Memoria de la Práctica de Procesadores de Lenguajes: Analizador Sintáctico

Diego José Abengózar Vilar, Alejandro García Castellanos, Ignacio Javier Encinas Ramos

${\rm Grupo}~82$

November 25, 2019

Índice

1	Diseño del Analizador Sintáctico	2
	1.1 Gramática	. 2
	1.2 Autómata Reconocedor de Prefijos Viables	. 3
	1.2.1 Estados del autómata	. 4
	1.3 Conflictos	. 6
	1.4 Matriz de Transiciones	. 8
2	Tabla de Símbolos: Estructura e implementación	8
3	Anexo de Pruebas	10

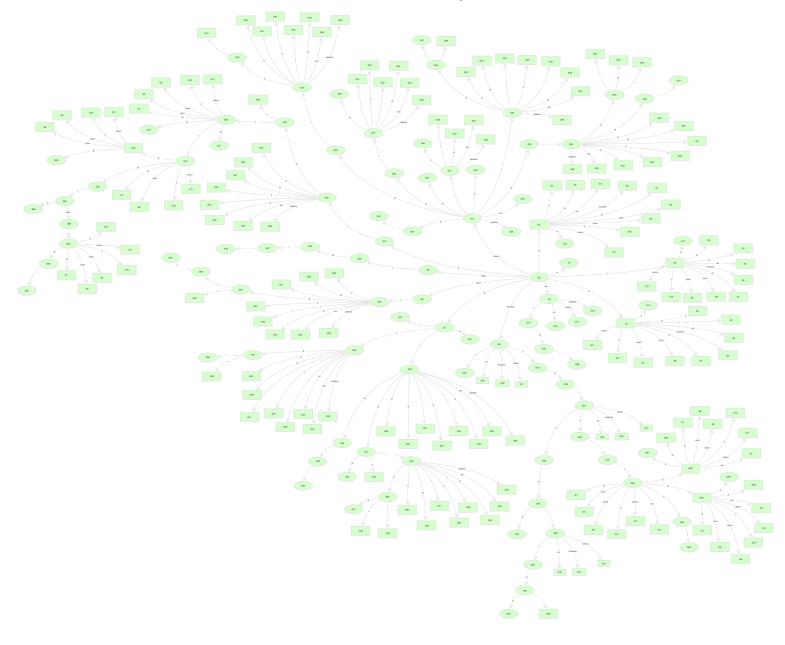
1 Diseño del Analizador Sintáctico

1.1 Gramática

```
Terminales = \{ ; \{ \} id ent cadena ( ) + < ! = | = var int \}
boolean string print input, return function if else }
NoTerminales = \{ PDTFT1AKCSLMQS1GXEURVS2 \}
Axioma = P
Producciones = \{
      P \rightarrow D P
      P \rightarrow F P
      P \rightarrow S P
      D \rightarrow var T id ;
      T \rightarrow int
      T \rightarrow string
      T \rightarrow boolean
      F \rightarrow function T1 id (A) \{C\}
      T1 \rightarrow \lambda
      T1 \rightarrow T
      A \rightarrow T id K
      A \rightarrow \lambda
      K \rightarrow \lambda
      K \rightarrow T id K
      C \rightarrow D C
      C \rightarrow S C
      C \rightarrow \lambda
      S \rightarrow id L E;
      S \rightarrow id \ (M);
      S \rightarrow print (E);
      S \rightarrow input \ (id);
      S \rightarrow if (E) S1
      S \rightarrow return X;
      L \rightarrow |=
      L \rightarrow =
     M \rightarrow E Q
     M \rightarrow \lambda
      Q \rightarrow \lambda
      Q \rightarrow E Q
      S1 \rightarrow \{S2\}G
      S1 \rightarrow S
      G \rightarrow else \{ S2 \}
      G \rightarrow \lambda
      X \rightarrow E
      X \rightarrow \lambda
      E \rightarrow E < U
      E \rightarrow U
      U \rightarrow U + R
      U \rightarrow R
```

```
R 
ightarrow ! V \ R 
ightarrow V \ V 
ightarrow (E) \ V 
ightarrow id \ V 
ightarrow id (M) \ V 
ightarrow ent \ V 
ightarrow cadena \ S2 
ightarrow S \ S2 \ S2 
ightarrow S \ P 
ightarrow \lambda
```

