Sesión 18

Autómata de Pila
(Pushdown Automata)

Autómata de Pila (AP, PDA)

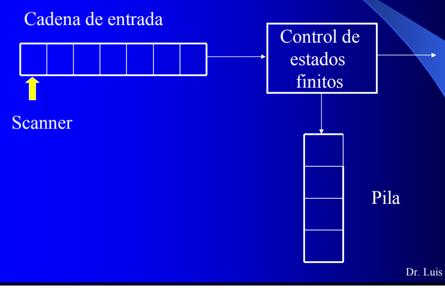
- Un AP es una máquina que acepta el lenguage generado por una *GLC*
- Consiste en un NFA-Λ aumentado con una pila (stack).

Una máquina para aceptar *Pal* pares

- $L = \{xx^r \mid x \in \{0, 1\}^*\}$
- $G_{pal\text{-}par} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid \Lambda\}$
  - $x = 101101$
- ¿Cómo podemos saber si la cadena está en el lenguaje?

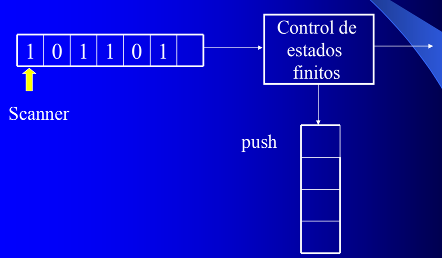
Una máquina para aceptar *Pal* pares

- Estado inicial: Leer la primera mitad de la cadena



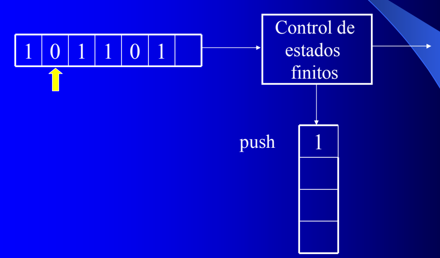
Una máquina para aceptar *Pal* pares

- Estado inicial: Leyendo la primera mitad de la cadena



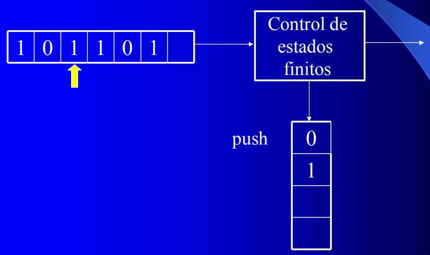
Una máquina para aceptar *Pal* pares

- Estado inicial: Leyendo la primera mitad de la cadena



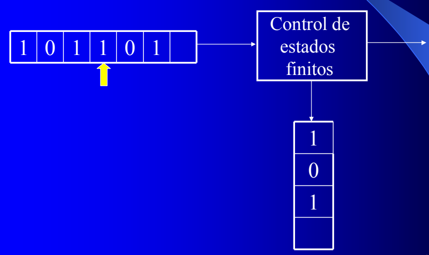
Una máquina para aceptar *Pal* pares

- Estado inicial: Leyendo la primera mitad de la cadena



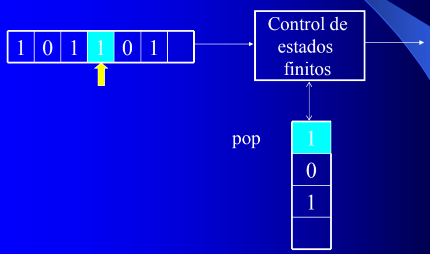
Una máquina para aceptar *Pal* pares

- Cambio de estado: Leyendo la segunda mitad de la cadena



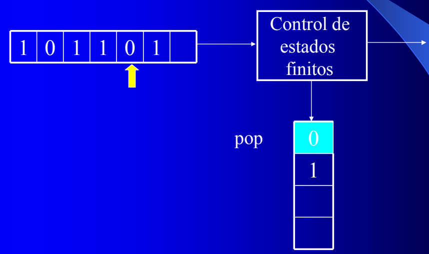
Una máquina para aceptar *Pal* pares

- Estado: Leyendo la segunda mitad de la cadena



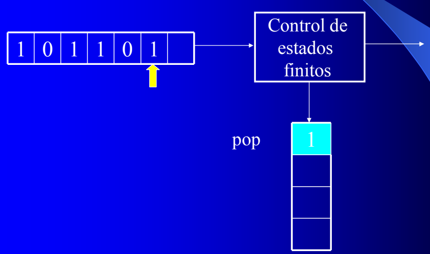
Una máquina para aceptar *Pal* pares

- Estado: Leyendo la segunda mitad de la cadena



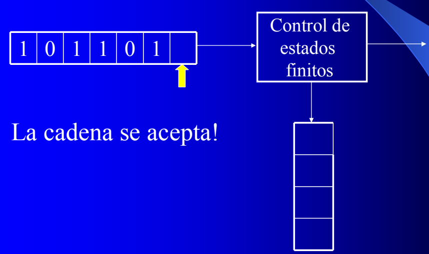
Una máquina para aceptar *Pal* pares

- Estado: Leyendo la segunda mitad de la cadena



Una máquina para aceptar *Pal* pares

- Estado Final: se leyeron todos los símbolos de la cadena, y la pila quedó vacía!



Noción de estado

- En FA (DFA, NFA & NFA-Λ) el siguiente estado depende del estado actual y el símbolo actual en la cinta de entrada: El estado es una fotografía completa de la máquina en cada punto (estado) de la computación!
