

República Bolivariana de Venezuela
Ministerio del Poder Popular para la Educación Superior
Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
Traductores e Interpretadores
Profesor: MSc Ricardo Monascal

INFORME

ETAPA 3

Autores:

Br. Jean Alexander 12-10848

Br. Diego Pedroza 12-11281

Caracas, Junio de 2016

Índice de contenido

IMPLEMENTACIÓN.....	4
Revisión Teórico-prácticas.....	4
PREGUNTA 1.....	5
PREGUNTA 2.....	8

IMPLEMENTACIÓN

La tercera etapa del proyecto de Traductores, es realizar una tabla de símbolo para el lenguaje Neo, el cual tiene palabras reservadas, identificadores, comentarios, caracteres, números y operadores. La finalidad del analizador es revisar si existe algún error estático, principalmente de tipo mostrando los sintácticos y los errores léxicos.

Para la segunda etapa, se usó Haskell como lenguaje de programación como en la entrega anterior, con la herramienta Alex, un analizador lexicográfico. Además, la herramienta Happy, un parser de la plataforma de Haskell. Primero se realizó la gramática correspondiente al lenguaje Neo, y luego siguiendo ejemplos y recomendaciones se desarrolló un parser para Neo.

Para la tercera etapa se aumentó la gramática dispuesta en la segunda etapa para que ejerciera una gramática de atributos. Del mismo modo que en entrega pasada, primero se pensó en la gramática, en los casos problemáticos y en el diseño, antes de la elaboración siguiendo los ejemplos de la herramienta Happy.

De esta forma, se procederá a responder las preguntas teórico-prácticas

Revisión Teórico-prácticas

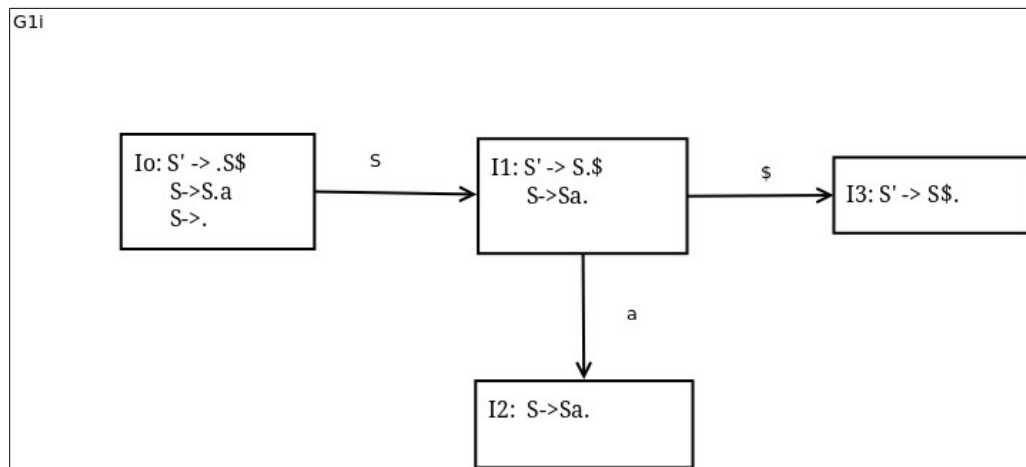
Se utilizó Dia de Gnome para realizar las imágenes y las tablas de libre office para la representación de las Tablas LR(1) y las de pilas. Se procederá a responder las preguntas.

1 PREGUNTA 1

Sea G_{li} la gramática recursiva-izquierda ($\{ S \}$, $\{ a \}$, $\{ S \rightarrow Sa, S \rightarrow \lambda \}$, S) y sea G_{ld} la gramática recursiva-derecha ($\{ S \}$, $\{ a \}$, $\{ S \rightarrow aS, S \rightarrow \lambda \}$, S) . Ambas generan el Lenguaje denotado por la expresión regular a^* , i.e. el lenguaje $L(a^*)$.

a) Muestre que ambas gramáticas son LR(1) y construya sus analizadores sintácticos.

