

## Sesión 20

AP correspondiente a una GLC

Dr. Luis Pineda, IIMAS, 2010

## AP & GLC

- Existe un AP  $M$  tal que  $L(M) = L(G)$  para toda GLC  $G$
- Existe una GLC  $G$  tal que  $L(G) = L(M)$  para todo AP  $M$
- El conjunto de todos los LLC generados por las GLC (ambiguas o no ambiguas) es el conjunto de todos los LLC aceptados por los AP.

Dr. Luis Pineda, IIMAS, 2010

## Para toda GLC hay un AP

- ¿Cuál es nuestra evidencia hasta ahora?:
  - $Pal$ ,  $Pal_{par}$ ,  $Pal_{marca}$ ?
  - ¿Qué hay de lenguajes con una estructura sintáctica más compleja?
  - ¿Qué hay del lenguaje natural?

Dr. Luis Pineda, IIMAS, 2010

## Para toda GLC hay un AP

- Existe un AP  $M$  tal que  $L(M) = L(G)$  para toda GLC  $G$
- Existe una CFG  $G$  tal que  $L(G) = L(M)$  para todo AP  $M$

Dr. Luis Pineda, IIMAS, 2010

## AP dada GLC para $Pal$

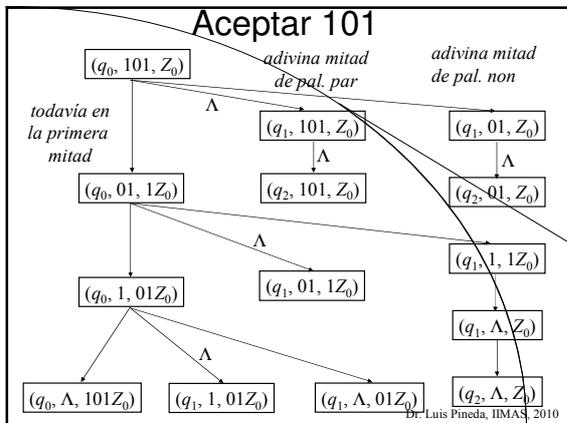
- El lenguaje:
 
$$Pal = \{x \mid x = x^r \in \{a, b\}^*\}$$
- La gramática:
 
$$G_{pal} = (\{P\}, \{0, 1\}, P, \{P \rightarrow 0P0 \mid 1P1 \mid 1 \mid 0 \mid \Lambda\})$$
- El AP  $M_{pal}$ :
 
$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, q_0, Z_0, \{q_2\}, \delta)$$

Dr. Luis Pineda, IIMAS, 2010

## Tabla de transición para $Pal$

Id	Estado	Entrada	Símbolo stack	Movida(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	$q_0$	$\Lambda$	$Z_0$	$(q_1, Z_0)$
8	$q_0$	$\Lambda$	0	$(q_1, 0)$
9	$q_0$	$\Lambda$	1	$(q_1, 1)$
10	$q_1$	0	0	$(q_1, \Lambda)$
11	$q_1$	1	1	$(q_1, \Lambda)$
12	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
Otras combinaciones				nada

Dr. Luis Pineda, IIMAS, 2010



- ### El AP para la GLC
- El AP para *Pal* es específico: alambrado!
  - Necesitamos un método general para construir APs a partir de la especificación de la gramática
  - Podemos guardar las producciones en una memoria, y simular la conducta de cualquier gramática utilizando un AP “universal”
  - De la simulación podemos inducir el AP específico
- Dr. Luis Pineda, IIMAS, 2010

- ### AP para GLC
- La estrategia:
    - Definir un AP que pueda simular una derivación para cada cadena  $x$  en  $L(G)$  para toda GLC
    - Para toda cadena  $x$  & GLC  $G$  simular el proceso de derivar  $x$  usando las producciones en  $P$  de  $G$
- Dr. Luis Pineda, IIMAS, 2010

- ### AP para GLC
- Ventajas de la simulación:
    - Verificar la membresía de una cadena en el lenguaje
    - Revelar los pasos de la derivación
  - Tipo de AP usado en la simulación:
    - En general no-determinístico
    - En algunos casos (pero importantes) es posible simular un AP-D
- Dr. Luis Pineda, IIMAS, 2010

- ### AP para GLC
- Cada paso de la simulación corresponde a la construcción de una parte del árbol de derivación; hay dos formas:
    - Top-down: expandir el símbolo inicial  $S$  hasta generar la cadena (con derivaciones más izquierdas)
    - Bottom-up: Construir el árbol hacia arriba a partir de la cadena  $x$  hasta el símbolo inicial  $S$  (derivaciones más derechas)
- Dr. Luis Pineda, IIMAS, 2010

- ### AP para GLC
- Cada paso de la simulación corresponde a la construcción de una parte del árbol de derivación; hay dos formas:
    - Top-down: expandir el símbolo inicial  $S$  hasta generar la cadena (con derivaciones más izquierdas)
    - Bottom-up: Construir el árbol hacia arriba a partir de la cadena  $x$  hasta el símbolo inicial  $S$  (derivaciones más derechas)
- Dr. Luis Pineda, IIMAS, 2010

### Simulación Top-Down

- Sea  $G = (V, \Sigma, S, P)$  una GLC. El AP  $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$  se define como sigue:
  - $Q = \{q_0, q_1, q_2\}$
  - $A = \{q_2\}$
  - $\Gamma = V \cup \Sigma \cup \{Z_0\}$  donde  $Z_0 \notin V \cup \Sigma$

### Simulación Top-Down

- Movida 0:
  - Movida inicial: poner  $S$  en el stack y moverse a  $q_1$ :  
 $\delta(q_0, \Lambda, Z_0) = \{(q_1, SZ_0)\}$
- Movida 3:
  - Movida final: pasar de  $q_1$  al estado aceptor  $q_2$  cuando el stack está vacío, excepto por  $Z_0$ :  
 $\delta(q_1, \Lambda, Z_0) = \{(q_2, Z_0)\}$

