática:

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1, E \rightarrow I \mid E^*E \mid E+E \mid (E)$$

Autómatas con pila deterministas

Los APD aceptan un conjunto de lenguajes a medio camino entre los lenguajes regulares y las GIC

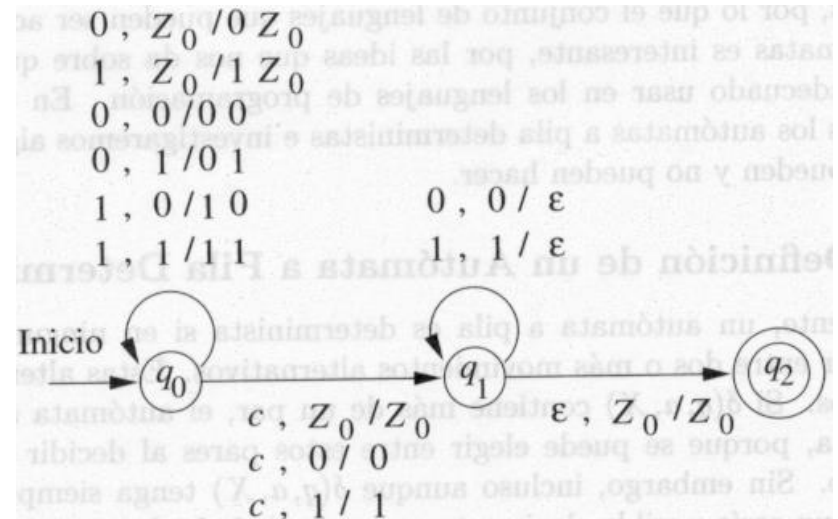
- los analizadores sintácticos se comportan generalmente como APD

Un AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ es determinista si:

- $\delta(q, a, X)$ tiene como máximo un elemento para cualquier q en Q , a en Σ o $a = \varepsilon$, y X en Γ
- si $\delta(q, a, X)$ no está vacío para algún a en Σ , $\delta(q, \varepsilon, X)$ debe estar vacío

Ejemplo:

$$L_{wcwr} = \{wcw^R \mid w \text{ está en } (0 + 1)^*\}$$



Problemas

1. Determinar si el siguiente AP es o no determinista: $P = (\{q, p\}, \{0, 1\}, \{Z_0, X\}, \delta, q, Z_0, \{p\})$, donde δ se define en la siguiente tabla

Q	Σ	Γ	Movimiento
q	0	Z_0	(q, XZ_0)
q	0	X	(q, XX)
q	1	X	(q, X)

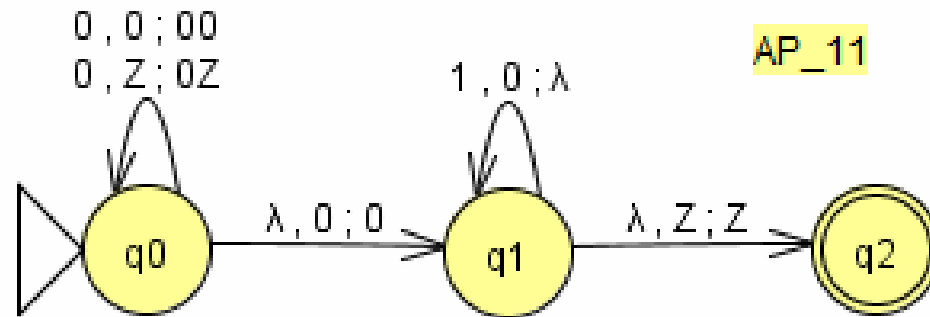
Q	Σ	Γ	Movimiento
q	ε	X	(p, ε)
p	ε	X	(p, ε)
p	1	X	(p, XX)
p	1	Z_0	(p, ε)

2. Diseñar un **APF** que acepte el lenguaje $\{0^n 1^n \mid n > 0\}$
 - Restricción: el alfabeto de la pila será igual al alfabeto de entrada más el símbolo inicial de pila.
3. Igual que el anterior, pero con $n \geq 0$.

