



PREGUNTAS TEST COMPILADORES - Gramáticas LR y LL (2º Grado en Informática)

1. Una *gramática con  $\lambda$ -reglas*...
  - a) ...puede ser LL y LR.
  - b) ...puede ser LL pero no LR.
  - c) ...puede ser LR pero no LL.
2. Elige la opción correcta, suponiendo que consideramos un único símbolo de anticipación para cada método:
  - a) Una *gramática LALR* es *SLR* aunque no tenemos seguridad de que sea *LR-canónica*.
  - b) Una *gramática LALR* es *LR-canónica* aunque no tenemos seguridad de que sea *SLR*.
  - c) Una *gramática LR-canónica* es *SLR* y *LALR*.
3. Si a partir del autómatá de una gramática LR-Canónica construimos el autómatá de la gramática LALR, al construir la tabla...
  - a) ...pueden aparecer conflictos shift/reduce y reduce/reduce.
  - b) ...sólo pueden aparecer conflictos reduce/reduce.
  - c) ...sólo pueden aparecer conflictos shift/reduce.
4. Una *gramática ambigua*
  - a) no puede ser LL ni LR.
  - b) puede ser LL pero no LR.
  - c) puede ser LR pero no LL.
5. Podemos afirmar que una gramática es LALR
  - a) si no es recursiva por la izquierda, ni ambigua ni está factorizada.
  - b) si es SLR.
  - c) si no es ambigua.
6. Dada la siguiente gramática:
$$\begin{array}{lcl} S & \rightarrow & S A \\ & | & A \\ A & \rightarrow & id = L ; \\ L & \rightarrow & id \\ & | & L = L \end{array}$$
elegir, de entre los siguientes, el conjunto de *items válidos* para el *prefijo viable*  $id =$ 
  - a)  $\{[A \rightarrow id = \cdot L ;], [L \rightarrow \cdot id], [L \rightarrow \cdot L = L]\}$
  - b)  $\{[S \rightarrow A \cdot]\}$
  - c)  $\{[A \rightarrow id = L \cdot;], [L \rightarrow L \cdot = L]\}$
7. Dada la misma gramática de la pregunta anterior, el pivote de la forma sentencial  $id = L = L = id$ ; es
  - a)  $id = L = L = \underline{id}$ ;
  - b)  $id = \underline{L} = \underline{L} = id$ ;
  - c)  $id = L = \underline{L} = id$ ;
8. Una **gramática recursiva por la izquierda**:
  - a) No puede ser LL, ni LR.
  - b) No puede ser LL, aunque sí SLR.
  - c) No puede ser SLR, aunque sí LR-Canónica.

9. Las **gramáticas LL y LR**:
- Pueden tener  $\lambda$ -reglas y ser *ambiguas*.
  - Pueden tener  $\lambda$ -reglas pero no pueden ser *ambiguas*.
  - No pueden tener  $\lambda$ -reglas ni ser *ambiguas*.
10. Elegir la opción correcta:
- El *método de análisis LL* es ascendente y predictivo.
  - El *método de análisis LR* es ascendente y no predictivo.
  - Los *métodos de análisis LL y LR* son ambos predictivos, de manera que el primero funciona obteniendo las derivaciones por la izquierda desde el símbolo inicial de la gramática y el segundo obteniendo las reducciones por la izquierda a partir de la cadena de entrada.
11. El **tratamiento de errores** a nivel de frase en cualquiera de los métodos estudiados:
- Es un método sistemático que no varía de una gramática a otra.
  - Requiere la determinación de posibles errores que puedan producirse en el lenguaje fuente, y para ello podemos hacer una llamada a un procedimiento particular en cada casilla vacía de la tabla de análisis.
  - Es un método que requiere la determinación previa de todas las posibles frases erróneas que puedan aparecer en un programa y que, además, permite corregirlas.
12. Supongamos que hemos calculado la colección LR(0) y la tabla SLR para la siguiente gramática:

$$E \rightarrow E * E \mid E + E \mid id$$

de modo que los conjuntos  $I_5$  e  $I_6$  contienen los siguientes items:

$$I_5 = \{E \rightarrow E * E\bullet, E \rightarrow E \bullet * E, E \rightarrow E \bullet + E\}$$

$$I_6 = \{E \rightarrow E \bullet * E, E \rightarrow E + E\bullet, E \rightarrow E \bullet + E\}$$

produciéndose una tabla SLR con conflictos. Elegir la acción adecuada para la casilla del estado 6 y el símbolo de entrada \*:

- d3.
- r1.
- r2.

13. Dada la siguiente gramática:

$$\begin{array}{ll} sent & \rightarrow \text{if } expr \text{ then } sent \\ & \mid \text{if } expr \text{ then } sent \text{ else } sent \\ & \mid S \\ expr & \rightarrow E \end{array}$$

- es posible encontrar una gramática equivalente no ambigua y LR.
- es posible encontrar una gramática equivalente no ambigua y LL.
- no es posible encontrar una gramática equivalente no ambigua.