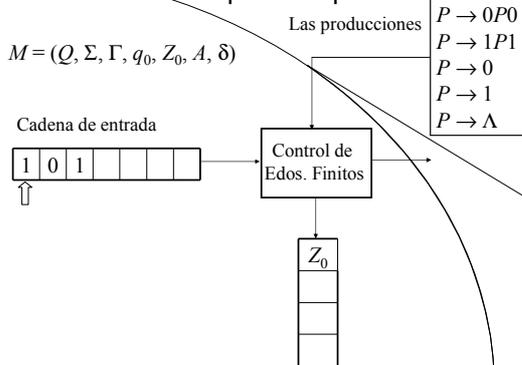
### Movida 1:

- Si la variable del lado izquierdo de una producción en  $P$  está hasta arriba de la pila, substituir dicha variable por la cadena que está del lado derecho de la producción (hasta arriba del stack), sin consumir ningún símbolo de la cadena de entrada:
  - Para todo  $A \in V$ ,  
 $\delta(q_1, \Lambda, A) = \{(q_1, \alpha) \mid A \rightarrow \alpha \in P\}$

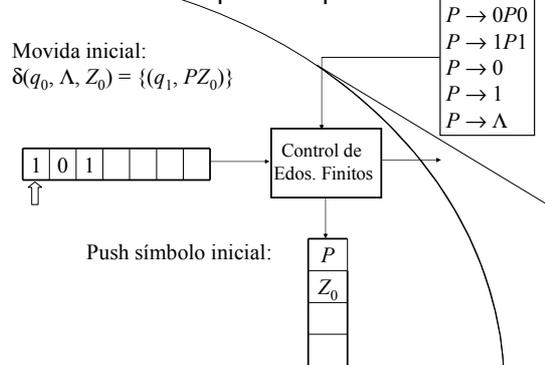
### Movida 2:

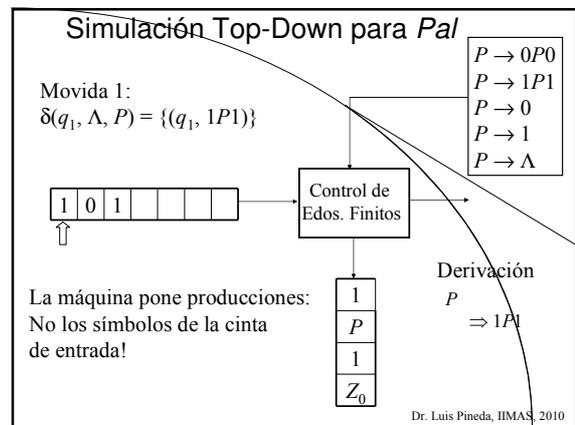
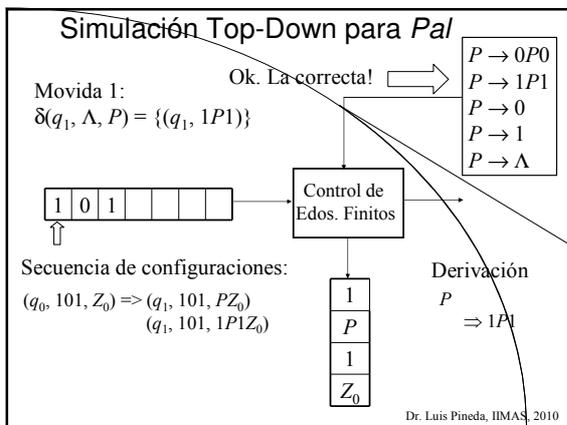
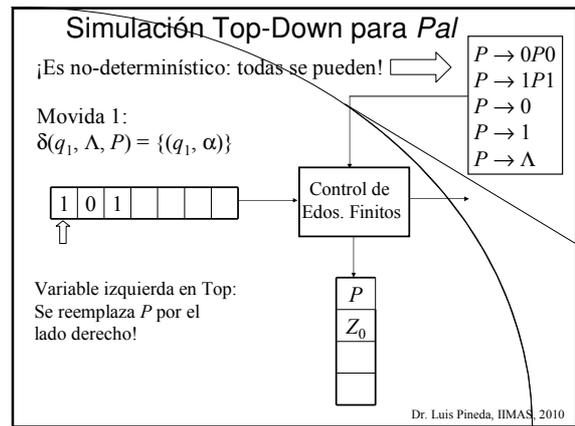
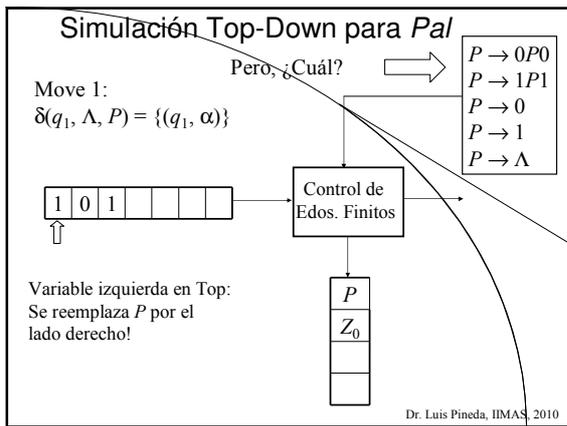
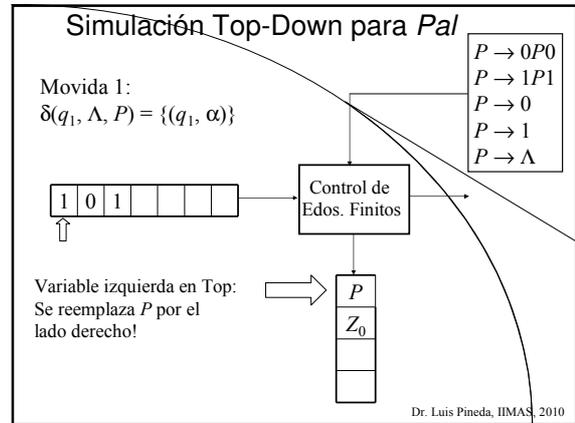
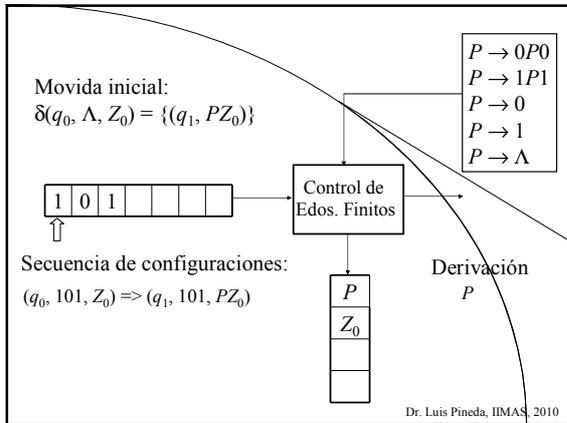
- Si hay un símbolo terminal hasta arriba de la pila, y dicho símbolo corresponde con el símbolo que se lee en el estado actual en la cinta de entrada, sacar (pop) el símbolo del stack y consumir el símbolo en la cadena de entrada:
  - Para todo  $a \in \Sigma$ ,  
 $\delta(q_1, a, a) = \{(q_1, \Lambda)\}$