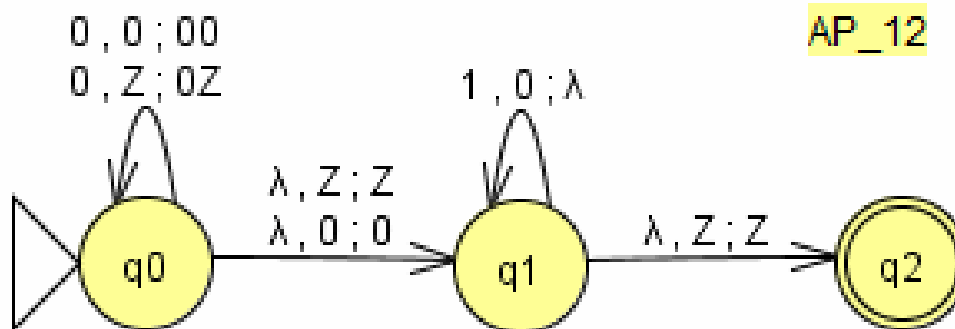
Problemas

2. Diseñar un **APF** que acepte el lenguaje $\{0^n 1^n \mid n > 0\}$

- Restricción: el alfabeto de la pila será igual al alfabeto de entrada más el símbolo inicial de pila.