14. Dada la gramática G siguiente:

$$\begin{aligned} Lista &\rightarrow [] \mid [ Termino ] \\ Termino &\rightarrow Termino , Termino \mid ID \mid Lista \end{aligned}$$

y la cadena de entrada  $w = [[a, b, c], [e, f]]$ , decidir cuales serían las tres primeras reducciones que realizaría un analizador ascendente para reconocer  $w$ , suponiendo que el operador ',' es asociativo por la izquierda:

- a)  $Lista \rightarrow [ Termino ]$   
 $Termino \rightarrow Termino , Termino$   
 $Termino \rightarrow Lista$
- b)  $Termino \rightarrow ID$   
 $Termino \rightarrow ID$   
 $Termino \rightarrow Termino , Termino$
- c)  $Termino \rightarrow ID$   
 $Termino \rightarrow ID$   
 $Termino \rightarrow ID$

15. Supongamos que hemos calculado la colección  $LR(1)$  para la gramática:

$$\begin{aligned} S &\rightarrow aAd \mid bBd \mid aBe \mid bAe \\ A &\rightarrow c \\ B &\rightarrow c \end{aligned}$$

de modo que los conjuntos  $I_6$  e  $I_9$  contienen los siguientes items:

$$\begin{aligned} I_6 &= \{[A \rightarrow c \bullet, d], [B \rightarrow c \bullet, e]\} \\ I_9 &= \{[A \rightarrow c \bullet, e], [B \rightarrow c \bullet, d]\} \end{aligned}$$

Sabiendo que la gramática es  $LR$ -canónica, indica la respuesta correcta:

- a) La gramática es  $LALR$  y  $SLR$ .
- b) La gramática no es  $LALR$  ni  $SLR$ .
- c) La gramática no es  $LALR$  pero si es  $SLR$ .

16. Si una gramática contiene (entre otras) las siguientes reglas:

$$\begin{aligned} A &\rightarrow a x \\ &\quad \mid \lambda \\ B &\rightarrow A a y \end{aligned}$$

- a) puede ser  $LL(1)$ .
- b) puede ser  $SLR(1)$ .
- c) no puede ser  $LL(1)$ .

17. Dada la siguiente gramática:

$$\begin{aligned} S &\rightarrow L = R \mid R \\ L &\rightarrow *R \mid id \\ R &\rightarrow L \end{aligned}$$

el pivote de la forma sentencial derecha  $*L = *id$  es:

- a)  $*L = *id$
- b)  $*L = *id$
- c)  $*L = *id$

18. ¿Cuál de las siguientes es una versión no ambigua de la gramática  $S \rightarrow SS \mid a \mid b$ ?

- a)  $S \rightarrow Sa \mid Sb \mid \lambda$
- b)  $S \rightarrow SS' \mid a \mid b$   
 $S' \rightarrow a \mid b$
- c)  $S \rightarrow Sa \mid S'$   
 $S' \rightarrow a \mid b$

19. Dada la gramática con producciones  $E \rightarrow E + E \mid id$ , ¿cuántos árboles de derivación distintos pueden crearse con la cadena de entrada  $id + id + id$ ?

- a) 1
- b) 2
- c) 3

20. Indica la respuesta correcta:

- a) Durante el análisis LR, es posible determinar la forma sentencial izquierda actual concatenando los símbolos de la pila y los símbolos pendientes de ser analizados en la entrada.
- b) El análisis LR produce la secuencia de derivaciones más a la derecha en orden inverso.
- c) El análisis LR opera mediante reducciones más a la derecha desde la cadena de entrada hasta el símbolo inicial.

21. Dada la siguiente gramática que genera cualquier palíndromo sobre  $\{a, b\}$ :

$$P \rightarrow a \mid b \mid a P a \mid b P b \mid \lambda$$

- a) el pivote de la forma sentencial derecha  $abbPbba$  es  $P$ .
- b) el pivote de la forma sentencial derecha  $abbPbba$  es  $bPb$ .
- c) el pivote de la forma sentencial derecha  $abbPbba$  es  $bbPbb$ .

22. Dada la gramática con producciones:

$$E \rightarrow E + E \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow id \mid (E)$$

¿qué tipo de ambigüedades presenta?

- a) Sólo ambigüedad debida a la precedencia de los operadores  $+$  y  $*$ .
- b) Sólo ambigüedad debida a la asociatividad del operador  $+$ .
- c) Ningún tipo de ambigüedad.

23. Dada la siguiente gramática:

$$\begin{array}{lcl} E & \rightarrow & id E' \\ E' & \rightarrow & \lambda \\ & & \mid \wedge E' \\ & & \mid . id E' \\ & & \mid [ E ] E' \end{array}$$

La forma sentencial  $id^\wedge . id^\wedge E'$ :

- a) Es una forma sentencial derecha.
- b) Es una forma sentencial izquierda.
- c) Es una forma sentencial derecha e izquierda.

24. Una gramática LR:

- a) Tiene que ser propia.
- b) Puede no ser propia.
- c) Tiene que ser no ambigua y  $\lambda$ -libre.

25. Una tabla LALR:

- a) Tiene un tamaño intermedio entre una SLR y una LR-Canónica para la misma gramática.
- b) Tiene necesariamente que tener un tamaño menor que una LR-Canónica para la misma gramática.
- c) Tiene necesariamente el mismo tamaño que una tabla SLR para la misma gramática.