1.2 Autómata Reconocedor de Prefijos Viables



1.2.1 Estados del autómata

```
S_0=\{P1 \rightarrow \bullet P, P \rightarrow \bullet DP, P \rightarrow \bullet SP, P \rightarrow \bullet, D \rightarrow \bullet var T id;,
         F \rightarrow \bullet \text{ function } T1 \text{ id}(A)\{C\}, S \rightarrow \bullet \text{ id } L E;, S \rightarrow \bullet \text{ id}(M);
         S \rightarrow \bullet \text{ print}(E); , S \rightarrow \bullet \text{ input}(id); , S \rightarrow \bullet \text{ if}(E) S1,
         S \rightarrow \bullet return X;
S_1 = \{P1 \rightarrow P \bullet \}
S_2 = \{P \rightarrow D \bullet P, P \rightarrow \bullet DP, P \rightarrow \bullet SP, P \rightarrow \bullet , D \rightarrow \bullet var T id;,
         F \rightarrow \bullet function T1 id(A){C}, S \rightarrow \bullet id L E;, S \rightarrow \bullet id(M);
         S \rightarrow \bullet print(E); S \rightarrow \bullet input(id); S \rightarrow \bullet if(E) S1,
         S \rightarrow \bullet return X;
S_3=\{P \rightarrow F \bullet P, P \rightarrow \bullet DP, P \rightarrow \bullet FP, P \rightarrow \bullet SP, P \rightarrow \bullet,
         D \rightarrow \bullet \text{ var } T \text{ id}; F \rightarrow \bullet \text{ function } T1 \text{ id}(A) \{C\},
         S \rightarrow \bullet \text{ id } L E;, S \rightarrow \bullet \text{ id}(M);, S \rightarrow \bullet \text{ print}(E);
         S \rightarrow \bullet \text{ input (id)}; S \rightarrow \bullet \text{ if (E)} S1, S \rightarrow \bullet \text{ return } X;
S_4=\{P \rightarrow S \bullet P, P \rightarrow \bullet DP, P \rightarrow \bullet FP, P \rightarrow \bullet SP, P \rightarrow \bullet,
         D \rightarrow \bullet \text{ var } T \text{ id};, F \rightarrow \bullet \text{ function } T1 \text{ id}(A)\{C\},
         S \rightarrow \bullet id L E;, S \rightarrow \bullet id (M);, S \rightarrow \bullet print (E);,
         S \ \rightarrow \ \bullet \ input(id); \, , \ S \ \rightarrow \ \bullet \ if(E) \ S1 \, , \ S \ \rightarrow \ \bullet \ return \ X; \}
S_5 = \{D \rightarrow var \bullet T id;, T \rightarrow \bullet int, T \rightarrow \bullet string, T \rightarrow \bullet boolean\}
S_6=\{F \rightarrow function \bullet T1 \ id(A)\{C\}, \ T1 \rightarrow \bullet, \ T1 \rightarrow \bullet T,
        T \rightarrow \bullet \text{ int}, T \rightarrow \bullet \text{ string}, T \rightarrow \bullet \text{ boolean}
S_7 = \{S \rightarrow id \bullet L E; , S \rightarrow id \bullet (M); , L \rightarrow |=, L \rightarrow \bullet =\}
S_8 = \{S \rightarrow print \bullet (E); \}
S_9 = \{S \rightarrow input \bullet (id); \}
S_{10} = \{S \rightarrow if \bullet (E) S1\}
S_{11}\!\!=\!\!\{S \ \rightarrow\! \texttt{return} \ \bullet X; \,, \ X \ \rightarrow \ \bullet, \ X \ \rightarrow \ \bullet \ E, \ E \ \rightarrow \ \bullet \ E < U, \ E \ \rightarrow \ \bullet \ U,
         U \rightarrow \bullet U + R, U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E),
         V \rightarrow \bullet \text{ id}, V \rightarrow \bullet \text{ id}(M), V \rightarrow \bullet \text{ ent}, V \rightarrow \bullet \text{ cadena}
S_{12} = \{P \rightarrow DP \bullet \}
S_{13} = \{P \rightarrow FP \bullet\}
S_{14} = \{P \rightarrow SP \bullet \}
S_{15}=\{D \rightarrow var \ T \bullet id;\}
S_{16} = \{T \rightarrow int \bullet \}
S_{17} = \{T \rightarrow string \bullet \}
S_{18} = \{T \rightarrow boolean \bullet \}
S_{19} = \{F \rightarrow function \ T1 \bullet id(A)\{C\}\}\
S_{20} {=} \{T1 \ \rightarrow T \ \bullet\}
S_{21} = \{L \rightarrow = \bullet\}
S_{22} = \{S \rightarrow id \ (\bullet M); , M \rightarrow \bullet E Q, M \rightarrow \bullet, E \rightarrow \bullet E < U, E \rightarrow \bullet U,
         U \rightarrow \bullet U + R, U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E),
         V \rightarrow \bullet \text{ id}, V \rightarrow \bullet \text{ id}(M), V \rightarrow \bullet \text{ ent}, V \rightarrow \bullet \text{ cadena}
S_{23} = \{L \rightarrow |= \bullet\}
S_{24} = \{S \rightarrow print \ (\bullet E); , E \rightarrow \bullet E < U, E \rightarrow \bullet U, U \rightarrow \bullet U + R,
         U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E), V \rightarrow \bullet id,
         V \rightarrow \bullet id(M), V \rightarrow \bullet ent, V \rightarrow \bullet cadena
S_{25} = \{S \rightarrow input \ ( \bullet id ) \}
S_{26} = \{S \rightarrow if \ (\bullet E) \ S1, \ E \rightarrow \bullet E < U, \ E \rightarrow \bullet U, \ U \rightarrow \bullet U + R,
         U \ \rightarrow \ \bullet \ R, \ R \ \rightarrow \ \bullet \ ! \ V, \ R \ \rightarrow \ \bullet \ V, \ V \ \rightarrow \ \bullet \ (E) \,, \ V \ \rightarrow \ \bullet \ id \,,