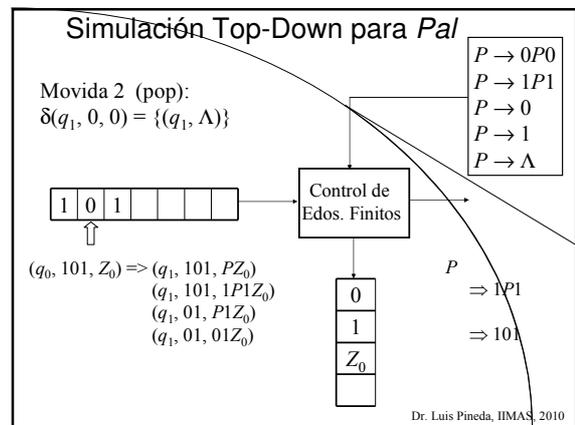
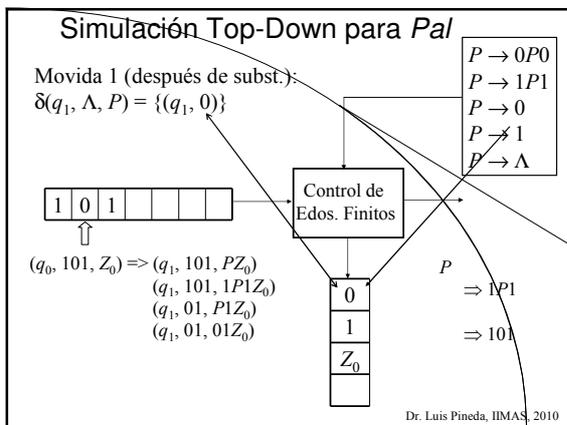
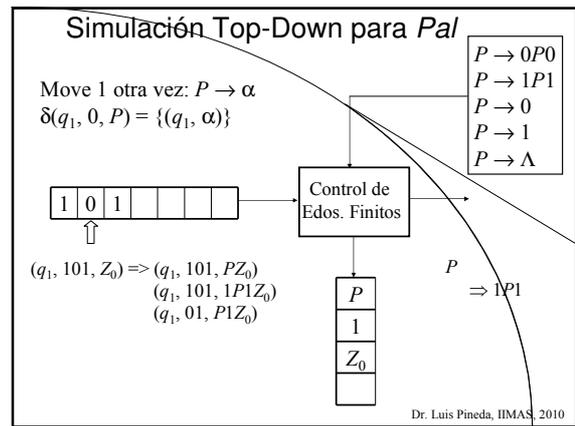
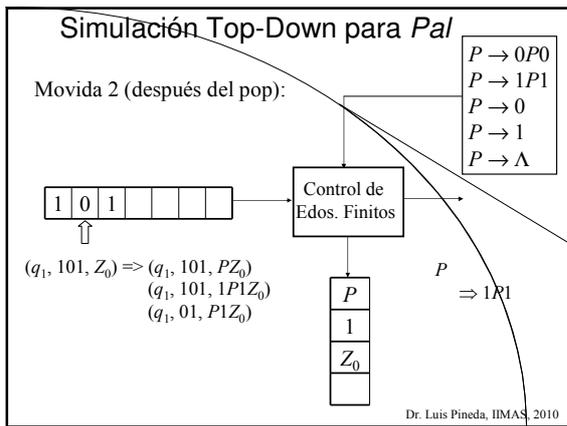
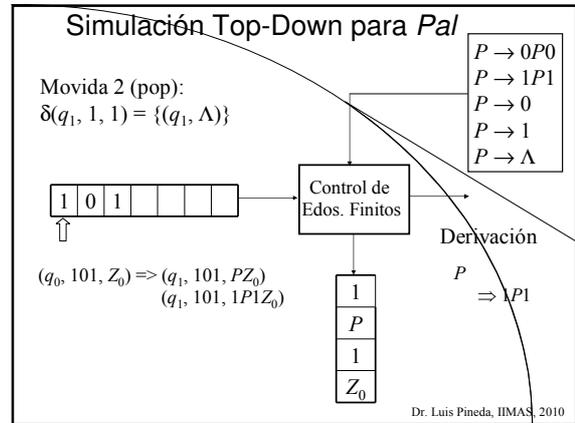
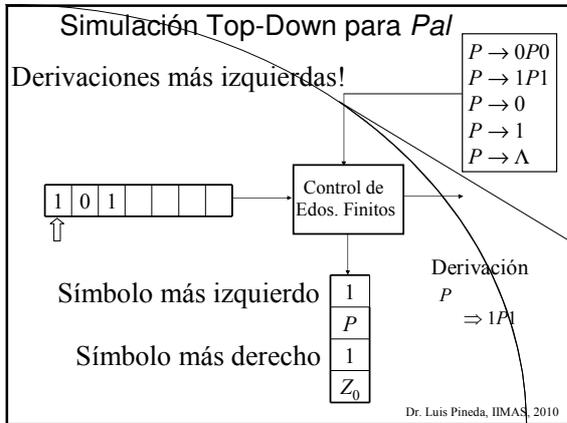
### Simulación Top-Down para Pal

