
PROCESADORES DE LENGUAJES - 1º parcial

16 de enero de 2012

1. Dada la siguiente gramática:

$$\begin{aligned} S &\rightarrow a S A \mid \epsilon \\ A &\rightarrow B b \\ B &\rightarrow A c \mid \epsilon \end{aligned}$$

- a) (0.5 ptos.) ¿Es una gramática LL(1)? Justificad vuestra respuesta.
- b) (1 pto.) Transformad la gramática anterior en otra equivalente y construid su tabla de análisis LL(1). La gramática resultante, ¿es LL(1)? ¿porqué?.
- c) (0.5 ptos.) Empleando la tabla del apartado anterior, haced la traza de análisis LL(1) para la cadena: $a b b$.

2. Para la gramática original de la pregunta anterior:

- a) (2 ptos.) Construid la colección canónica de conjuntos de elementos LR(1).
- b) (1 pto.) A partir de dicha colección, construid la tabla de análisis LALR(1).
- c) (0.5 ptos.) Haced la traza de análisis LALR(1) para la cadena: $a b b$.

3. (2 ptos.) La siguiente gramática genera expresiones matemáticas:

$$S \rightarrow (\text{print } E) \quad E \rightarrow (Op\ E) \mid (Op\ E\ E) \mid num \quad Op \rightarrow + \mid - \mid * \mid /$$

- a) (1 pto.) Diseñad un ETDS que calcule el valor numérico de una expresiones y la imprima.

Ejemplo: $(\text{print } (/ 20 (* 2 (- 5)))) \rightarrow \text{Imprimirá } -2$

- b) (1 pto.) Añadid acciones semánticas para detectar divisiones por cero e informar del número del operador en el que se ha producido.

Ejemplo: $(\text{print } (+ (* 2 10) (/ 5 (- 3 3)))) \rightarrow \text{Error: División por cero en operador número: } 3$

4. Cuestiones teóricas (contestad brevemente):

- a) (0.5 ptos.) Describid las principales ventajas y desventajas de los compiladores frente a los intérpretes.
- b) (0.5 ptos.) Dada la gramática del Ejercicio 1, indicad cuales son los elementos LR(1) válidos para el prefijo viable $a a S$.
- c) (0.5 ptos.) Explicad cuál es la diferencia entre la equivalencia de tipos por nombre y estructural.
- d) (1 pto.) Indicad cuáles de las siguientes afirmaciones son ciertas:
 - i) Toda gramática LL(1) es LR(1).
 - ii) No hay ninguna gramática LL(1) que no sean SLR(1).
 - iii) Toda gramática LR(1) es LALR(1).
 - iv) No hay ninguna gramática LR(k) que sea ambigua.

1. a) La gramática tiene una recursividad a izquierdas indirecta: $A \Rightarrow Bb \Rightarrow Acb$
b) Haciendo explícita (directa) la recursividad a izquierdas y aplicando las transformaciones correspondientes, queda:

$$\begin{array}{lll} S \rightarrow aSA \mid \epsilon & S \rightarrow aSA \mid \epsilon & S \rightarrow aSA \mid \epsilon \\ A \rightarrow Bb & A \rightarrow Acb \mid b & A \rightarrow bA' \\ B \rightarrow Ac \mid \epsilon & & A' \rightarrow cbA' \mid \epsilon \end{array}$$

La tabla de análisis LL(1) será:

	a	b	c	\$
S	(aSA, R_1)	(ϵ, R_2)		(ϵ, R_2)
A		(bA', R_3)		
A'		(ϵ, R_5)	(cbA', R_4)	(ϵ, R_5)

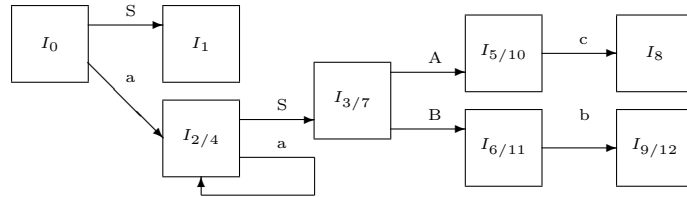
La tabla de análisis LL(1) no tiene entradas múltiples; por tanto, la gramática resultante es LL(1).

