

Interprete de BOT

Etapas 4

Meggie Sánchez 11-10939, Jorge Marcano 11-10566

30/03/2016

1. Introducción

En el presente proyecto, se nos ha pedido crear un intérprete para el lenguaje llamado “BOT”. El desarrollo para el intérprete se dará en 4 etapas: (I) Análisis lexicográfico, (II) Análisis sintáctico y construcción del árbol sintáctico abstracto, (III) Análisis de contexto e (IV) Intérprete final del lenguaje.

Para esta cuarta etapa, (intérprete final del lenguaje), se interpretarán finalmente los programas escritos en BOT. Además, se verifican los errores dinámicos que corresponden a ejecuciones incorrectas y que no podían ser detectados estáticamente.

Para la interpretación final de los programas en BOT, se ha dado uso de la implementación de todos los analizadores anteriores: analizador lexicográfico, analizador sintáctico y analizador de contexto para el lenguaje BOT. Esta etapa incluye la modificación en la tabla de símbolos creada anteriormente, para guardar además, la información necesaria de las distintas variables declaradas en el programa, tales como si un bot está activado o no, la posición en la que se encuentra y el valor que posee. De igual forma se implementaron los mecanismos necesarios para detectar los posibles errores dinámicos presentes en cualquier programa BOT.

Ya dicho anteriormente, ha sido realizado en el lenguaje de programación Python (versión 3.4), con la herramienta ofrecida por este mismo llamada PLY.

2. Implementación

Para la implementación del intérprete, fueron necesarios dos procedimientos nuevos, para el caso de los árboles que representan expresiones, se creó el procedimiento evaluate, que está encargado de evaluar las expresiones y realizar la operación correspondiente (digase las operaciones aritméticas y booleanas) y dar un valor resultante, y para el caso de los árboles que representan instrucciones, se creó el procedimiento run, que está encargado de hacer que se ejecute la instrucción correspondiente utilizando el árbol respectivo y cumpliendo con las restricciones descritas (verificación de errores dinámicos) en el enunciado de esta misma etapa. De esta forma, se recorrerá el árbol sintáctico abstracto generado por el parser y se ejecutarán las instrucciones y las evaluaciones de las expresiones presentes en cualquier programa BOT.

Para la estructura tabla de símbolos, se hizo modificaciones dentro de la misma, y se decidió crear una estructura de arreglo dentro de esta tabla de símbolos, para agregar toda la información referente al robot, tales como el tipo del robot, su valor, su estado (0 para indicar que el bot esta inactivo y 1 para activo), la posición x y la posición y para indicar en que posición de la matriz

se encuentra el robot.

Otro aspecto importante a mencionar, es la creación de otra estructura tabla hash llamada superficie, para hacer uso de ella como la “matriz” dentro de nuestro lenguaje BOT, donde la clave es de la forma (x,y) que posee la posición x , y la posición y , el valor es el valor asociado al robot, y así poder hacer uso de ella en instrucciones que requieran de esta tabla como collect, y drop.

Referencias

- [1] David M. Beazley - Documentación de Python-PLY
<http://www.dabeaz.com/ply/>
- [2] T. Sudkamp. Languages and Machines . Second Edition. Addison-Wesley, 1997.