



Universidad Simón Bolívar  
Coordinación de Ingeniería de la Computación  
Curso: Traductores E Interpretadores  
Profesor: Ricardo Monascal

## INFORME PROYECTO 1 (Traductores E Interpretadores)

Autores: Arleyn Goncalves 10-10290  
Francisco Sucre 10-10717

Sartenejas, Febrero del 2016

# **INFORME DE IMPLEMENTACIÓN DE PROYECTO**

## **Herramientas utilizadas**

- Python 3.5
- Eclipse ver 4.5 (Mars) como IDE para el desarrollo del proyecto, con el complemento “PyDev” para integrar el IDE con el idioma Python.
- Paquete PLY de Python con algunas modificaciones hechas para cumplir con lo necesario del proyecto.
- GitHub como herramienta de control de versiones, utilizando Egit (Complemento de Git para eclipse), RabbitSVN(Linux) y TortoiseGit (Windows) como herramientas para manejar el repositorio.

## **Detalle de la implementación**

- Se importo la librería PLY al proyecto en un archivo aislado llamado “Parser.py”, en el cual se diseña la gramática para el lenguaje del BOT del parser y en donde se crea el árbol sintáctico abstracto.
  - La gramática implementada se encuentra en el archivo “Gramatica.txt” en la carpeta de Proyecto2/Documentos.
- Se importo el parser creado en el archivo “SintBot.py” el cual se encarga de recorrer el contenido del archivo de entrada, y de imprimir el resultado de la ejecución del analizador sintáctico.
- Se creo la clase “Instruction.py” y “Expression.py” para la implementación de las clases que van a imprimir el árbol sintáctico abstracto .

## **Instalación de PLY**

- Ya que el proyecto utiliza una librería PLY personalizada para el proyecto, es necesario

aplicarle los cambios a la librería PLY que se use en la maquina, podemos:

- Agregar en la linea 144 el siguiente codigo

- # Atributos agregados

```
self.error_Found = False    # Booleano que avisa si se consiguió un error.
```

```
self.current = -1           # Usado para el calculo de columna interno.
```

- Reemplazar el contenido del lex.py original por el contenido del archivo lex.py dentro del rar que se encuentra en la carpeta de documentos.
- Instalar la librería personalizada mediante el comando “python3 setup.py install” en linux y “ python3 setup.py install” en windows, en un terminal abierto en la carpeta de PLY.

### **Instrucciones De Uso**

- Abrir una ventana de terminal.
- Trasladarse a la carpeta de “scripts”.
- En Linux escribir ./SintBot <texto> donde <texto> es el nombre del archivo donde se encuentra el código a analizar.
- En Windows escribir SintBot <texto> donde <texto> es el nombre del archivo donde se encuentra el código a analizar.
- El archivo <texto> debe estar guardado en la carpeta de Input\_Cases.
- La pantalla del terminal se vaciara y el programa imprimirá el resultado.

## REVISIÓN TEÓRICO – PRÁCTICA

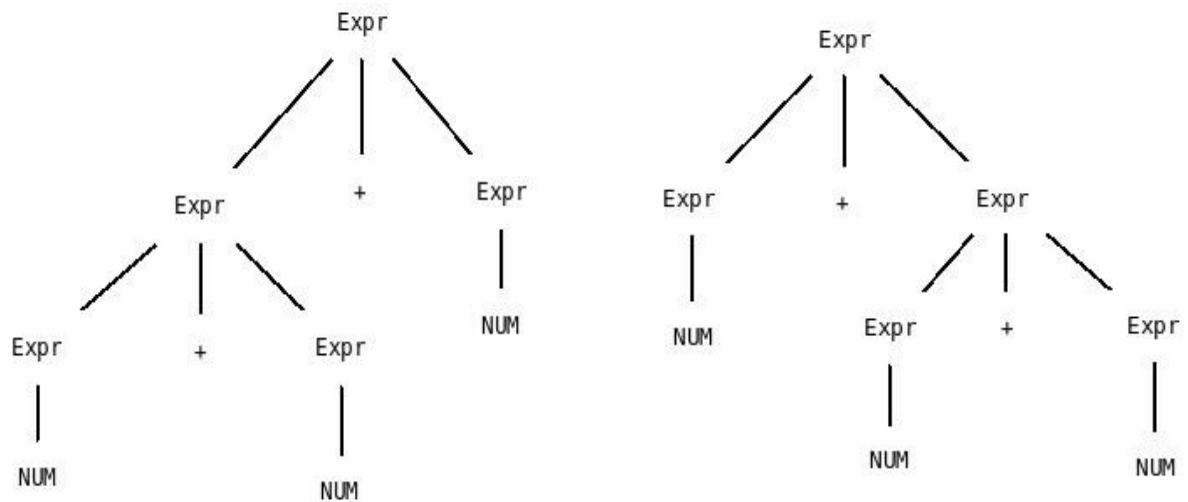
1. Sea  $G_1$  la gramática  $(\{ \text{Expr} \}, \{ +, \text{NUM} \}, P_1, \text{Expr})$ , con  $P_1$  compuesto por las siguientes producciones:

$\text{Expr} \rightarrow \text{Expr} + \text{Expr}$

$\text{Expr} \rightarrow \text{NUM}$

(a) Muestre que la frase  $\text{NUM} + \text{NUM} + \text{NUM}$  de  $G_1$  es ambigua

Se puede decir que una frase es ambigua cuando puede ser creada por dos o más árboles de derivación, a continuación le mostraremos dos árboles diferentes que pueden crear la frase  $\text{NUM} + \text{NUM} + \text{NUM}$ .



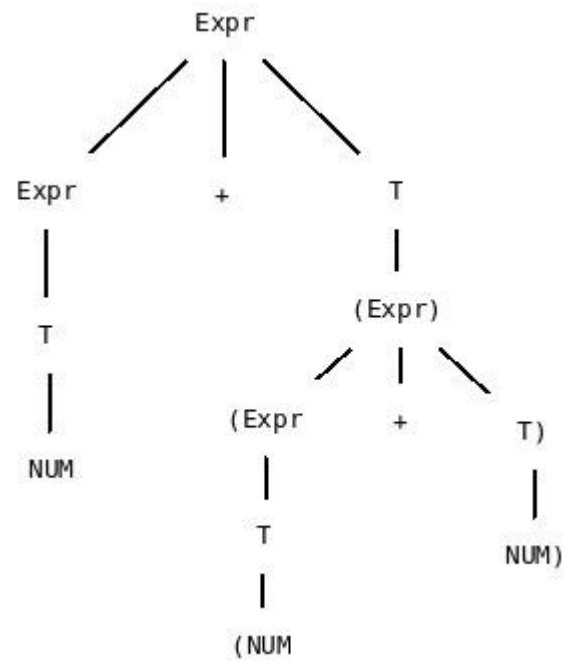
(b) Dé una gramática no ambigua Izq(G1) que genere el mismo lenguaje que G1 y que asocie las expresiones aritméticas generadas hacia la izquierda. Dé también una gramática Der(G1) con las mismas características pero que asocie a la derecha

Der(G1)

$\text{Expr} \rightarrow \text{Expr} + \text{T}$   
 $\quad \quad \quad | \text{T}$

$\text{T} \rightarrow (\text{Expr})$   
 $\quad \quad \quad | \text{NUM}$

Este árbol genera la frase:  
 NUM + (NUM + NUM)

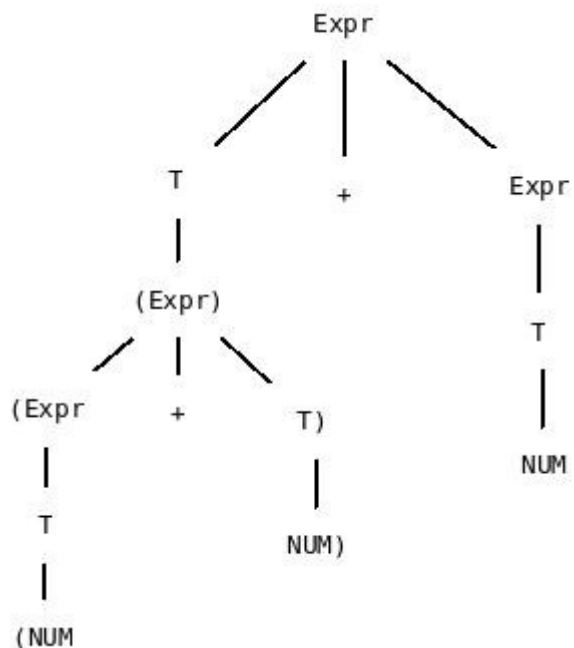


Izq(G1)

$\text{Expr} \rightarrow \text{T} + \text{Expr}$   
 $\quad \quad \quad | \text{T}$

$\text{T} \rightarrow (\text{Expr})$   
 $\quad \quad \quad | \text{NUM}$

Este árbol genera la frase:  
 (NUM + NUM) + NUM



(c) ¿ Importa si se asocian las expresiones hacia la izquierda o hacia la derecha ? Considere el caso del operador “ - ” o el caso del operador “ / ”

Si importa cuando se asocian las expresiones hacia la derecha o hacia la izquierda, ya que debido a las asociaciones el resultado puede ser diferente especialmente en el caso del operador “ - ” o el operador “ / ”.

