Análisis Sintáctico

Grupo 55

Aarón Cabero Blanco
Daniel Tomás Sanchez
Alejandro Cuadrón Lafuente

7 de diciembre de 2020

Índice

1. Diseño del Analizador Sintáctico

1.1. Gramática

Y → P	V → – id
P → B P	V → id
P → F P	V → (E)
P → λ	V → id (L)
B → if (E) S	V → ent
B → let T id ;	V → cad
B → S	V → log
B → for (D ; E ; Z) C	X → E
S → id = E ;	X → λ
S → id (L) ;	L → E Q
S → alert (E) ;	L → λ
S → input (id) ;	Q → , E Q
S → return X ;	Q → λ
T → number	D → id = E
T → boolean	D → λ
T → string	Z → id = E
F → F1 F2 F3	Z → – id R
F1 → function H id	Z → λ
F2 → (A)	H → T
F3 → C	H → λ
E → E && R	A → T id K
E → R	A → λ
R → R == U	K → , T id K
R → U	K → λ
U → U - V	C → B C
U → U - V	C → λ
U → V	

1.2. Tabla LR

LR table

State	ACTION																											GOTO																									
	lambda	if	(	)	let	id	;	for	{	}	=	alert	input	return	number	boolean	string	function	&&	==	-	--	ent	cad	log	,	$	Y	P	B	S	T	F	F1	F2	F3	E	R	U	V	X	L	Q	D	Z	H	A	K	C
0	s4	s5			s6	s10		s8				s11	s12	s13				s14										1	2	7		3	9																		
1																											acc																								
2	s4	s5			s6	s10		s8				s11	s12	s13				s14										15	2	7		3	9																		
3	s4	s5			s6	s10		s8				s11	s12	s13				s14										16	2	7		3	9																		
4																										r_3																								
5			s17																																															
6															s19	s20	s21														18																			
7	r_6	r_6			r_6	r_6		r_6				r_6	r_6	r_6				r_6																																
8			s22																																															
9			s24																																23															
10			s26								s25																																						
11			s27																																															
12			s28																																															
13	s31		s37			s36																s35	s38	s39	s40													30	32	33	34	29							
14	s43														s19	s20	s21													42																	41			
15																										r_1																								
16																										r_2																								
17			s37			s36																s35	s38	s39	s40													44	32	33	34								
18						s45																																													
19						r_{13}																																												
20						r_{14}																																												
21						r_{15}																																												
22	s48					s47																																												
23									s50																																								
24	s53														s19	s20	s21													52																	51			
25			s37			s36																s35	s38	s39	s40													54	32	33	34								
26	s57		s37			s36																s35	s38	s39	s40													56	32	33	34		55						
27			s37			s36																s35	s38	s39	s40													58	32	33	34								
28						s59																																												
29							s60																																											
30							r_{33}												s61																																
31							r_{34}																																												
32	r_{21}			r_{21}			r_{21}												r_{21}	s62						r_{21}																									
33	r_{23}			r_{23}			r_{23}												r_{23}	r_{23}	s63					r_{23}																									
34	r_{25}			r_{25}			r_{25}												r_{25}	r_{25}	r_{25}					r_{25}																									
35						s64																																												
36	r_{27}		s65	r_{27}			r_{27}												r_{27}	r_{27}	r_{27}					r_{27}																									
37			s37			s36																s35	s38	s39	s40													66	32	33	34								
38	r_{30}			r_{30}			r_{30}												r_{30}	r_{30}	r_{30}					r_{30}																									
39	r_{31}			r_{31}			r_{31}												r_{31}	r_{31}	r_{31}					r_{31}																									

