Memoria de la Práctica de Procesadores de Lenguajes: Analizador Sintáctico

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

Grupo 82

November 25, 2019

Índice

1 Diseño del Analizador Sintáctico

1.1 Gramática

$Terminales = \{ \; ; \; \{ \; \} \; id \; ent \; cadena \; (\;) \; + \; < \; ! \; = \; |= \; var \; int \; boolean \; string \; print \; input \; , \; return \; function \; if \; else \; \}$

$NoTerminales = \{ \; P \; D \; T \; F \; T1 \; A \; K \; C \; S \; L \; M \; Q \; S1 \; G \; X \; E \; U \; R \; V \; S2 \; \}$

$Axioma = P$

$Producciones = \{$

$P \rightarrow D \; P$

$P \rightarrow F \; P$

$P \rightarrow S \; P$

$D \rightarrow var \; T \; id \; ;$

$T \rightarrow int$

$T \rightarrow string$

$T \rightarrow boolean$

$F \rightarrow function \; T1 \; id \; (\; A \;) \; \{ \; C \; \}$

$T1 \rightarrow \lambda$

$T1 \rightarrow T$

$A \rightarrow T \; id \; K$

$A \rightarrow \lambda$

$K \rightarrow \lambda$

$K \rightarrow , \; T \; id \; K$

$C \rightarrow D \; C$

$C \rightarrow S \; C$

$C \rightarrow \lambda$

$S \rightarrow id \; L \; E \; ;$

$S \rightarrow id \; (\; M \;) \; ;$

$S \rightarrow print \; (\; E \;) \; ;$

$S \rightarrow input \; (\; id \;) \; ;$

$S \rightarrow if \; (\; E \;) \; S1$

$S \rightarrow return \; X \; ;$

$L \rightarrow |=$

$L \rightarrow =$

$M \rightarrow E \; Q$

$M \rightarrow \lambda$

