

Sesión 19

Autómata de pila determinístico

Dr. Luis Pineda, IIMAS, 2010

Tres versiones de *Pal*

- $Pal_{marca} = \{xcx^r \mid x \in \{0, 1\}^*\}$
 - 0110c0110
- $Pal_{par} = \{xx^r \mid x \in \{0, 1\}^*\}$
 - 00111100
- $Pal = \{x \mid x = x^r \in \{0, 1\}^*\}$
 - 00111100
 - 001101100

Dr. Luis Pineda, IIMAS, 2010

Una máquina para aceptar *Pal*

- El lenguaje:

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

- Definir M_{pal} :

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

- Dos tipos de no-determinismo:

- Hay más de una movida para un estado, símbolo de entrada y top del stack
- Hay estados en los que la máquina tienen que elegir entre consumir un símbolo o hacer una transición- Λ

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	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				nada

Dr. Luis Pineda, IIMAS, 2010

Más de un siguiente estado

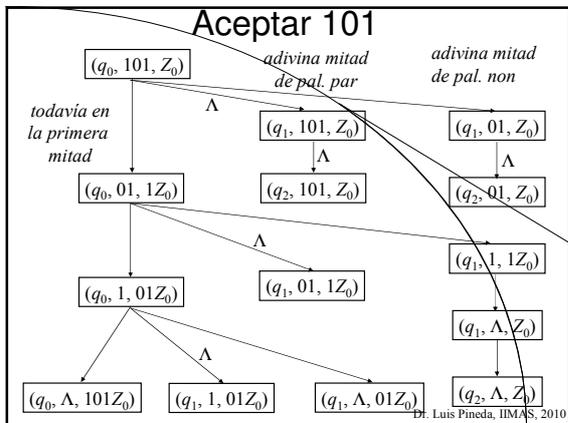
Id	Estado	Entrada	Símbolo stack	Movida(s)
1	q_0	0	Z_0	$(q_0, 0Z_0), (q_1, Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0), (q_1, Z_0)$
3	q_0	0	0	$(q_0, 00), (q_1, 0)$
4	q_0	1	0	$(q_0, 10), (q_1, 0)$
5	q_0	0	1	$(q_0, 01), (q_1, 1)$
6	q_0	1	1	$(q_0, 11), (q_1, 1)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
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				nada

Dr. Luis Pineda, IIMAS, 2010

Transición- Λ

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	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				nada

Dr. Luis Pineda, IIMAS, 2010



Una máquina para acepta Pal_{par}

- El lenguaje:

$$Pal_{par} = \{xx^r \mid x \in \{0, 1\}^*\}$$
- Definir $M_{pal-par}$:

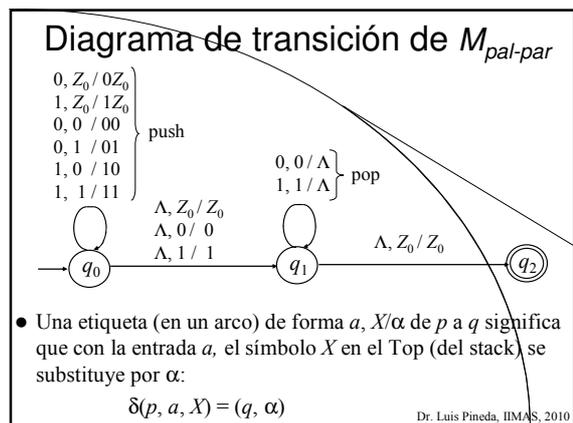
$$M_{pal-par} = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, q_0, Z_0, \{q_2\}, \delta)$$
- Sólo un tipo de no-determinismo:
 - Hay estados en los que la máquina tienen que elegir entre consumir un símbolo o hacer una transición- Λ

Dr. Luis Pineda, IIMAS, 2010

Tabla de transición para $M_{pal-par}$

Id	Estado	Entrada	Símbolo 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				nada

Dr. Luis Pineda, IIMAS, 2010



Elección de transición- Λ

Id	Estado	Entrada	Símbolo 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				nada