40	r_{32}			r_{32}			r_{32}												r_{32}	r_{32}	r_{32}					r_{32}																							
41						s67																																											
42						r_{44}																																											
43						r_{45}																																											
44				s68															s61																														
45							s69																																										
46							s70																																										
47											s71																																						
48							r_{40}																																										
49	r_{16}	r_{16}			r_{16}	r_{16}		r_{16}				r_{16}	r_{16}	r_{16}				r_{16}																															
50	s74	s5			s6	s10		s8				s11	s12	s13																73	7																		72
51				s75																																													
52						s76																																											
53				r_{47}																																													
54							s77												s61																														
55				s78																																													
56	s81																		s61							s80																	79						
57				r_{36}																																													
58				s82															s61																														
59				s83																																													
60	r_{12}	r_{12}			r_{12}	r_{12}		r_{12}				r_{12}	r_{12}	r_{12}				r_{12}																															
61			s37			s36																s35	s38	s39	s40													84	33	34									
62			s37			s36																s35	s38	s39	s40														85	34									
63			s37			s36																s35	s38	s39	s40															86									
64	r_{26}			r_{26}			r_{26}												r_{26}	r_{26}	r_{26}					r_{26}																							
65	s57		s37			s36																s35	s38	s39	s40												56	32	33	34		87							
66				s88															s61																														
67			r_{17}																																														
68						s10						s11	s12	s13																	89																		
69	r_{5}	r_{5}			r_{5}	r_{5}		r_{5}				r_{5}	r_{5}	r_{5}				r_{5}																															
70			s37			s36																s35	s38	s39	s40												90	32	33	34									
71			s37			s36																s35	s38	s39	s40												91	32	33	34									
72										s92																																							
73	s74	s5			s6	s10		s8				s11	s12	s13																73	7																		93
74										r_{51}																																							
75									r_{18}																																								
76	s96																									s95																						94	
77	r_{8}	r_{8}			r_{8}	r_{8}		r_{8}				r_{8}	r_{8}	r_{8}				r_{8}																															
78							s97																																										
79				r_{35}																																													
80			s37			s36																s35	s38	s39	s40												98	32	33	34									
81				r_{38}																																													
82							s99																																										
83							s100																																										

84	r_{20}			r_{20}			r_{20}												r_{20}	s62						r_{20}																							
85	r_{22}			r_{22}			r_{22}												r_{22}	r_{22}	s63					r_{22}																							
86	r_{24}			r_{24}			r_{24}												r_{24}	r_{24}	r_{24}					r_{24}																							
87				s101																																													
88	r_{28}			r_{28}			r_{28}												r_{28}	r_{28}	r_{28}					r_{28}																							
89	r_4	r_4			r_4	r_4		r_4				r_4	r_4	r_4				r_4																															
90							s102												s61																														
91							r_{39}												s61																														
92	r_{19}	r_{19}			r_{19}	r_{19}		r_{19}				r_{19}	r_{19}	r_{19}				r_{19}																															
93										r_{50}																																							
94				r_{46}																																													
95															s19	s20	s21																103																	
96				r_{49}																																													
97	r_9	r_9			r_9	r_9		r_9				r_9	r_9	r_9				r_9																															
98	s81																		s61							s80																	104						
99	r_{10}	r_{10}			r_{10}	r_{10}		r_{10}				r_{10}	r_{10}	r_{10}				r_{10}																															
100	r_{11}	r_{11}			r_{11}	r_{11}		r_{11}				r_{11}	r_{11}	r_{11}				r_{11}																															
101	r_{29}			r_{29}			r_{29}												r_{29}	r_{29}	r_{29}					r_{29}																							
102	s108					s106																s107																							105				
103						s109																																											
104				r_{37}																																													
105				s110																																													
106											s111																																						
107						s112																																											
108				r_{43}																																													
109	s96																									s95																						113	
110									s114																																								
111			s37			s36																s35	s38	s39	s40												115	32	33	34									
112				r_{42}																																													
113				r_{48}																																													
114	s74	s5			s6	s10		s8				s11	s12	s13																73	7																		116
115				r_{41}															s61																														
116										s117																																							
117	r_7	r_7			r_7	r_7		r_7				r_7	r_7	r_7				r_7																															

Como puede observarse en la tabla, esta gramática es adecuada para este tipo de Analizador sintáctico, puesto que no se produce ningún tipo de conflicto.

2. Anexo

A continuación se presentarán los casos de prueba para comprobar el funcionamiento de la gramática de la sección 1.1.