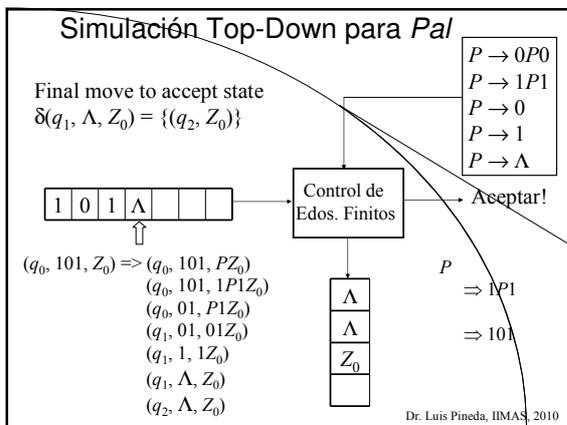
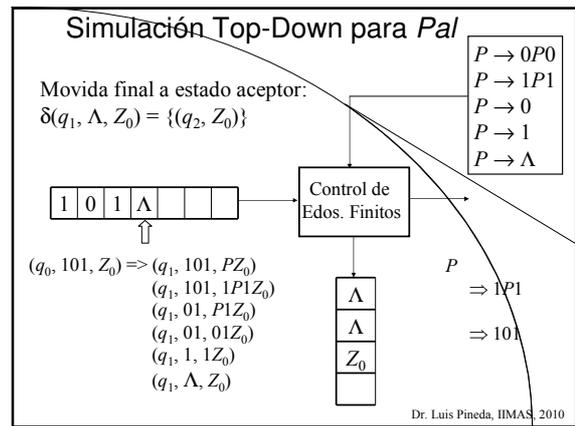
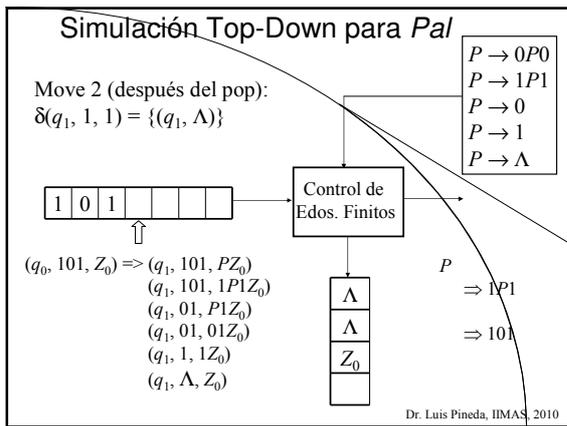
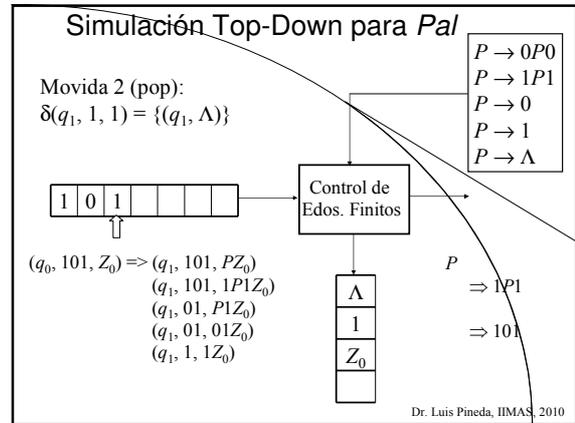
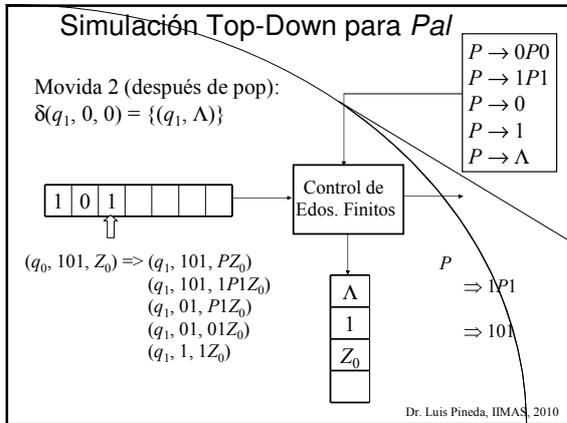


### Simulación Top-Down para Pal





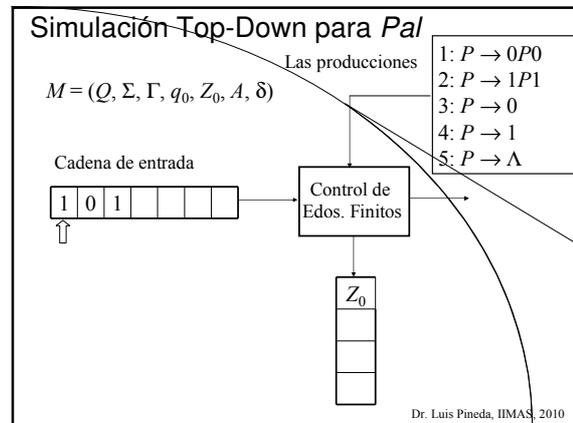
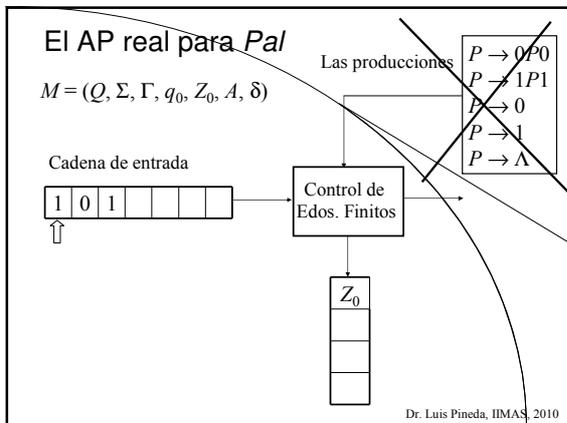




