

Tema 23

Parseo Top-down

Dr. Luis A. Pineda
ISBN: 970-32-2972-7

Proceso de parseo

- Del Latin: Partes del habla o categorías gramaticales
- Si G es una CFG sobre Σ & $x \in \Sigma^*$, parsear x consiste en encontrar una derivación para x en G o determinar que ésta no existe (i.e. por lo que x no está en el lenguaje de G).

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Proceso de parseo

- Antecedentes: simulación de derivaciones en G por un a PDA:
 - Top-down (derivaciones más izquierdas)
 - Bottom-up (derivaciones más derechas)
- Pero, estos procesos son no determinísticos y no proveen de un algoritmo directamente!

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Algoritmos de parseo

- Algoritmo trivial: explorar todas las trayectorias (en el espacio no determinístico) en cierto orden (i.e. depth first o breath first) & ver si alguna trayectoria termina exitosamente
 - Muy costoso: búsqueda exponencial!

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Algoritmos de parseo

- Enfrentar el no-determinismo directamente:
 - Utilizar toda la información disponible en cada paso de la computación (local) y seleccionar la mejor alternativa (en algunos casos determinísticamente)
 - Capitalizar la forma de la gramática: Para seleccionar la siguiente movida considerar no sólo el símbolo hasta arriba del stack, sino también uno o más de los símbolos que siguen en la cinta!

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Top-down *Lookahead* Parser

- Movida 1:
 - Si el símbolo hasta arriba del stack es una variable en el lado izquierdo de una producción, reemplazar dicha variable por el lado derecho de la producción
 - Inspeccionar el símbolo actual (i.e. consumir) y escoger sólo producciones que tengan a dicho símbolo como el más izquierdo del lado derecho!
 - Usar no-determinismo sólo si no hay opción!

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Top-down *Lookahead* Parser

- Movida 2:
 - Si el símbolo de entrada corresponde con el símbolo hasta arriba del stack, consumir el símbolo y pop.

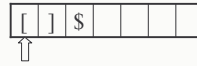
Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Simulación top-down

$L =$ El lenguaje de los paréntesis balanceados

$S \rightarrow T\$$
 $T \rightarrow [T]T$
 $T \rightarrow \Lambda$

Cadena de entrada



Control de edos. finitos

Una gramática no-ambigua

La lógica de la máquina:
 Cancelar “[” en la cinta con “[” en el lado derecho de una producción; igual para “]”



Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

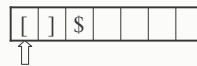
AP no determinístico

Id	estado	entrad	top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TZ_0)
3	q_1	Λ	T	$(q_1, [T]T), (q_1, \Lambda)$
4	q_1	[[(q_1, Λ)
5	q_1]]	(q_1, Λ)
6	q_1	\$	\$	(q_1, Λ)
7	q_1	Λ	Z_0	(q_2, Z_0)
Otras combinaciones				ninguna

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Simulación top-down

$S \rightarrow T\$$
 $T \rightarrow [T]T$
 $T \rightarrow \Lambda$



Control de edos. finitos

Movida 0:



S

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

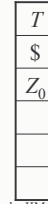
Simulación top-down

$S \rightarrow T\$$
 $T \rightarrow [T]T$
 $T \rightarrow \Lambda$



Control de edos. finitos

Movida 1:

