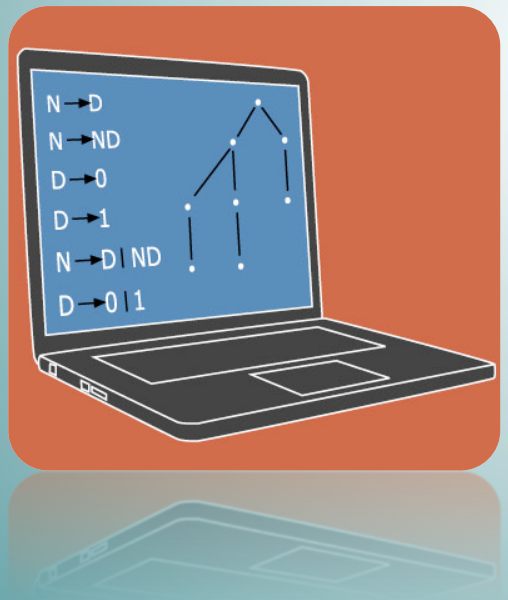
Procesadores de Lenguaje

Analizador LR(0)



Cristina Tirnauca

Domingo Gómez Pérez

Dpto. de Matemáticas,

Estadística y Computación

Prefijos viables (Repaso)

Posibles contenidos de la pila

Para cada regla de la gramática:

Hemos de decidir cuándo conviene usarla para reducir.

- Prefijos viables de los pasos intermedios: no se pasan de la parte derecha que hay que reducir para obtener el paso anterior.
- El conjunto de prefijos viables para cada regla es regular (Knuth, 1965).
- Usaremos autómatas finitos para recorrer la pila e indicarnos cuándo hay que reducir y por qué regla hemos de hacerlo.
- Nos permitirán aplicar *shift* mientras nos encontremos en un prefijo viable.

Analizadores Ascendentes LR

Habíamos dicho una mentirijilla...

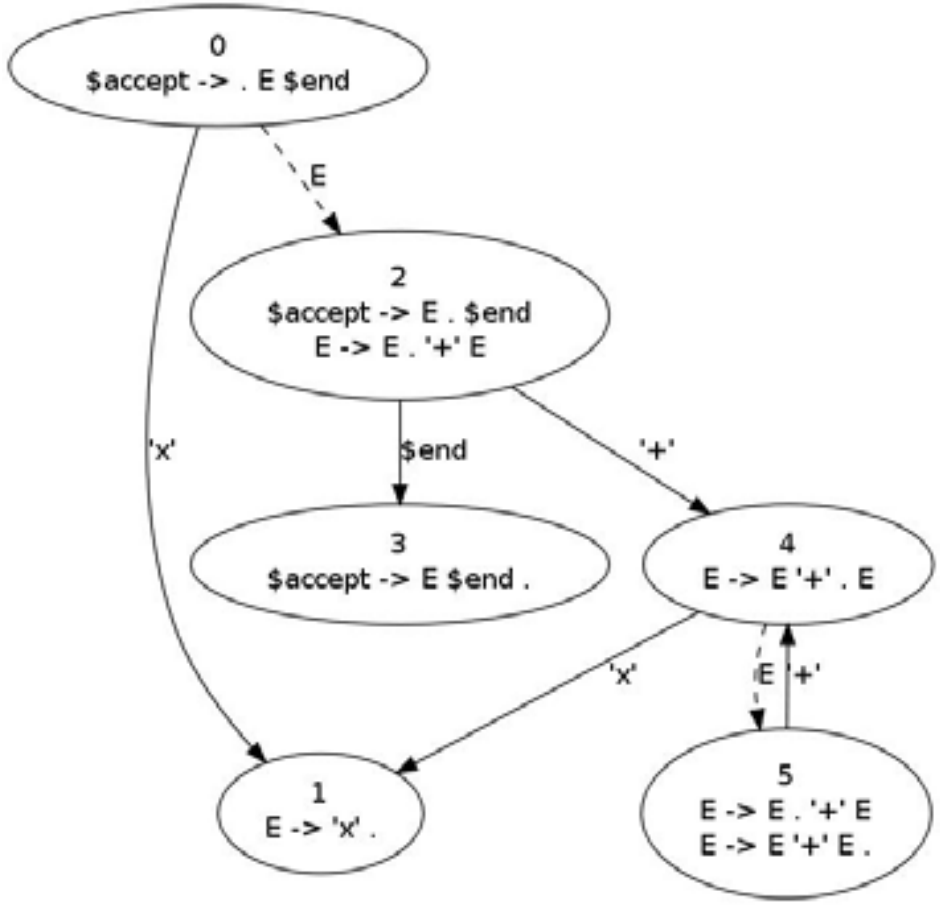
Características principales, una vez más:

- Construyen un árbol sintáctico de las hojas hacia la raíz.
- Leen la entrada de izquierda a derecha (*Left to right*); reconstruyen hacia atrás la (¿una?) derivación derecha (*Rightmost*).
- Operaciones básicas: desplazar (*shift*) un símbolo terminal de la entrada a la pila y reducir (*reduce*) una parte derecha de regla que hay en la cima de la pila, sustituyéndola por la correspondiente parte izquierda de la misma regla.
- A cada vuelta del bucle principal, la pila contiene los estados por los que pasa el autómata LR(0) al leer el prefijo viable que corresponde a la derivación derecha de la parte ya leída de la entrada.

Ejemplo de Autómata LR(0)

Expresiones muy sencillas

```
%%
E : E '+' E
  | 'x' ;
```



(Conflicto en el estado 5 resuelto en favor de reducir.)