### Proceso Top-Down 101

cadena	edo.	símb.	Stack	movida	Conf.	Producción
101	$q_0$	$\Lambda$	$Z_0$	0: $(q_1, P)$	$(q_0, 101, Z_0)$	
101	$q_1$	$\Lambda$	$P$	1: $(q_1, 1P1)$	$(q_1, 101, PZ_0)$	$P \rightarrow 1P1$
101	$q_1$	1	1	2: $(q_1, \Lambda)$	$(q_1, 101, 1P1Z_0)$	
01	$q_1$	$\Lambda$	$P$	1: $(q_1, 0)$	$(q_1, 01, P1Z_0)$	$P \rightarrow 0$
01	$q_1$	0	0	2: $(q_1, \Lambda)$	$(q_1, 01, 01Z_0)$	
1	$q_1$	1	1	2: $(q_1, \Lambda)$	$(q_1, 1, 1Z_0)$	
$\Lambda$	$q_1$	$\Lambda$	$Z_0$	3: $(q_2, Z_0)$	$(q_1, \Lambda, Z_0)$	
$\Lambda$	$q_2$	$\Lambda$	$Z_0$		$(q_2, \Lambda, Z_0)$	

Dr. Luis Pineda, IIMAS, 2010



### El PD legal para $Pal$

Id	Estado	Entrada	Símbolo stack	Movida(s)
1	$q_0$	$\Lambda$	$Z_0$	<del><math>(q_1, PZ_0)</math></del>
2	$q_1$	$\Lambda$	$P$	<del><math>(q_{11}, 0)</math></del>
3	$q_{11}$	$\Lambda$	$0$	<del><math>(q_{11}, P0)</math></del>
4	$q_{11}$	$\Lambda$	$P$	<del><math>(q_{12}, 0P)</math></del>
...	...	...	...	...
$n-1$	$q_1$	$0$	$0$	$(q_1, \Lambda)$
$n-2$	$q_1$	$1$	$1$	$(q_1, \Lambda)$
$n$	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
Otras combinaciones				nada

Pop  $P$   
Push  $0P0$

Movidas para producción 1:  $P \rightarrow 0P0$

Dr. Luis Pineda, IIMAS, 2010

### El PD específico: no-determinístico

Id	Estado	Entrada	Símbolo stack	Movida(s)
1	$q_0$	$\Lambda$	$Z_0$	$(q_1, PZ_0)$
2	$q_1$	$\Lambda$	$P$	$\{(q_{11}, 0), (q_{12}, 1), (q_1, 0), (q_1, 1), (q_1, \Lambda)\}$
3	$q_{11}$	$\Lambda$	$0$	$(q_{11}, P0)$
4	$q_{11}$	$\Lambda$	$P$	$(q_{11}, 0P)$
5	$q_{12}$	$\Lambda$	$1$	$(q_{12}, P1)$
6	$q_{12}$	$\Lambda$	$P$	$(q_1, 1P)$
7	$q_1$	$0$	$0$	$(q_1, \Lambda)$
8	$q_1$	$1$	$1$	$(q_1, \Lambda)$
9	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
Otras combinaciones				nada

Movidas para todas las producciones con  $P$

Dr. Luis Pineda, IIMAS, 2010

- ### Simulación Top-down
- Un paso de la simulación corresponde a la construcción de una porción del árbol de derivación.
  - Top-down: se expande el símbolo inicial  $S$  hasta la cadena de entrada
  - La secuencia de configuraciones del AP corresponde a una derivación más izquierda en la gramática!
- Dr. Luis Pineda, IIMAS, 2010

- ### AP para $GLC$
- Cada paso de la simulación corresponde a la construcción de una parte del árbol de derivación; hay dos formas:
    - ✓ Top-down: expandir el símbolo inicial  $S$  hasta generar la cadena (con derivaciones más izquierdas)
    - Bottom-up: Construir el árbol hacia arriba a partir de la cadena  $x$  hasta el símbolo inicial  $S$  (derivaciones más derechas)
- Dr. Luis Pineda, IIMAS, 2010

## AP para GLC

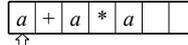
- Cada paso de la simulación corresponde a la construcción de una parte del árbol de derivación; hay dos formas:
  - ✓ Top-down: expandir el símbolo inicial  $S$  hasta generar la cadena (con derivaciones más izquierdas)
  - Bottom-up: Construir el árbol hacia arriba a partir de la cadena  $x$  hasta el símbolo inicial  $S$  (derivaciones más derechas)

Dr. Luis Pineda, IIMAS, 2010

## Simulación bottom-up en Exp

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

Cadena de entrada



Las producciones

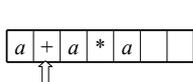
$S \rightarrow S + T$
$S \rightarrow T$
$T \rightarrow T * a$
$T \rightarrow a$

La máquina chequea si el lado derecho de una producción corresponde a una subcadena hasta arriba del stack; si es el caso, substituye dicha subcadena por la variable en el lado izquierdo de la producción: *reduce*.  
O, alternativamente, push el símbolo actual en la cinta: *shift*

Dr. Luis Pineda, IIMAS, 2010

## Simulación bottom-up en Exp

Movida 1: Shift



Las producciones

$S \rightarrow S + T$
$S \rightarrow T$
$T \rightarrow T * a$
$T \rightarrow a$

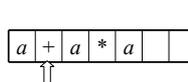
En la simulación bottom-down los símbolos de la cadena si visitan al stack!



Dr. Luis Pineda, IIMAS, 2010

## Simulación bottom-up en Exp

Movida 2: Reduce



Las producciones

$S \rightarrow S + T$
$S \rightarrow T$
$T \rightarrow T * a$
$T \rightarrow a$

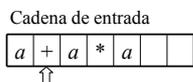
El top del stack corresponde al lado derecho de una producción:  $a$



Dr. Luis Pineda, IIMAS, 2010

## Simulación bottom-up en Exp

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



Las producciones

$S \rightarrow S + T$
$S \rightarrow T$
$T \rightarrow T * a$
$T \rightarrow a$

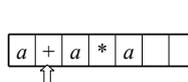
No-determinismo: *Shift* o *Reduce*



Dr. Luis Pineda, IIMAS, 2010

## Simulación bottom-up en Exp

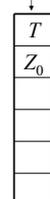
Move 2: Reduce!



