



Apellidos, nombre:

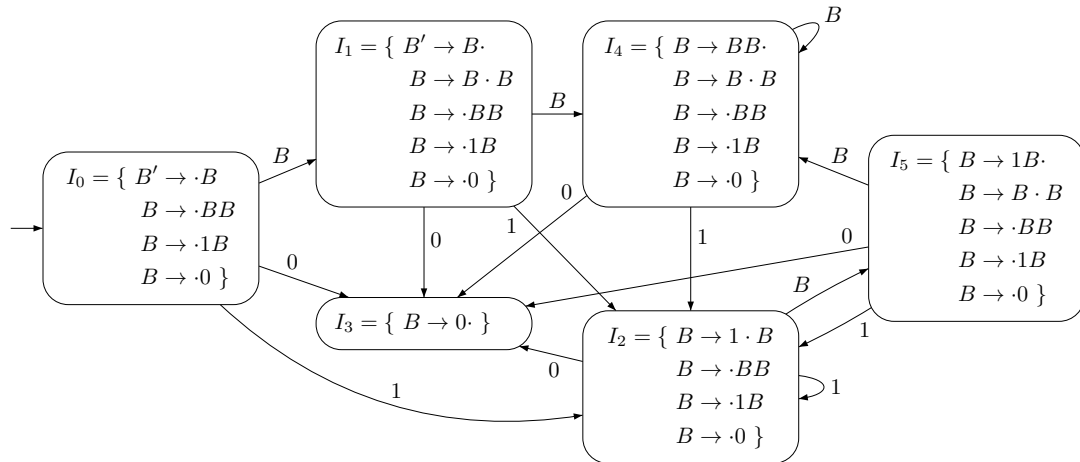
DNI:

Instrucciones: Este enunciado y todos los folios usados deben entregarse al salir

Grupo:

Parte I: PREGUNTAS TIPO TEST. 40%. Cada dos respuestas incorrectas anulan una correcta.

1. A continuación se muestra el autómata de la colección LR(0) de la gramática  $B \rightarrow BB \mid 1B \mid 0$ . Supongamos que se quiere emplear la concatenación de símbolos con asociatividad izquierda. Con entradas válidas, algunas de las transiciones del autómata no se usarían tras resolver los conflictos. Indica la respuesta correcta.



- a) No se usaría la transición de  $I_0$  a  $I_3$ .
- b) Desde  $I_5$  no se usarían las transiciones que llegan a  $I_2$  e  $I_3$ .
- c) No se usaría la transición que llega a  $I_2$  desde él mismo.
2. Dada la colección LR(0) anterior, considera la entrada 10\$. Marca la respuesta correcta:
- a) Al terminar de reconocer la entrada se encontraría en el estado  $I_1$ .
- b) Al terminar de reconocer la entrada se encontraría en el estado  $I_3$ .
- c) El autómata no usaría en ningún caso el estado  $I_5$ .
3. Supongamos que se tiene una gramática  $G$  que es LALR. Indica la respuesta correcta:
- a) La tabla SLR de  $G$  puede tener más acciones de desplazamiento que la LALR.
- b) La tabla SLR de  $G$  puede tener menos estados que la LALR.
- c) La tabla SLR de  $G$  puede tener conflictos reduce/reduce.
4. Supongamos que tenemos una gramática  $G$  con las reglas  $P \rightarrow P \wedge P \mid P \vee P \mid \text{true} \mid \text{false}$ , siendo  $V_T = \{\wedge, \vee, \text{true}, \text{false}\}$  y  $V_N = \{P\}$ . Supongamos ahora que realizamos transformaciones en la gramática para realizar un análisis LL(1), obteniendo una gramática  $G'$  equivalente que es LL(1). Indica la respuesta correcta:
- a) Se puede definir una DDS S-atribuida sobre  $G'$  para evaluar el valor lógico de las entradas válidas.
- b) No se puede definir una DDS S-atribuida sobre  $G'$ , pero sí sobre  $G$ .
- c) Es necesaria una DDS L-atribuida sobre  $G'$  para evaluar el valor lógico de las entradas válidas.
5. Supongamos que implementamos con Flex un analizador léxico de la gramática anterior, sustituyendo  $\wedge$  por `&&` y  $\vee$  por `||`, y añadimos el símbolo terminal `;` para separar expresiones lógicas:

```
%{
#include <stdio.h>
#include "sintactico.tab.h"
int inicio_error = 0;
void msg_error();
int contador_errores = 0;
%}
espacio      [ \n\r\t]
%x error_lex
%%
```

```

{espacio}+      { }
"&&"          { return AND; }
"||"           { return OR; }
";"            { return PYC; }
"true"         { yylval.ent = 1; return VAL; }
"false"        { yylval.ent = 0; return VAL; }
.              { inicio_error = yylineno; yymore(); BEGIN(error_lex); }
<error_lex>{espacio}|";" { yyless(yyleng-1); msg_error(); BEGIN(0); }
<error_lex>.     { yymore(); }
<error_lex><<EOF>> { msg_error(); return 0; }
%%
void msg_error() {
    printf("Error lexico en linea %d: %s\n", inicio_error, yytext);
    contador_errores++;
}