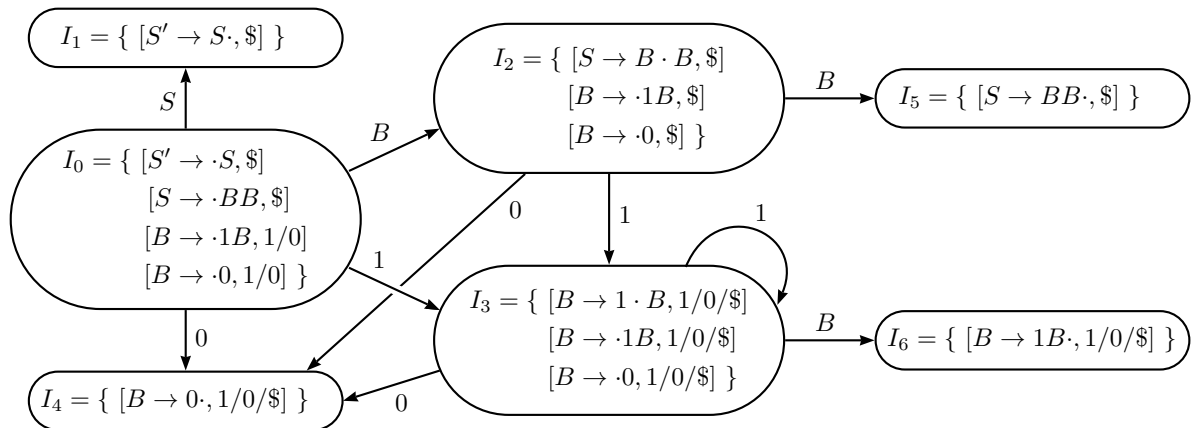
26. Supongamos que construimos un analizador LALR de la siguiente gramática:

$S \rightarrow BB$

$B \rightarrow 1B$

$B \rightarrow 0$

y obtenemos los siguientes conjuntos de ítems:



Indicar la respuesta correcta:

- Sólo se puede afirmar que la gramática es LALR.
  - Sólo se puede afirmar que la gramática es LR(1).
  - Se puede afirmar que la gramática es SLR y LR(1).
27. Continuando con el analizador LALR del ejemplo anterior, ¿en qué estado se encontraría el analizador al terminar de procesar la subcadena de entrada 011?
- $I_3$
  - $I_5$
  - $I_6$
28. Los métodos generales de análisis sintáctico que analizan cualquier gramática libre de contexto:
- No existen.
  - No siempre pueden usarse puesto que requieren transformar las gramáticas previamente. Por ejemplo, el método CYK analiza gramáticas en *forma normal de Chomsky*.
  - No suelen usarse por falta de eficiencia.
29. Dada la gramática G con las producciones:
- $$S \rightarrow S ( S ) S \mid \lambda$$
- que genera el conjunto de paréntesis balanceados,
- Existen dos derivaciones por la derecha y dos por la izquierda para generar la sentencia '()()'.
  - Existe una derivación por la derecha y una por la izquierda para generar la sentencia '()()'.
  - La sentencia '()()' no puede ser generada por la gramática G.
30. Considera la siguiente gramática:
- $$S \rightarrow A ( S ) B \mid \lambda$$
- $$A \rightarrow S \mid S B \mid x \mid \lambda$$
- $$B \rightarrow S B \mid y$$
- Indica la respuesta correcta:
- $\text{SIGUIENTE}(S) = \{y, x, (, )\}$
  - $\text{SIGUIENTE}(B) = \{x, y, (, )\}$
  - $\text{PRIMERO}(S) = \{x, y, (, \lambda\}$

31. Dada la siguiente gramática:

$$E \rightarrow T * E \mid T$$

$$T \rightarrow \text{int} + T \mid \text{int} \mid (E)$$

¿Cuál es el pivote de la forma sentencial derecha  $((\text{int} + T) * \text{int})$ :

a)  $((\underline{\text{int}} + T) * \text{int})$

b)  $((\text{int} + \underline{T}) * \text{int})$

c)  $((\text{int} + \underline{T}) * \text{int})$

32. Dada la gramática de la pregunta anterior, y la cadena de entrada  $((\text{int} + \text{int}) * \text{int})$ , ¿cuántas reducciones y desplazamientos se emplean para reconocerla en un análisis ascendente?

a) 9 desplazamientos y 9 reducciones.

b) 8 desplazamientos y 9 reducciones.

c) 8 desplazamientos y 8 reducciones.