Las producciones

$S \rightarrow S + T$
$S \rightarrow T$
$T \rightarrow T * a$
$T \rightarrow a$

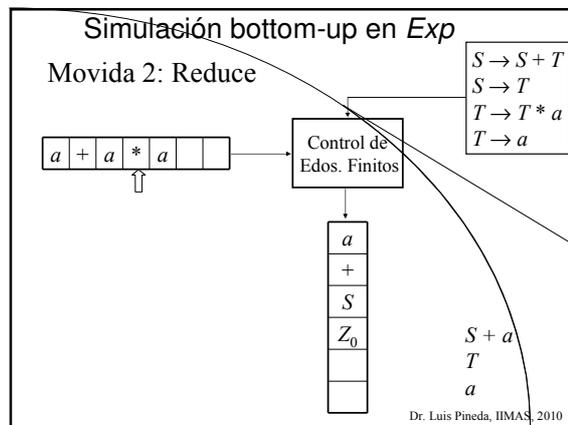
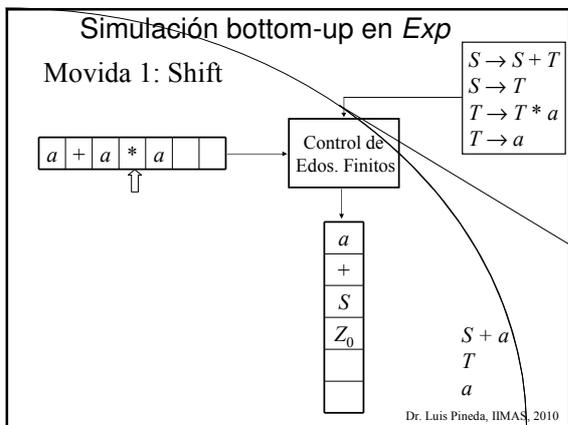
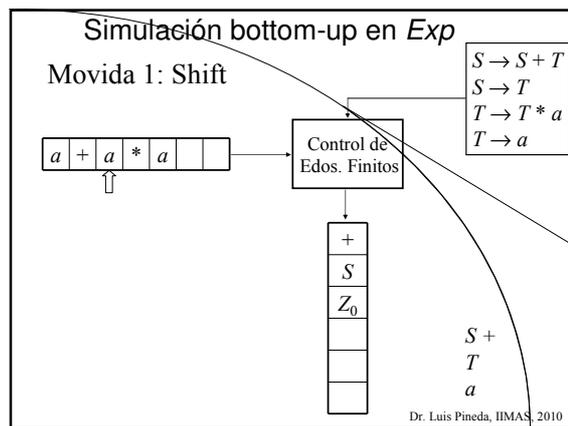
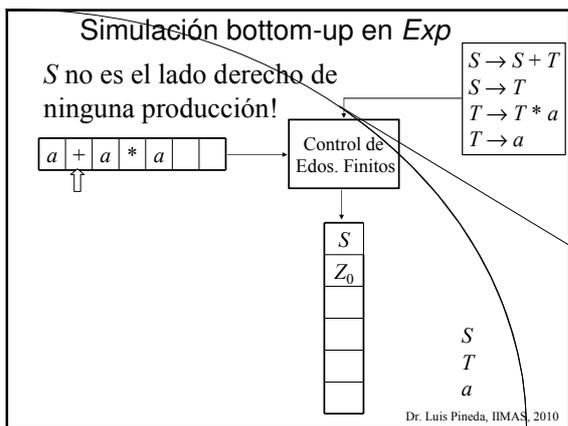
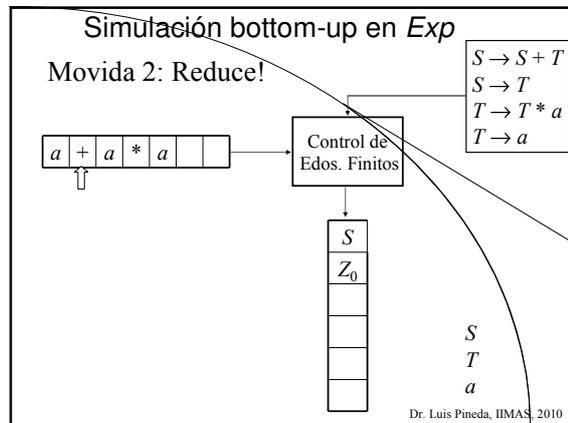
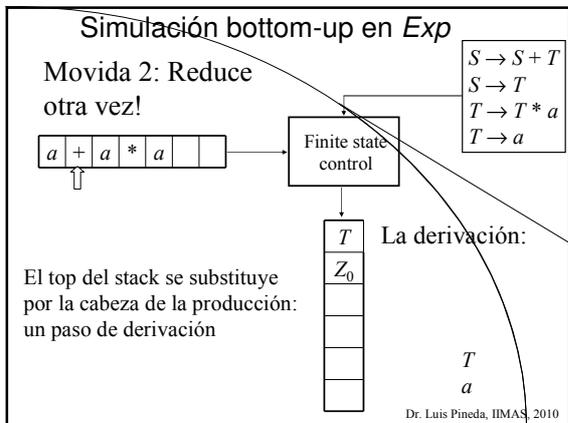
El top del stack se substituye por la cabeza de la producción: un paso de derivación