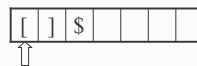


$S \Rightarrow T\$$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

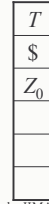
Simulación top-down

$S \rightarrow T\$$
 $T \rightarrow [T]T$
 $T \rightarrow \Lambda$



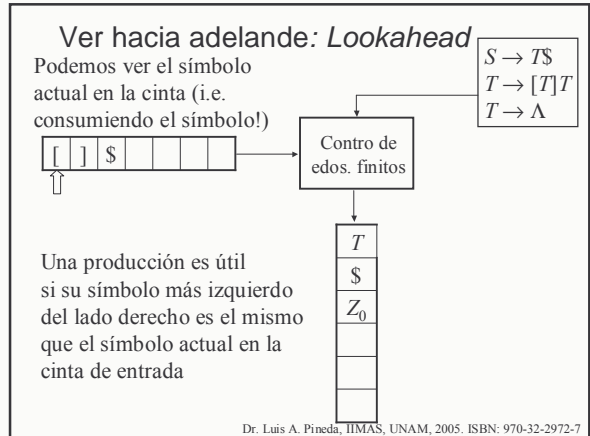
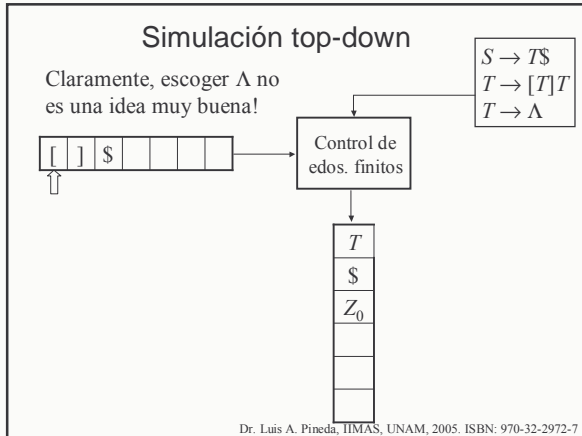
Control de edos. finitos

La decisión es no-determinística cuando T está hasta arriba del stack: podemos escoger $[T]T$ o Λ



$S \Rightarrow T\$$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7



- ### ¡Ver hacia adelante!
- Remover el no-determinismo inspeccionando un símbolo hacia adelante
 - Sólo considerar producciones que tengan a dicho símbolo como el más izquierdo del lado derecho de la producción
 - Diseñar una AP determinístico que corresponda con el AP no-determinístico original
- Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

AP lookahead determinístico

Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	[T	$(q_1, [T]T)$
4	q_1	Λ	[(q_1, Λ)
5	q_1]	T	(q_1, Λ)
6	q_1	Λ]	(q_1, Λ)
7	q_1	S	T	(q_5, Λ)
8	q_5	Λ	S	(q_1, Λ)
9	q_1	[[(q_1, Λ)
10	q_1]]	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
otras combinaciones				ninguna

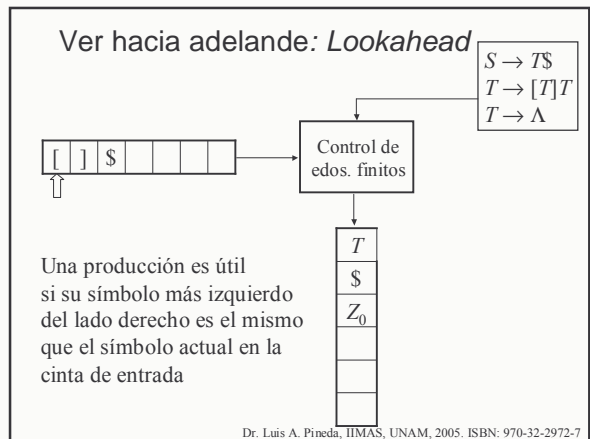
Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

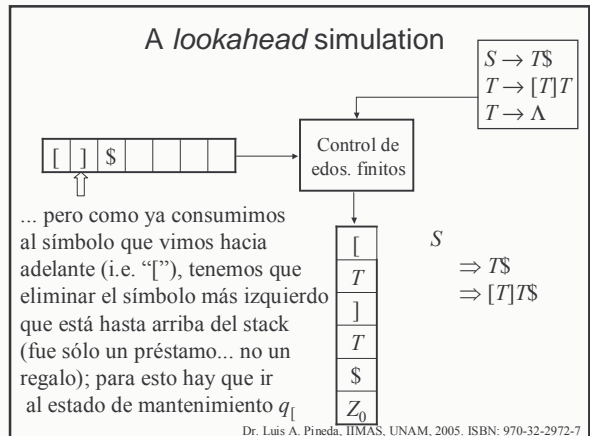
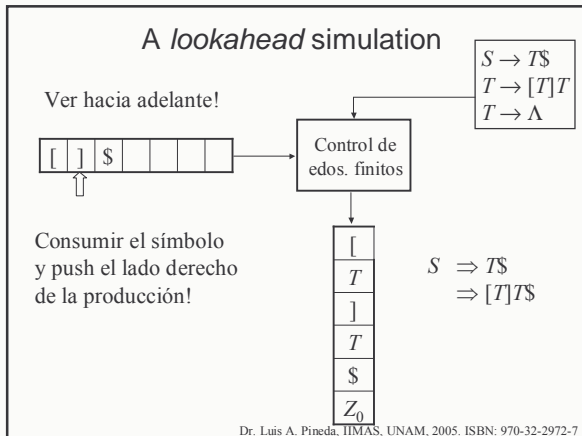
Consume lookahead symbol