33. Supongamos que estamos creando la colección LR(1) de una gramática, y tenemos que aplicar la operación de clausura al ítem  $[A \rightarrow B \cdot C D E, x]$ . La gramática tiene las siguientes reglas de producción:

$$C \rightarrow y \mid z$$

$$D \rightarrow d \mid \lambda$$

$$E \rightarrow e \mid \lambda$$

¿Cuáles de los siguientes ítems serían añadidos por aplicación de la operación clausura?

a)  $[C \rightarrow \cdot y, d/e/x], [C \rightarrow \cdot z, d/e/x]$

b)  $[C \rightarrow \cdot y, d/\lambda], [C \rightarrow \cdot z, d/\lambda]$

c)  $[C \rightarrow \cdot y, y/z], [C \rightarrow \cdot z, y/z]$

34. Considérese la siguiente gramática:

$$S \rightarrow X S a \mid b$$

$$X \rightarrow \lambda \mid z$$

¿Cuál es la respuesta correcta?

a) La gramática es recursiva por la izquierda, y por tanto no puede ser LL(1).

b) La gramática es SLR y LALR.

c) La gramática es LR(1).

35. Supongamos que para una cierta gramática G aparece un conflicto d/r en una tabla LALR.

a) Puesto que G no es una gramática LALR, L(G) no se podría analizar con este método.

b) Sí podría analizarse L(G) con el método LALR, eligiendo el desplazamiento en cualquier caso.

c) Dependiendo de la gramática, podría ser posible analizarla estudiando el conflicto y eligiendo una de las dos acciones.

36. Sea G la gramática con las producciones:

$$S \rightarrow 0 S 1 \mid 0 1$$

El pivote de la sentencia '00001111' sería:

a) 00001111

b) 00001111

c) 00001111

37. Sea G la gramática con las producciones:

$$S \rightarrow 0 S 1 \mid 0 1$$

Un analizador ascendente predictivo realizaría la siguiente secuencia de reducciones para reconocer la sentencia '000111':

- a) r2 r1 r1
- b) r1 r2 r1
- c) r2 r2 r1

38. Considera la siguiente gramática:

$$\begin{aligned} S &\rightarrow A ( S ) B \mid \lambda \\ A &\rightarrow S \mid S B \mid x \mid \lambda \\ B &\rightarrow S B \mid y \end{aligned}$$

¿Cuál es el conjunto de ítems  $I_0$  de la colección LR(0) de la gramática de la pregunta anterior?

- a)  $\{ [ S' \rightarrow \cdot S ], [ S \rightarrow \cdot A ( S ) B ], [ S \rightarrow \cdot ], [ A \rightarrow \cdot S ], [ A \rightarrow \cdot S B ], [ A \rightarrow \cdot x ], [ A \rightarrow \cdot ], [ B \rightarrow \cdot S B ], [ B \rightarrow \cdot y ] \}$
- b)  $\{ [ S' \rightarrow \cdot S ], [ S \rightarrow \cdot A ( S ) B ], [ S \rightarrow \cdot ], [ A \rightarrow \cdot S ], [ A \rightarrow \cdot S B ], [ A \rightarrow \cdot x ], [ A \rightarrow \cdot ] \}$
- c)  $\{ [ S' \rightarrow \cdot S ], [ S \rightarrow \cdot A ( S ) B ], [ A \rightarrow \cdot S ], [ A \rightarrow \cdot S B ], [ A \rightarrow \cdot x ] \}$

39. ¿Qué tipo de conflictos se producen en el conjunto  $I_0$  de la pregunta anterior?

- a) Reduce-reduce.
- b) Desplaza-reduce.
- c) Desplaza-reduce y reduce-reduce.

40. Dada la siguiente gramática:

$$\begin{aligned} stmt &\rightarrow var \mid if\_stmt \\ if\_stmt &\rightarrow if\ var\ then\ stmt \mid if\ var\ then\ stmt\ else\ stmt \\ var &\rightarrow a \mid b \mid win \mid loss \end{aligned}$$

¿Qué tipo de conflictos se producen al analizar la entrada "if a then if b then win else loss"?

- a) Reduce-reduce.
- b) Desplaza-reduce.
- c) Ninguno.

41. Sea G la gramática con las producciones:

$$S \rightarrow S ( S ) S \mid \lambda$$

que genera el conjunto de paréntesis balanceados.

- a) G es propia, no LL y no LR-Canónica.
- b) G es no propia, no LL, no SLR, pero sí LR-Canónica.
- c) G es no propia, ambigua, no LL y no SLR.

42. La siguiente gramática:

$$\begin{aligned} S &\rightarrow S A \\ &\mid A \\ A &\rightarrow id = L ; \\ L &\rightarrow id \\ &\mid L = L \end{aligned}$$

- a) Es propia.
- b) No es propia, pues es recursiva por la izquierda.
- c) No es propia, pues es ambigua.

43. La gramática anterior:

- a) Es LL(1) y SLR(1).
- b) No es LL(1) aunque sí LR(1).
- c) No es ni LL(1) ni SLR(1).

44. Si en la colección LR(0) de la gramática anterior obtenemos el estado

$$I_{10} \equiv \{ L \rightarrow L = L \bullet, L \rightarrow L \bullet = L \}$$

y la siguiente tabla:

ESTADO	accion				ir_a		
	id	=	;	\$	S	A	L
<b>0</b>	d3				1	2	
<b>1</b>	d3			aceptar		4	
<b>2</b>	r2			r2			
<b>3</b>		d5					
<b>4</b>	r1			r1			
<b>5</b>	d7						6
<b>6</b>		d9	d8				
<b>7</b>		r4	r4				
<b>8</b>	r3			r3			
<b>9</b>	d7						10
<b>10</b>		d9/r5	r5				

Para conseguir que el operador = sea asociativo por la derecha, en la casilla [10,=] debemos elegir la acción:

- a) r5
- b) d9
- c) Ninguna valdría.