- c) La traza para la cadena abb será:
 $(S\$, abb\$, -) \vdash (aSA\$, abb\$, 1) \vdash (SA\$, bb\$, 1) \vdash (A\$, bb\$, 12) \vdash (bA'\$, bb\$, 123) \vdash (A'\$, b\$, 123) \vdash (\$, b\$, 1235) \vdash ERROR$

2. a) La colección de elementos LR(1)

I_0	<table><tr><td>$S' \rightarrow \cdot S$</td><td>$\\$</td></tr><tr><td>$S \rightarrow \cdot aSA$</td><td>$\\$</td></tr><tr><td>$S \rightarrow \cdot$</td><td>$\\$</td></tr></table>	$S' \rightarrow \cdot S$	$\$$	$S \rightarrow \cdot aSA$	$\$$	$S \rightarrow \cdot$	$\$$	I_3	<table><tr><td>$S \rightarrow aS \cdot A$</td><td>$\\$</td></tr><tr><td>$A \rightarrow \cdot Bb$</td><td>$\\$,c$</td></tr><tr><td>$B \rightarrow \cdot Ac$</td><td>$b$</td></tr><tr><td>$B \rightarrow \cdot$</td><td>$b$</td></tr></table>	$S \rightarrow aS \cdot A$	$\$$	$A \rightarrow \cdot Bb$	$\$,c$	$B \rightarrow \cdot Ac$	b	$B \rightarrow \cdot$	b	I_5	<table><tr><td>$S \rightarrow aSA \cdot$</td><td>$\\$</td></tr><tr><td>$B \rightarrow A \cdot c$</td><td>b</td></tr></table>	$S \rightarrow aSA \cdot$	$\$$	$B \rightarrow A \cdot c$	b	I_8	<table><tr><td>$B \rightarrow Ac \cdot$</td><td>b</td></tr></table>	$B \rightarrow Ac \cdot$	b				
$S' \rightarrow \cdot S$	$\$$																														
$S \rightarrow \cdot aSA$	$\$$																														
$S \rightarrow \cdot$	$\$$																														
$S \rightarrow aS \cdot A$	$\$$																														
$A \rightarrow \cdot Bb$	$\$,c$																														
$B \rightarrow \cdot Ac$	b																														
$B \rightarrow \cdot$	b																														
$S \rightarrow aSA \cdot$	$\$$																														
$B \rightarrow A \cdot c$	b																														
$B \rightarrow Ac \cdot$	b																														
I_1	<table><tr><td>$S' \rightarrow S \cdot$</td><td>$\\$</td></tr></table>	$S' \rightarrow S \cdot$	$\$$			I_6	<table><tr><td>$A \rightarrow B \cdot b$</td><td>$\\$,c$</td></tr></table>	$A \rightarrow B \cdot b$	$\$,c$	I_9	<table><tr><td>$A \rightarrow Bb \cdot$</td><td>$\\$,c$</td></tr></table>	$A \rightarrow Bb \cdot$	$\$,c$																		
$S' \rightarrow S \cdot$	$\$$																														
$A \rightarrow B \cdot b$	$\$,c$																														
$A \rightarrow Bb \cdot$	$\$,c$																														
I_2	<table><tr><td>$S \rightarrow a \cdot SA$</td><td>$\\$</td></tr><tr><td>$S \rightarrow a \cdot aSA$</td><td>$b$</td></tr><tr><td>$S \rightarrow a \cdot$</td><td>$b$</td></tr></table>	$S \rightarrow a \cdot SA$	$\$$	$S \rightarrow a \cdot aSA$	b	$S \rightarrow a \cdot$	b	I_4	<table><tr><td>$S \rightarrow a \cdot SA$</td><td>b</td></tr><tr><td>$S \rightarrow a \cdot aSA$</td><td>$b$</td></tr><tr><td>$B \rightarrow \cdot$</td><td>$b$</td></tr></table>	$S \rightarrow a \cdot SA$	b	$S \rightarrow a \cdot aSA$	b	$B \rightarrow \cdot$	b	I_7	<table><tr><td>$S \rightarrow aS \cdot A$</td><td>b</td></tr><tr><td>$A \rightarrow \cdot Bb$</td><td>b,c</td></tr><tr><td>$B \rightarrow \cdot Ac$</td><td>b</td></tr><tr><td>$B \rightarrow \cdot$</td><td>b</td></tr></table>	$S \rightarrow aS \cdot A$	b	$A \rightarrow \cdot Bb$	b,c	$B \rightarrow \cdot Ac$	b	$B \rightarrow \cdot$	b	I_{10}	<table><tr><td>$S \rightarrow aSA \cdot$</td><td>b</td></tr><tr><td>$B \rightarrow A \cdot c$</td><td>b</td></tr></table>	$S \rightarrow aSA \cdot$	b	$B \rightarrow A \cdot c$	b
$S \rightarrow a \cdot SA$	$\$$																														
$S \rightarrow a \cdot aSA$	b																														
$S \rightarrow a \cdot$	b																														
$S \rightarrow a \cdot SA$	b																														
$S \rightarrow a \cdot aSA$	b																														
$B \rightarrow \cdot$	b																														
$S \rightarrow aS \cdot A$	b																														
$A \rightarrow \cdot Bb$	b,c																														
$B \rightarrow \cdot Ac$	b																														
$B \rightarrow \cdot$	b																														
$S \rightarrow aSA \cdot$	b																														
$B \rightarrow A \cdot c$	b																														
						I_{11}	<table><tr><td>$A \rightarrow B \cdot b$</td><td>b,c</td></tr></table>	$A \rightarrow B \cdot b$	b,c																						
$A \rightarrow B \cdot b$	b,c																														
						I_{12}	<table><tr><td>$A \rightarrow Bb \cdot$</td><td>b,c</td></tr></table>	$A \rightarrow Bb \cdot$	b,c																						
$A \rightarrow Bb \cdot$	b,c																														