{ (i) $S \rightarrow Sa$, (ii) $S \rightarrow \lambda$, (iii) $S' \rightarrow S \$$ }

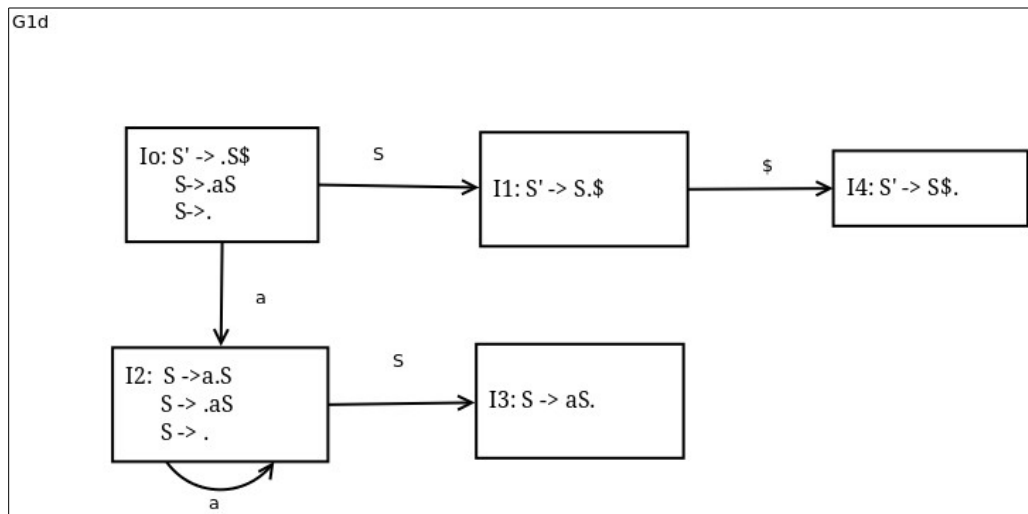


FOLLOW(S') : { \$ }

FOLLOW(S) : { \$,a }

	a	\$		S	S'
I0	Reduce ii			1	
I1	Shift 2	Shift3			
I2	Reduce i				
I3		Accept			

{ (d) $S \rightarrow aS$, (ii) $S \rightarrow \lambda$, (iii) $S' \rightarrow S \$$ }



FOLLOW(S'):{ \$ }

FOLLOW(S): { \$ }

	a	\$		S	S'
I0	Shift 2	Reduce ii		1	
I1		Shift 4			
I2	Shift 2	Reduce ii		3	
I3		Reduce i			
I4		Accept			

Como ninguno tiene conflictos sin resolver, entonces son LR(1)

b) Compare la eficiencia de ambos analizadores en términos de espacio, i.e. los tamaños de sus tablas y la cantidad de pila utilizada para reconocer cada frase de $L(a^*)$, y de tiempo, i.e. cantidad de movimientos realizados por el autómata de pila para reconocer cada frase de $L(a^*)$.

Indicador	G1i	G1d
Espacio	$O(n)$	$O(n)$
Cantidad de Pila	$O(1)$	$O(n)$
De Tiempo	$O(n)$	$O(n)$

La cantidad de espacio y de tiempo dependen del tamaño de la entrada. Sin embargo, se notó que la cantidad de pila en G1i por cada shift había un reduce lo que limita la pila en una constante mientras que la G1d se ampliaba.

Se utilizó los ejemplos aaa y aaaaaaa

2 PREGUNTA 2

Sea G_2 la gramática $(\{ Instr \}, \{ ;, IS \}, P_2, Instr)$, con P_2 compuesto por:

$$\begin{aligned} Instr &\rightarrow Instr ; Instr \\ &\quad | IS \end{aligned}$$

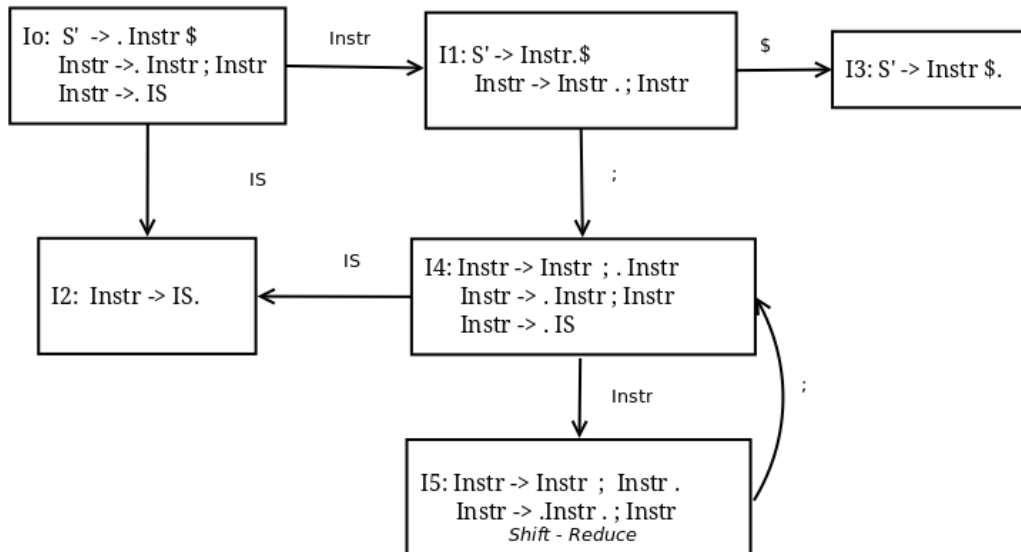
a) Muestre que G_2 no es una gramática LR(1), intentando construir un analizador sintáctico LR(1) para ella y consiguiendo que tal analizador tendrá

un conflicto.

