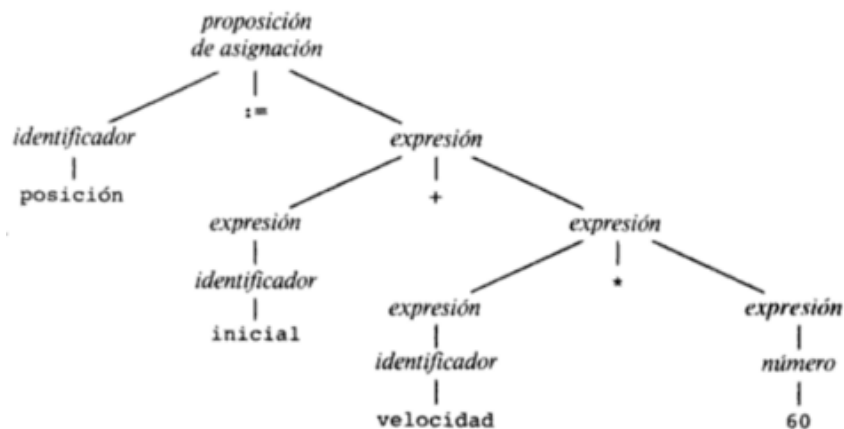




Definición de Variables en el analizador sintáctico

Curso: Compiladores



Arbol de análisis sintáctico para posición := inicial + velocidad * 60.

De: Ruelas Lope, Rodrigo Alonso

Ciclo: 3

Semestre: V

2 de Abril del 2024

Contents

1	Introducción	2
1.1	Justificación	2
1.2	Objetivo	2
2	Propuesta	3
2.1	Realizando la gramática LL1	3
3	Ejemplos	4
3.1	Prueba 1	4
3.2	Prueba 2	4
3.3	Prueba 3	5
3.4	Prueba 4	5
3.5	Prueba 5	6

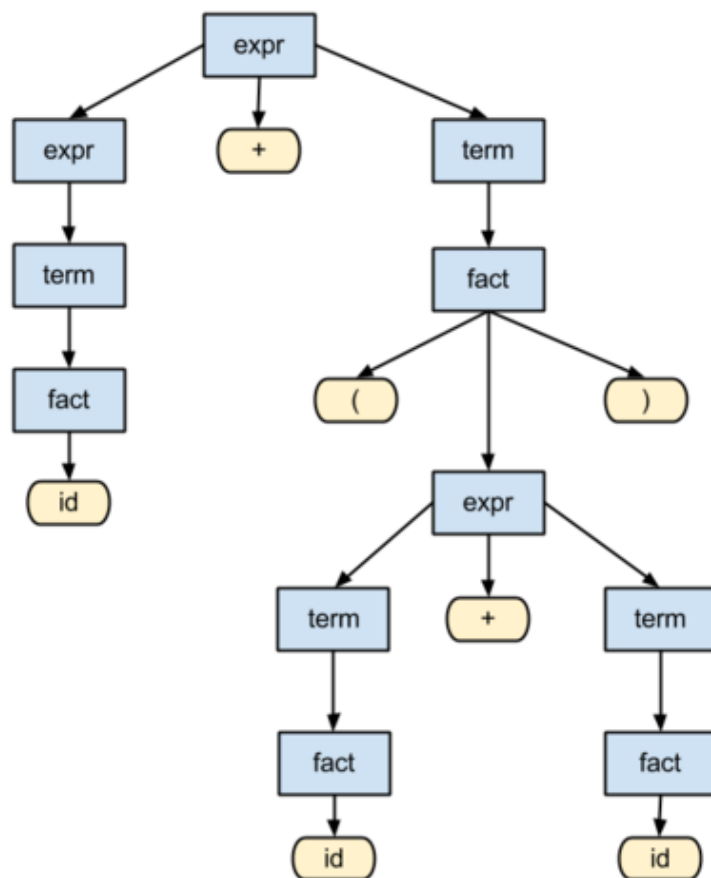
1 Introducción

1.1 Justificación

Ya hemos realizado el analizador léxico de nuestro nuevo lenguaje de programación PC, ahora vamos con el siguiente paso, elaborar el analizador sintáctico. Del analizador sintáctico vamos a obtener el árbol sintáctico que es construido por los tokens generados por el analizador léxico.

1.2 Objetivo

Como primer paso, vamos a elaborar una gramática para definir variables en nuestro lenguaje PC, con la ayuda de la gramática LL1 y los tokens de nuestro lenguaje.



2 Propuesta

2.1 Realizando la gramática LL1

Vamos a elaborar la gramática LL1 para nuestra definición de variables en el lenguaje PC. Para ello, vamos a necesitar nuestros atributos, los ID's, los signos aritméticos y lógicos (para int, float, bool), y los textos (para string).