- b) A partir de este conjunto de elementos LR(1) se pueden fusionar estados con el criterio LALR(1) quedando el siguiente autómata de prefijos viables y la tabla LALR(1):



	a	b	c	\$	S	A	B
0	$d_{2/4}$			r_2	1		
1				acc			
2/4	$d_{2/4}$	r_2			3/7		
3/7		r_5				5/10	6/11
5/10		r_1	d_8	r_1			
6/11		$d_{9/12}$					
8		r_4					
9/12		r_3	r_3	r_3			

- c) La traza para la cadena abb será:
 $(0, abb\$, -) \vdash (0a2, bb\$, -) \vdash (0a2S3, bb\$, 2) \vdash (0a2S3B6, bb\$, 25) \vdash (0a2S3B6b9, b\$, 25) \vdash (0a2S3A5, b\$, 253) \vdash (0S1, b\$, 2531) \vdash ERROR$

3.-

S -> (print E)	E.cont := 0 print (E.valor);
E -> (op E ₁)	E ₁ .cont := E.cont+1 if (op.o = mas) E.valor := E ₁ .valor else if (op.o = menos) E.valor := - E ₁ .valor else MemError("Operador no válido"); E.sintcont := E ₁ .sintcont
E -> (op E ₁ E ₂)	E ₁ .cont := E.cont+1 ; E ₂ .cont := E ₁ .sintcont if (op.o = mas) E.valor := E ₁ .valor + E ₂ .valor else if (op.o = menos) E.valor := E ₁ .valor - E ₂ .valor else if (op.o = por) E.valor := E ₁ .valor * E ₂ .valor else if (op.o = divi) if E ₂ .valor = 0 { MemError("División por cero en op num.", E.cont +1); E.valor := 0; } else E.valor := E ₁ .valor / E ₂ .valor E.sintcont := E ₂ .sintcont
E -> num	E.valor = num.lexval; E.sintcont := E.cont
op -> +	op.o := mas
op -> -	op.o := menos
op -> *	op.o := por
op -> /	op.o := divi

4.- a) VENTAJAS COMPILADOR

- Generan código que consume menos recursos
- Durante la ejecución no es necesario realizar comprobaciones léxico, sintácticas ni semánticas que se han podido realizar en tiempo de compilación.
- No es necesario que esté presente el compilador para ejecutar el programa.
- No es necesario disponer del código fuente para ejecutar el programa

DESVENTAJAS

- Código ejecutable no portable entre distintas plataformas.
- Hasta que el programa no está libre de errores no es posible ejecutar ninguna parte de él
- Es más difícil portar el compilador a otra plataforma

b) [S -> a S . A, b], [A -> . B b, b/c] , [B -> . A c, b] y [B -> . , b]

c) En la *equivalencia por nombre* dos expresiones de tipo son equivalentes si tienen el mismo nombre. En cambio, en la *equivalencia estructural* dos expresiones de tipo son equivalentes si representan expresiones estructuralmente equivalentes después de sustituir todos los nombres por las expresiones que representan.

d)

- i) Cierto
- ii) Falso
- iii) Falso
- iv) Cierto