Teoría de Lenguajes - Segundo Parcial

Primer cuatrimestre de 2022

No usar celulares.
Hacer cada ejercicio en hojas separadas.
Poner nombre, número de libreta y firma en cada página.
Justificar todas las respuestas.
El examen es a libro abierto.
Se aprueba con al menos 65 puntos.

1. (32 pts) Dada la gramática extendida $G =< \{S\}, \{1,2,3,4\}, P, S >$, con P:

   $$S \rightarrow (1 \mid 2 \mid 3)^+(3 \mid 4)$$

   a) Determinar si G es ELL(1). Si no lo es, dar otra gramática extendida G' equivalente que sí lo sea y tal que $|V_N(G')| \leq 2$.

   b) Exhibir un parser iterativo recursivo para $L(G)$.

2. (34 pts) Dada la siguiente gramática $G =< \{S, A, B, C\}, \{a, b\}, P, S >$, con P:

   $$\begin{aligned} &S \rightarrow CC \mid aASA \mid \lambda \\ &A \rightarrow a \\ &B \rightarrow S \mid \lambda \\ &C \rightarrow bC \mid CA \end{aligned}$$

   a) Eliminar renombramientos y símbolos inútiles.

   b) Determinar si la gramática resultante es LALR(1), indicar todos los conflictos que pudiera haber asociados a la tabla de parsing y discutir la posibilidad de solucionarlos sin alterar el lenguaje.

   c) Determinar si el lenguaje $L(G)$ es SLR(1).

3. (34 pts) Dada la siguiente gramática: $G =< \{E, M, N\}, \{(,), p\}, P, E >$, con P:

   $$\begin{aligned} &E \rightarrow E(Np \mid E(M) \mid \lambda \\ &M \rightarrow M(M) \mid \lambda \\ &N \rightarrow (M)N \mid (N \mid \lambda \end{aligned}$$

   Agregar a G atributos y/o reglas semánticas de modo que resulte una gramática de atributos que permita sintetizar en el atributo *exp* de S la expresión reconocida pero en la cual se ha reemplazado cada carácter p por los paréntesis de cierre consecutivos necesarios para que todos los paréntesis queden balanceados, y sintetizar en el atributo *cuantas* de S la cantidad de subcadenas iguales a () que había en la expresión original. Por ejemplo, para la cadena de entrada ()((p(()((p(p la cadena resultante será ()(())(()(()))() y la cantidad de subcadenas () será 2.

MICELI, JUAN PABLO
LU: 424/19

2°P TLENG JPm

HOJA N° 3
FECHA 23/06/2022

1) a) $S \rightarrow (1|2|3)^{+}(3|4)$

(11)

1	2	3	tot
17	18	34	69

A

		SD
$A_1 = (1\|2\|3)^{+}$ $A_2 = (3\|4)$	$S \rightarrow A_1 A_2$	$\{1,2,3\}$
$A_3 = (1\|2\|3)$ $A_4 = (1\|2\|3)^{*}$	$A_1 \rightarrow A_3 A_4$	$\{1,2,3\}$
	$A_3 \rightarrow 1 \mid$	$\{1\}$
	$2 \mid$	$\{2\}$
	3	$\{3\}$
	$A_4 \rightarrow A_3 A_4 \mid$	$\{1,2,3\}$
	λ	$\{3,4\}$
	$A_2 \rightarrow 3 \mid$	$\{3\}$
	4	$\{4\}$

conflicto

hubiera sido bueno mostrar que problema hay ahí.

Esta gramática no es ELL(1) porque su derivada no es LL(1). Entonces buscando una gramática que genere números de por lo menos 2 cifras que terminen en 3 o en 4 y que las demás posiciones tengan sólo 1, 2, o 3. Existirá una gramática así ELL(1)? Sólo dios lo sabe, porque yo no la encontré.

b)

```
S():
    do:
        if tc == "1":
            match("1")
        elif tc == "2":
            match("2")
        elif tc == "3":
            match("3")
        else error
    while (tc in {"1", "2", "3"})
    if tc == "3":
        match("3")
    elif tc == "4":
        match("4")
    else error
    match($)
```

Obviamente este parser no anda porque no tenemos una gramática ELL(1), pero bueno, era el único parser que podía hacer.

NOTA

MICELI, JUAN PABLO

LU: 424/19

2°P TLENG

HOJA N° 2

FECHA 23/06/2022

2) $S \to \cancel{CC} \mid a\,ASA \mid \lambda$

11

$A \to a$

~~$B \to S \mid \lambda$~~ indecorgable

~~$C \to bC \mid CA$~~ inactivo

Primero era renombres, $B \to S$

a) Elimino símbolos inactivos:

$\to \{A, B, S\}$ pues $A \to a$, $B \to \lambda$ y $S \to \lambda$.