(Se responderá con la **b**))

b) A pesar de que la gramática G2 no es LR(1), se puede construir un analizador a sintáctico LR(1) con conflictos para ella (lo cual corresponde a un autómata de pila no determinístico). Construya tal analizador sintáctico, especificando cuál es el conflicto y de qué tipo (i.e. shift-reduce o reduce-reduce) es.

G2
 (i) $S' \rightarrow Instr \$$
 (ii) $Instr \rightarrow Instr ; Instr$
 (iii) $Instr \rightarrow IS$



FOLLOW(S) : { \$ }

FOLLOW(S'): { \$ }

	;	IS	\$		S	Instr
I0		Shift 2				1
I1	Shift 4		Shift3			
I2	Reduce iii		Reduce iii			
I3			Accept			
I4		Shift 2				5
I5	Shift 4 / Reduce ii		Reduce ii			

Contiene un Conflicto Shift-Reduce, por lo tanto no es LR(1).

c) Considere las dos alternativas de eliminación del conflicto (i.e. en favor del shift o en favor del reduce en caso de un conflicto shift-reduce, o en favor de una producción o de otra en caso de un conflicto reduce-reduce). Muestre, para ambas alternativas de eliminación del conflicto, la secuencia de reconocimiento de la frase IS ; IS ; IS dando como salida la secuencia de producciones reducidas. ¿A qué corresponde cada una de las alternativas: a asociar el operador de secuenciación hacia la izquierda o hacia la derecha?

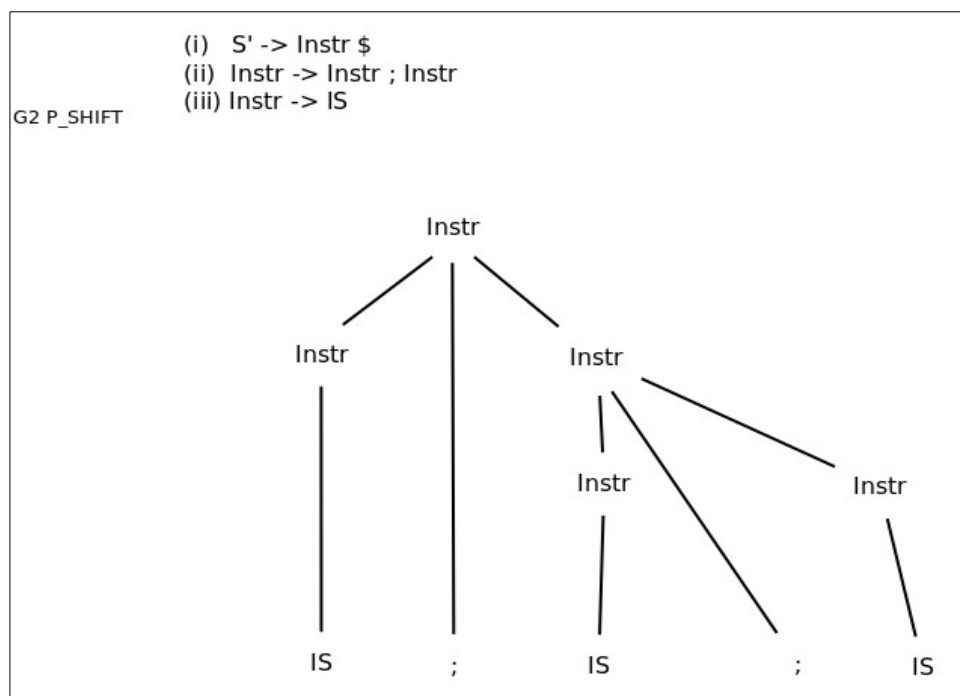
IS ; IS ; IS

Conflicto Shift-Reduce (Primera Prioridad Shift)

Pila	Entrada	Acción
0 \$	IS ; IS ; IS \$	Shift 2
20 \$; IS ; IS \$	Reduce iii
10 \$; IS ; IS \$	Shift 4
410 \$	IS ; IS \$	Shift 2

2 410 \$; IS \$	Reduce iii
5 410 \$; IS \$	Shift 4
45 410 \$	IS \$	Shift 2
245 410 \$	\$	Reduce iii
545 410 \$	\$	Reduce ii
5 410 \$	\$	Reduce ii
10 \$	\$	Shift 3
310 \$	\$	Accept