A continuación mostraremos un ejemplo para el caso del operador “ - ” y el operador “ / ”, y demostramos como influye las asociaciones en el resultado.

- Caso del operador “ - ”

Tenemos que en Izq(G1) = (NUM - NUM) - NUM  
Der(G1) = NUM - (NUM - NUM)

$$(NUM - NUM) - NUM = NUM - (NUM - NUM)$$

$$\begin{aligned}(9 - 7) - 4 &= 9 - (7 - 4) \\ 2 - 4 &= 9 - 3 \\ -2 &\neq 6\end{aligned}$$

Se puede observar que debido a la asociación los resultados son totalmente diferentes.

- Caso del operador “ / ”

Tenemos que en Izq(G1) = (NUM / NUM) / NUM  
Der(G1) = NUM / (NUM / NUM)

$$(NUM / NUM) / NUM = NUM / (NUM / NUM)$$

$$\begin{aligned}(9 / 3) / 3 &= 9 / (3 / 3) \\ 3 - 3 &= 9 - 1 \\ 1 &\neq 9\end{aligned}$$

Se puede observar que debido a la asociación los resultados son totalmente diferentes.

2. Suponga que para el manejo de este operador se decide utilizar la gramática G2 definida como  $(\{ Instr \}, \{ ;, IS \}, P2, Instr)$ , con P2 compuesto por las siguientes producciones

$Instr \rightarrow Instr ; Instr$

$Instr \rightarrow IS$

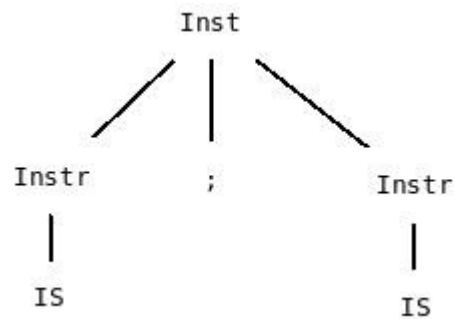
(a) ¿Presenta G2 los mismos problemas de ambigüedad que G1? ¿Cuales son las unicas frases no ambiguas de G2?

Si, G2 presenta los mismos problemas de ambigüedad que G1. A continuación mostraremos los árboles y las únicas dos frases que no son ambiguas de G2.

FRASE : IS



FRASE: IS ; IS

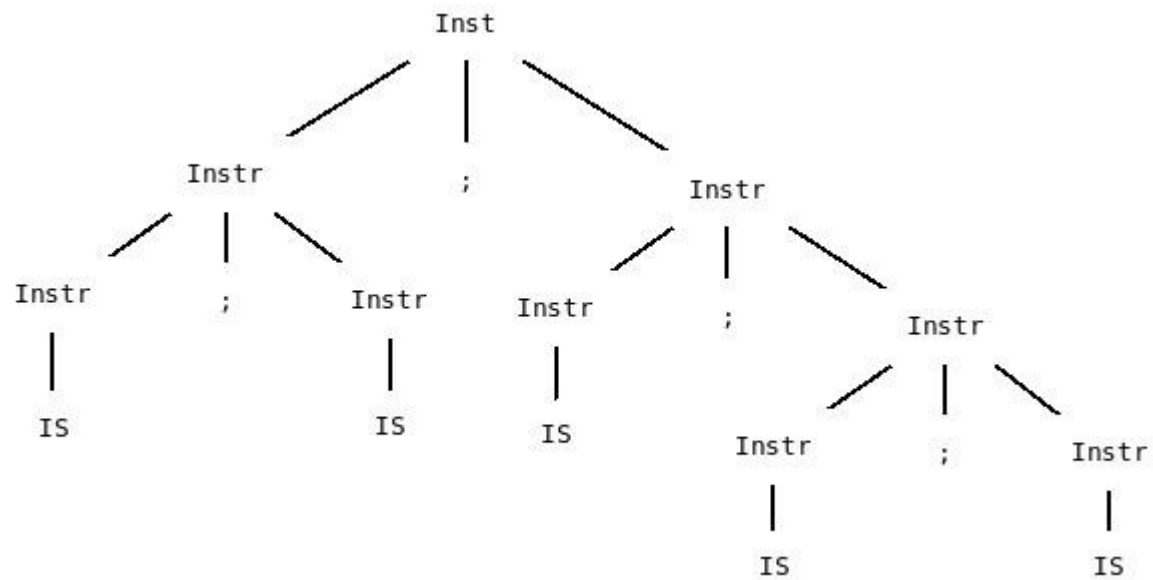


(b) ¿Importa si la ambigüedad se resuelve con asociación hacia la izquierda o hacia la derecha?