$\{\} \to \{A\} \to \{A, B\} \to \{A, B, S\} \to \{A, B, S\}$ Elimino a C

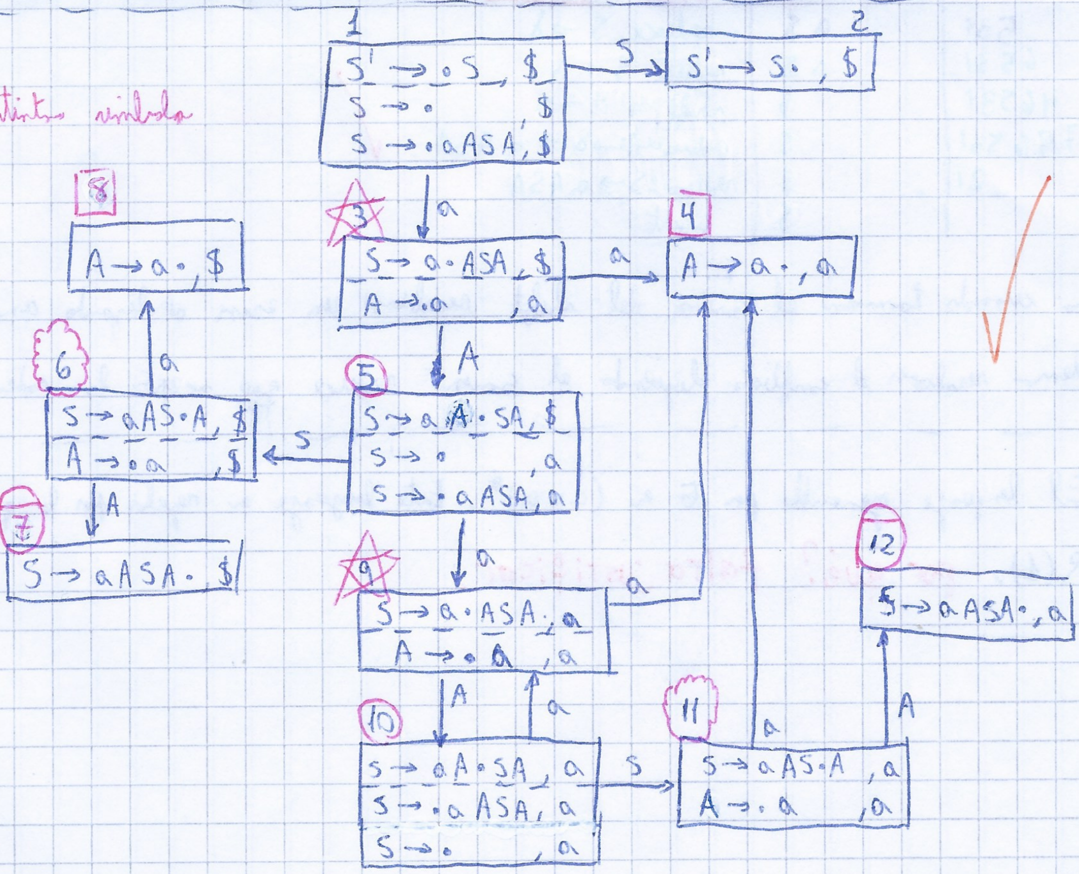
Elimino símbolos indecorgables:

$\{S\} \to \{S, A\} \to \{S, A\}$ Elimino a B

No hay renombramientos $\Longrightarrow$ $\boxed{\begin{array}{l} S \to a\,ASA \mid \lambda \\ A \to a \end{array}}$

b)

Marco con distintos símbolos los estados mergeables en LALR(1)



3 y 9 ⇒ 3

4 y 8 ⇒ 4

5 y 10 ⇒ 5

6 y 11 ⇒ 6

7 y 12 ⇒ 7

NOTA

	a	$	S	A
1	s3	r(S→λ)	2	
2		accept		
3	s4			5
4	~~r(A→a)~~ r(A→a)	r(A→a)		
5	s3 / r(S→λ)		6	
6	s4			7
7	r(S→aASA)	r(S→aASA)		

G No es LALR(1) ya que hay un conflicto shift-reduce en el estado 6. ✓

Veamos si podemos solucionar los conflictos:

Probemos analizar la cadena w=aaa.

pila	cadena	acción
1	aaa$	shift 3
31	aa$	shift 4
431	a$	reduce A→a
531	a$	shift 3
3531	$	error
531	a$	reduce S→λ
6531	a$	shift 4
46531	$	reduce A→a
76531	$	reduce S→aASA
21	$	accept

pero si probabas con r(S→λ) para "aaaaaa", no iba a funcionar

Como cuando tomamos el camino del shift recibimos un error analizando una cadena válida, podemos resolver el conflicto eligiendo el camino reduce que acepta la cadena normalmente.

c) El lenguaje generado por G es $(aaa)^*$. Este lenguaje es regular por lo que también es SLR(1). por qué? falta justificar.