         V \rightarrow \bullet id(M), V \rightarrow \bullet ent, V \rightarrow \bullet cadena)
S_{27} = \{S \rightarrow return \ X \bullet ; \}
S_{28} = \{X \rightarrow E \bullet, E \rightarrow E \bullet < U\}
S_{29} = \{E \rightarrow U \bullet, U \rightarrow U \bullet + R\}
S_{30} = \{U \rightarrow R \bullet \}
S_{31} = \{R \rightarrow ! \bullet V, V \rightarrow \bullet (E), V \rightarrow \bullet id, V \rightarrow \bullet id(M),
        V \rightarrow \bullet \text{ ent}, V \rightarrow \bullet \text{ cadena}
S_{32} = \{R \rightarrow V \bullet \}
S_{33}=\{V \rightarrow (\bullet E), E \rightarrow \bullet E < U, E \rightarrow \bullet U, U \rightarrow \bullet U + R,
```

```
U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E), V \rightarrow \bullet id,
         V \rightarrow \bullet id(M), V \rightarrow \bullet ent, V \rightarrow \bullet cadena
S_{34} = \{V \rightarrow id \bullet, V \rightarrow id \bullet (M)\}
S_{35} = \{V \rightarrow ent \bullet \}
S_{36} = \{V \rightarrow cadena \bullet \}
S_{37}=\{D \rightarrow var \ T \ id \bullet;\}
S_{38} = \{F \rightarrow function T1 id \bullet (A)\{C\}\}\
S_{39}=\{S \rightarrow id L \bullet E, E \rightarrow \bullet E < U, E \rightarrow \bullet U, U \rightarrow \bullet U + R,
         U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E), V \rightarrow \bullet id,
         V \rightarrow \bullet id(M), V \rightarrow \bullet ent, V \rightarrow \bullet cadena
S_{40} = \{S \rightarrow id (M \bullet); \}
S_{41}=\{M \rightarrow E \bullet Q, E \rightarrow E \bullet < U, Q \rightarrow \bullet, Q \rightarrow \bullet, EQ\}
S_{42} = \{S \rightarrow print(E \bullet);, E \rightarrow E \bullet < U\}
S_{43} = \{S \rightarrow input(id \bullet);\}
S_{44} = \{S \rightarrow if(E \bullet) S1, E \rightarrow E \bullet < U\}
S_{45} = \{S \rightarrow return X; \bullet \}
S_{46}=\{E \rightarrow E < \bullet U, U \rightarrow \bullet R, U \rightarrow \bullet U + R, R \rightarrow \bullet ! V, R \rightarrow \bullet V\}
         V \rightarrow \bullet (E), V \rightarrow \bullet id, V \rightarrow \bullet id (M), V \rightarrow \bullet ent, V \rightarrow \bullet cadena}
S_{47}=\{U \rightarrow U + \bullet R, R \rightarrow \bullet ! V, V \rightarrow \bullet (E), V \rightarrow \bullet id(M),
         V \rightarrow \bullet \text{ ent}, V \rightarrow \bullet \text{ cadena}
S_{48} = \{R \rightarrow ! V \bullet \}
S_{49} = \{V \rightarrow (E \bullet), E \rightarrow E \bullet < U\}
S_{50} = \{ V \rightarrow id \ ( \bullet \ M ) \ , \ M \rightarrow \bullet \ E \ Q, \ M \rightarrow \bullet \ , \ E \rightarrow \bullet \ E < U, \ E \rightarrow \bullet \ U,
         U \rightarrow \bullet U + R, U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E),
         V \rightarrow \bullet \text{ id}, V \rightarrow \bullet \text{ id}(M), V \rightarrow \bullet \text{ ent}, V \rightarrow \bullet \text{ cadena}
S_{51}=\{F \rightarrow function \ T1 \ id (\bullet \ A)\{C\}, \ A \rightarrow \bullet \ T \ id \ K, \ A \rightarrow \bullet, \ T \rightarrow \bullet \ int,
         T \rightarrow \bullet \text{ string}, T \rightarrow \bullet \text{ boolean}
S_{52} = \{S \rightarrow id \ L \ E \bullet ; , E \rightarrow E \bullet < U\}
S_{53} = \{S \rightarrow id(M) \bullet ; \}
S_{54}=\{M \rightarrow E Q \bullet \}
S_{55}=\{Q \rightarrow , \bullet E Q, E \rightarrow \bullet E < U, E \rightarrow \bullet U,
         U \rightarrow \bullet U + R, U \rightarrow \bullet R, R \rightarrow \bullet ! V, R \rightarrow \bullet V, V \rightarrow \bullet (E),
         V \rightarrow \bullet \text{ id}, V \rightarrow \bullet \text{ id}(M), V \rightarrow \bullet \text{ ent}, V \rightarrow \bullet \text{ cadena}
S_{56} = \{S \rightarrow print(E) \bullet ;\}
S_{57} = \{S \rightarrow input(id) \bullet ;\}
S_{58} \hspace{-0.05cm} = \hspace{-0.05cm} \{S \rightarrow i\, f\, (E) \mid \bullet \ S1 \,, \ S1 \rightarrow \bullet \ \{S2\}G, \ S1 \rightarrow \bullet \ S, \ S \rightarrow \bullet \ i\, d \ L \ E; \,,
         S \rightarrow \bullet id(M);, S \rightarrow \bullet print(E);, S \rightarrow \bullet input(id);,
         S \ \rightarrow \ \bullet \ if (E) S1 \, , \ S \ \rightarrow \ \bullet \ return \ X \ ; \}
S_{59} = \{E \rightarrow E < U \bullet, U \rightarrow U \bullet + R\}
S_{60} = \{U \rightarrow U + R \bullet \}
S_{61} {=} \{ V \rightarrow (E) \quad \bullet \, \}
S_{62} = \{V \rightarrow id (M \bullet)\}
S_{63}=\{M \rightarrow E \bullet Q, E \rightarrow E \bullet < U, Q \rightarrow \bullet, Q \rightarrow \bullet, EQ\}