- En el Autómata de Pila la fotografía completa depende, además, del contenido del stack!

La máquina no es determinística

- ¿Cómo sabe la máquina cuando termina de leer la primera parte de la cadena?
  - No sabe: ¡adivina!

La máquina no es determinística

- Al leer cada símbolo, la máquina:
  - Lee el símbolo y realiza la operación correspondiente (push o pop).
  - Hace una transición vacía y cambia de estado para leer la segunda mitad de la cadena!

La máquina no es determinística

- ¿Cómo sabe cuando el stack está vacio?
  - Se usa un símbolo especial!

Operación de la Máquina

- Dependiendo de:
  - El símbolo actual en la cinta
  - El estado actual
  - El símbolo hasta arriba (top) de la pila
- Acción:
  - Seleccionar el siguiente estado
  - Realizar operación (push, pop o nada)

Operación de la Máquina

- Aceptar una cadena si:
  - Se leen todos los símbolos en la cinta
  - Se llega a un estado aceptor o con el stack vacío

Definición formal de un AP

- Un Autómata de Pila es un 7-teto: $(Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ donde:
  - Q es un conjunto finito de estados
  - Σ es el alfabeto del lenguaje reconocido por el AP
  - Γ es el alfabeto de las cadenas que ocurren en el stack
  - $q_0 \in Q$ (el estado inicial)
  - $Z_0 \in \Gamma$ (el símbolo inicial del stack)
  - $A \subseteq Q$ (el conjunto de estados aceptores)
  - δ la función de transición de tipo:

    $Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma \rightarrow$ subconjuntos finitos de $Q \times \Gamma^*$

    donde la cadena <u>hasta arriba del stack</u> $\in \Gamma^*$

Función de transición

- Función de transición para DFA:

  δ: $Q \times \Sigma \rightarrow Q$
- Función de transición para NFA:

  δ: $Q \times \Sigma \rightarrow 2^Q$
- Función de transición para NFA-Λ:

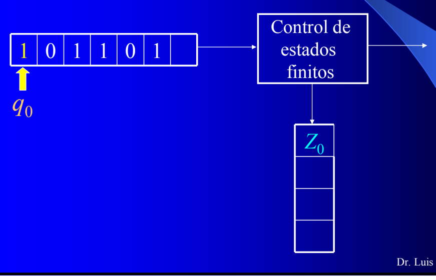
  δ: $Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q$
- Función de transición para PDA:

  δ: $Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma \rightarrow$ c.f. $Q \times \Gamma^*$

Una transición

- Una transición:

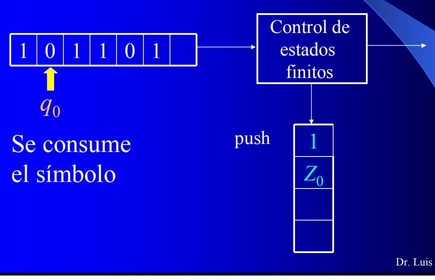
  $\delta(q_0, 1, Z_0)$



Una transición

- Una transicón: Push!

  $\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$

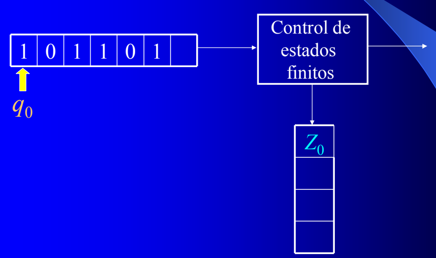


Se consume el símbolo

Transición no-determinística

- Una transición:

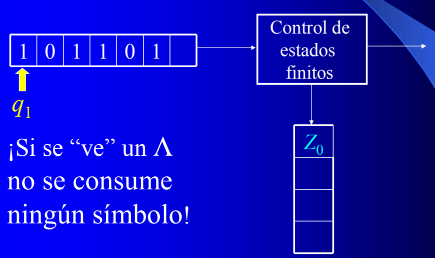
  $\delta(q_0, \Lambda, Z_0)$



Transición no-determinística

- Una transición-Λ:

  $\delta(q_0, \Lambda, Z_0) = (q_1, Z_0)$



¡Si se "ve" un Λ no se consume ningún símbolo!

Tabla de transición: $L = \{xx^r \mid x \in \{0, 1\}^*\}$

Id	Estado	Entrada	Top (stack)	Movida(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Tabla de transición: $L = \{xx^r \mid x \in \{a, b\}^*\}$

Push

Id	Estado	Entrada	Top (stack)	Movida(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Tabla de transición: $L = \{xx^r \mid x \in \{a, b\}^*\}$

Mitad de la cadena

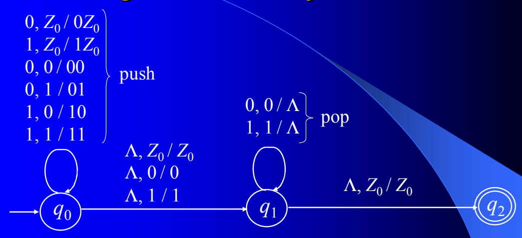
Id	Estado	Entrada	Top (stack)	Movida(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Tabla de transición: $L = \{xx^r \mid x \in \{a, b\}^*\}$

Pop

Id	Estado	Entrada	Top (stack)	Movida(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Diagrama de flujo de PDA

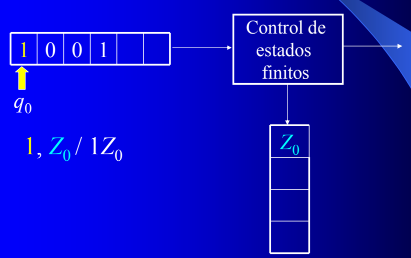


- Una etiqueta (en un arco) de forma $a, X/\alpha$ de p a q significa que con la entrada a, el símbolo X en el Top (del stack) se substituye por α:

$$\delta(p, a, X) = (q, \alpha)$$

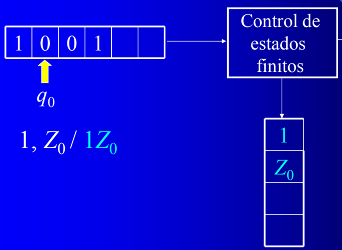
Una máquina para aceptar *Pal* pares

$$\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$$



Una máquina para aceptar *Pal* pares

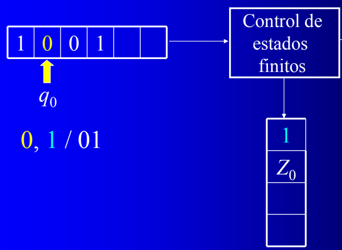
$\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$