$T \rightarrow [T]T$

Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	[T	$(q_1, [T]T)$
4	q_1	Λ	[(q_1, Λ)
5	q_1]	T	(q_1, Λ)
6	q_1	Λ]	(q_1, Λ)
7	q_1	S	T	(q_5, Λ)
8	q_5	Λ	S	(q_1, Λ)
9	q_1	[[(q_1, Λ)
10	q_1]]	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
otras combinaciones				ninguna

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7



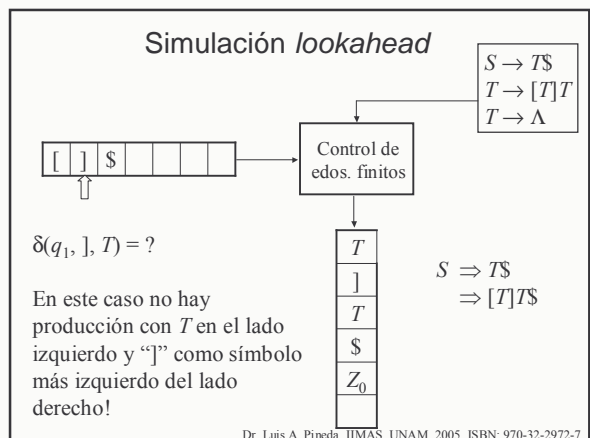
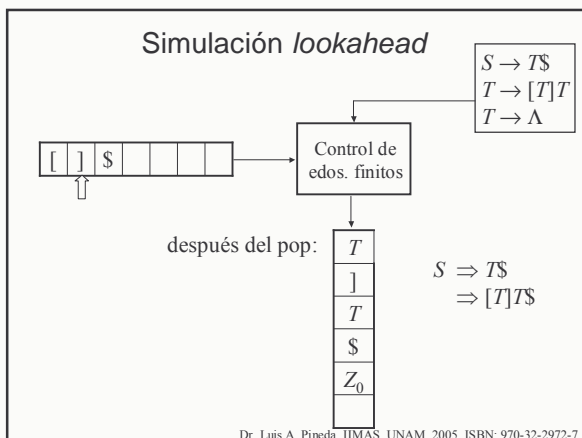
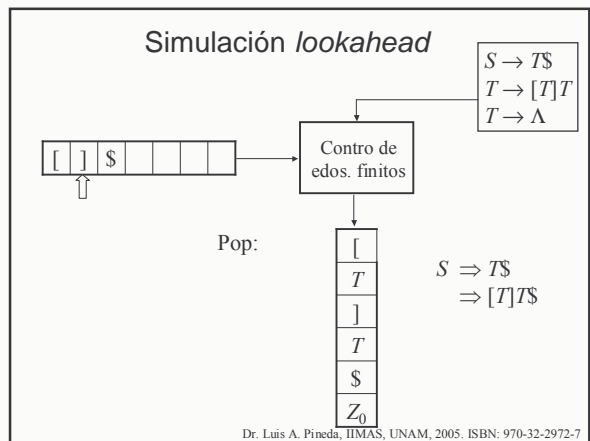


