

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ciencias

Integrantes:

Adrián Aguilera Moreno
Sebastián Alejandro Gutierrez Medina



Compiladores

Tarea 04

(2pts.) Considera la siguiente gramática donde E es el símbolo inicial:

$$\begin{aligned} E &\rightarrow Aa \\ A &\rightarrow BC \mid BCf \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

1. Construye los conjuntos canónicos de items **LR(0)**.
2. Con estos conjuntos construye el autómata finito mostrando las transiciones con la función GoTo.
3. Muestra la tabla de parsing que se genera usando el autómata anterior.

Solución. Primero obtenemos la FNC

$$\begin{aligned} E &\rightarrow Aa \\ A &\rightarrow BC \mid A'f \\ A' &\rightarrow BC \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

Ahora encontremos los items asociados:

$$\begin{aligned} E' &\rightarrow \bullet E \\ E &\rightarrow \bullet Aa \\ A &\rightarrow \bullet BC \mid \bullet A'f \\ A' &\rightarrow \bullet BC \\ B &\rightarrow \bullet b \\ C &\rightarrow \bullet c \end{aligned}$$

$$I_0 = \text{Closure}(\{E' \rightarrow \bullet E\}) =$$

Luego, para I_1

$$I_1 = \text{GoTo}(\{I_0, E\}) = E' \rightarrow E\bullet$$

así, tenemos que I_2

$$I_2 = \text{GoTo}(\{I_0, A\}) = E \rightarrow A\bullet a$$

observemos que I_3

$$I_3 = \text{GoTo}(\{I_0, A'\}) = A \rightarrow A'\bullet f$$

posteriormente, veamos que I_4

$$I_4 = \text{GoTo}(\{I_0, B\}) = \begin{aligned} A &\rightarrow B\bullet C \\ A' &\rightarrow B\bullet C \end{aligned}$$

y para I_5

$$I_5 = \text{GoTo}(\{I_0, C\}) = \begin{aligned} A &\rightarrow BC\bullet \\ A' &\rightarrow BC\bullet \end{aligned}$$

con I_6

$$I_6 = \text{GoTo}(\{I_0, f\}) = A \rightarrow A'f\bullet$$

y para I_7

$$I_7 = \text{GoTo}(\{I_0, a\}) = E \rightarrow Aa\bullet$$

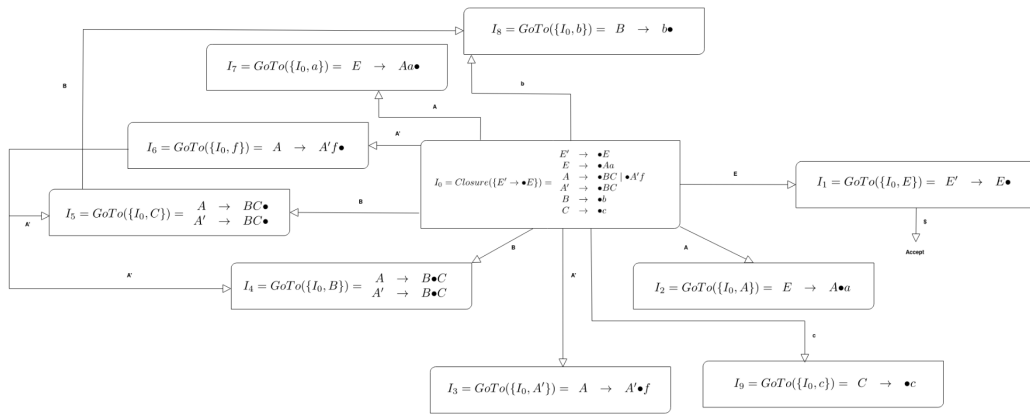
con I_8

$$I_8 = \text{GoTo}(\{I_0, b\}) = B \rightarrow b\bullet$$

Por último, tenemos que I_9

$$I_9 = \text{GoTo}(\{I_0, c\}) = C \rightarrow \bullet c$$

A continuación se presenta el autómata generado por GoTo de R(0):



Para obtener la tabla de parsing supondremos que 0 corresponde al estado de I_0 y así para cada estado. A continuación se muestra la tabla de parsing.

Estado	Acción					GoTo					
	a	b	c	f	\$	E'	E	A	A'	B	C
0		S8	S9				1	2,7	3,6	4,5	
1					Accept						
2	r1	r1	r1								
3	r2										
4			r3								
5											
6				r4					4,5		
7	r5										
8		r6									
9			r7								

Considera la siguiente gramática donde A es el símbolo inicial:

$$\begin{aligned} A &\rightarrow bB \\ B &\rightarrow cC \\ B &\rightarrow cCe \\ C &\rightarrow dA \\ A &\rightarrow a \end{aligned}$$