$1, Z_0 / 1Z_0$

Una máquina para aceptar *Pal* pares

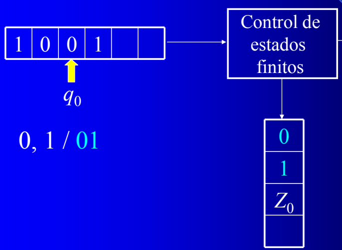
$\delta(q_0, 0, 1) = (q_0, 01)$



$0, 1 / 01$

Una máquina para aceptar *Pal* pares

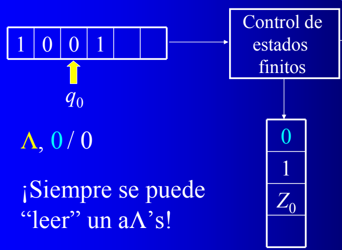
$\delta(q_0, 0, 1) = (q_0, 01)$



$0, 1 / 01$

Una máquina para aceptar *Pal* pares

$\delta(q_0, \Lambda, 0) = (q_1, 0)$

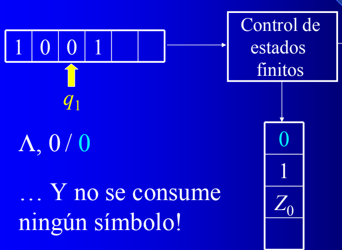


$\Lambda, 0 / 0$

¡Siempre se puede "leer" un aΛ's!

Una máquina para aceptar *Pal* pares

$\delta(q_0, \Lambda, 0) = (q_1, 0)$ Sólo se cambia de estado

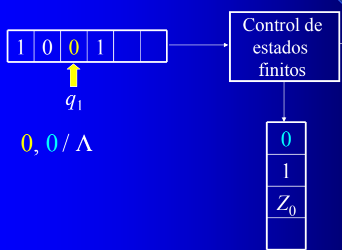


$\Lambda, 0 / 0$

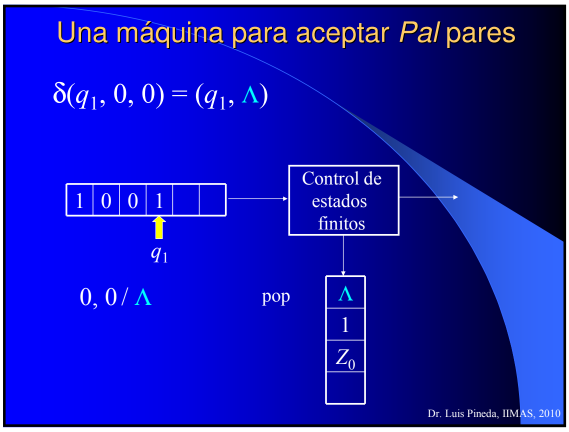
… Y no se consume ningún símbolo!