```

donde `yyless(n)` devuelve a la entrada los caracteres de `yytext` desde `n` en adelante. Indica la respuesta verdadera suponiendo que la entrada a este analizador es `true && (false true || true)`;

- a) El contador de errores termina tomando el valor 1.
- b) El contador de errores termina tomando el valor 2.
- c) El analizador léxico devuelve siete tokens.

6. Supongamos la siguiente implementación de un analizador sintáctico con Bison para la gramática de expresiones lógicas, que se combina con el analizador léxico anterior:

```

%{
#include <stdio.h>
extern int yylineno;
extern int yylex();
extern int yyparse();
extern int contador_errores;
void yyerror(const char *msg);
}%
%union {
    int ent;
}
%token AND OR PYC
%token <ent> VAL
%type <ent> predicado
%left OR
%left AND
%%
entrada      : predicado PYC                { if (contador_errores == 0)
                                              printf("Valor: %d\n", $1);
                                              }
;
predicado    : predicado AND predicado { $$ = $1 && $3; }
              | predicado OR predicado { $$ = $1 || $3; }
              | VAL                    { $$ = $1; }
;
%%
void yyerror(const char *msg) {
    printf("Error sintactico en linea %d: %s\n", yylineno, msg);
}
int main(int argc, char **argv) {
    yyparse();
}

```

Supongamos la entrada `true && false || true`; . Indica la respuesta verdadera:

- a) Primero se evaluaría el operador `&&` y luego el operador `||`.
- b) Primero se evaluaría el operador `||` y luego el operador `&&`.
- c) Habría un conflicto deslaza/reduce sin resolver.

7. Si se construye un analizador LALR de la siguiente gramática para representar operaciones de conjuntos:

$$\begin{aligned}
 C &\rightarrow C \cup C \mid C \cap C \mid (C) \mid \{L\} \\
 L &\rightarrow L, L \mid id \mid \lambda
 \end{aligned}$$

se obtiene una tabla que contiene, entre otras, las siguientes filas:

ESTADO	ACCIÓN						IR-A				
	(	)	,	id	$\cap$	$\cup$	{	}	\$	$\mathcal{C}$	$\mathcal{L}$
9				r1	r1/d5	r1/d4			r1		
10				r2	r2/d5	r2/d4			r2		
14				r5/d13					r5		

siendo los conjuntos de ítems de los estados indicados los siguientes:

$$I_9 = \{ [ C \rightarrow C \cup C \bullet, \$/\cap/\cup ], [ C \rightarrow C \bullet \cup C, \$/\cap/\cup ], [ C \rightarrow C \bullet \cap C, \$/\cap/\cup ] \}$$

$$I_{10} = \{ [ C \rightarrow C \cap C \bullet, \$/\cap/\cup ], [ C \rightarrow C \bullet \cap C, \$/\cap/\cup ], [ C \rightarrow C \bullet \cup C, \$/\cap/\cup ] \}$$

$$I_{14} = \{ [ L \rightarrow L, L \bullet, \}/, ], [ L \rightarrow L \bullet, L, \}/, ] \}$$

Considera que siempre se utiliza asociatividad izquierda, y que  $\cap$  tiene más precedencia que  $\cup$ . Se emplea  $M[s, c]$  para indicar la casilla de la tabla en la fila del estado  $s$  y la columna del símbolo  $c$ . Indica la forma de resolver los conflictos:

- $M[9, \cap] = r1$  ;  $M[9, \cup] = r1$  ;  $M[10, \cup] = d5$  ;  $M[10, \cap] = r2$  ;  $M[14, ,] = r5$
- $M[9, \cap] = r1$  ;  $M[9, \cup] = r1$  ;  $M[10, \cup] = r2$  ;  $M[10, \cap] = r2$  ;  $M[14, ,] = r5$
- $M[9, \cap] = d5$  ;  $M[9, \cup] = r1$  ;  $M[10, \cup] = r2$  ;  $M[10, \cap] = r2$  ;  $M[14, ,] = r5$

8. Dada la siguiente gramática  $G$ , con  $V_T = \{a, b, x, y\}$  y  $V_N = \{S, A, B, C\}$ , y  $P$ :

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

respecto a la simulación descendente predictiva, señalar la afirmación incorrecta:

- Si en la pila de análisis aparece \$CBA y en la entrada b\$, en el siguiente paso de cálculo, la pila contendría \$CB y la entrada b\$.
- Si en la pila de análisis aparece \$CBA y en la entrada yb\$, en el siguiente paso de cálculo, la pila contendría \$CB y la entrada b\$.
- Si en la pila de análisis aparece \$CBA y en la entrada yb\$, en el siguiente paso de cálculo, la pila contendría \$CBA y la entrada b\$.

9. Dada la siguiente gramática  $G$  y su correspondiente tabla LALR:

$$\begin{aligned} (1) S &\rightarrow s ( A ) \\ (2) A &\rightarrow A , B \\ (3) &\mid B \\ (4) B &\rightarrow b \end{aligned}$$

ESTADO	ACCIÓN					IR-A				
	,	(	)	b	s	\$	A	B	S	
0						d2				1
1						acepta				
2						d3				
3						d6	4	5		
4	d8					d7				
5	r3					r3				
6	r4					r4				
7						r1				
8						d6				9
9	r2					r2				

- $G$  es LL, LALR y LR-Canónica.
- $G$  es SLR, LALR, LR-Canónica.
- $G$  es LALR y LR-Canónica, pero no SLR.

10. Dada la gramática  $G$  de la pregunta anterior, y con respecto a la simulación LR:

- Si el contenido de la pila de un analizador LR es 0s2(3A4,8B9 y el de la entrada ,b)\$, la siguiente configuración tendrá en la pila 0s2(3A4 y en la entrada b)\$.
- Si el contenido de la pila de un analizador LR es 0s2(3A4,8B9 y el de la entrada (b)\$, la siguiente configuración tendrá en la pila 0s2(3A4,8B9 y en la entrada )\$.
- Si el contenido de la pila de un analizador LR es 0s2(3A4,8b6 y el de la entrada ,b)\$, la siguiente configuración tendrá en la pila 0s2(3A4,8B9 y en la entrada b)\$.

**Parte II: EJERCICIOS. 60 %.**

La siguiente gramática  $G_1$ , con  $V_T = \{i, (, ), =\}$ ,  $V_N = \{S, A\}$  y  $P$ :

- (1)  $S \rightarrow i = A$
- (2)  $A \rightarrow A = A$
- (3)  $A \rightarrow i$
- (4)  $A \rightarrow (A)$

sirve para representar sentencias de asignación en donde el operador  $=$  es asociativo por la derecha.

1. (0,5 p.) Indica si  $G_1$  tiene alguna propiedad (o propiedades) que, de partida, le impidan ser LL(1). En caso de ser así, especificar cuáles.

La regla  $A \rightarrow A = A$  de la gramática  $G$  provoca dos propiedades que impiden que sea LL(1):

- Es ambigua. Por ejemplo, la cadena  $i = i = i = i$  se puede derivar de varias formas.
- Es recursiva por la izquierda.

2. (1 p.) Teniendo en cuenta la asociatividad del operador  $=$ , realiza transformaciones para obtener una gramática  $G'_1$  equivalente a  $G_1$  y que se pueda analizar con el método LL(1).

