



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3725 - Traductores e Interpretadores
Enero-Marzo 2016

Proyecto 3

Traductores

Samuel Arleo
10-10969

Sergio Terán
11-11020

07 de Marzo, 2016

1. Formulación e Implementación

2. Revisión Teórico-Practica

Pregunta 1

(a)

$$(a.1) \ G_1 = (\{S\}, \{a\}, \{S \rightarrow Sa, S \rightarrow \lambda\}, S)$$

Determinemos si la gramática

$$\begin{array}{l} S \rightarrow Sa \\ S \rightarrow \lambda \end{array}$$

Es LR(1) y construyamos su analizador sintáctico. Comenzamos por aumentar la gramática con el símbolo S' y agregando el símbolo $\$$ al final de la primera entrega. A demás enumeramos las producciones.

$$\begin{array}{lll} (i) & S' & \rightarrow S\$ \\ (ii) & S & \rightarrow Sa \\ (iii) & S & \rightarrow \lambda \end{array}$$

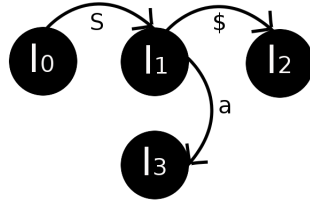
Construimos los conjuntos FIRST y FOLLOW para los simbolos no terminales:

$$\begin{array}{l} \text{FIRST}(S') = \{ \lambda, a \} \\ \text{FIRST}(S) = \{ \lambda, a \} \\ \text{FOLLOW}(S') = \{ \$ \} \\ \text{FOLLOW}(S) = \{ a, \$ \} \end{array}$$

El conjunto de clauduras nos queda:

$$\begin{array}{lll} I_0 : & S' & \rightarrow \cdot S\$ \\ & S & \rightarrow \cdot Sa \\ & S & \rightarrow \cdot \\ I_1 : & S' & \rightarrow S \cdot \$ \\ & S & \rightarrow S \cdot a \\ I_2 : & S' & \rightarrow S\$ \cdot \\ I_3 : & S & \rightarrow Sa \cdot \end{array}$$

Construimos el automata de prefijos viables, que nos queda de la forma:



Ahora podemos construir la tabla de parsing $SLR(1)$:

	Acciones		Goto	
	a	\$	S'	S
I_0	r(iii)	r(iii)		1
I_1	s(3)	s(2)		
I_2	acc			
I_3	r(ii)	r(ii)		

$$(a.2) \ G_1 = (\{S\}, \{a\}, \{S \rightarrow aS, S \rightarrow \lambda\}, S)$$

Determinemos si la gramática

$$\begin{aligned} S &\rightarrow aS \\ S &\rightarrow \lambda \end{aligned}$$

Es LR(1) y construyamos su analizador sintáctico. Comenzamos por aumentar la gramática con el símbolo S' y agregando el símbolo $\$$ al final de la primera entrega. A demás enumeramos las producciones.

$$\begin{aligned} (i) \quad S' &\rightarrow S\$ \\ (ii) \quad S &\rightarrow aS \\ (iii) \quad S &\rightarrow \lambda \end{aligned}$$

Construimos los conjuntos FIRST y FOLLOW para los simbolos no terminales:

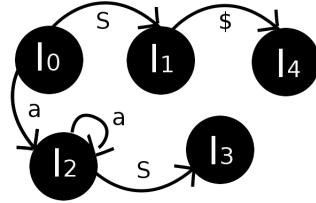
$$\begin{aligned} \text{FIRST}(S') &= \text{FIRST}(S) = \{ \lambda, a \} \\ \text{FOLLOW}(S') &= \text{FOLLOW}(S) = \{ \$ \} \end{aligned}$$

El conjunto de clauduras nos queda:

$$\begin{aligned} I_0 : \quad S' &\rightarrow \cdot S\$ \\ &S \rightarrow \cdot aS \\ &S \rightarrow \cdot \\ I_1 : \quad S' &\rightarrow S \cdot \$ \\ I_2 : \quad S &\rightarrow a \cdot S \\ &S \rightarrow \cdot aS \\ &S \rightarrow \cdot \end{aligned}$$

$$\begin{array}{lcl} I_3 & : & S \longrightarrow aS \cdot \\ I_4 & : & S' \longrightarrow S\$ \cdot \end{array}$$

Construimos el automata de prefijos viables, que nos queda de la forma:



Ahora podemos construir la tabla de parsing $SLR(1)$:

	Acciones		Goto	
	a	\$	S'	S
I_0	s(2)	r(iii)		1
I_1		s(4)		
I_2	s(2)	r(iii)		3
I_3		r(ii)		
I_4		acc		

(b) ANALISIS *SERIO* DE EFICIENCIA

Pregunta 2

- (a)
- (i) $S' \longrightarrow Instr$
 - (ii) $Instr \longrightarrow Instr ; Instr$
 - (iii) $Instr \longrightarrow IS$

$$\text{FIRST}(S') = \text{FIRST}(Instr) = \{ IS \}$$

$$\text{FOLLOW}(S') \{ \$ \}$$

$$\text{FOLLOW}(Instr) \{ ;, \$, IS \}$$

$$\begin{array}{lcl} I_0 & : & S' \longrightarrow \cdot Instr \$ \\ & & Instr \longrightarrow \cdot Instr ; Instr \\ & & Instr \longrightarrow \cdot IS \\ I_1 & : & S' \longrightarrow Instr \cdot \$ \\ & & Instr \longrightarrow Instr \cdot ; Instr \\ I_2 & : & Instr \longrightarrow IS \cdot \\ I_3 & : & S' \longrightarrow Instr \$ \cdot \\ I_4 & : & Instr \longrightarrow Instr ; \cdot Instr \\ & & Instr \longrightarrow \cdot Instr ; Instr \\ & & Instr \longrightarrow \cdot IS \end{array}$$

$$I_5 : \quad \begin{array}{l} Instr \longrightarrow Instr ; Instr \cdot \\ Instr \longrightarrow Instr \cdot ; Instr \end{array}$$

En la regla I_5 vemos que existe un conflicto

- (b) Este conflicto, del tipo *shift/reduce*, se puede solucionar usando el algoritmo de SLR(1), apoyandonos con los FIRST y FOLLOW que ya hemos calculado.
- (c)
- (d)