3. Igual que el anterior, pero con $n \geq 0$.



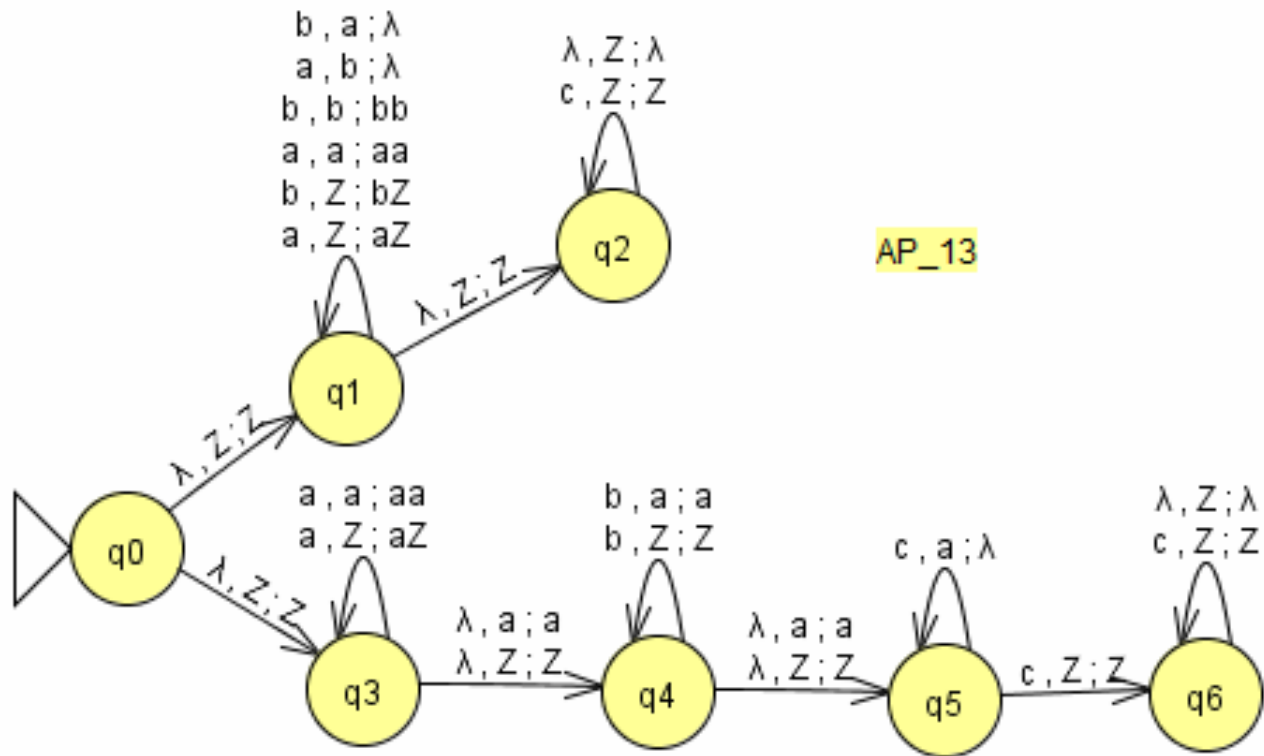
Problemas

1. Diseñar un **APN** que acepte el lenguaje formado por aquellas cadenas que cumplen alguno de los siguientes criterios:
 - contienen igual número de símbolos ***a*** y ***b***, entrando estos en cualquier orden, y finalizan con un número k de símbolos ***c***, $k \geq 0$.
 - $a^i b^j c^k / k > i$

Restricción: el alfabeto de la pila será igual al alfabeto de entrada más el símbolo inicial de pila.

Problemas

Solución:



Problemas

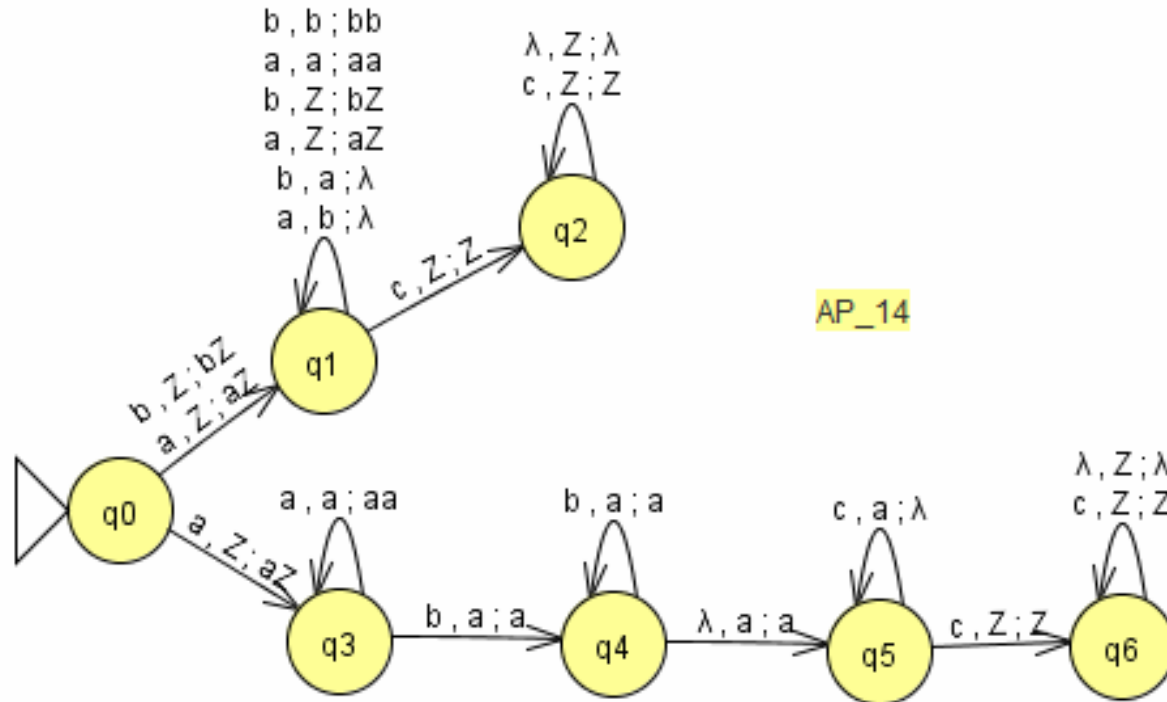
1. Igual que el anterior, pero las cadenas deben tener ahora al menos un símbolo de cada tipo.
2. Diseñar el **APF** sobre el alfabeto $\{a, b\}$ que acepte el lenguaje $\{a^i b^j \mid 2i = j; i, j > 0\}$.