Teniendo en cuenta que  $=$  es asociativo por la derecha, se puede eliminar la ambigüedad de la siguiente forma:

- $$\begin{aligned} S &\rightarrow i = A \\ A &\rightarrow F = A \mid F \\ F &\rightarrow i \mid (A) \end{aligned}$$

A continuación, es necesario eliminar los factores comunes de las reglas de  $A$ , obteniendo  $G'_1$ :

- (1)  $S \rightarrow i = A$
- (2)  $A \rightarrow F A'$
- (3)  $A' \rightarrow = A$
- (4)  $A' \rightarrow \lambda$
- (5)  $F \rightarrow i$
- (6)  $F \rightarrow (A)$

3. (0,75 p.) Calcula los conjuntos PRIMERO, SIGUIENTE y PREDICT de los no terminales de  $G'_1$ .

PRIMERO( $S$ ) =  $\{i\}$   
 PRIMERO( $A$ ) =  $\{i, (\}$   
 PRIMERO( $A'$ ) =  $\{=, \lambda\}$   
 PRIMERO( $F$ ) =  $\{i, (\}$

SIGUIENTE( $S$ ) =  $\{\$ \}$   
 SIGUIENTE( $A$ ) =  $\{\$, )\}$   
 SIGUIENTE( $A'$ ) =  $\{\$, )\}$   
 SIGUIENTE( $F$ ) =  $\{=, \$, )\}$

PREDICT(1) =  $\{i\}$   
 PREDICT(2) =  $\{i, (\}$   
 PREDICT(3) =  $\{=\}$

PREDICT(4) =  $\{\$, )\}$   
 PREDICT(5) =  $\{i\}$   
 PREDICT(6) =  $\{(\}$

4. (0,75 p.) Crea la tabla de análisis LL de  $G'_1$  y razona si es LL(1).

NO TERM	TERMINAL				
	$i$	$=$	$($	$)$	$\$$
$S$	1				
$A$	2		2		
$A'$		3		4	4
$F$	5		6		

La tabla no tiene ningún conflicto. Por tanto, podemos decir que  $G'_1$  es LL(1).

Dada la siguiente gramática  $G_2$ , con  $V_T = \{a, b\}$ ,  $V_N = \{S, A, B, C\}$  y  $P$ :

- (1)  $S \rightarrow A B$
- (2)  $S \rightarrow B$
- (3)  $S \rightarrow C a$
- (4)  $A \rightarrow b B a$
- (5)  $B \rightarrow A$
- (6)  $B \rightarrow b$
- (7)  $C \rightarrow A$

5. (1 p.) Completa la colección LR(1) para la gramática  $G_2$  calculando los conjuntos  $I_0$ ,  $I_5$  e  $I_8$  según los GOTO que se indican. Calcula también los GOTO que faltan, es decir, que conducen a estados previamente calculados.

$$I_0 = \{ [ S' \rightarrow \cdot S, \$ ] \\ [ S \rightarrow \cdot AB, \$ ] \\ [ S \rightarrow \cdot B, \$ ] \\ [ S \rightarrow \cdot Ca, \$ ] \\ [ A \rightarrow \cdot bBa, b/\$/a ] \\ [ B \rightarrow \cdot A, \$ ] \\ [ B \rightarrow \cdot b, \$ ] \\ [ C \rightarrow \cdot A, a ] \}$$

$$I_1 = \text{GOTO}(I_0, S) = \{ [ S' \rightarrow S \cdot, \$ ] \}$$

$$I_2 = \text{GOTO}(I_0, A) = \{ [ S \rightarrow A \cdot B, \$ ] \\ [ B \rightarrow A \cdot, \$ ] \\ [ C \rightarrow A \cdot, a ] \\ [ B \rightarrow \cdot A, \$ ] \\ [ B \rightarrow \cdot b, \$ ] \\ [ A \rightarrow \cdot bBa, \$ ] \}$$

$$I_3 = \text{GOTO}(I_0, B) = \{ [ S \rightarrow B \cdot, \$ ] \}$$

$$I_4 = \text{GOTO}(I_0, C) = \{ [ S \rightarrow C \cdot a, \$ ] \}$$

$$I_5 = \text{GOTO}(I_0, b) = \{ [ A \rightarrow b \cdot Ba, b/\$/a ] \\ [ B \rightarrow b \cdot, \$ ] \\ [ B \rightarrow \cdot A, a ] \\ [ B \rightarrow \cdot b, a ] \\ [ A \rightarrow \cdot bBa, a ] \}$$