Dr. Luis Pineda, IIMAS, 2010

Elección de transición- Λ

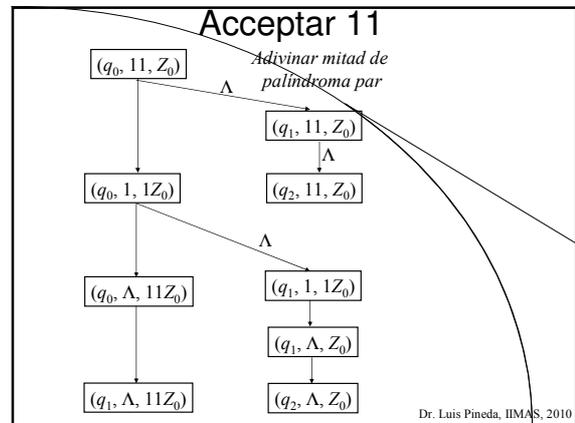
Id	Estado	Entrada	Símbolo stack	Movida(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2,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

Dr. Luis Pineda, IIMAS, 2010

Elección de transición- Λ

Id	Estado	Entrada	Símbolo 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				nada

Dr. Luis Pineda, IIMAS, 2010



Una máquina para aceptar $M_{pal-marca}$

- El lenguaje:

$$Pal_{marca} = \{xcx^r \mid x \in \{0, 1\}^*\}$$
- Definir $M_{pal-marca}$:

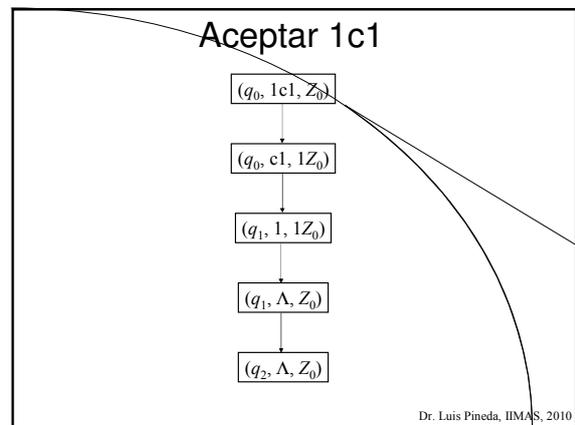
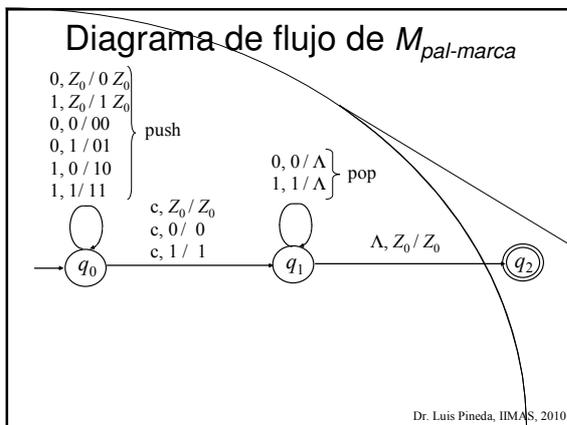
$$M_{pal-mark} = (\{q_0, q_1, q_2\}, \{0, 1, c\}, \{0, 1, Z_0\}, q_0, Z_0, \{q_2\}, \delta)$$
- ¡No hay no-determinismo!