S_{64} = \{F \rightarrow function T1 id(A \bullet)\{C\}\}
S_{65}=\{A \rightarrow T \bullet id K\}
S_{66} = \{S \rightarrow id \ L \ E \ ; \bullet \}
S_{67} = \{S \rightarrow id (M); \bullet \}
S_{68}=\{Q \rightarrow E \bullet Q, E \rightarrow E \bullet C, Q \rightarrow \bullet, Q \rightarrow \bullet, EQ\}
S_{69} = \{S \rightarrow print(E); \bullet \}
S_{70} = \{S \rightarrow input(id); \bullet\}
S_{71}=\{S \rightarrow if(E) S1 \bullet \}
S_{72} = \{S1 \rightarrow \{\bullet \ S2\}G, \ S2 \rightarrow \bullet \ S \ S2, \ S2 \rightarrow \bullet \ S, \ S \rightarrow \bullet \ id \ L \ E;, \}
         S \ \rightarrow \ \bullet \ id \, (M) \, ; \, , \ S \ \rightarrow \ \bullet \ print \, (E) \, ; \, , \ S \ \rightarrow \ \bullet \ input \, (id \,) \, ; \, ,
         S \rightarrow \bullet \text{ if } (E)S1, S \rightarrow \bullet \text{ return } X ; 
S_{73} = \{S1 \rightarrow S \bullet \}
S_{74}=\{S2 \rightarrow S \bullet S2 , S2 \rightarrow S \bullet , S2 \rightarrow \bullet S S2, S2 \rightarrow \bullet S,
         \dot{S} \rightarrow ullet id \ L \ E \ ; , \ S \rightarrow ullet id \ (\ M \ ) \ ; , \ S \rightarrow ullet print \ (\ E \ ) \ ; ,
         S \rightarrow \bullet \text{ input (id)};, S \rightarrow \bullet \text{ if (E)} S1, S \rightarrow \bullet \text{ return } X;
```

```
S_{75} = \{F \rightarrow function T1 id (K) \bullet \{C\}\}\
S_{76}=\{A \rightarrow T \text{ id } \bullet K , K \rightarrow \bullet, K \rightarrow \bullet , T \text{ id } K\}
S_{77}=\{Q \rightarrow , E Q \bullet \}
S_{78} = \{S1 \rightarrow \{S2 \bullet \} G\}
S_{79} = \{S2 \rightarrow S \bullet S2 , S2 \rightarrow S \bullet , S2 \rightarrow \bullet S S2 , S \rightarrow \bullet \text{ id } L E; ,
          S \rightarrow \bullet id(M);, S \rightarrow \bullet print(E); S \rightarrow \bullet if(E)S1;
          S \ \rightarrow \ \bullet \ input(id); , \ S \ \rightarrow \ return \ X; \}
S_{80} = \{F \rightarrow function T1 id (K) \{ \bullet C \}, C \rightarrow \bullet D C, C \rightarrow \bullet \}
         D \rightarrow \bullet \text{ var } T \text{ id } ;, S \rightarrow \bullet \text{ id } L E , S \rightarrow \bullet \text{ id } (M);,
          S \rightarrow \bullet \text{ print } (E);, S \rightarrow \bullet \text{ input } (id);, S \rightarrow \bullet \text{ if } (E) S1,
         S \rightarrow \bullet return X ; 
S_{81} = \{A \rightarrow T \text{ id } K \bullet \}
S_{82} = \{K \rightarrow, \bullet T \text{ id } K, T \rightarrow \bullet \text{ int }, T \rightarrow \bullet \text{ string }, T \rightarrow \bullet \text{ boolean}\}
S_{83} = \{S1 \rightarrow \{S2\} \bullet G , G \rightarrow \bullet \text{ else } \{S2\} , G \rightarrow \bullet \}
S_{84} = \{F \rightarrow function T1 id (K) \{C \bullet \}\}
S_{85}=\{C \rightarrow D \bullet C, C \rightarrow \bullet D C, C \rightarrow \bullet S C, C \rightarrow \bullet, D \rightarrow \bullet \text{ var T id } ;,
          S \rightarrow \bullet \text{ id } L E ;, S \rightarrow \bullet \text{ id } (M) ;, S \rightarrow \bullet \text{ print } (E) ;,
          S \rightarrow • input ( id ) ;, S \rightarrow • if ( E ) S1, S \rightarrow • return X ;}
S_{86}=\{C \rightarrow S \bullet C, C \rightarrow \bullet D C, C \rightarrow \bullet S C, C \rightarrow \bullet, D \rightarrow \bullet \text{ var T id } ;,
         S \rightarrow • id L E ;, S \rightarrow • id ( M ) ;, S \rightarrow • print ( E ) ;,
          S \rightarrow \bullet input ( id ) ;, S \rightarrow \bullet if ( E ) S1, S \rightarrow \bullet return X ;}
S_{87} = \{K \rightarrow , T \bullet id K\}
S_{88} = \{S1 \rightarrow \{S2\} G \bullet \}
S_{89} = \{G \rightarrow else \bullet \{S2\}\}\
S_{90} = \{F \rightarrow function T1 id (K) \{C\} \bullet \}
\begin{array}{l} S_{91} = \{K \rightarrow, \ T \ \text{id} \ \bullet K, \ K \rightarrow \bullet \ , \ K \rightarrow \bullet \ , \ T \ \text{id} \ K\} \\ S_{92} = \{G \rightarrow \text{else} \ \{ \ \bullet \ S2 \}, \ S2 \rightarrow \bullet \ S \ S2 \ , \ S2 \rightarrow \bullet \ \text{id} \ L \ E \ ; \ , \\ S \rightarrow \bullet \ \text{id} \ (M \ ) \ ; \ , \ S \rightarrow \bullet \ \text{print} \ (E \ ) \ ; \ , \ S \rightarrow \bullet \ \text{input}(\text{id}); \ , \end{array}
         S \rightarrow • if ( E ) S1, S \rightarrow • return X ;}
S_{93} = \{K \rightarrow , T \text{ id } K \bullet \}
S_{94} = \{G \rightarrow else \{ S2 \bullet \} \}
S_{95} = \{G \rightarrow else \{ S2 \} \bullet \}
S_{96} = \{C \rightarrow D C \bullet \}
S_{97} = \{C \rightarrow S \ C \bullet \}
S_{98} = \{D \rightarrow var \ T \ id ; \bullet \}
S_{99} = \{S2 \rightarrow S \ S2 \bullet \}
```