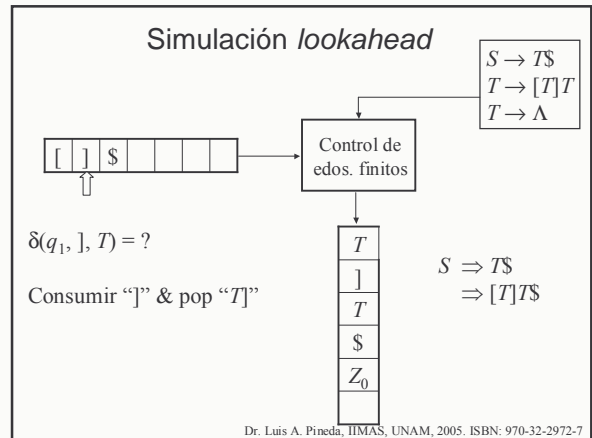
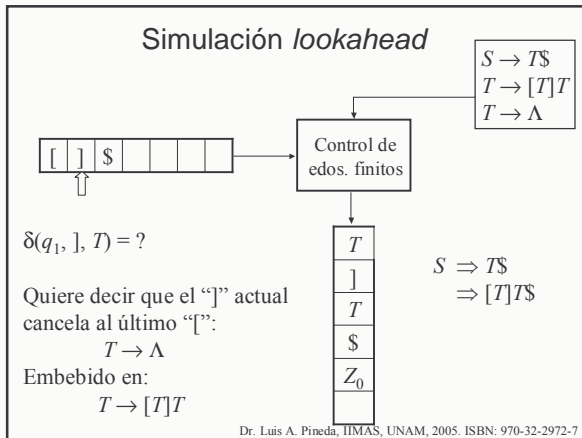
Consume lookahead symbol

Id	State	Input	Stack symbol	Move(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	[T	$(q_1, [T]T)$
4	q_1	Λ	[(q_1, Λ)
5	q_1]	T	(q_1, Λ)
6	q_1	Λ]	(q_1, Λ)
7	q_1	S	T	(q_s, Λ)
8	q_s	Λ	S	(q_1, Λ)
9	q_1	[[(q_1, Λ)
10	q_1]]	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Other combinations				non

$T \rightarrow [T]T$

Dr. Luis A. Pineda, HIMAS, UNAM, 2005. ISBN: 970-32-2972-7

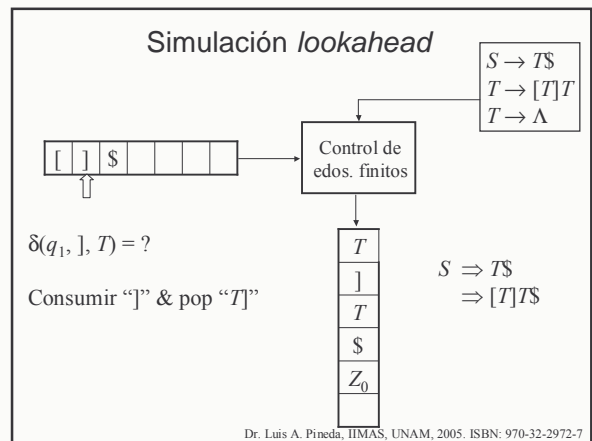




Pop "T]"

Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	[T	$(q_1, [T]T)$
4	q_1	Λ	[(q_1, Λ)
5	q_1]	T	(q_1, Λ)
6	q_1	Λ]	(q_1, Λ)
7	q_1	S	T	(q_s, Λ)
8	q_s	Λ	S	(q_1, Λ)
9	q_1	[[(q_1, Λ)
10	q_1]]	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
otras combinaciones				ninguna