45. Sea G la gramática con las producciones:

$$S \rightarrow S ( S ) S \mid \lambda$$

que genera el conjunto de paréntesis balanceados. El conjunto  $I_0$  de la colección LR(1) es:

- a)  $\{ [ S' \rightarrow \cdot S, \$ ], [ S \rightarrow \cdot S ( S ) S, \$ ] \}$
- b)  $\{ [ S' \rightarrow \cdot S, \$ ], [ S \rightarrow \cdot S ( S ) S, \$ ( ] ], [ S \rightarrow \cdot, \$ ( ] ] \}$
- c)  $\{ [ S' \rightarrow \cdot S, \$ ], [ S \rightarrow \cdot S ( S ) S, \$ ( ] ], [ S \rightarrow \cdot, \$ ( ] ] \}$

46. Si, para la misma gramática, calculamos la colección LR(0) y aparece el siguiente estado:

$$I_j = \{ [ S \rightarrow S ( S ) S \cdot ], [ S \rightarrow S \cdot ( S ) S ] \}$$

Entonces:

- a) Se produciría un conflicto desplaza-reduce en la casilla [j,(] de la tabla.
- b) Se produciría un conflicto desplaza-reduce en la casilla [j, ] de la tabla.
- c) Se produciría un conflicto reduce-reduce en la casilla [j,(] de la tabla.

47. La herramienta Bison realiza un análisis:

- a) SLR con dos tokens de anticipación.
- b) LALR usando un token de anticipación sólo cuando existen acciones diferentes en un mismo estado de la tabla.
- c) LR-Canónica con un token de anticipación.



## Parte II: PREGUNTAS CORTAS.

- Supongamos que hemos calculado la colección LR(0) y la tabla SLR para la siguiente gramática:

$$E \rightarrow E \wedge E \mid E \vee E \mid id$$

de modo que los conjuntos  $I_5$  e  $I_6$  contienen los siguientes items:

$$I_5 = \{E \rightarrow E \wedge E\bullet, E \rightarrow E \bullet \wedge E, E \rightarrow E \bullet \vee E\}$$

$$I_6 = \{E \rightarrow E \bullet \wedge E, E \rightarrow E \vee E\bullet, E \rightarrow E \bullet \vee E\}$$

produciéndose en la tabla SLR los siguientes conflictos:

ESTADO	accion		...
	$\wedge$	$\vee$	...
...	...	...	...
<b>5</b>	$r1/d3$	$r1/d4$	...
<b>6</b>	$r2/d3$	$r2/d4$	...

Resolverlos dando mayor prioridad al operador  $\wedge$  que al  $\vee$  y considerando que ambos son asociativos por la izquierda.

- Si una gramática  $G$  es LR-Canónica pero no es LALR, en la tabla LALR sólo pueden aparecer conflictos reducción/reducción y no desplazamiento/reducción. Proponer un ejemplo (concreto o genérico) en el que al obtener un estado  $I_{ij}$  para una tabla LALR a partir de dos estados sin conflictos  $I_i$  e  $I_j$  de una colección LR(1), en el primero (el  $I_{ij}$ ) aparezca un conflicto reducción/reducción.
- Dada la gramática  $G$  con  $V_T = \{int, float, ident, , \}$ ,  $V_N = \{DECLARACION, TIPO, LISTA\_VAR\}$ , símbolo inicial  $DECLARACION$  y el siguiente conjunto de producciones:

$$\begin{aligned} DECLARACION &\rightarrow TIPO LISTA\_VAR \\ TIPO &\rightarrow int \\ &\mid float \\ LISTA\_VAR &\rightarrow ident, LISTA\_VAR \\ &\mid ident \end{aligned}$$

¿Es  $G$  una gramática **propia**? ¿Por qué? ¿Puede ser una gramática no-propia LR? ¿Y LL? Justificar las respuestas.

- Considerar la siguiente gramática  $G$  que genera *expresiones regulares*:

$$\begin{aligned} R &\rightarrow R' \mid R \\ &\mid R R \\ &\mid R' *' \\ &\mid a \\ &\mid b \end{aligned}$$

Suponemos que, una vez calculada la colección LR(1) correspondiente a dicha gramática, el conjunto  $I_5$  es el siguiente:

$$\begin{aligned} I_5 = \{ &[R \rightarrow R \cdot \mid R, a/b/ \mid * / \$] \\ &[R \rightarrow RR \cdot, a/b/ \mid * / \$] \\ &[R \rightarrow R \cdot R, a/b/ \mid * / \$] \\ &[R \rightarrow R \cdot *, a/b/ \mid * / \$] \\ &[R \rightarrow \cdot R \mid R, a/b/ \mid * / \$] \\ &[R \rightarrow \cdot RR, a/b/ \mid * / \$] \\ &[R \rightarrow \cdot R *, a/b/ \mid * / \$] \\ &[R \rightarrow \cdot a, a/b/ \mid * / \$] \\ &[R \rightarrow \cdot b, a/b/ \mid * / \$] \} \end{aligned}$$