Dr. Luis Pineda, IIMAS, 2010

Función de transición para $M_{pal-marca}$

Id	Estado	Entrada	Símbolo 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	c	Z_0	(q_1, Z_0)
8	q_0	c	0	$(q_1, 0)$
9	q_0	c	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				nada

Dr. Luis Pineda, IIMAS, 2010



Definición de AP-Determinístico

- Sea $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ un AP. M es determinístico si y sólo si no existen configuraciones en las que se pueda escoger más de una movida. Para esto M tiene que satisfacer dos condiciones:
 1. Para todo $q \in Q, a \in \Sigma \ \& \ X \in \Gamma$, el valor de $\delta(q, a, X)$ tiene cuando más un elemento
 2. Para todo $q \in Q \ \& \ X \in \Gamma$, Si $\delta(q, \Lambda, X) \neq \Phi$, entonces $\delta(q, a, X) = \Phi$ para todo $a \in \Sigma$

Dr. Luis Pineda, IIMAS, 2010

Definición de AP-Determinístico

- Un lenguaje L es un lenguaje libre de contexto determinístico (LLC-D) si existe un AP-Determinístico (AP-D) que acepta a L

Dr. Luis Pineda, IIMAS, 2010

Definición de AP-Determinístico

- Sea $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ un AP-D. M es determinístico si y sólo si:
 - Una movida como máximo en cada configuración
 - No elección entre consumir un símbolo o hacer una transition- Λ

Dr. Luis Pineda, IIMAS, 2010

Clases de no-determinismo

- ¿En el lenguaje?
 - $L_{pal} = \{x \mid x = x^r \in \{0, 1\}^*\}$
 - $Pal_{par} = \{xx^r \mid x \in \{0, 1\}^*\}$
 - $Pal_{marca} = \{xcx^r \mid x \in \{0, 1\}^*\}$
- ¿Depende de la forma del lenguaje en sí o del formalismo para expresar al lenguaje?

Dr. Luis Pineda, IIMAS, 2010

Clases de no-determinismo

- ¿En la gramática?
 - $G_{pal} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid 0 \mid 1 \mid \Lambda)$
 - $G_{pal-par} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid \Lambda)$
 - $G_{pal-marca} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid c)$
- Dos decisiones:
 - Qué variable re-escribir
 - Qué producción utilizar

Dr. Luis Pineda, IIMAS, 2010

Clases de no-determinismo

- $G_{pal-marca}$
 - $G_{pal-marca} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid c)$
 - Una sólo variable
 - Dos producciones, pero sólo una es relevante para generar o reconocer una cadena dada (i.e. en un proceso de parseo, la producción usada en la derivación corresponde con un símbolo terminal en el lado derecho de la producción)

Dr. Luis Pineda, IIMAS, 2010

Clases de no-determinismo

- Pero en G_{pal}
 - $G_{pal} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid 0 \mid 1 \mid \Lambda)$
- ¿Qué producción se utiliza para generar cierto símbolo en la derivación?
 - 001100 $P \rightarrow 0P0$ o $P \rightarrow 0$?
 - 001 Λ 100 $P \rightarrow \Lambda$ se utiliza, ¿pero cuándo?

Dr. Luis Pineda, IIMAS, 2010

¿Eliminando el no-determinismo?

- Eliminar las producciones- Λ :
 - $G_{pal} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid 0 \mid 1 \mid 00 \mid 11)$
 - $G_{pal-par} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid 00 \mid 11)$
- No es suficiente!

Dr. Luis Pineda, IIMAS, 2010

¿Eliminando el no-determinismo?

- Para una variable en la cabeza de la producción se tienen todavía varios lados derechos que producen el mismo símbolo en la derivación:

$$P \rightarrow 0P0 \mid 0 \mid 00$$
- Al inspeccionar una cadena no podemos saber que producción se usó para generar un símbolo específico usando tan sólo información local (i.e. el estado, el símbolo de entrada y el símbolo en hasta arriba de la pila).
- Abstracción disyuntiva en las producciones!

Dr. Luis Pineda, IIMAS, 2010

El AP para Pal_{marca} es determinístico!

- $Pal_{marca} = \{xcx^r \mid x \in \{0, 1\}^*\}$
 - $G_{pal-mark} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid c)$
- No hay producciones que introduzcan no-determinismo:
 - $P \rightarrow 0P0 \mid 1P1$: Un push y un pop!
 - $P \rightarrow c$: Llegando a la mitad
- Ninguna producción con la misma cabeza (variable del lado izquierdo) se reescribe con el mismo símbolo más izquierdo del lado derecho!