2.1. Casos de prueba correctos

2.1.1. Caso de prueba 1

Programa introducido:

```
function boolean bisiesto (number a)
{
  return (a - 4 == 0 && a - 100 == 0 && a - 400 == 0);
}
```

Parse:

Ascendente 15 45 18 14 50 47 19 28 26 31 25 24 31 26 23 22 28 26 31 25 24 31 26 23 21 28 26 31 25 24 31 26 23 21 29 26 24 22 34 13 7 52 51 20 17 4 3 1

VASt:

Árbol resultante de esta prueba en la sección 2.3.1 que se encuentra en la página 11

2.1.2. Caso de prueba 2

Programa introducido:

```
/* prueba correcta */
let number entero;
let string cadena;
let boolean logico;
let number abcdefghij________;
function FuncionNumero ()
{
  let number i;
  let number j;

  logico = i == j;
}
```

Parse:

Ascendente 14 6 16 6 15 6 14 6 46 18 48 19 14 6 14 6 28 26 24 28 26 23 22 9 7 52 51 51 51 20 17 4 3 2 2 2 2 1 l l

VASt:

Árbol resultante de esta prueba en la sección 2.3.2 que se encuentra en la página 14

2.1.3. Caso de prueba 3

Programa introducido:

```
function FuncionSentencia (number b, boolean z)
{
  for (b=1;z; )
  {
  alert (88);
  }
}
function Funcion (number x, boolean b)
{
  if (x == 0)FuncionSentencia(x,b);
  alert
    (cadena);return;
}
```

Parse:

Ascendente 46 18 14 15 50 49 47 19 31 26 24 22 40 28 26 24 22 44 31 26 24 22 11 7 52 51 8 52 51 20 17 46 18 14 15 50 49 47 19 28 26 24 31 26 23 22 28 26 24 22 28 26 24 22 39 38 36 10 5 28 26 24 22 11 7 35 13 7 52 51 51 51 20 17 4 3 3 1

VASt:

Árbol resultante de esta prueba en la sección 2.3.3 que se encuentra en la página 17

2.2. Casos de prueba incorrectos

2.2.1. Caso de prueba 1

Programa introducido:

```
function FuncionSentencia (number b, boolean z)
{
  for (b=1;z; ))
  {
  alert (88);
  }
}
```

Parse:
Ascendente 46 18 14 15 50 49 47 19 31 26 24 22 40 28 26 24 22 44

Error sintáctico:
Token ilegal CEPAREN en la linea 3

2.2.2. Caso de prueba 2

Programa introducido:

```
let number entero;
let string cadena;
let boolean logico;
let number abcdefghij________;
function FuncionNumero ()
{
  let number i;
  let number j;

  logico = i = j;
}
function string FuncionRetorno (string hola, number x, boolean z)
{
  let number i;

  input (i);
  alert (hola);
  return string;
}
```

Parse:
Ascendente 14 6 16 6 15 6 14 6 46 18 48 19 14 6 14 6

Error sintáctico:
Token ilegal OPASIG en la linea 9

2.2.3. Caso de prueba 3

Programa introducido:

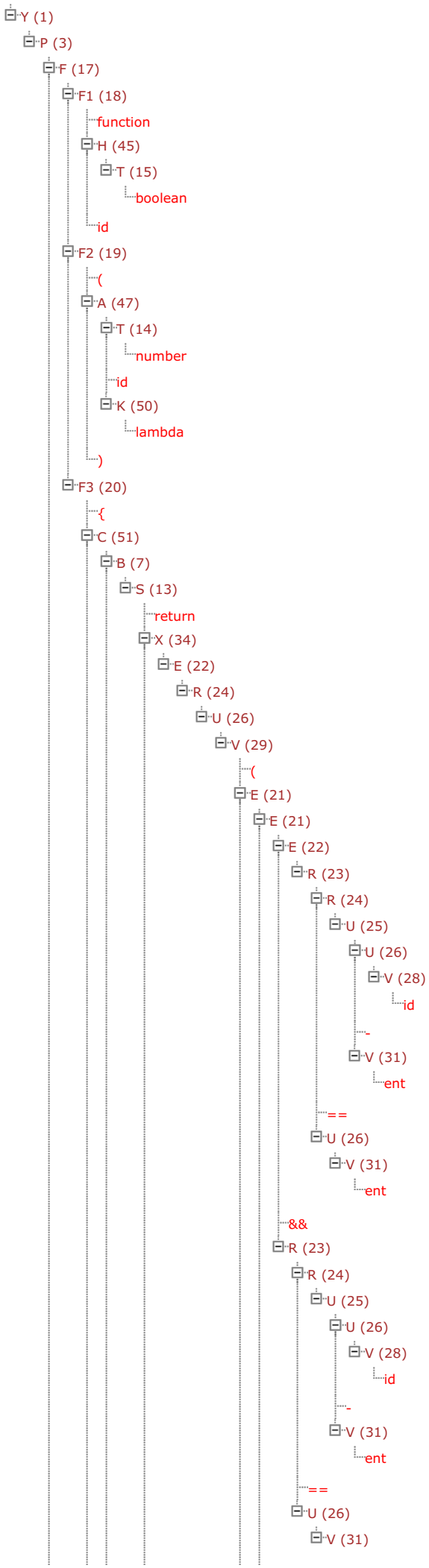
```
let number iii;
let boolean bbb;
/* comentario
multi
linea ***/
input(iii);
alert (iii) ;

iii = iii function H (no&7/);
bbb = bbb;

Funcion
 ( iii,
   bbb );
```

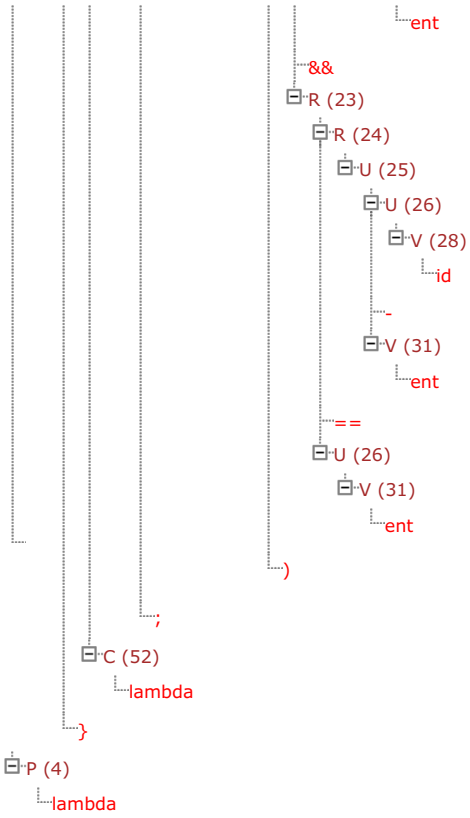
Parse:
Ascendente 14 6 15 6 12 7 28 26 24 22 11 7

Error sintáctico:
Token ilegal FUNCTION en la linea 9

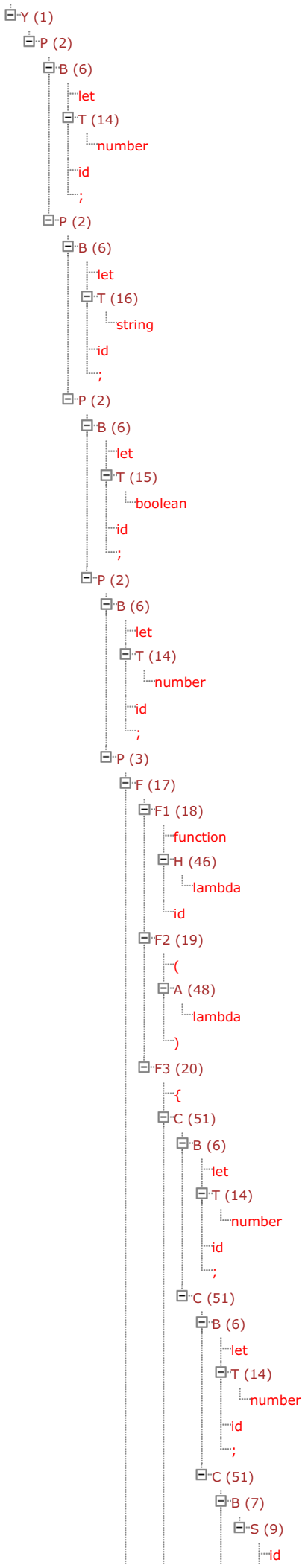
2.3. Árboles generados por la herramienta VASt