Restricción: el alfabeto de la pila será igual al alfabeto de entrada más el símbolo inicial de pila.

3. Igual que el anterior, pero para la condición $i = 2j$.

Problemas

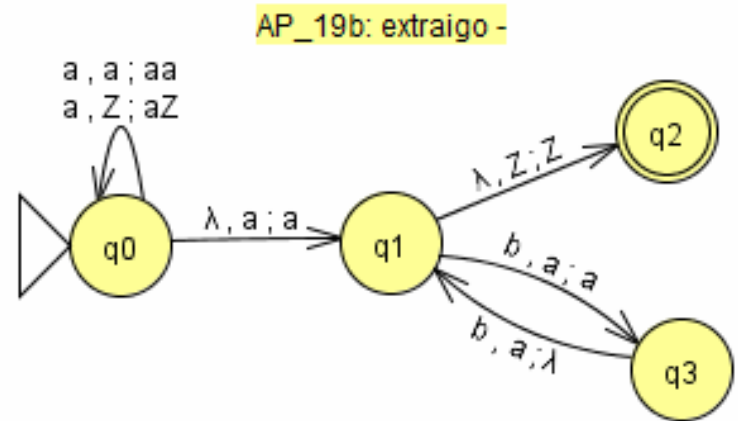
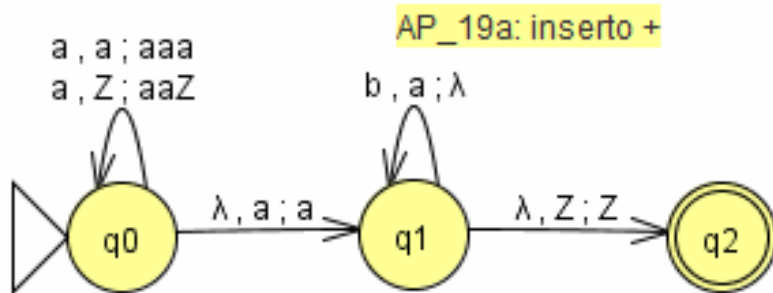
1. Igual que el anterior, pero las cadenas deben tener ahora al menos un símbolo de cada tipo.



Problemas

Diseñar el **APF** sobre el alfabeto $\{a, b\}$ que acepte el lenguaje $\{a^i b^j \mid 2i = j; i, j > 0\}$.

Restricción: el alfabeto de la pila será igual al alfabeto de entrada más el símbolo inicial de pila.