No, no importa si se resuelve con asociación hacia la izquierda o hacia la derecha debido al que el operador de secuenciación es un monoide “;” y tiene las características de que es asociativo no importa el orden en que se operen las parejas de elementos, mientras no se cambie el orden de los elementos, siempre dará el mismo resultado.

(c) De una derivación “más a la izquierda ” (leftmost) y una derivación “más a la derecha” (rightmost) de G2 para la frase IS;IS;IS

Al hacer una derivación a la izquierda y una a la derecha se genera la frase IS ; IS ; IS ; IS.





3. Sea  $G_3$  la gramática  $(\{ Instr \}, \{ :, IF, :, ELSE, Bool, IS \}, P_3, Instr)$ ,  $P_3$  compuesto por:

$Instr \rightarrow Instr ; Instr$

$Instr \rightarrow IF Bool : Instr$

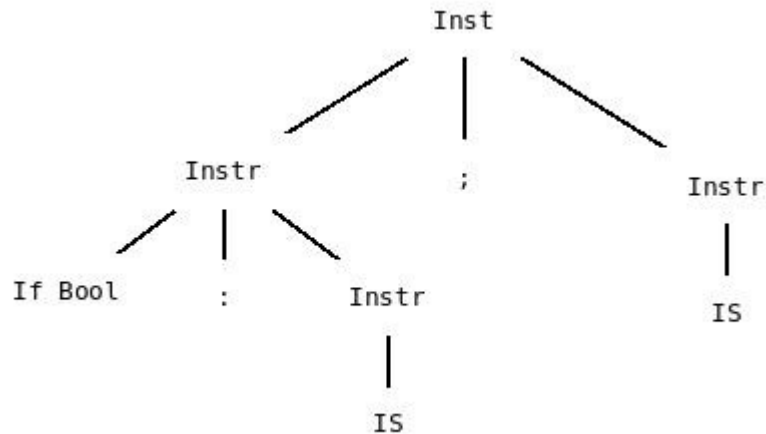
$Instr \rightarrow IF Bool : Instr ELSE : Instr$

$Instr \rightarrow IS$

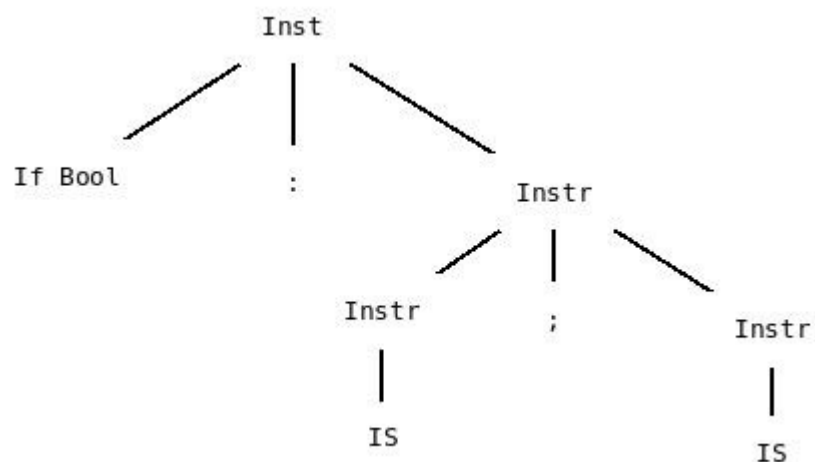
(a) Sea  $f$  la frase “IF Bool : IS ; IS”. Muestre que  $f$  es una frase ambigua de  $G_3$ .

Como ya hemos mencionado antes, sabemos que una frase es ambigua cuando, puede ser creada por dos o más árboles de derivación.