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

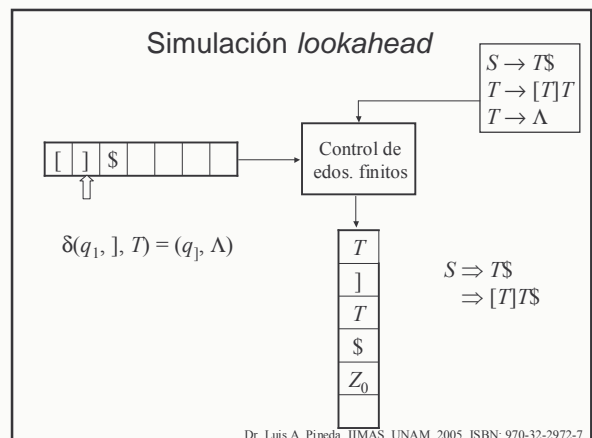


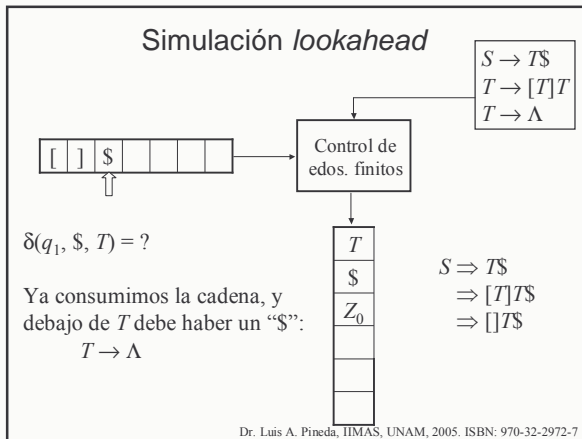
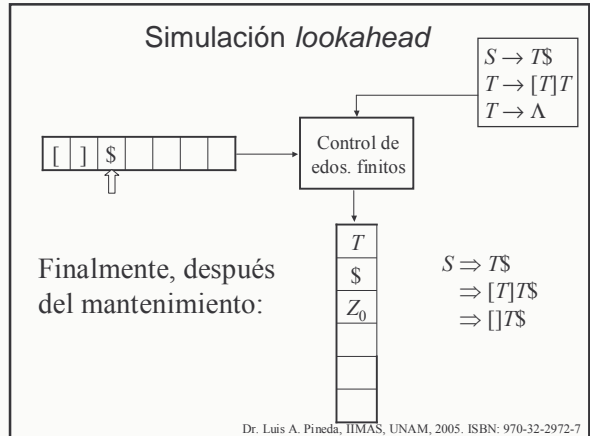
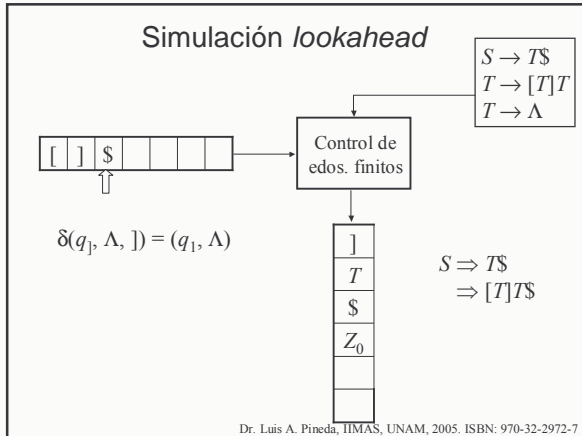
No hay entrada para $T \rightarrow \Lambda$

Id	estado	entrada	Top del stack	Movida(s)
3	q_1	[T	$(q_1, [T]T)$
X	q_1	Λ	T	(q_1, Λ)

No hay entrada en la tabla para la regla: $T \rightarrow \Lambda$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7



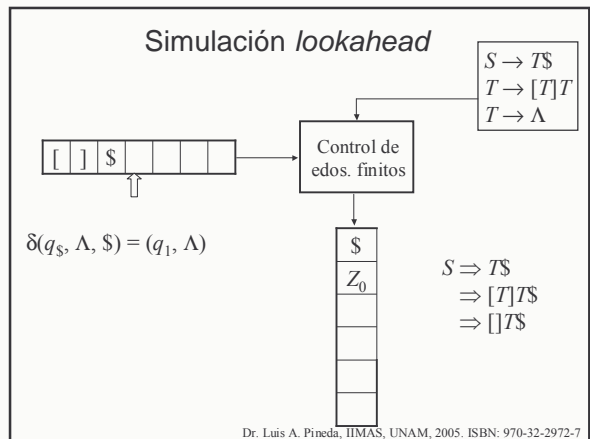
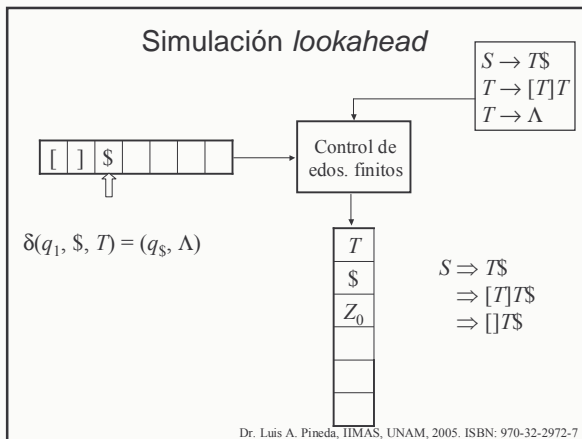