Listing 1: Gramática LL1

```
S → A ID E

A → A_int
A → A_float
A → A_bool
A → A_string

E → S_igual V
E → ' '

V → Sig_( V Sig_ ) V1
V → ID V1
V → Num V1
V → Tex
V → B

V1 → OP V
V1 → ' '

B → Bool_True
B → Bool_False

OP → S_sum
OP → S_res
OP → S_multi
OP → S_div
```

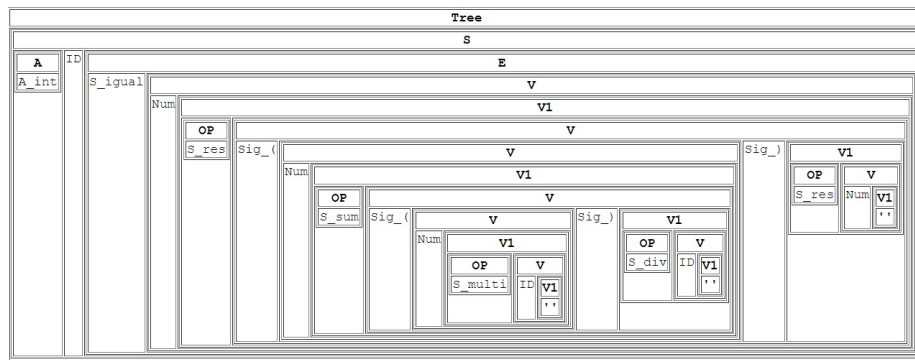
3 Ejemplos

3.1 Prueba 1

Listing 2: Prueba 1

Codigo: `int id = num - (num + (num * id) / id) - num`

Gramatica: `A_int ID S_igual Num S_res Sig_(Num S_sum Sig_(Num S_multi ID Sig_) S_div ID Sig_) S_res Num`

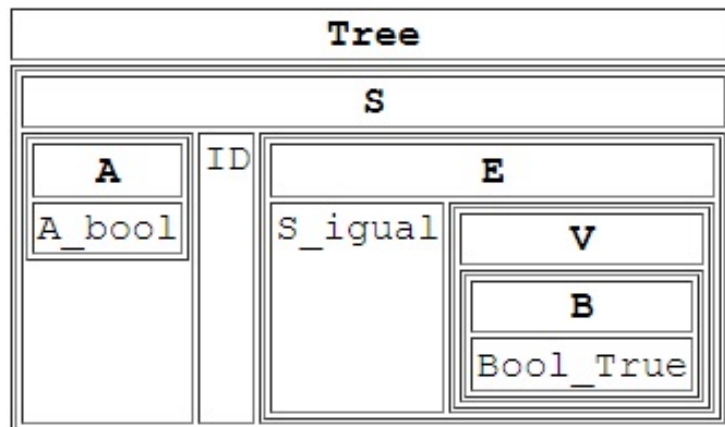


3.2 Prueba 2

Listing 3: Prueba 2

Codigo: `bool id = True`

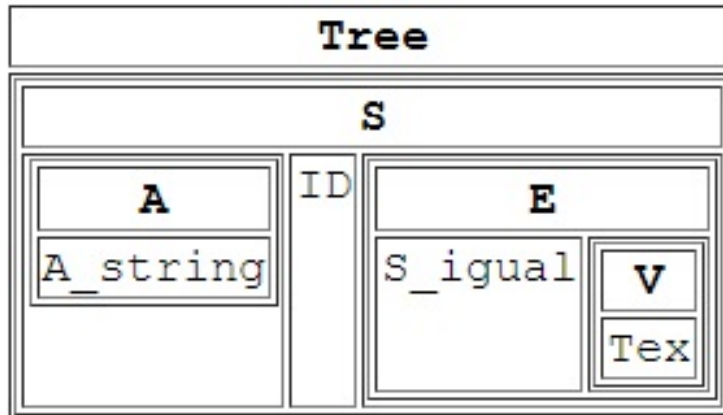
Gramatica: `A_bool ID S_igual Bool_True`



3.3 Prueba 3

Listing 4: Prueba 3

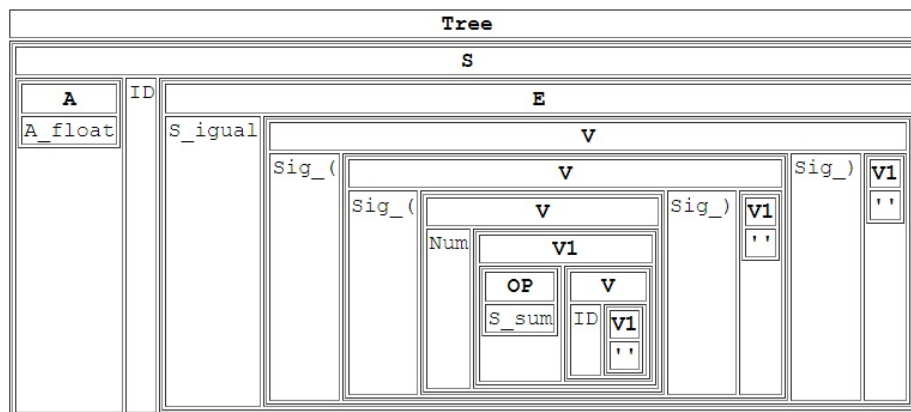
Codigo: `string id = "Hola-Mundo"`
 Gramatica: `A_string ID S_igual Tex`



3.4 Prueba 4

Listing 5: Prueba 4

Codigo: `float id = ((num + id))`
 Gramatica: `A_float ID S_igual Sig-(Sig-(Num S_sum ID Sig-) Sig-)`



3.5 Prueba 5

Listing 6: Prueba 5

Codigo: `int id = ((num + id)) / (id - num + num)`

Gramatica: A_float ID S_igual Sig_ (Sig_ (Num S_sum ID Sig_) Sig_) S_div
Sig_ (ID S_res Num S_sum Num Sig_)