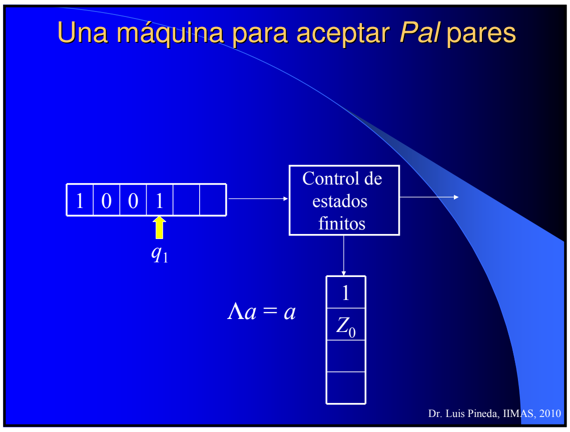
Una máquina para aceptar *Pal* pares

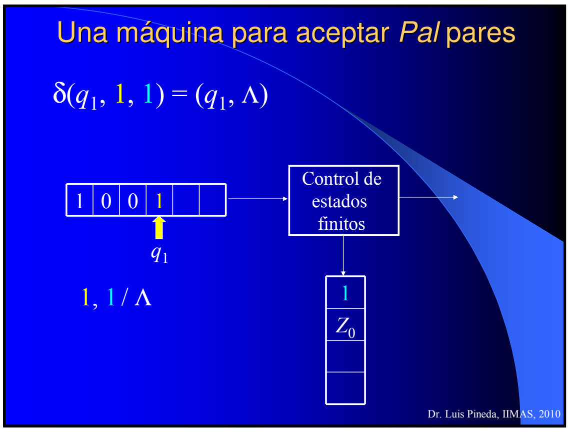
$\delta(q_1, 0, 0) = (q_1, \Lambda)$

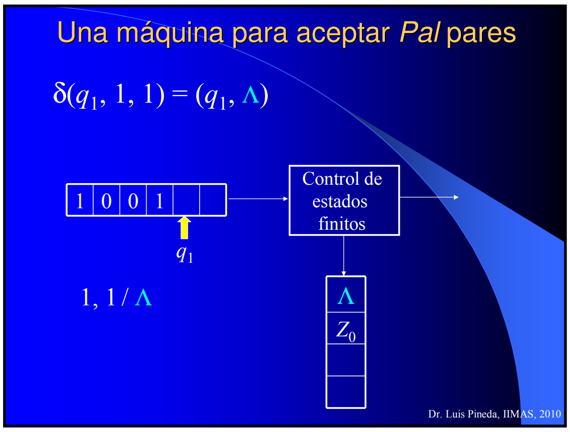


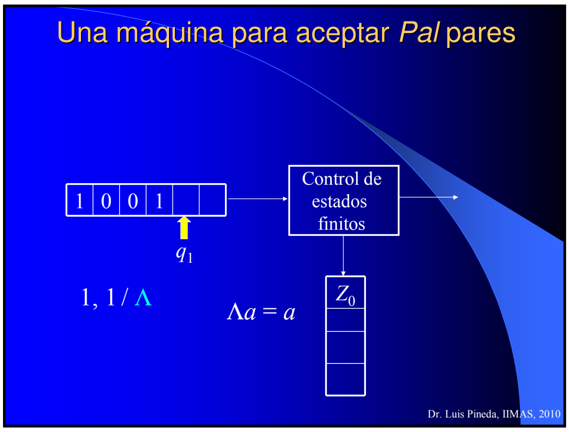
$0, 0 / \Lambda$

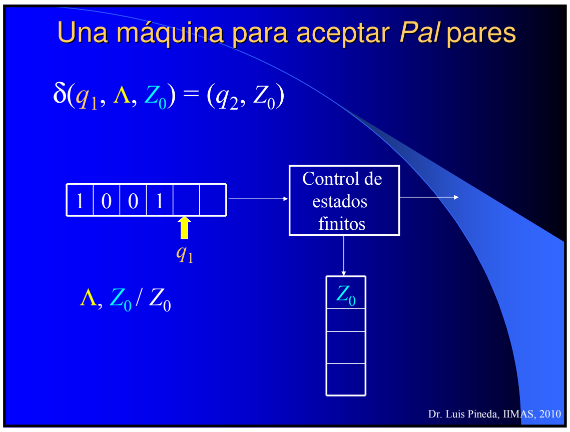

Una máquina para aceptar *Pal* pares
$\delta(q_1, 0, 0) = (q_1, \Lambda)$
1 0 0 1
Control de estados finitos
q_1
$0, 0 / \Lambda$
pop
Λ
1
Z_0
Dr. Luis Pineda, IIMAS, 2010