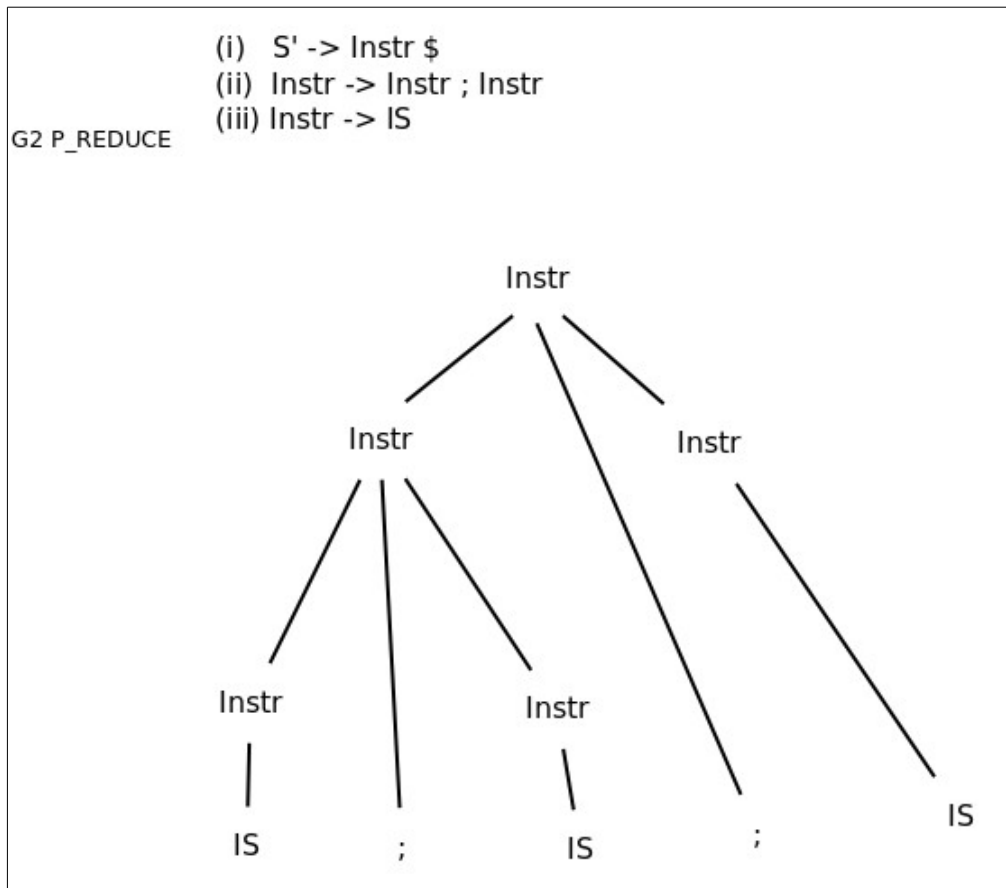
El árbol de sintáctico de esta gramática tomando en cuenta el shift como prioridad, asocia para la derecha como se puede observar:



Conflicto Shift-Reduce (Primera Prioridad Reduce)

Pila	Entrada	Acción
0 \$	IS ; IS ; IS \$	Shift 2
20 \$; IS ; IS \$	Reduce iii
10 \$; IS ; IS \$	Shift 4
410 \$	IS ; IS \$	Shift 2
2 410 \$; IS \$	Reduce iii
5 410 \$; IS \$	Reduce ii
10 \$; IS \$	Shift 4
410 \$	IS \$	Shift 2
2 410 \$	\$	Reduce iii
5 410 \$	\$	Reduce ii
10 \$	\$	Shift 3
310 \$	\$	Accept

El árbol de sintáctico de esta gramática tomando en cuenta el reduce como prioridad, asocia para la izquierda como se puede observar:



d) En la Etapa II se concluyó que era indiferente resolver esta ambigüedad hacia la izquierda o hacia la derecha. Compare ahora la eficiencia de ambas alternativas, en términos de la cantidad de pila y del tiempo que se utiliza para reconocer frases de la forma $IS (; IS)^n$ con n un número natural. ¿Cuál alternativa conviene entonces utilizar?

Indicador	Prioridad Shift	Prioridad Reduce
Tiempo	$O(n)$	$O(n)$
Tamaño de la Pila	$O(n)$	$O(1)$

Nuevamente el tamaño de la pila difiere en los ejemplos, cuando se prioriza el reduce, se nota que el tamaño de la pila se mantiene de forma $O(1)$ mientras que el shift recibe más estados en la pila.