Tablas de Análisis LR(0)

La misma información, en formato tabular

```
%%
E : E '+' E
  | 'x' ;
```

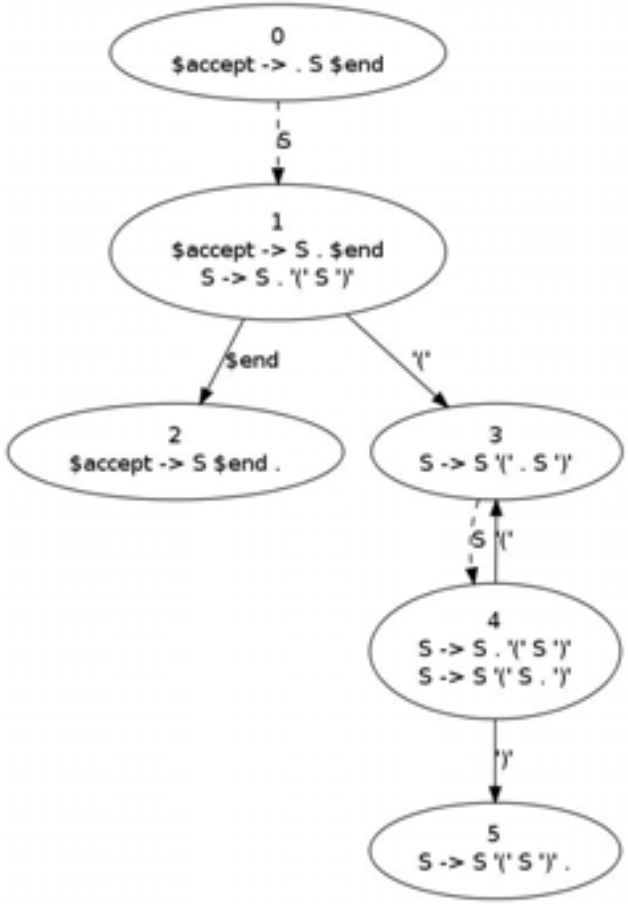
Lo que nos dice el autómata LR(0):

estado	x	+	$	E	reduce:
0	shift: 1			go to: 2	
2		shift: 4	shift: 3		
4	shift: 1			go to: 5	
1					reduce `E:'x'`
3					accept
5					reduce `E:E'+'E`

Otro Ejemplo de Autómata LR(0)

Lenguaje de Dyck, gramática recursiva izquierda habitual

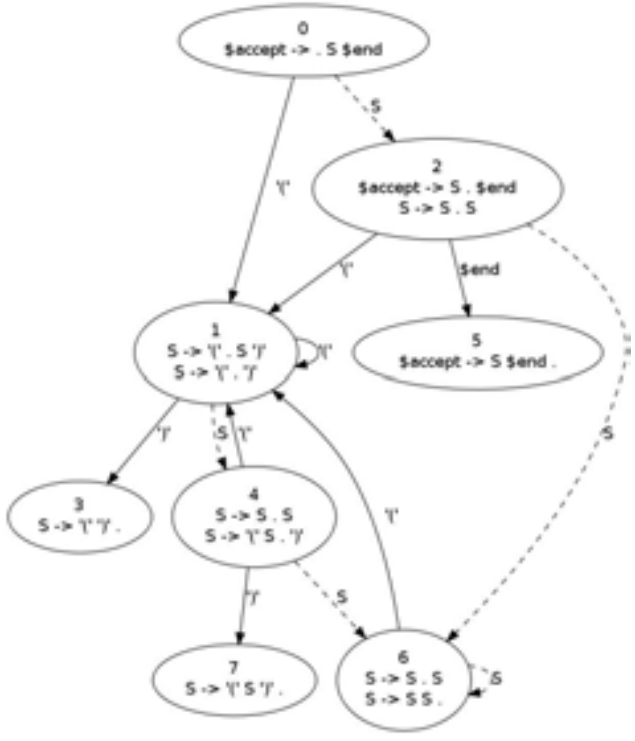
```
%%
S : S '(' S ')'
  | ;
```



Y Otro Ejemplo de Autómata LR(0)

Lenguaje de Dyck sin λ, gramática ambigua

```
%%
S : S S
  | '(' S ')'
  | '(' ')' ;
```



(Conflicto en el estado 6 resuelto en favor de reducir.)

Analizadores Ascendentes LR

La versión más reciente de las que hemos visto

Características principales, una vez más:

- Construyen un árbol sintáctico de las hojas hacia la raíz.
- Leen la entrada de izquierda a derecha (*Left to right*); reconstruyen hacia atrás una derivación derecha (*Rightmost*).
- Operaciones básicas: desplazar (*shift*) un símbolo terminal de la entrada a la pila y reducir (*reduce*) una parte derecha de regla que hay en la cima de la pila, sustituyéndola por la correspondiente parte izquierda de la misma regla.
- A cada vuelta del bucle principal, la pila contiene los estados por los que pasa el autómata LR(0) al leer el prefijo viable que corresponde a la derivación derecha de la parte ya leída de la entrada.

Anulabilidad, FIRST y FOLLOW

Herramientas Conceptuales Auxiliares

Tres nociones importantes

Sobre palabras formadas por símbolos de la gramática, tanto terminales como no terminales.

- Anulabilidad de una palabra (capacidad para "desaparecer"),
- FIRST de una palabra (terminales por los que puede empezar una parte de la entrada que derive de esa palabra),
- FOLLOW de una palabra (terminales que pueden aparecer en una entrada válida justo a continuación de una parte de la entrada que derive de esa palabra).