Dr. Luis Pineda, IIMAS, 2010

AP-D & Lenguajes regulares

- Un AP-D puede simular un FA:
 - Definir una AP que sólo use sus estados (i.e. NO utiliza a su stack!)
 - Si $A = (Q, \Sigma, q_0, A, \delta_A)$ es un FA entonces A tiene un AP-D equivalente

$$P = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta_P)$$
 tal que

$$\delta_P(q, a, Z_0) = \{(p, Z_0)\}$$
 para todos los estado p & q en Q tales que:

$$\delta_A(q, a) = p$$

Dr. Luis Pineda, IIMAS, 2010

LLG-Determinísticos

- Pal_{marca} es un LLC-D (tiene un AP-D correspondiente)
- Pal_{marca} no es un LR:
 - El lema del bombeo: Considerar $w = 0^n c 0^n$: escoger uv como dos grupos de 0's de la primera mitad, bombear el grupo v , y la cadena resultante no está en el lenguaje!

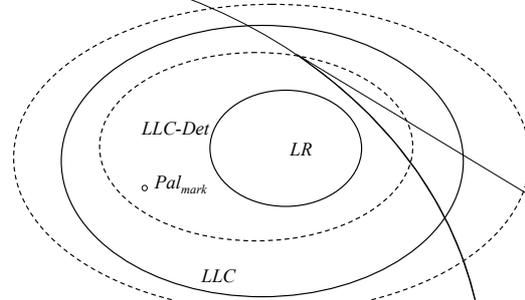
Dr. Luis Pineda, IIMAS, 2010

LLC-Determinísticos

- Por otro lado, Pal & Pal_{pal} son LLC para los que no existe in AP-D equivalente
- La jerarquía del determinismo:
 - LLC incluye propiamente a LLC-Det
 - LLC-Det incluye propiamente a LR

Dr. Luis Pineda, IIMAS, 2010

La clase de los LLC Determinísticos



- ¿Son equivalentes la clase de los LLC-Det y la clase de los LLC no-ambiguos?

Dr. Luis Pineda, IIMAS, 2010

Determinismo & ambigüedad en LLCs

- $L_{pal} = \{x \mid x = x^r \in \{0,1\}^*\}$
 - $G_{pal} = (\{P\}, \{0, 1\}, P, R)$

$$R \rightarrow \{0P0 \mid 1P1 \mid 0 \mid 1 \mid 00 \mid 11\}$$
- El AP para G_{pal} es no-determinístico!
- Pero G_{pal} no es ambiguo: Las derivaciones más izquierdas son únicas!

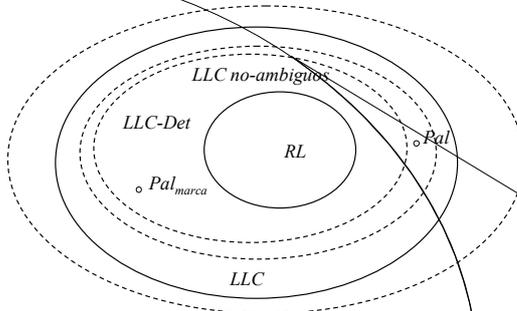
Dr. Luis Pineda, IIMAS, 2010

Determinismo & ambigüedad en LLCs

- Hay muchas derivaciones pero sólo una estructura sintáctica:
 - Podemos hacer varias elecciones en cada paso, pero todas las equivocadas van a morir eventualmente antes de llegar al estado aceptor o a vaciar el stack
 - En cada paso de derivación (más izquierda o más derecha) podemos seleccionar más de una producción, pero sólo una derivación producirá la cadena final!
 - ¡Sólo hay una estructura sintáctica para cada cadena!

Dr. Luis Pineda, IIMAS, 2010

Determinismo & ambigüedad en LLCs



- ¿Son equivalentes la clase de los LLC-Det y la clase de los LLC no-ambiguos?: ¡NO!

Dr. Luis Pineda, IIMAS, 2010