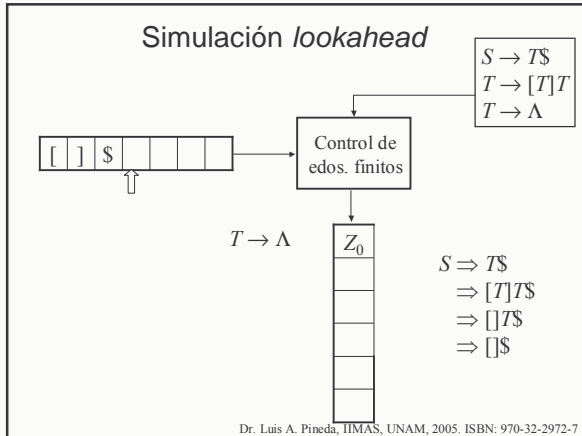
lookahead el symbol \$

Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	[T	$(q_1, [T]T)$
4	q_1	Λ	[(q_1, Λ)
5	q_1]	T	(q_1, Λ)
6	q_1	Λ]	(q_1, Λ)
7	q_1	S	T	(q_s, Λ)
8	q_s	Λ	S	(q_1, Λ)
9	q_1	[[(q_1, Λ)
10	q_1]]	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_s, Z_0)
otras combinaciones				ninguna

$T \rightarrow \Lambda$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7





Operaciones de pop

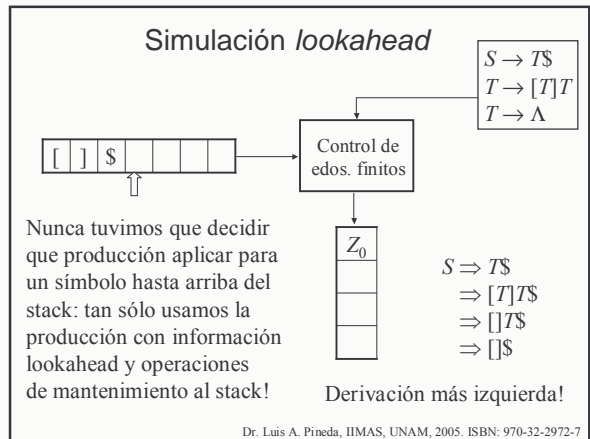
Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	$[$	T	$(q_1, [T]T)$
4	q_1	Λ	$[$	(q_1, Λ)
5	q_1	$]$	T	(q_1, Λ)
6	q_1	Λ	$]$	(q_1, Λ)
7	q_1	S	T	(q_5, Λ)
8	q_5	Λ	S	(q_1, Λ)
9	q_1	$[$	$[$	(q_1, Λ)
10	q_1	$]$	$]$	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
otras combinaciones				ninguna

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Estado final

Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	$[$	T	$(q_1, [T]T)$
4	q_1	Λ	$[$	(q_1, Λ)
5	q_1	$]$	T	(q_1, Λ)
6	q_1	Λ	$]$	(q_1, Λ)
7	q_1	S	T	(q_5, Λ)
8	q_5	Λ	S	(q_1, Λ)
9	q_1	$[$	$[$	(q_1, Λ)
10	q_1	$]$	$]$	(q_1, Λ)
11	q_1	S	S	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
otras combinaciones				ninguna