1.3 Conflictos

Como podemos observar en la tabla de decisión no hay ningún conflicto. Los posibles conflictos son:

Reducción-Reducción

Podemos ver como en los posibles estados con este conflicto, S...., se verifica que $\forall \{A \to \alpha \bullet, B \to \beta \bullet\} \subset S_x \Rightarrow \text{Follow}(A) \cap \text{Follow}(B) = \emptyset$ (Esto lo podemos observar al no tener dos entradas de reducción en la misma celda de cada fila de S_x)

Reducción-Desplazamiento

Podemos ver como en los posibles estados con este conflicto, S...., se verifica que $\forall \{A \to \alpha \bullet b \gamma, C \to \beta \bullet\} \subset S_x \Rightarrow b \notin Follow(C)$ (Esto lo podemos observar al no tener una entrada de desplazamiento y otra de reducción en la misma celda de cada fila de S_x)

	{ } id enterc	entero cadena ()	= 	= var	int boolean string print input	. return function if else	\$ P D T F	T' A K C S L M Q S G X E U R V S"
os	d7		-	- GP		d11 d6 d10	r49 1 2 3	4
S1							в	
S2	47			d5		- qe	: =	4
S3	d7		_	d5	6p 8p	d11 d6 d10	r49 13 2 3	4
S4	d7	_ _ _		d5	6p 8p	d11 d6 d10	r49 14 2 3	4
S5	_	_ _ _	_	d16 d18	8 d17		15	
Se	61	_ _ _	_	d16 d18	8 d17		20 19	
S7	_	d22	d21	d23	_ _ _ _	_		
88	_ _ _	d24	_ _ _	_ _ _	_ _ _ _	 		
6S	_	d25	_					
S10	_	d26	_					
S11 r35	d34 d35	d36 d33		_ _ _				
S12	_	_	_	_ _ _	_	_	11	
S13	_	_	_	_	_	_	r2	
814				_			r3	
S15	d37							
S16	r5							
S17	lu6	_	_	_	_	_		
818	Tr	_	_	_				
819	829	- - -				_		
S20	r10	- - -		_		_		
S21	r25 r25	r25 r25	r25	_				
S22	l_	989						
S23	l	r24	r24			_		
S24	d34 d35	d36		_ _ _	_	_		
S25	d43	_ _ _			_ _ _			
826	d34 d35	d36 d33	d31	_ _ _	_ _ _	_ _ _		
S27 d45	_ _ _	_ _ _ _	_ _ _	_ _ _	_ _ _ _	_ _ _ _		
S28 r34	_	_ _ _	d46	_	_ _ _ _	_	_ _ _ _	
S29 r37	_ _ _	137	r37 d47 r37	_ _ _		r37		
S30 r39	_	139	r39 r39 r39	_		r39	 	
831	d34 d35	d36 d33	_ _ _	_	_ _ _ _	_	_ _ _ _ _	
S32 r41	_	_	r41 r41 r41	_		r41	_ _ _ _ _	
S33	d34 d35	d36					_ _ _ _	
S34 r43	_	d50 r43	d50 r43 r43 r43	_		r43	_ _ _ _ _	
S35 r45	_	r45	r45 r45 r45	_		r45	_ _ _ _ _	
S36 r46		146	r46 r46 r46		_	r46	_ _ _ _ _ _	
S37 d98	_ _ _	- - -	_ _ _		_ _ _	_ _ _	_ _ _ _ _ _	
838		_	_ _ _			_ _ _	_ _ _ _ _	_
839	d34 d35	d36 d33	d31	_	_ _ _ _	_ _ _ _	_ _ _ _ _	
S40	_ _ _	d53	_ _ _	_ _ _	_ _ _ _	_ _ _		
S41		r28			p	d55	_ _ _ _	
S42	_		d46	_	_	_		
S43		d57						
S44		82P 128	446					
S45			_	r23	r23 r23	r23 r23 r23	r23	
S46		9EP				_ _ _ _	_ _ _ _ _	
847	d34 d35	d36 d33		_		_ _ _ _		
S48 r40		1.40	r40			r40		
S49		d61						