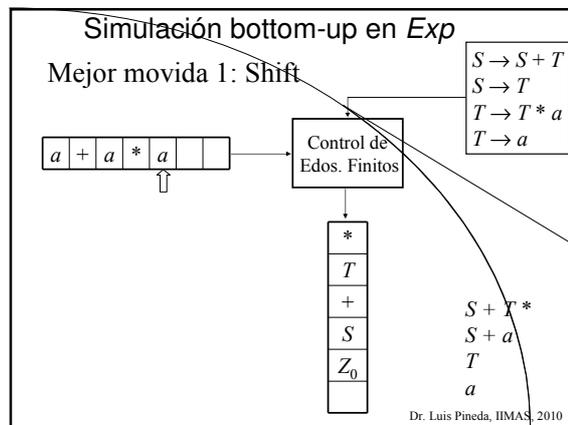
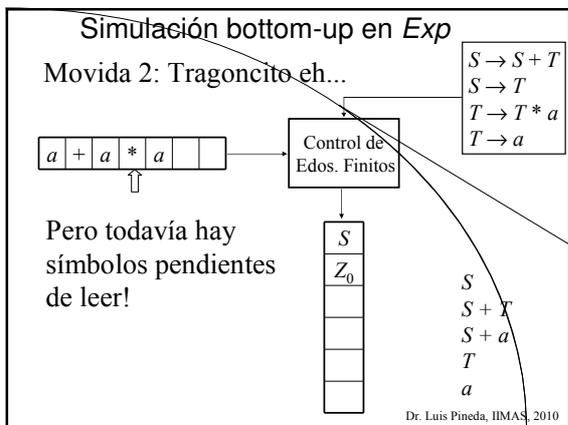
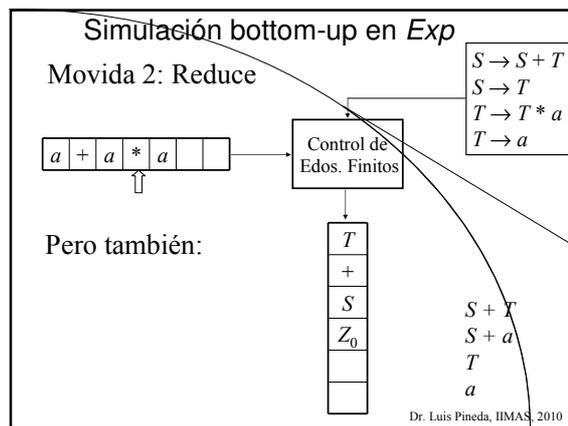
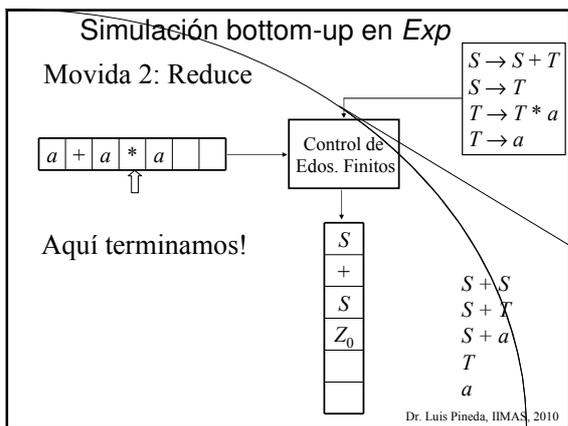
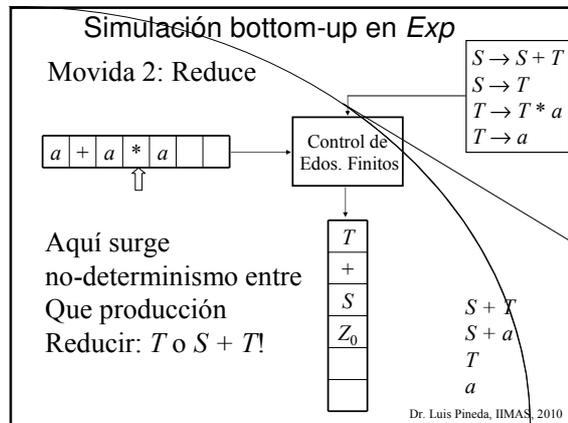
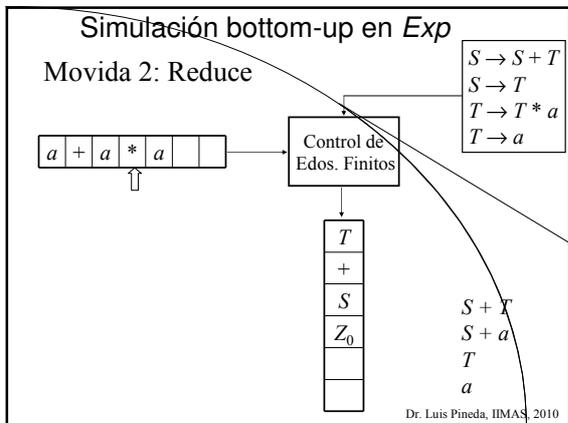