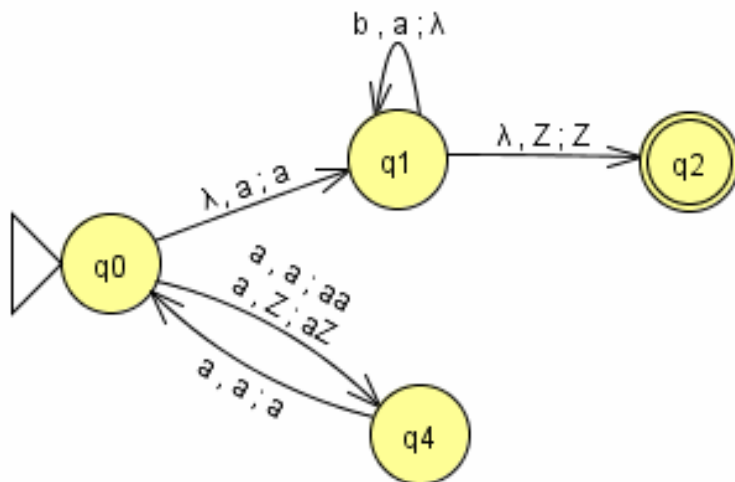


Problemas

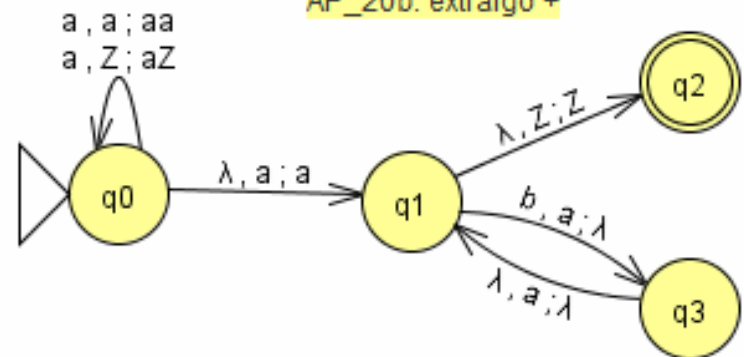
Diseñar el **APF** sobre el alfabeto $\{a, b\}$ que acepte el lenguaje $\{a^i b^j \mid i = 2j; i, j > 0\}$.

Restricción: el alfabeto de la pila será igual al alfabeto de entrada más el símbolo inicial de pila.

AP_20a: inserto -



AP_20b: extraigo +



Lema de bombeo para LIC

- Para un LIC, el cumplimiento del lema de bombeo (LB) es una condición **necesaria**, pero **no suficiente**
- Teorema: sea L un LIC. Entonces existe una constante n tal que si z es cualquier cadena de L de longitud $|z| \geq n$, podemos escribir $z=uvwxy$, con las siguientes condiciones:
 1. $|vwx| \leq n$
 2. $vx \neq \varepsilon$
 3. Para todo $k \geq 0$, uv^kwx^ky está en L

Aplicación del lema de bombeo

1. Elegimos L del que queremos demostrar que no es LIC
2. El valor de n es desconocido, por lo que debemos considerar cualquier posible valor
3. Elegimos z (podemos usar n como parámetro)
4. Repetir para todas las descomposiciones:
 1. Escoger una descomposición de z en $uvwxy$, sujeta a las restricciones:
 1. $vx \neq \varepsilon$
 2. $|vwx| \leq n$
 2. Si uv^kwx^ky pertenece a L para todo valor de k
 1. Se verifica el LB
 2. No se puede afirmar que el lenguaje sea independiente del contexto
 3. No es necesario probar con otras descomposiciones (finaliza el algoritmo)
5. Si 4.2 no se ha cumplido para ninguna descomposición, no se verifica el LB y, por tanto, el lenguaje no es un LIC

Aplicación del lema de bombeo

Verificar si se cumple el lema del bombeo para: $L = \{a^n b^n c^n \mid n \geq 1\}$

1. Si L es un LIC, entonces existe una constante n tal que si z es cualquier cadena de L de longitud $|z| \geq n$, podemos escribir $z = uvwxy$, cumpliendo las condiciones antes vistas.
2. Elegimos $z = a^n b^n c^n$
3. Al tener que cumplirse: $|vwx| \leq n$, vwx no pueden contener al mismo tiempo los tres símbolos del alfabeto. Entonces tampoco vx , que al menos, eso sí, contendrá un símbolo.
4. Por tanto, si consideramos $k=0$, resulta $uv^0wx^0y = uwy$, y esta cadena no podrá pertenecer a L ya que le faltarán los elementos de vx para estar equilibrada en número de “aes”, “bes” y “ces”.

Ejemplos de Lenguajes **no** IC

- $L = \{0^p 1^p 2^p \mid p \geq 1\}$
 - un LIC no puede emparejar tres grupos de símbolos de acuerdo con su igualdad o desigualdad
- $L = \{0^i 1^j 2^i 3^j \mid i \geq 1 \text{ y } j \geq 1\}$
 - un LIC no puede emparejar dos pares de números iguales de símbolos que se entrelacen
- $L = \{ss \mid s \in (0+1)^*\}$
 - un LIC no puede emparejar dos cadenas de longitud arbitraria si las cadenas se eligen de un alfabeto de más de un símbolo

Problemas

Dados los siguientes lenguajes:

- Razonar si son LIC, es decir, si es posible reconocerlos con un autómatata con pila.

1. $L = \{0^p 1^p \mid p \geq 1\}$

2. $L = \{0^p 1^p 2^p \mid p \geq 1\}$