Una máquina para aceptar *Pal* pares
1 0 0 1
Control de estados finitos
q_1
$\Lambda a = a$
1
Z_0
Dr. Luis Pineda, IIMAS, 2010


Una máquina para aceptar *Pal* pares
$\delta(q_1, 1, 1) = (q_1, \Lambda)$
1 0 0 1
Control de estados finitos
q_1
$1, 1 / \Lambda$
1
Z_0
Dr. Luis Pineda, IIMAS, 2010

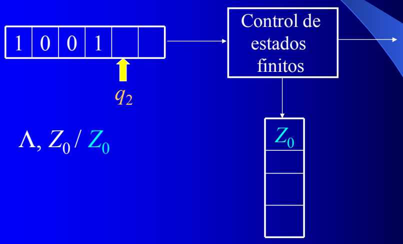

Una máquina para aceptar *Pal* pares
$\delta(q_1, 1, 1) = (q_1, \Lambda)$
1 0 0 1
Control de estados finitos
q_1
$1, 1 / \Lambda$
Λ
Z_0
Dr. Luis Pineda, IIMAS, 2010


Una máquina para aceptar *Pal* pares
1 0 0 1
Control de estados finitos
q_1
$1, 1 / \Lambda$
$\Lambda a = a$
Z_0
Dr. Luis Pineda, IIMAS, 2010


Una máquina para aceptar *Pal* pares
$\delta(q_1, \Lambda, Z_0) = (q_2, Z_0)$
1 0 0 1
Control de estados finitos
q_1
$\Lambda, Z_0 / Z_0$
Z_0
Dr. Luis Pineda, IIMAS, 2010

Una máquina para aceptar *Pal* pares

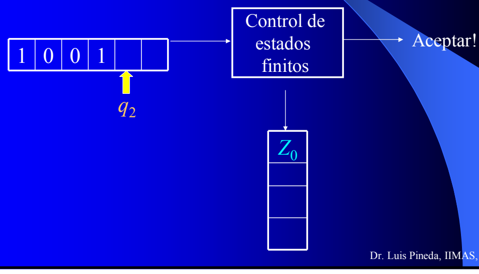
$\delta(q_1, \Lambda, Z_0) = (q_2, Z_0)$



$\Lambda, Z_0 / Z_0$

Una máquina para aceptar *Pal* pares

- El estado q_2 es aceptor
- Se leyó toda la cadena!
- Z_0 está hasta arriba del stack!



Configuraciones de un AP

- Una configuración de una AP

  $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

  es una tercia:

  (q, x, α)

  donde
  - $q \in Q$ es el estado actual
  - $x \in \Sigma^*$ es la parte de la entrada <u>que no se ha leído</u>
  - $\alpha \in \Gamma^*$ es el contenido actual de <u>TODO</u> el stack

Configuraciones de un AP

- Una <u>movida</u> de una configuración a la siguiente:

  $(p, x, \alpha) \Rightarrow_M (q, y, \beta)$
- Una secuencia de cero o más movidas:

  $(p, x, \alpha) \Rightarrow^*_M (q, y, \beta)$

El lenguaje aceptado por un AP

- Aceptando por estado final:
  - Si $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ es un AP, $L(M)$, el lenguaje aceptado por M por *estado final*, es:

    $L(M) = \{w \mid (q_0, w, Z_0) \Rightarrow^*_M (q, \Lambda, \alpha)\}$

    para algún $\alpha \in \Gamma^*$ & algún $q \in A$. El stack puede o no estar vacío cuando w es aceptada (i.e. α puede ser Λ, pero no necesariamente).