$Q \rightarrow \lambda$

$Q \rightarrow , \; E \; Q$

$S1 \rightarrow \{ \; S2 \; \} \; G$

$S1 \rightarrow S$

$G \rightarrow else \; \{ \; S2 \; \}$

$G \rightarrow \lambda$

$X \rightarrow E$

$X \rightarrow \lambda$

$E \rightarrow E < U$

$E \rightarrow U$

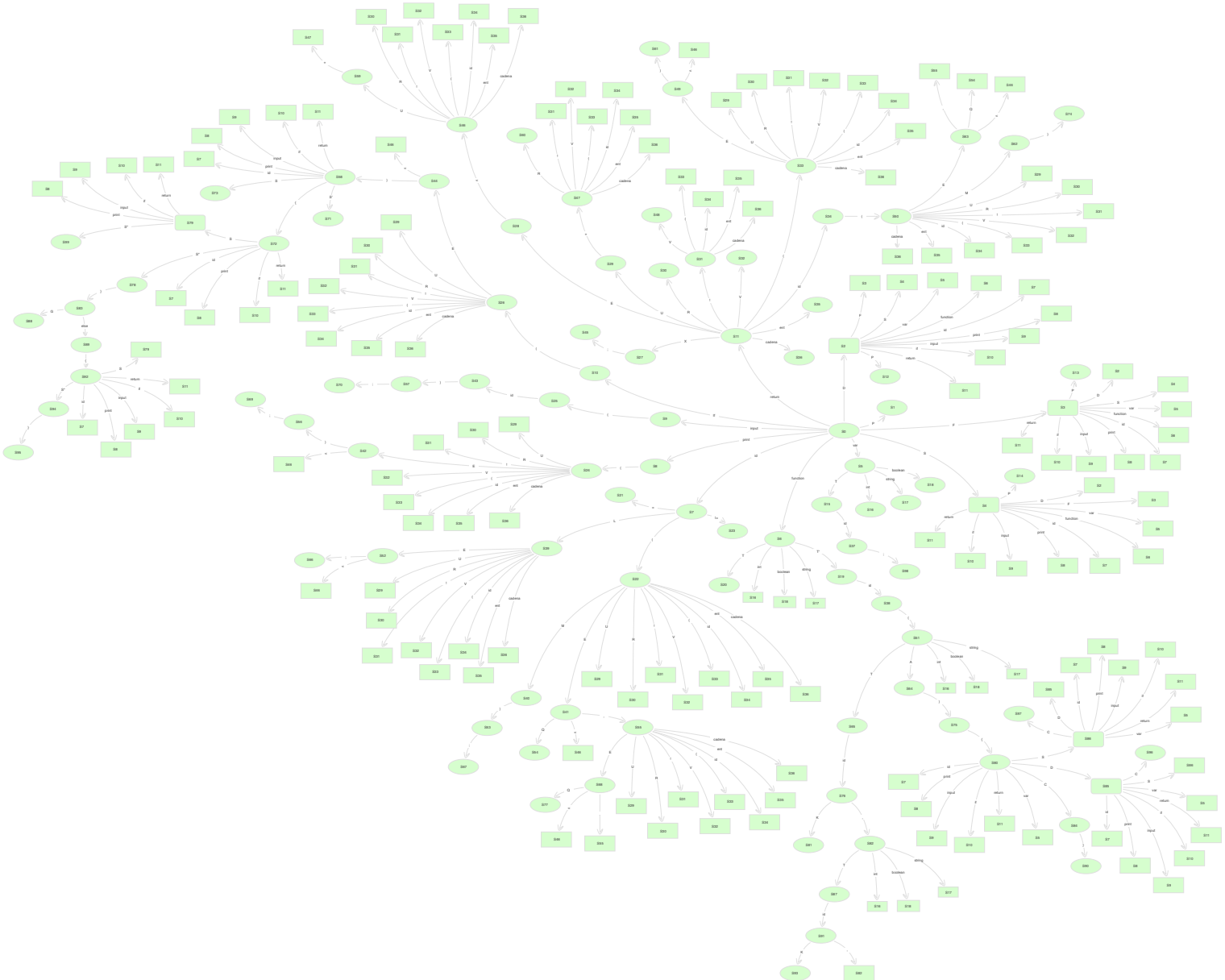
$U \rightarrow U + R$

$U \rightarrow R$

$R \rightarrow\ !\ V$
$R \rightarrow\ V$
$V \rightarrow\ (\ E\ )$
$V \rightarrow\ id$
$V \rightarrow\ id\ (\ M\ )$
$V \rightarrow\ ent$
$V \rightarrow\ cadena$
$S2 \rightarrow\ S\ \ S2$
$S2 \rightarrow\ S$
$P \rightarrow\ \lambda$

}

1.2 Autómata Reconocedor de Prefijos Viables



1.2.1 Estados del autómata

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

U → • R, R → • ! V, R → • V, V → • (E), V → • id,
V → • id(M), V → • ent, V → • cadena}

S_{34}={V →id •, V → id • (M)}

S_{35}={V →ent •}

S_{36}={V →cadena •}

S_{37}={D →var T id •;}

S_{38}={F →function T1 id • (A){C}}

S_{39}={S →id L • E, E → • E < U, E → • U, U → • U + R,
U → • R, R → • ! V, R → • V, V → • (E), V → • id,
V → • id(M), V → • ent, V → • cadena}

S_{40}={S →id (M •);}

S_{41}={M →E • Q, E → E • < U, Q → •, Q → • ,EQ}

S_{42}={S →print(E •);, E → E • < U}

S_{43}={S →input(id •);}

S_{44}={S →if(E •) S1, E → E • < U}

S_{45}={S →return X; •}

S_{46}={E →E < • U, U → • R, U → • U + R, R→ • ! V, R → • V
V → • (E), V → • id, V → • id(M), V → • ent, V → • cadena}