Y la tabla LR-Canónica correspondiente al estado 5 es la siguiente:

ESTADO	ACCIÓN					IR-A
	*	<i>a</i>	<i>b</i>		\$	<i>R</i>
0						5
1						
2						
3						
4						
5	r2/d6	r2/d2	r2/d3	r2/d4	r2	
6						
7						

Resolver los conflictos existentes en las casillas de la tabla  $[5, *]$ ,  $[5, a]$  y  $[5, |]$ , eliminando las acciones correspondientes a las entradas múltiples con el objeto de dotar a cada operador de la precedencia y asociatividad usual en expresiones regulares.

### Parte III: PROBLEMAS.

- Supongamos que se desea formalizar un lenguaje de matrices numéricas en el que, por ejemplo, una matriz de dos filas y tres columnas como

$$\begin{pmatrix} -1 & 3 & 7 \\ 4 & 6 & 0 \end{pmatrix}$$

quedaría representada de la siguiente forma:

$$(-1 \ 3 \ 7 ; 4 \ 6 \ 0 ; )$$

La siguiente gramática,  $G$ , con  $V_T = \{ (, ), num, ; \}$ ,  $V_N = \{ MATRIZ, FILA, FILAS, NUMEROS \}$ , símbolo inicial  $MATRIZ$  y el siguiente conjunto  $P$  de producciones:

$$\begin{array}{ll} MATRIZ & \rightarrow ( FILA FILAS ) \\ FILA & \rightarrow num NUMEROS ; \\ FILAS & \rightarrow FILA FILAS \\ & | \lambda \\ NUMEROS & \rightarrow num NUMEROS \\ & | \lambda \end{array}$$

sirve para generar matrices con el formato anterior.

Realiza los siguientes ejercicios:

- Construir la colección LR(1) para  $G$  y la tabla LR-canónica. Indicar si  $G$  es una gramática LR-canónica justificando la respuesta.
  - Indicar si  $G$  es una gramática LALR y/o SLR justificando la respuesta, y sin calcular ninguna colección de ítems adicional.
- Dada la gramática  $G$ , con  $V_T = \{ id, =, ; \}$ ,  $V_N = \{ S, A, L \}$ , símbolo inicial  $S$  y el siguiente conjunto  $P$  de producciones:

$$\begin{array}{ll} S & \rightarrow S A \\ & | A \\ A & \rightarrow id = L ; \\ L & \rightarrow id \\ & | L = L \end{array}$$

responder a las siguientes cuestiones:

- Decir, justificando las respuestas, y sin construir ninguna tabla de análisis, si  $G$  es:
  - Propia.
  - Ambigua.
  - LR-Canónica(1).
  - LALR(1).
  - SLR(1).
  - LL(1).

En caso de que  $G$  no cumpla alguna de estas propiedades, dar todas las razones que se conozcan para justificarlo.

- Construir la colección LR(0) y la tabla de análisis SLR. Modificar esta tabla para conseguir que el operador de asignación ( $=$ ) sea asociativo por la derecha. Si usáramos *Bison* para generar la tabla, ¿cómo podría conseguirse esa asociatividad por la derecha?
- Simular el algoritmo de análisis ascendente predictivo no recursivo para las cadenas de entrada  $w_1 \equiv a = b = c = d$ ; y  $w_2 \equiv a == b$ ;, realizando recuperación en modo pánico en caso de error. Comenzar la simulación en ambos casos sustituyendo cada lexema por el código de token correspondiente.

3. La siguiente gramática  $G$ , con  $V_T = \{ (, ), :=, \text{id}, \text{num}, , \}$ ,  $V_N = \{ L, R \}$ , símbolo inicial  $A$  y el siguiente conjunto  $P$  de producciones:

$$\begin{array}{lcl} A & \rightarrow & ( L ) := ( R ) \\ L & \rightarrow & \text{id} \\ & | & L , \text{id} \\ R & \rightarrow & \text{num} \\ & | & R , \text{num} \end{array}$$

permite definir asignaciones múltiples (es una característica de algunos lenguajes de programación, como Perl). Por ejemplo, la siguiente sentencia:

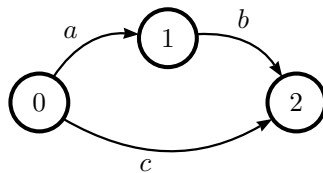
$(a, b) := (1, 2)$

asigna simultáneamente a la variable  $a$  el valor  $1$ , y a la variable  $b$  el valor  $2$ . Responder a las siguientes cuestiones:

- Obtener la colección LR(1) para la gramática  $G$  inicial y construir la tabla LR-canónica. Indicar si  $G$  es una gramática LR-canónica justificando la respuesta.
  - Indicar si  $G$  es una gramática LALR y/o SLR, justificando la respuesta, y sin calcular ninguna colección de ítems adicional.
4. La siguiente gramática  $G$ , con  $V_T = \{ \text{id}, \text{num}, \&, ;, :, > \}$ ,  $V_N = \{ A, S, D, L, B \}$ , símbolo inicial  $A$  y el siguiente conjunto  $P$  de producciones:

$$\begin{array}{lcl} A & \rightarrow & A S \\ & | & S \\ S & \rightarrow & \text{num} : D ; \\ D & \rightarrow & L \\ & | & \lambda \\ L & \rightarrow & L \& B \\ & | & B \\ B & \rightarrow & \text{id} > \text{num} \end{array}$$

permite representar autómatas finitos textualmente. Por ejemplo, el siguiente autómata:



se representa de la siguiente forma:

0 :  $a > 1 \& c > 2$  ;  
 1 :  $b > 2$  ;  
 2 : ;

Responder a las siguientes cuestiones:

- Construir la colección LR(1) para  $G$ , y la tabla LR-canónica. Indicar si  $G$  es una gramática LR-canónica justificando la respuesta.
- Simular el reconocimiento de la cadena  $w \equiv 0 : a >$  ;
- Indicar si  $G$  es una gramática LALR y/o SLR, justificando la respuesta, y sin calcular ninguna colección de ítems adicional.

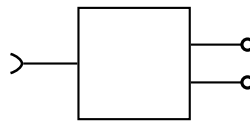
5. La siguiente gramática  $G$ , con  $V_T = \{&, <, >, ;, (, ), \text{int}\}$ ,  $V_N = \{P, S, E, I, O\}$ , símbolo inicial  $P$  y el siguiente conjunto de producciones:

$$\begin{aligned}
 P &\rightarrow S \\
 &\quad | \quad S \& P \\
 S &\rightarrow E \\
 &\quad | \quad E S \\
 E &\rightarrow < I ; O > \\
 &\quad | \quad ( P ) \\
 I &\rightarrow \text{int} \\
 O &\rightarrow \text{int}
 \end{aligned}$$

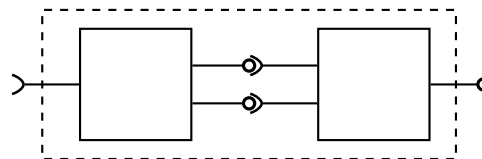
permite representar grafos de componentes agrupados de forma serie o paralela. Un componente se describe mediante un bloque que posee interfaces de entrada y salida. Textualmente, un componente básico se representa mediante una cadena de la forma indicada por la primera alternativa de E:

`< n interfaces entrada ; n interfaces de salida >`

Por ejemplo, un componente `<1;2>` representa al bloque de la siguiente figura:

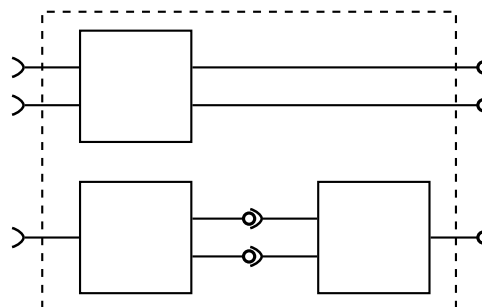


Se pueden conectar en serie dos o más componentes. Para ello, basta con concatenarlos uno seguido de otro, empleando las opciones del no terminal S. Por ejemplo, el componente anterior se puede conectar con otro que tenga dos interfaces de entrada. La cadena `<1;2><2;1>` representa el siguiente grafo:



La agrupación serie de dos componentes da lugar a un nuevo componente con tantos interfaces de entrada como el primer componente, y tantos interfaces de salida como el último componente.

También es posible agrupar en paralelo dos o más componentes. Para ello es necesario concatenarlos empleando las opciones del no terminal P. Por ejemplo, la agrupación serie anterior se puede disponer en paralelo con un nuevo componente. La cadena `<2;2>&<1;2><2;1>` representa el siguiente grafo:



La agrupación en paralelo de dos componentes da lugar a un nuevo componente con tantos interfaces de entrada como la suma de interfaces de entrada de los componentes agrupados, y tantos interfaces de salida como la suma de interfaces de salida de los componentes agrupados.

Responder a las siguientes cuestiones:

- a) Considérese la siguiente gramática, que contiene algunas simplificaciones con respecto a la inicial:  $V_T = \{ \&, <, >, , ;, (, ), \text{int} \}$ ,  $V_N = \{ P, S, E \}$ , símbolo inicial  $P$  y el siguiente conjunto de producciones:

$$\begin{array}{lcl} P & \rightarrow & S \\ & | & S \& P \\ S & \rightarrow & E \\ & | & E S \\ E & \rightarrow & < \text{int} ; \text{int} > \\ & | & ( P ) \end{array}$$

Completar la colección LR(0) de esta gramática a partir de los conjuntos de ítems que se indican a continuación, y construir su tabla de análisis SLR. Indicar, justificadamente, si la gramática es SLR.