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7



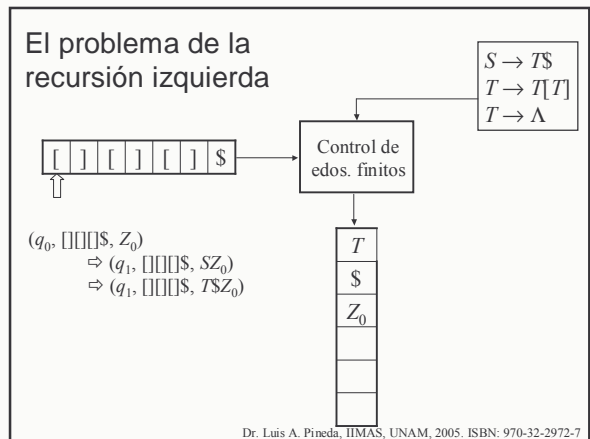
- ### El problema de la recursión izquierda
- Recursión izquierda:
 - Un problema con la simulación top-down lookahead:
 - Otra gramática no ambigua para L

$$S \rightarrow T\$$$

$$T \rightarrow T[T] \mid \Lambda$$
 - Una derivación más izquierda:

$$S \Rightarrow T\$ \Rightarrow T[T] \$ \Rightarrow T[T][T] \$ \Rightarrow T[T][T][T] \$ \dots$$

$$\dots \Rightarrow [][][] \$$$
 - Pero $T \rightarrow \Lambda$ realmente no existe en el AP determinístico!
- Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

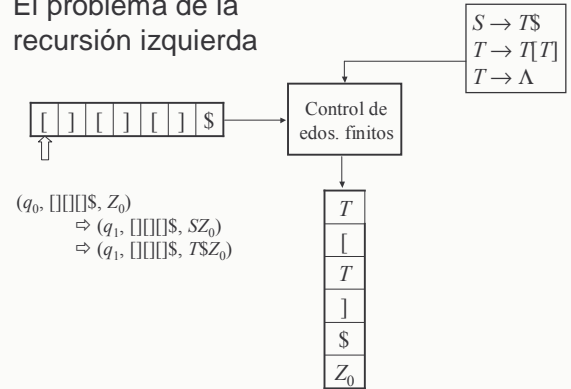


AP no-determinístico

Id	estado	entrada	Top del stack	Movida(s)
1	q_0	Λ	Z_0	(q_1, SZ_0)
2	q_1	Λ	S	(q_1, TSZ_0)
3	q_1	$[$	T	$(q_1, T[T])$
4	q_1	Λ	$[$	(q_1, Λ)
5	q_1	$]$	T	(q_1, Λ)
6	q_1	Λ	$]$	(q_1, Λ)
7	q_1	$\$$	T	(q_s, Λ)
8	q_s	Λ	$\$$	(q_1, Λ)
9	q_1	$[$	$[$	(q_1, Λ)
10	q_1	$]$	$]$	(q_1, Λ)
11	q_1	$\$$	$\$$	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
otras combinaciones				ninguna

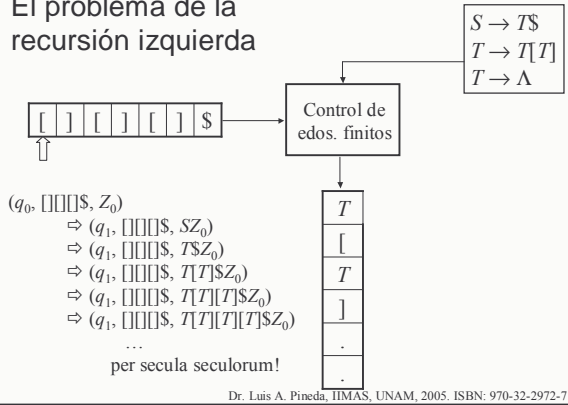
Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

El problema de la recursión izquierda



Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

El problema de la recursión izquierda



Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Eliminando la recursión izquierda

- La solución:
 - Encontra una gramática equivalente sin recursión izq.
- Considerar las producciones-T (β no empieza con T)

$$T \rightarrow T\alpha \mid \beta$$
- Es posible generar $\beta\alpha^n$:

$$T \rightarrow T\alpha, T \rightarrow T\alpha\alpha, \dots, T \rightarrow T\alpha^n$$
 pero $T \rightarrow \beta$ & $T \rightarrow \beta\alpha^n$
- T genera la concatenación de β con la cerradura de α :

$$T \rightarrow \beta U \text{ \& } U \rightarrow \alpha^n$$

$$T \rightarrow \beta U \text{ \& } U \rightarrow \alpha U \mid \Lambda$$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Eliminando la recursión izquierda