S_{47}={U →U + • R, R → • ! V, V → • (E), V → • id(M),
V → • ent, V → • cadena}

S_{48}={R →! V •}

S_{49}={V →(E •), E → E • < U}

S_{50}={V →id(• M), M → • E Q, M → •, E → • E < U, E → • U,
U → • U + R, U → • R, R → • ! V, R → • V, V → • (E),
V → • id, V → • id(M), V → • ent, V → • cadena}

S_{51}={F →function T1 id(• A){C}, A → • T id K, A → •, T → • int,
T → • string, T → • boolean}

S_{52}={S →id L E • ;, E → E • < U}

S_{53}={S →id(M) • ;}

S_{54}={M →E Q •}

S_{55}={Q →, • E Q, E → • E < U, E → • U,
U → • U + R, U → • R, R → • ! V, R → • V, V → • (E),
V → • id, V → • id(M), V → • ent, V → • cadena}

S_{56}={S →print(E) • ;}

S_{57}={S →input(id) • ;}

S_{58}={S →if(E) • S1, S1 → • {S2}G, S1 → • S, S → • id L E;,
S → • id(M);, S → • print(E);, S → • input(id);,
S → • if(E)S1, S → • return X ;}

S_{59}={E →E < U •, U → U • + R}

S_{60}={U →U + R •}

S_{61}={V →(E) •}

S_{62}={V →id(M •)}

S_{63}={M →E • Q, E → E • < U, Q → •, Q → • ,EQ}

S_{64}={F →function T1 id(A •){C}}

S_{65}={A →T • id K}

S_{66}={S →id L E ; •}

S_{67}={S →id (M); •}

S_{68}={Q →,E • Q, E → E • < U, Q → •, Q → • ,EQ}

S_{69}={S →print(E); •}

S_{70}={S →input(id); •}

S_{71}={S →if(E) S1 •}

S_{72}={S1 →{• S2}G, S2 → • S S2, S2 → • S, S → • id L E;,
S → • id(M);, S → • print(E);, S → • input(id);,
S → • if(E)S1, S → • return X ;}

S_{73}={S1 →S •}

S_{74}={S2 →S • S2 , S2 → S • , S2 → • S S2, S2 → • S,
S → • id L E ;, S → • id (M) ;, S → • print (E) ;,
S → • input (id) ;, S → • if (E) S1, S → • return X ;}

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

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 $\rightarrow \alpha \bullet$, B $\rightarrow \beta \bullet$} $\subset S_x \Rightarrow$ Follow(A) $\cap$ 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 $\rightarrow \alpha \bullet$ b γ, C $\rightarrow \beta \bullet$} $\subset S_x \Rightarrow$ 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	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"
S0				d7										d5				d8	d9		d11	d6	d10		r49	1	2		3				4												
S1																									a																				
S2				d7										d5				d8	d9		d11	d6	d10		r49	12	2		3				4												
S3				d7										d5				d8	d9		d11	d6	d10		r49	13	2		3				4												
S4				d7										d5				d8	d9		d11	d6	d10		r49	14	2		3				4												
S5															d16	d18	d17											15																	
S6				r9											d16	d18	d17											20		19																
S7							d22					d21	d23																					39											
S8							d24																																						
S9							d25																																						
S10							d26																																						
S11	r35			d34	d35	d36	d33				d31																													27	28	29	30	32	
S12																									r1																				
S13																									r2																				
S14																									r3																				
S15				d37																																									
S16				r5																																									
S17				r6																																									
S18				r7																																									
S19				d38																																									
S20				r10																																									
S21				r25	r25	r25	r25				r25																																		
S22				d34	d35	d36	d33	r27			d31																															41	29	30	32	
S23				r24	r24	r24	r24				r24																																		
S24				d34	d35	d36	d33				d31																															42	29	30	32	
S25				d43																																									
S26				d34	d35	d36	d33				d31																															44	29	30	32	
S27	d45																																												
S28	r34										d46																																		
S29	r37							r37	d47	r37										r37																									
S30	r39							r39	r39	r39										r39																									
S31				d34	d35	d36	d33																																					48	
S32	r41							r41	r41	r41										r41																									
S33				d34	d35	d36	d33				d31																															49	29	30	32	
S34	r43						d50	r43	r43	r43										r43																									
S35	r45							r45	r45	r45										r45																									
S36	r46							r46	r46	r46										r46																									
S37	d98																																												
S38							d51																																						
S39				d34	d35	d36	d33				d31																															52	29	30	32	
S40								d53																																					
S41								r28		d46										d55																									
S42								d56		d46																																				
S43								d57																																					
S44								d58		d46																																				
S45			r23	r23										r23				r23	r23		r23	r23	r23		r23																			
S46				d34	d35	d36	d33				d31																																59	30	32	
S47				d34	d35	d36	d33				d31																																	60	32	
S48	r40							r40	r40	r40										r40																									
S49								d61		d46																																			