(Interpretación 1 de  $f$ )



(Interpretación 2 de  $f$ )

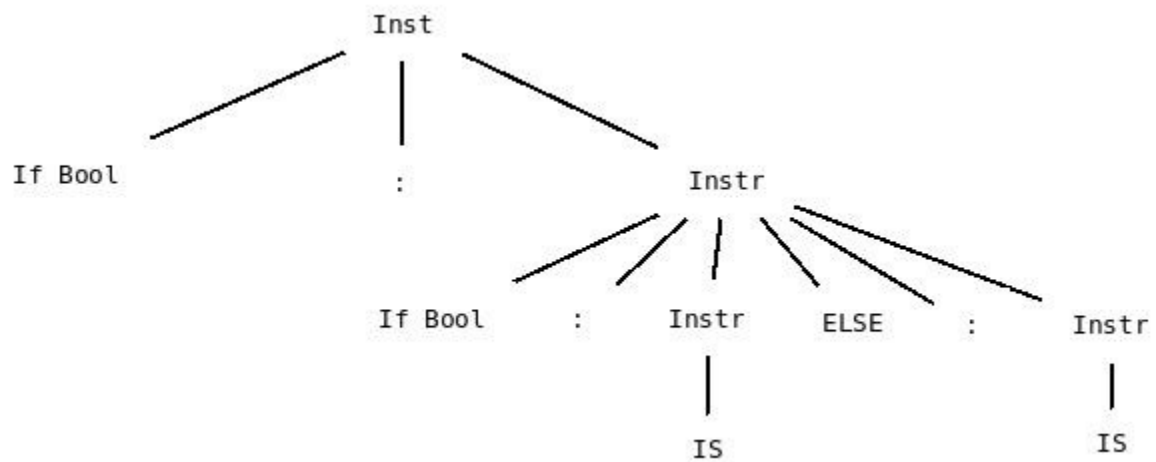


(b) De una frase g de G3 sin ocurrencias de “;” que se ambigua, y muestre que lo es.

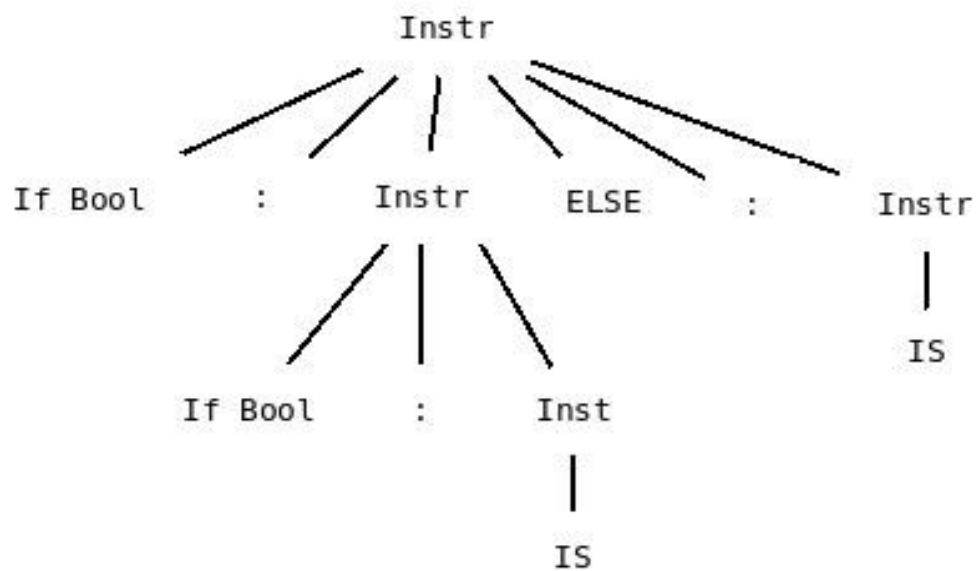
Una frase ambigua de G3 que no contenga la ocurrencia “;” es:

IF Bool : IF Bool : IS ELSE : IS

(Interpretación 1 de g)



(Interpretación 2 de g)



(c) ¿ Como se escribirían las dos interpretaciones de f y las dos interpretaciones de g usando las llaves como separadores?

En los ejercicios anteriores se especifica cuales son las dos interpretaciones de f y las dos interpretaciones de g

- En el caso de f

Interpretación 1: { {IF Bool} : {IS} ; {IS} }

Interpretación 2: { {IF Bool} : { {IS} ; {IS} } }

- En el caso de g:

Interpretación 1: { {IF Bool} : { IF Bool : {IS} ELSE : {IS} } }

Interpretación 2: { IF Bool : { IF Bool : {IS} } ELSE : {IS} }

(d) En BOT, que utiliza los terminadores “end” en la sintaxis de sus condiciones, ¿ cómo se escribiría las dos interpretaciones de f y las dos interpretaciones de g?

En los ejercicios anteriores se especifica cuales son las dos interpretaciones de f y las dos interpretaciones de g

- En el caso de f

Interpretación 1: IF Bool : IS End ; IS

Interpretación 2: Bool: IS ; IS End

- En el caso de g:

Interpretación 1: IF Bool: IF Bool : IS ELSE : IS End End

Interpretación 2: IF Bool : IF Bool : IS End ELSE : IS End