El lenguaje aceptado por un AP

- Se dice que un lenguaje $L \subseteq \Sigma^*$ se acepta por M si y sólo si L es exactamente el conjunto de cadenas aceptadas por M, y escribimos $L = L(M)$

El lenguaje aceptado por un AP

- Aceptando por stack vacío:
  - Si $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ es un PDA, $N(M)$, el lenguaje aceptado por M por *stack vacío*, si:

    $N(M) = \{w \mid (q_0, w, Z_0) =>^*_M (q, \Lambda, \Lambda)\}$

    Aquí, el estado final es irrelevante!

El lenguaje aceptado por un AP

- Si $L \subseteq \Sigma^*$ se acepta por un AP M_F por estado final, existe un AP M_N que acepta a L por stack vacío
- Para un AP M dado, el lenguaje que acepta por estado final es normalmente diferente del lenguaje que acepta (el mismo AP M) por stack vacío

Una máquina para aceptar *Pal* (todas) (por estado final)

- El lenguaje:

  $Pal = \{x \mid x = x^r \in \{1, 0\}^*\}$

- La gramática:

  $G_{pal} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid 1 \mid 0 \mid \Lambda\}$

- Definimos M_{pal}:

  $M_{pal} = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, q_0, Z_0, \{q_2\}, \delta)$

Una máquina para aceptar *Pal* (todas) (por estado final)

- Opciones durante el proceso de la primera parte de la cadena:
  - Todavía ahí: seguir en el estado, consumir símbolo & push
  - Se llega a la mitad de una palíndrome par: cambiar de estado con una transición-Λ
  - Se llega a la mitad de una palíndrome non: cambiar de estado & consumir símbolo pero no poner (push) el símbolo en el stack!

Tabla de transición: $L = \{x \mid x = x^r\}$

Id	Estado	Entrada	Top (stack)	Movidas(s)
1	q_0	0	Z_0	$(q_0, 0Z_0), (q_1, Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0), (q_1, Z_0)$
3	q_0	0	0	$(q_0, 00), (q_1, 0)$
4	q_0	1	0	$(q_0, 10), (q_1, 0)$
5	q_0	0	1	$(q_0, 01), (q_1, 1)$
6	q_0	1	1	$(q_0, 11), (q_1, 1)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Todavía en la primera parte...

Id	Estado	Entrada	Top (stack)	Movidas(s)
1	q_0	0	Z_0	$(q_0, 0Z_0), (q_1, Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0), (q_1, Z_0)$
3	q_0	0	0	$(q_0, 00), (q_1, 0)$
4	q_0	1	0	$(q_0, 10), (q_1, 0)$
5	q_0	0	1	$(q_0, 01), (q_1, 1)$
6	q_0	1	1	$(q_0, 11), (q_1, 1)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Mitad de cadena par...

Id	Estado	Entrada	Top (stack)	Movidas(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$, (q_1, Z_0)
2	q_0	1	Z_0	$(q_0, 1Z_0)$, (q_1, Z_0)
3	q_0	0	0	$(q_0, 00)$, $(q_1, 0)$
4	q_0	1	0	$(q_0, 10)$, $(q_1, 0)$
5	q_0	0	1	$(q_0, 01)$, $(q_1, 1)$
6	q_0	1	1	$(q_0, 11)$, $(q_1, 1)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Mitad de cadena non...

Id	Estado	Entrada	Top (stack)	Movidas(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$, (q_1, Z_0)
2	q_0	1	Z_0	$(q_0, 1Z_0)$, (q_1, Z_0)
3	q_0	0	0	$(q_0, 00)$, $(q_1, 0)$
4	q_0	1	0	$(q_0, 10)$, $(q_1, 0)$
5	q_0	0	1	$(q_0, 01)$, $(q_1, 1)$
6	q_0	1	1	$(q_0, 11)$, $(q_1, 1)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Aceptar 101