- La solución:
 - Encontra una gramática equivalente sin recursión izq.
- Considerar las producciones-T (β no empieza con T)

$$T \rightarrow T\alpha \mid \beta$$
- Reemplazar por:

$$T \rightarrow \beta U \text{ \& } U \rightarrow \alpha U \mid \Lambda$$
- Ejemplo:

$$T \rightarrow T[T] \mid \Lambda \quad (\alpha = [T], \beta = \Lambda)$$
- Reemplazar por:

$$T \rightarrow U \text{ \& } U \rightarrow [T]U \mid \Lambda$$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Factorización

- Otra vez la gramática original:

$$S \rightarrow T\$, T \rightarrow [T]T \mid \Lambda$$
- Eliminar las producciones- Λ

$$S \rightarrow T\$, T \rightarrow [T]T \mid [T] [T] []$$
- Considerar el parseo por un lookahead AP:

$$S \rightarrow T\$, T \rightarrow [T]T \mid [T] [T] []$$
 - Cuando T está hasta arriba del stack hay 4 opciones
- Factorizar el primer símbolo del lado derecho:

$$S \rightarrow T\$, T \rightarrow [U, U \rightarrow [T]T \mid [T] []$$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Factorización

- Los lados derechos que empiezan con T se pueden factorizar:

$$S \rightarrow T\$, \quad T \rightarrow [U, \quad U \rightarrow T]T \mid]T \mid T \mid]$$

- Factorizar otra vez los lados derechos que empiezan igual:

$$\text{Sea } W \rightarrow T \mid \Lambda$$

$$\text{Entonces en } U \rightarrow T]W \mid]W$$

Consecuentemente:

$$S \rightarrow T\$, \quad T \rightarrow [U, \quad U \rightarrow T]W \mid]W \quad \& \quad W \rightarrow T \mid \Lambda$$

- Eliminando a T ($T \rightarrow [U$):
 $S \rightarrow [U\$, \quad U \rightarrow [U]W \mid]W \quad \& \quad W \rightarrow [U \mid \Lambda$
- El primer símbolo del lado derecho de todas las producciones con el mismo lado izquierdo es siempre diferente

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Gramáticas $LL(1)$

- Gramáticas $LL(1)$:
 - Sin recursión izquierda
 - Factorizadas: Para todas las producciones $\alpha \rightarrow a_i\beta$, donde a_i es diferentes para todas las producciones con el mismo lado izquierdo α

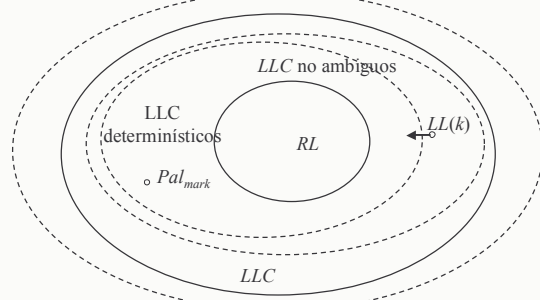
Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Gramáticas $LL(1)$

- Se pueden parsear determinísticamente con un AP top-down viendo un símbolo hacia delante en la cinta de entrada
- Gramáticas $LL(k)$:
 - Parseo top-down determinístico
 - Viendo k símbolos hacia delante en la cinta de entrada

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

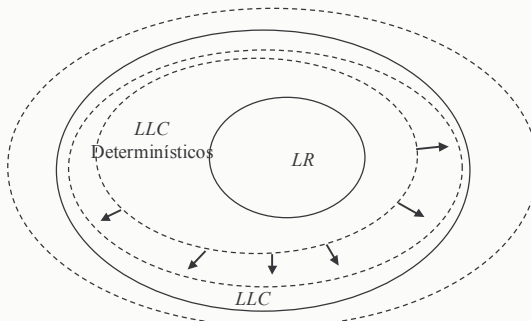
LLC no ambiguos



- LLC pueden parsearse determinísticamente viendo k hacia delante en la cinta de entrada

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7

Aumentando el espacio de las $LLC-D$



El parseo determinístico extiende la clase de los $LLC-D$

Dr. Luis A. Pineda, IIMAS, UNAM, 2005. ISBN: 970-32-2972-7