1.4 Matriz de Transiciones

MT_AFD		letra	digito	,	/	_	carácter	*	delimitador
$\rightarrow 0$	1 lee	2 C	3A	4 lee	5 lee	-1 error	-1 error	-1 error	0 lee
1	-1 error								
2	9 G2	2 C	2 C	9 G2	9 G2	2 C	9 G2	9 G2	9 G2
3	10 G3	10 G3	3 B	10 G3					
4	4 C	4 C	4 C	11 G4	4 C	4 C	4 C	4 C	4 C
5	-1 error	6 lee	-1 error						
6	6 lee	7 lee	6 lee						
7	6 lee	6 lee	6 lee	6 lee	0 lee	6 lee	6 lee	7 lee	6 lee

MT_AFD	;	{	}	()	+	<	!	=	,
$\rightarrow 0$	12 G5	13 G6	14 G7	15 G8	16 G9	17 G10	18 G11	19 G12	20 G13	21 G14
1	-1 error	8 G1	-1 error							
2	9 G2									
3	10 G3									
4	4 C	4 C	4 C	4 C	4 C	4 C	4 C	4 C	4 C	4 C
5	-1 error									
6	6 lee									
7	6 lee									

2 Tabla de Símbolos: Estructura e implementación

Contiene la información de los identificadores, de los cuales se guardan los campos: lexema, tipo y desplazamiento. Para las funciones, además, se guardará el número de parámetros, su tipo, la forma de paso de parámetros y el tipo del valor de retorno.

La tabla de símbolos estará formada por dos matrices de tamaño dinámico; la primera contendran los indentificadores de ámbito global y la segunda del local. Así pues, esta segunda se creará al encontrar la declaración de una función y se borrará al acabar de ser declarada. También se utiliza un flag de declaración o uso (FlagDeclUso), un flag para saber cual es la tabla actual y dos más para el valor del desplazamiento en cada una de las tablas.

Sin embargo, en la implementación actual sólo se usa una tabla y siempre se supone que está el FlagDeclUso = Uso, pero en el caso de que no este declarada la variable se insertará en la tabla actual, ya que requerimos de la implementacion del Analizador Semántico para poder saber cuando se cambia de ámbito y cuando se están declarando o usando identificadores. Así que, la acción semántica que genera los tokens de los identificadores quedaría temporalmente así:

```
G_2: if (lex \in palRes)

GEN_TOKEN(palRes, -)

else if ((p:= BUSCA_TS(lex))=NULL)

p:=INSERTAR_TS(lex)

GEN_TOKEN(ID, p)
```

: { } id entero cadena () + i ! = = var int	boolean string	print input . return	rn function	if else \$	PIDIT	F T' A	K C S L	M O S' G	X E U B	N S.
d35 d36 d33 r27 d31								62	63 29	32
	d18 d17	_	_	_	65	5 64	- 	_		_
S52 d66	_	_	_	_	_	_	_	_	_	_
S53 d67	_	_	_		_		_		_	
	_	_ _ _	_	_	 	_ _	 	_ _ _ _		
S55	_	_	_	_	_	_ _ _	_ _ _	_ _ _ _	68 29 30	32
	_	_	_	_	_ _ _	_ _ _	_ _ _	_ _ _ _	_ _ _	
S57 d70	_	_	_	_	_	_ _ _	_ _ _	_ _ _	_ _ _	_
S58	p	d8 d9 d11		d10			73	71		_
S59 r36		r36								_
S60 r38 r38		r38	_		_ _ _	_ _ _				_
r42	_	r42	_	_	_ _ _	_ _ _	- - -	_ _ _	_ _ _	- -
S62	_	_		_	- -	- -	- - -		- -	- -
S63	_	d55		_	_	_	_ _ _	54	_ _ _	_
	_	_		_	_		_	_	_	_
94P				_	_				_	
S66 r18 r18 r18	r1	r18 r18 r18	s r18	r18 r18	_ _ _		- - -	_ _ _	- - -	- -
S67	rl	r19 r19 r19	el1 1	r19 r19	_	_	_ _ _	_ _ _	_	_
S68	_	d55	_	_	- - -	- - -	- - -	77	- - -	_ _
S69 r20 r20	r2	r20 r20 r20	_	r20 r20	_		_	_	_	_
S70 r21 r21 r21	r2	r21	r21	r21 r21	_		_		_	
	r2		-		_		_			
		6P	-		_		62			182
r31		r31	r31	r31 r31	-				- -	
r44	-	r44	-	_			-		-	
S75 d80	_	_		_	_		_		_	
S76		d82		_	_		81	_		
S77	- -	_	_	_ _	- - -	- - -	- - -	_ _ 	- - -	- -
S78	_	_ _ _	_	_	_ _ _	_ _ _	_ _ _	_ _ _	_	
S79	8p	8 d9 d11	_	d10	_		62	_ _ _	_	66
S80 r17 d7	P	d8 d9 d11		d10	85	_ _ _	84 86		_	_
S81	_	_	_	_			_ _ _	_ _ _ _	_	_
S82	d18 d17	_	_	_ _	87		_ _ _	_ _ _ _	_ _ _	
_		r33 r33 r33	r33	r33 d89 r33			_ _ _	- 88	_	
S84 d90	_	_	_	_	_	_ _ _	_ _ _	_ _ _ _	_	_
117	p	dp	_	d10	_	_ _ _	98 96	_ _ _	_	_
117	P —	d8 d9 d11	_	d10	82	_ _ _	98 26	_ _ _ _	_ _ _	_
	_ _ _	_ _	-		_ _ _	_ _ _	_ _ _	_ _ _ _	_ _ _	_
S88 r30 r30		r30 r30 r30	r30	r30 r30	_	_ _ _	_ _ _	_ _ _	_ _ _	_
S89 d92	_	_ _ _	_	_	_	_ _ _	_ _ _	_ _ _ _	_ _ _	_
S90		r8 r8 r8	r8	r8 r8	_	_ _ _	_ _ _	_ _ _	_	
S91	_	-	_	_	_ _		93	_ _ _ _	_	
d7	P —	d8 d9 d11	_	d10	_ _ _	_ _ _	62	_ _ _ _	_ _ _	94
_	_	_	_	_	_ _ _	_ _ _	_ _ _		_ _ _	_
d95	_	_	_				_ _ _	_ _ _	_	
_	r3	r32 r32 r32	r32	r32 r32	_ _		_ _ _	_ _ _ _	_	
S96	_	_	_	_	_ _ _	_ _ _	_ _ _	_ _ _ _	_ _ _	_
S97	_	_	_	_	_ _ _	_ _ _	_ _ _		_ _ _	_
_		r4 r4 r4	r4	r4 r4			_ .		_ - -	
S99 147	_	_		_			_ _ _	_		