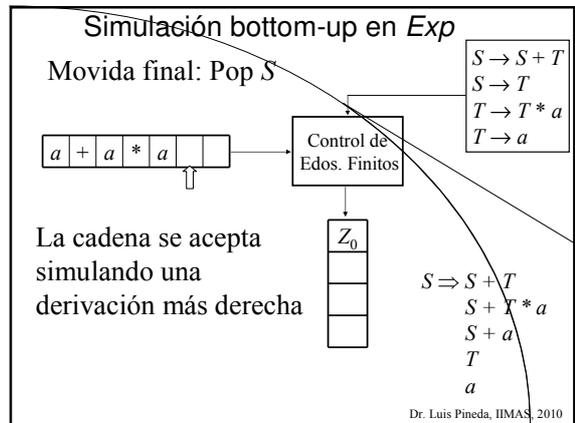
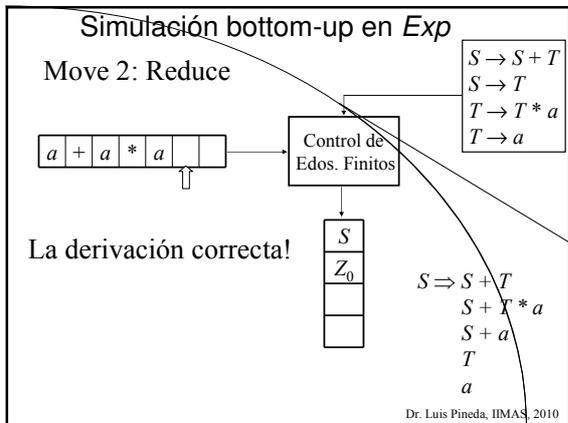
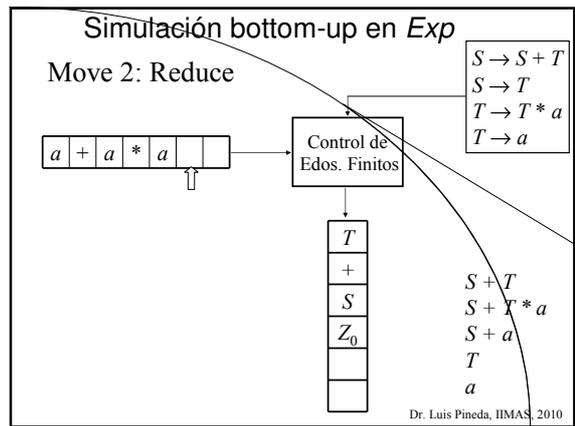
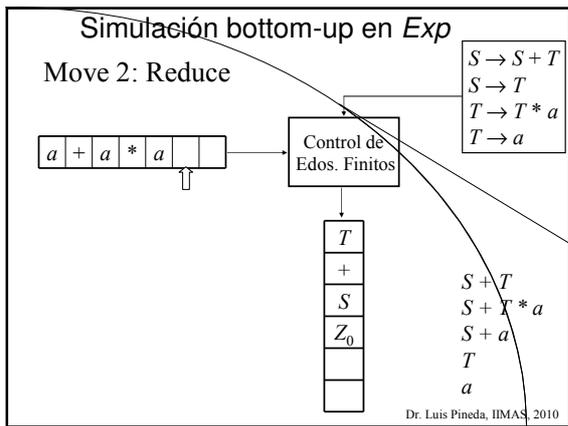
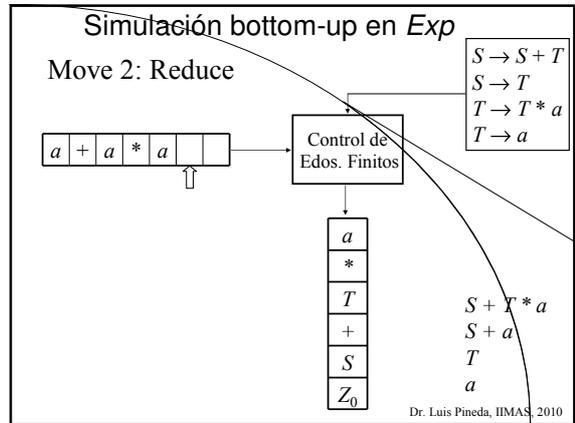
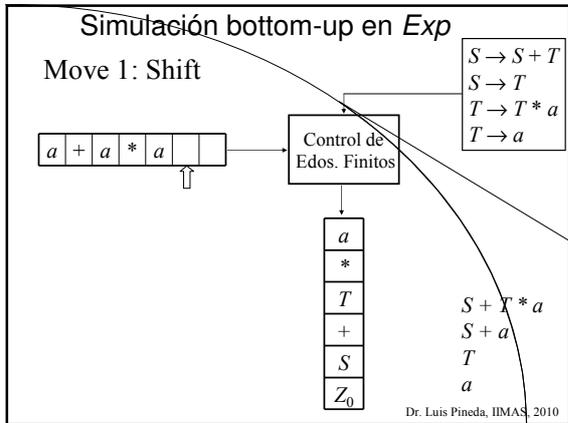
La derivación:

$T$   
 $a$

Dr. Luis Pineda, IIMAS, 2010







### Proceso bottom-up de $a + a * a$

Id	Movida	Producción	Stack	por leer
1			$Z_0$	$a + a * a$
2	Shift		$aZ_0$	$+ a * a$
3	reduce	$T \rightarrow a$	$TZ_0$	$+ a * a$
4	reduce	$S \rightarrow T$	$SZ_0$	$+ a * a$
5	Shift		$+SZ_0$	$a * a$
6	Shift		$a+SZ_0$	$* a$
7	reduce	$T \rightarrow a$	$T+SZ_0$	$* a$
8	Shift		$*T+SZ_0$	$a$
9	Shift		$a*T+SZ_0$	-
10	reduce	$T \rightarrow T * a$	$T+SZ_0$	-
11	reduce	$S \rightarrow S + T$	$SZ_0$	-
12	(pop S)		$Z_0$	-
13	accept			-

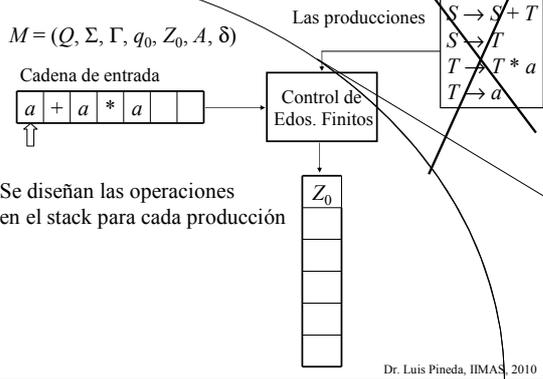
Dr. Luis Pineda, IIMAS, 2010

### Simulación bottom-up

- Un paso de la simulación corresponde a la construcción de una porción del árbol de derivación.
- Bottom-up: Construir el árbol hacia arriba a partir de la cadena  $x$  y hasta el símbolo inicial  $S$
- La secuencia de configuraciones en el PDA corresponde a una derivación más derecha en la gramática!

Dr. Luis Pineda, IIMAS, 2010

### El AP bottom-up real para $Exp$



Dr. Luis Pineda, IIMAS, 2010

### AP para $GLC$

- Cada paso de la simulación corresponde a la construcción de una parte del árbol de derivación; hay dos formas:
  - ✓ Top-down: expandir el símbolo inicial  $S$  hasta generar la cadena (con derivaciones más izquierdas)
  - ✓ Bottom-up: Construir el árbol hacia arriba a partir de la cadena  $x$  hasta el símbolo inicial  $S$  (derivaciones más derechas)

Dr. Luis Pineda, IIMAS, 2010

### AP correspondiente a $GLC$

- ✓ Existe un AP  $M$  tal que  $L(M) = L(G)$  para toda  $GLC G$
- Existe una  $CFG G$  tal que  $L(G) = L(M)$  para todo AP  $M$

Dr. Luis Pineda, IIMAS, 2010