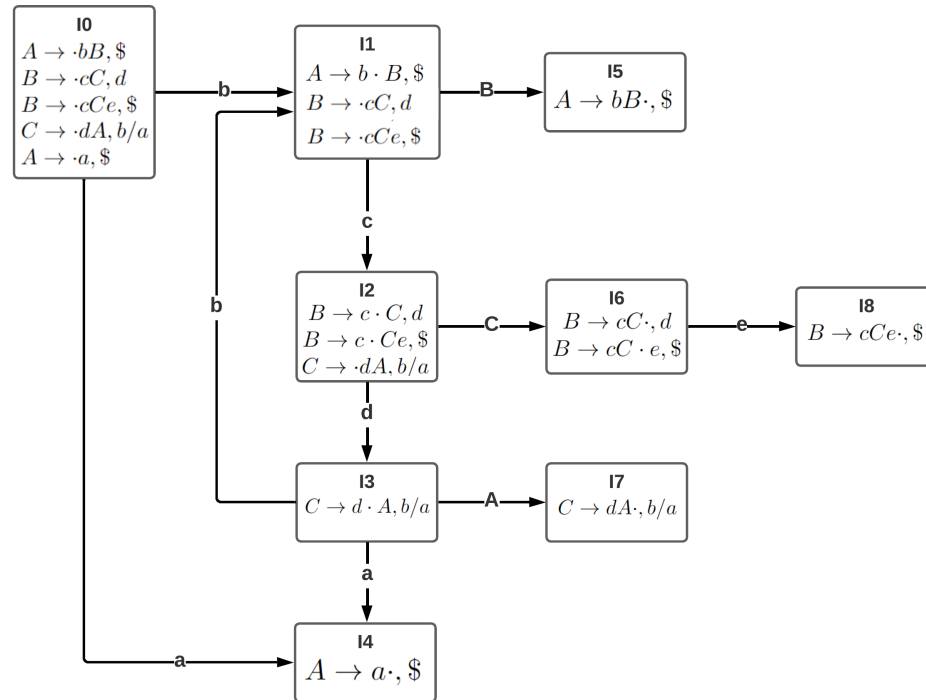
1. (1pt.) Describe formalmente el lenguaje que acepta la gramática.

Sea L el lenguaje que acepta la gramática anterior

$$L = (\{A, B, C\}, \{a, b, c, d, e\}, P, A)$$

$$P = \{A \rightarrow bB, B \rightarrow cC, B \rightarrow cCe, C \rightarrow dA, A \rightarrow a\}$$

2. (3pts.) Proporciona el autómata para construir la tabla de parsing LR(1).



Debido a que hay un conflicto en la transición de $C \rightarrow dA$ no es posible que sea LR

3. (1pt.) De ser posible, analiza una cadena no trivial y de longitud al menos 4, mostrando la secuencia de acciones del autómata mediante una tabla que incluya al menos la actualización de la cadena de entrada y la actualización de la pila.

Cadena : $bcdae$

Table 1: Tabla del progreso del automata

Pila del Parser	Cadena	Regla aplicada
\$A	<i>bcd</i> ae\$	
\$Bb	<i>bcd</i> ae\$	$A \rightarrow bB$
\$B	<i>cd</i> ae\$	
\$eCc	<i>cd</i> ae\$	$B \rightarrow cCe$
\$eC	<i>d</i> ae\$	
\$eAd	<i>d</i> ae\$	$C \rightarrow dA$
\$eA	<i>a</i> e\$	
\$ea	<i>a</i> e\$	$A \rightarrow a$
\$e	<i>e</i> \$	
\$	\$	
\$	\$	ACEPTADA

(3pts.) Muestra que la siguiente gramática pertenece a la clase **LL(1)** pero no a la clase **SLR** (E es el símbolo inicial).

$$E \rightarrow A a A b \mid B b B a \qquad A \rightarrow \varepsilon \qquad B \rightarrow \varepsilon$$

Solución. Nótese que la gramática dada es recursiva por la izquierda, en base a la definición marcada de $LL(1)$. Entonces, la gramática cumple con pertenecer a $LL(1)$. Sin embargo, la gramática no cumple ser $LR(1)$ sencillo, pues la derivación no es por la derecha.

(Hasta 1.5pts. extra) Realiza una tabla comparativa entre los estilos de parsing LL, LR0, SLR y LR1. Incluye características o descripciones breves de las coincidencias o diferencias entre ellos.

No olvides agregar las referencias, libros o páginas web, consultadas.

Table 2: Tabla comparativa de estilos de parsing

Estilo de Parsing	Características	Ventajas	Desventajas
LL	Utiliza una estrategia de análisis descendente, basada en la construcción de una tabla de análisis sintáctico.	Sencillo de entender y de implementar. Útil para gramáticas simples.	No puede manejar gramáticas ambiguas. Limitado en su capacidad de manejar ciertos tipos de gramáticas.
LR0	Utiliza una estrategia de análisis descendente que utiliza un autómata finito.	Puede manejar un amplio rango de gramáticas.	No puede manejar gramáticas ambiguas. Puede generar muchos conflictos de reducción-desplazamiento.
SLR	Basado en LR0, pero utiliza información adicional sobre los símbolos de entrada para resolver conflictos.	Mayor capacidad para manejar gramáticas ambiguas que LR0.	No puede manejar ciertos tipos de gramáticas complejas.
LR1	Basado en SLR, pero utiliza información adicional sobre los símbolos de entrada y de pila para resolver conflictos.	Puede manejar una amplia variedad de gramáticas, incluyendo aquellas con recursión a izquierda y ambigüedad.	Más complejo y difícil de implementar que otros estilos de parsing. Requiere más memoria y tiempo de procesamiento.

Entre las características que tienen en común estos estilos de parsing se encuentran :

- Utilizan una pila para almacenar símbolos de la gramática.
- Utilizan una tabla de análisis para decidir qué acción tomar en función del estado actual de la pila y el símbolo de entrada.
- Tienen como objetivo construir un árbol de análisis sintáctico que represente la estructura sintáctica de una cadena de entrada dada

Referencias

- "Parsing Techniques: A Practical Guide" by Dick Grune and Ceriel J.H. Jacobs.