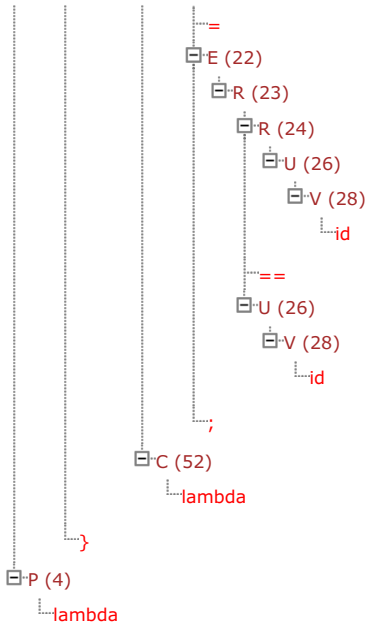
2.3.1. Árbol generado por la primera prueba correcta


Y (1)
P (3)
F (17)
F1 (18)
function
H (45)
T (15)
boolean
id
F2 (19)
(
A (47)
T (14)
number
id
K (50)
lambda
)
F3 (20)
{
C (51)
B (7)
S (13)
return
X (34)
E (22)
R (24)
U (26)
V (29)
(
E (21)
E (21)
E (22)
R (23)
R (24)
U (25)
U (26)
V (28)
id
-
V (31)
ent
==
U (26)
V (31)
ent
&&
R (23)
R (24)
U (25)
U (26)
V (28)
id
-
V (31)
ent
==
U (26)
V (31)

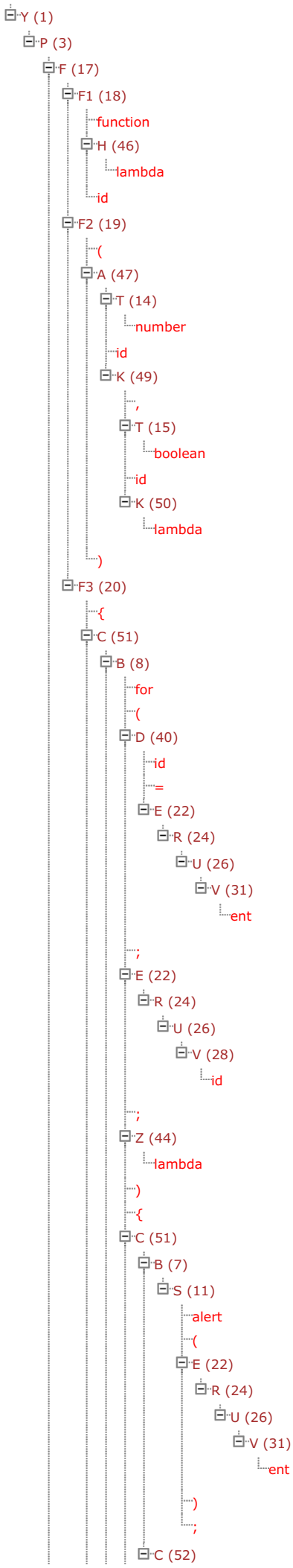

ent
&&
R (23)
R (24)
U (25)
U (26)
V (28)
id
-
V (31)
ent
==
U (26)
V (31)
ent
)
;
C (52)
lambda
}
P (4)
lambda