3 Anexo de Pruebas

```
Error 1: Número fuera de rango.
```

```
Fuente:
   var int a;
   var int b;
   a = 33333;
   b = a;
   if (a < b) b = 1;
   if (b < a) b = 8;
   a = a + b;
   print (a);
   print (b);
Tokens:
   <DEC, >
   <TipoVarENT, >
   < ID, 0 >
   <PuntoComa, >
   <DEC, >
   <TipoVarENT, >
   < ID, 1 >
   <PuntoComa, >
   < ID, 0 >
   \langle ASIG, \rangle
Tabla Simbolos#1:
* LEXEMA: 'a'
* LEXEMA: 'b'
Error 2: Transición no prevista
Fuente:
   var string texto; /*Comentario bueno*/
   function imprime (string msg)
   print (msg);
   / Comentario malo*/
   function pideTexto ()
   print ( 'Introduce un texto' );
   input (texto);
   pideTexto();
   var string textoAux;
   textoAux = texto;
   imprime (textoAux);
```

Tokens:

<DEC, >

```
<TipoVarCAD, >
   \langle ID, 0 \rangle
   <PuntoComa, >
   <DECFunc, >
   <ID, 1>
   <ParentesisAbrir, >
   <TipoVarCAD, >
   <ID, 2>
   < Parentesis Cerrar, >
   <CorcheteAbrir, >
   <Print, >
   <ParentesisAbrir, >
   <ID, 2>
   < Parentesis Cerrar, >
   <PuntoComa, >
   <CorcheteCerrar, >
Tabla Simbolos #1:
* LEXEMA: 'texto'
* LEXEMA: 'imprime'
* LEXEMA: 'msg'
Error 3: Transición no prevista
Fuente:
   var string texto;
   function pideTexto ()
   { print ('Introduce un texto);
   input (texto);
   } function imprime (string msg)
   print ('Mensage introducido:');
   print (msg);
   pideTexto();
   var string textoAux;
   textoAux = texto;
   imprime (textoAux);
Tokens:
   <DEC, >
   <TipoVarCAD, >
   \langle ID, 0 \rangle
   <PuntoComa, >
   <DECFunc, >
   <ID, 1>
   <ParentesisAbrir, >
   <ParentesisCerrar, >
   <CorcheteAbrir, >
   <Print, >
```

```
<ParentesisAbrir, >
   <CAD, "Introduce un texto);
   input (texto);
   function imprime (string msg)
   print (" >
   <ID, 2>
   <ID, 3>
Tabla Simbolos #1:
* LEXEMA: 'texto'
* LEXEMA: 'pideTexto'
* LEXEMA: 'Mensage'
* LEXEMA: 'introducido'
Pruebas pasadas con éxito:
1:
Fuente:
   var int a;
   var int b;
   a = 3;
   b = a;
   var boolean c;
   c = a < b;
   if (c) b = 1;
   c = b < a;
   if (c) b = 4;
   a = a + b;
   print (a);
   print (b);
Tokens:
   <DEC, >
   <TipoVarENT, >
   <ID, 0>
   <PuntoComa, >
   <DEC, >
   <TipoVarENT, >
   <ID, 1>
   <PuntoComa, >
   < ID, 0 >
   \langle ASIG, \rangle
   <ENT, 3>
   <PuntoComa, >
   <ID, 1>
   \langle ASIG, \rangle
   < ID, 0 >
   <PuntoComa, >
```