$$\begin{array}{ll} \text{GOTO}(I_0, P) = I_1 = \{ P' \rightarrow P \bullet \} & \text{GOTO}(I_2, \&) = I_6 = \{ P \rightarrow S \& \bullet P \\ & P \rightarrow \bullet S \\ & P \rightarrow \bullet S \& P \\ & S \rightarrow \bullet E \\ & S \rightarrow \bullet E S \\ & E \rightarrow \bullet < \text{int} ; \text{int} > \\ & E \rightarrow \bullet ( P ) \} \\ \text{GOTO}(I_0, S) = I_2 = \{ P \rightarrow S \bullet \\ & P \rightarrow S \bullet \& P \} \\ \text{GOTO}(I_0, <) = I_4 = \{ E \rightarrow < \bullet \text{int} ; \text{int} > \} \\ \text{GOTO}(I_3, S) = I_7 = \{ S \rightarrow E S \bullet \} \end{array}$$

6. Considerar la siguiente gramática:

$$\begin{array}{lcl} S & \rightarrow & LP ; E \\ LP & \rightarrow & LP ; num \\ & | & num \\ E & \rightarrow & num \\ & | & var \\ & | & fun ( E ) \end{array}$$

que genera frases consistentes en una lista de puntos (LP) y una expresión matemática sencilla en la que se emplean únicamente funciones **seno** y **coseno**. Por ejemplo, la gramática podría generar la siguiente cadena:

$$0,1;0,2;0,3; \text{seno}(\text{seno}(x))$$

- Calcular los conjuntos PRIMERO y SIGUIENTE.
- Calcular la tabla SLR(1) y comprobar si se trata de una gramática de este tipo.
- ¿Es la gramática LALR? ¿Y LR-Canónica?

7. Considerar la siguiente gramática:

$$\begin{array}{lcl} P & \rightarrow & [ L ] ( num ) \\ L & \rightarrow & E \\ & | & E , L \\ E & \rightarrow & num \\ & | & P \end{array}$$

que genera un lenguaje para evaluación de polinomios. Por ejemplo, la gramática podría generar la siguiente cadena:

$$[1, 3, [2, 1](2)](1)$$

donde los números de la lista entre corchetes representarían los coeficientes de un polinomio de una variable, y el número entre paréntesis, el valor que se quiere asignar a la variable para evaluar el polinomio. Por ejemplo, la frase  $[1, 2, 6](5)$  representaría al polinomio  $x^2 + 2x + 6$ , que deberá evaluarse asignando a  $x$  el valor 5.

- Calcular la colección LR(0) y la tabla de análisis para comprobar si se trata de una gramática SLR.
- Simular el comportamiento de algoritmo ascendente predictivo para reconocer la cadena  $'[2(3)](1)'$ , aplicando el método de recuperación de errores en modo pánico en caso de error.
- ¿Es la gramática LALR? ¿Y LR-Canónica?

8. Considerar la siguiente gramática G:

$$\begin{array}{lcl} P & \rightarrow & [ L ] ( num ) \\ L & \rightarrow & E \\ & | & L , E \\ E & \rightarrow & num \\ & | & P \end{array}$$

que genera un lenguaje para evaluación de polinomios. Por ejemplo, la gramática podría generar la siguiente cadena:

$$[1, 3, [2, 1](2)](1)$$

donde los números de la lista entre corchetes representarían los coeficientes de un polinomio de una variable, y el número entre paréntesis, el valor que se quiere asignar a la variable para evaluar el polinomio. Por ejemplo, la frase  $[1, 2, 6](5)$  representaría al polinomio  $x^2 + 2x + 6$ , que deberá evaluarse asignando a  $x$  el valor 5.

Construir la colección de items LR(1) para la gramática G original y comprobar si se trata de una gramática LR-Canónica y LALR. Razonar, también, sin construir la colección LR(0), si la gramática es SLR.

9. En el sistema formal denominado  $\lambda$ -Cálculo se definen los *números de Church* de la siguiente forma:

$$\begin{array}{l} \underline{0} \equiv \lambda f.(\lambda x.x) \\ \underline{1} \equiv \lambda f.(\lambda x.(fx)) \\ \underline{2} \equiv \lambda f.(\lambda x.(f(fx))) \\ \underline{3} \equiv \lambda f.(\lambda x.(f(f(fx)))) \\ \dots \end{array}$$

Considerar la siguiente gramática G para generar *números de Church*, con  $V_T = \{\lambda, ., (, ), f, x\}$ ,  $V_N = \{N, F, X, C\}$ , N el símbolo inicial, y P el siguiente conjunto de producciones<sup>1</sup>:

$$\begin{array}{lcl} N & \rightarrow & F . ( X . C ) \\ F & \rightarrow & \lambda f \\ X & \rightarrow & \lambda x \\ C & \rightarrow & ( f C ) \\ & | & x \end{array}$$

Construir la colección de items LR(1) para la gramática G original y comprobar si se trata de una gramática LR-Canónica y LALR. Razonar, también, sin construir la colección LR(0), si la gramática es SLR.

<sup>1</sup>Observar que en esta gramática, tanto el símbolo  $\lambda$  como el símbolo  $.$  forman parte de los tokens del lenguaje. Por tanto, no debéis confundir en este caso el símbolo  $\lambda$  con la cadena vacía.