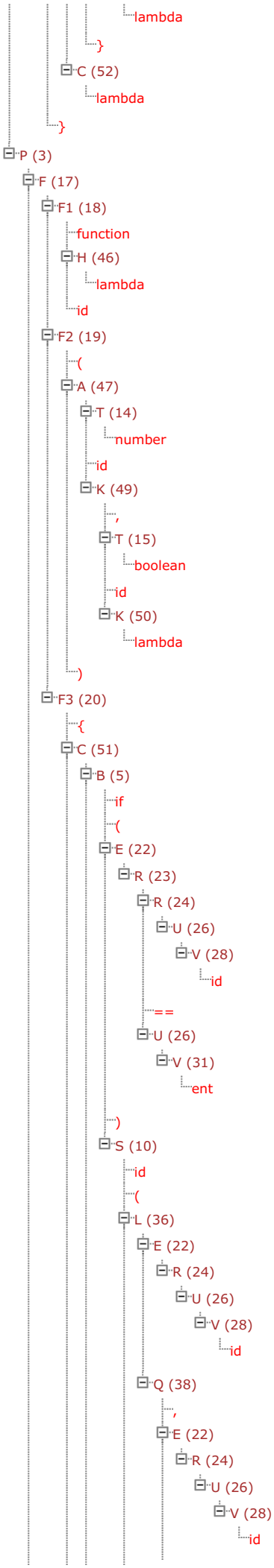
2.3.2. Árbol generado por la segunda prueba correcta

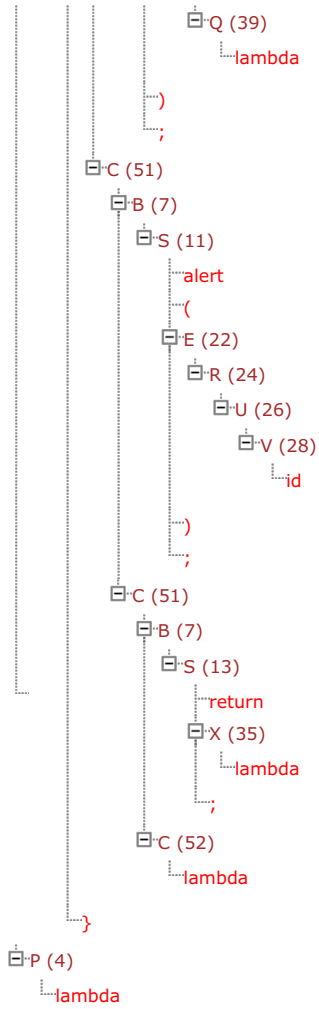

Y (1)
P (2)
B (6)
let
T (14)
number
id
;
P (2)
B (6)
let
T (16)
string
id
;
P (2)
B (6)
let
T (15)
boolean
id
;
P (2)
B (6)
let
T (14)
number
id
;
P (3)
F (17)
F1 (18)
function
H (46)
lambda
id
F2 (19)
(
A (48)
lambda
)
F3 (20)
{
C (51)
B (6)
let
T (14)
number
id
;
C (51)
B (6)
let
T (14)
number
id
;
C (51)
B (7)
S (9)
id


=
E (22)
R (23)
R (24)
U (26)
V (28)
id
==
U (26)
V (28)
id
;
C (52)
lambda
}
P (4)
lambda

2.3.3. Árbol generado por la tercera prueba correcta


Y (1)
P (3)
F (17)
F1 (18)
function
H (46)
lambda
id
F2 (19)
(
A (47)
T (14)
number
id
K (49)
,
T (15)
boolean
id
K (50)
lambda
)
F3 (20)
{
C (51)
B (8)
for
(
D (40)
id
=
E (22)
R (24)
U (26)
V (31)
ent
;
E (22)
R (24)
U (26)
V (28)
id
;
Z (44)
lambda
)
{
C (51)
B (7)
S (11)
alert
(
E (22)
R (24)
U (26)
V (31)
ent
)
;
C (52)


lambda
}
C (52)
lambda
}
P (3)
F (17)
F1 (18)
function
H (46)
lambda
id
F2 (19)
(
A (47)
T (14)
number
id
K (49)
,
T (15)
boolean
id
K (50)
lambda
)
F3 (20)
{
C (51)
B (5)
if
(
E (22)
R (23)
R (24)
U (26)
V (28)
id
==
U (26)
V (31)
ent
)
S (10)
id
(
L (36)
E (22)
R (24)
U (26)
V (28)
id
Q (38)
,
E (22)
R (24)
U (26)
V (28)
id


Q (39)
lambda
)
;
C (51)
B (7)
S (11)
alert
(
E (22)
R (24)
U (26)
V (28)
id
)
;
C (51)
B (7)
S (13)
return
X (35)
lambda
;
C (52)
lambda
}
P (4)
lambda