$$I_6 = \text{GOTO}(I_2, B) = \{ [ S \rightarrow AB \cdot, \$ ] \}$$

$$I_7 = \text{GOTO}(I_2, A) = \{ [ B \rightarrow A \cdot, \$ ] \}$$

$$I_8 = \text{GOTO}(I_2, b) = \{ [ B \rightarrow b \cdot, \$ ] \\ [ A \rightarrow b \cdot Ba, \$ ] \\ [ B \rightarrow \cdot A, a ] \\ [ B \rightarrow \cdot b, a ] \\ [ A \rightarrow \cdot bBa, a ] \}$$

$$I_9 = \text{GOTO}(I_4, a) = \{ [ S \rightarrow Ca \cdot, \$ ] \}$$

$$I_{10} = \text{GOTO}(I_5, B) = \{ [ A \rightarrow bB \cdot a, b/\$/a ] \}$$

$$I_{11} = \text{GOTO}(I_5, A) = \{ [ B \rightarrow A \cdot, a ] \}$$

$$I_{12} = \text{GOTO}(I_5, b) = \{ [ A \rightarrow b \cdot Ba, a ] \\ [ B \rightarrow b \cdot, a ] \\ [ B \rightarrow \cdot A, a ] \\ [ B \rightarrow \cdot b, a ] \\ [ A \rightarrow \cdot bBa, a ] \}$$

$$I_{13} = \text{GOTO}(I_8, B) = \{ [ A \rightarrow bB \cdot a, \$ ] \}$$

$$I_{14} = \text{GOTO}(I_{10}, a) = \{ [ A \rightarrow bBa \cdot, b/\$/a ] \}$$

$$I_{15} = \text{GOTO}(I_{12}, B) = \{ [ A \rightarrow bB \cdot a, a ] \}$$

$$I_{16} = \text{GOTO}(I_{13}, a) = \{ [ A \rightarrow bBa \cdot, \$ ] \}$$

$$I_{17} = \text{GOTO}(I_{15}, a) = \{ [ A \rightarrow bBa \cdot, a ] \}$$

$$\begin{aligned} \text{GOTO}(I_8, b) &= I_{12} \\ \text{GOTO}(I_8, A) &= I_{11} \\ \text{GOTO}(I_{12}, b) &= I_{12} \\ \text{GOTO}(I_{12}, A) &= I_{11} \end{aligned}$$

6. (0,75 p.) Calcula las filas correspondientes a los estados 2, 5 y 14 de la tabla LR-Canónica.

ESTADO	ACCIÓN			IR-A			
	$a$	$b$	$\$$	$S$	$A$	$B$	$C$
2	r7	d8	r5		7	6	
5		d12	r6		11	10	
14	r4	r4	r4				

7. (0,75 p.) Indica qué conjuntos se agrupan, y su contenido, para obtener la colección LALR.

$$I_{5-8-12} = \{ [ A \rightarrow b \cdot Ba , b/\$/a ] \\ [ B \rightarrow b \cdot , \$/a ] \\ [ B \rightarrow \cdot A , a ] \\ [ B \rightarrow \cdot b , a ] \\ [ A \rightarrow \cdot bBa , a ] \}$$

$$I_{10-13-15} = \{ [ A \rightarrow bB \cdot a , b/\$/a ] \}$$

$$I_{7-11} = \{ [ B \rightarrow A \cdot , \$/a ] \}$$

$$I_{14-16-17} = \{ [ A \rightarrow bBa \cdot , b/\$/a ] \}$$

8. (0,5 p.) Justifica si la gramática es LR-Canónica, LALR y/o SLR.

Analizando la colección LR(1) obtenida en el apartado 5 se puede comprobar que los estados con acciones de reducción y desplazamiento no presentan conflictos r/r ni d/r. Estos estados son  $I_2$ ,  $I_5$ ,  $I_8$  e  $I_{12}$ . Por tanto, la gramática es LR-Canónica.

Analizando los estados obtenidos en el apartado 7 se puede comprobar que no se producen conflictos r/r. Por tanto, la gramática es LALR.

Analizando el estado  $I_2$ , y teniendo en cuenta que  $\text{SIGUIENTE}(B) = \{\$, a\}$  y  $\text{SIGUIENTE}(C) = \{a\}$ , podemos ver que se producirá un conflicto r/r con el símbolo  $a$  en la colección SLR entre los ítems  $B \rightarrow A \cdot$  y  $C \rightarrow A \cdot$ . Por tanto, la gramática no es SLR.