<DEC, ><TipoVarLOG, > $\langle ID, 2 \rangle$ <PuntoComa, > < ID, 2 > $\langle ASIG, \rangle$ $\langle ID, 0 \rangle$ <MENOR, >< ID, 1 ><PuntoComa, > $\langle IF, \rangle$ <ParentesisAbrir, > <ID, 2><ParentesisCerrar, > < ID, 1 > $\langle ASIG, \rangle$ <ENT, 1> <PuntoComa, > <ID, 2> $\langle ASIG, \rangle$ < ID, 1 ><MENOR, >< ID, 0 ><PuntoComa, > $\langle IF, \rangle$ <ParentesisAbrir, > < ID, 2 ><ParentesisCerrar, > $\langle ID, 1 \rangle$ $\langle ASIG, \rangle$ $\langle ENT, 4 \rangle$ <PuntoComa, > < ID, 0 > $\langle ASIG, \rangle$ < ID, 0 > $\langle SUMA, \rangle$ <ID, 1> <PuntoComa, > <Print, ><ParentesisAbrir, > < ID, 0 ><ParentesisCerrar, >

<PuntoComa, >

<ParentesisAbrir, >

<ParentesisCerrar, > <PuntoComa, >

<Print, >

<ID, 1>

13

```
Tabla Simbolos #1:
* LEXEMA: 'a'
* LEXEMA: 'b'
* LEXEMA: 'c'
2:
Fuente:
   var int a;
   var int b;
   var int c;
   print ('Introduce el primer operando');
   input (a);
   print ('Introduce el segundo operando');
   input (b);
   function int suma (int num1, int num2)
   return num1+num2;
   c = suma (a, b);
   print (c);
Tokens:
   <DEC, >
   <TipoVarENT, >
   \langle ID, 0 \rangle
   <PuntoComa, >
   <DEC, >
   <TipoVarENT, >
   < ID, 1 >
   <PuntoComa, >
   <DEC, >
   <TipoVarENT, >
   < ID, 2 >
   <PuntoComa, >
   <Print, >
   <ParentesisAbrir, >
   <CAD, "Introduce el primer operando">
   < Parentesis Cerrar, >
   <PuntoComa, >
   <Input, >
   <ParentesisAbrir, >
   <ID, 0>
   <ParentesisCerrar, >
   <PuntoComa, >
   <Print, >
   <ParentesisAbrir, >
   <CAD, "Introduce el segundo operando">
   < Parentesis Cerrar, >
```

```
<PuntoComa, >
   <Input, >
   <ParentesisAbrir, >
   \langle ID, 1 \rangle
   < Parentesis Cerrar, >
   <PuntoComa, >
   <DECFunc, >
   <TipoVarENT, >
   <ID, 3>
   <ParentesisAbrir, >
   <TipoVarENT, >
   \langle ID, 4 \rangle
   <Coma, >
   <TipoVarENT, >
   < ID, 5 >
   <ParentesisCerrar, >
   <CorcheteAbrir, >
   <Return, >
   \langle ID, 4 \rangle
   \langle SUMA, \rangle
   < ID, 5 >
   <PuntoComa, >
   <CorcheteCerrar, >
   < ID, 2 >
   \langle ASIG, \rangle
   \langle ID, 3 \rangle
   <ParentesisAbrir, >
   <ID, 0>
   <Coma, >
   <ID, 1>
   <ParentesisCerrar, >
   <PuntoComa, >
   <Print, >
   <ParentesisAbrir, >
   < ID, 2 >
   <ParentesisCerrar, >
   <PuntoComa, >
Tabla Simbolos #1:
* LEXEMA: 'a'
* LEXEMA: 'b'
* LEXEMA: 'c'
* LEXEMA: 'suma'
* LEXEMA: 'num1'
* LEXEMA: 'num2'
```

3:

Fuente:

```
var int a;
   var int b;
   a = 3;
   b = a;
   if (a < b) b = 1;
   a = a + b;
   print (a);
   print (b);
Tokens:
   <DEC, >
   <TipoVarENT, >
   <ID, 0>
   <PuntoComa, >
   <DEC, >
   <TipoVarENT, >
   < ID, 1 >
   <PuntoComa, >
   \langle ID, 0 \rangle
   \langle ASIG, \rangle
   \langle ENT, 3 \rangle
   <PuntoComa, >
   < ID, 1 >
   \langle ASIG, \rangle
   < ID, 0 >
   <PuntoComa, >
   \langle IF, \rangle
   <ParentesisAbrir, >
   < ID, 0 >
   <MENOR, >
   \langle ID, 1 \rangle
   <ParentesisCerrar, >
   < ID, 1 >
   \langle ASIG, \rangle
   <ENT, 1>
   <PuntoComa, >
   < ID, 0 >
   <ASIG, >
   < ID, 0 >
   \langle SUMA, \rangle
   \langle ID, 1 \rangle
   <PuntoComa, >
   <Print, >
   <ParentesisAbrir, >
   < ID, 0 >
   <ParentesisCerrar, >
   <PuntoComa, >
   <Print, >
   <ParentesisAbrir, >
```

```
<ID, 1>
<ParentesisCerrar, >
<PuntoComa, >

Tabla Simbolos #1:

* LEXEMA: 'a'

* LEXEMA: 'b'
```