	;	{	}	id	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"
S50				d34	d35	d36	d33	r27			d31																								62					63	29	30	32	
S51								r12							d16	d18	d17				65			64																						
S52	d66									d46																																			
S53	d67																																												
S54								r26																																					
S55				d34	d35	d36	d33				d31																														68	29	30	32	
S56	d69																																												
S57	d70																																												
S58		d72		d7														d8	d9		d11		d10											73				71							
S59	r36							r36	d47	r36										r36																									
S60	r38							r38	r38	r38										r38																									
S61	r42							r42	r42	r42										r42																									
S62								d74																																					
S63								r28		d46										d55																				54						
S64								d75																																					
S65				d76																																									
S66			r18	r18										r18				r18	r18		r18	r18	r18		r18																			
S67			r19	r19										r19				r19	r19		r19	r19	r19		r19																			
S68								r28		d46										d55																				77						
S69			r20	r20										r20				r20	r20		r20	r20	r20		r20																			
S70			r21	r21										r21				r21	r21		r21	r21	r21		r21																			
S71			r22	r22										r22				r22	r22		r22	r22	r22		r22																			
S72				d7														d8	d9		d11		d10											79										78
S73			r31	r31										r31				r31	r31		r31	r31	r31		r31																			
S74	r44							r44	r44	r44										r44																									
S75		d80																																											
S76								r13												d82																									
S77								r29																																					
S78			d83																																									
S79			r48	d7														d8	d9		d11		d10											79										99
S80			r17	d7										d5				d8	d9		d11		d10				85						84	86									
S81								r11																																					
S82															d16	d18	d17											87																
S83			r33	r33										r33				r33	r33		r33	r33	r33	d89	r33													88						
S84			d90																																										
S85			r17	d7										d5				d8	d9		d11		d10										96	86									
S86			r17	d7										d5				d8	d9		d11		d10				85						97	86									
S87				d91																																									
S88			r30	r30										r30				r30	r30		r30	r30	r30		r30																			
S89		d92																																											
S90				r8										r8				r8	r8		r8	r8	r8		r8																			
S91								r13												d82													93											
S92				d7														d8	d9		d11		d10											79										94
S93								r14																																					
S94			d95																																										
S95			r32	r32										r32				r32	r32		r32	r32	r32		r32																			
S96			r15																																										
S97			r16																																										
S98			r4	r4										r4				r4	r4		r4	r4	r4		r4																			
S99			r47																																										

2 Anexo de Pruebas

```
1  var int a;
2  var int b;
3  var int c;
4  print ('Introduce el primer operando');
5  input (a);
6  print ('Introduce el segundo operando');
7  input (b);
8  function int suma (int num1, int num2)
9  {
10   var int res;
11   res = num1+num2;
12   return res;
13 }
14 c = suma (a, b);
15 print (c);
```