



**Universidad Simón Bolívar**  
**CI-3725 Traductores e Intérpretes**

**Proyecto BOT - Entrega 3:**  
**Análisis Semántico + ASA Decorado**

Gustavo Gutiérrez 11-10428  
José Pascarella 11-10743

Sartenejas, 07 de Marzo de 2016

## **Introducción**

En el siguiente informe, se explica brevemente la implementación de la identificación de errores estáticos en los programas escritos en BOT. También llamado análisis semántico. Son llamados estaticos porque se pueden identificar lexicograficamente, antes de la fase de ejecución. Para esto se recibe el programa ya convertido previamente en un ASA y mediante el uso de una nueva estructura, llamada tabla de símbolos que se llena en las declaraciones de las variables, se procede a chequear las instrucciones, ya sea de un comportamiento o de controlador que hagan uso de dichas variables en sus expresiones.

Que las declaraciones posean los tipos de las variables permite que se hagan algunos chequeos a nivel lexicográfico, como que las expresiones esten bien construidas (operandos consonos son su tipo), además de los más comunes como las redeclaraciones, usos de variables que no esten declaradas, entre otros.

## Detalles de la implementación

Ya elaborado el ASA se debe continuar con el análisis semántico. Luego del chequeo lexicográfico y sintáctico, ahora sabemos que estamos recibiendo un programa bien construido en BOT. Continuamos con el chequeo de tipos, que se implanto de la siguiente forma:

Se diseñó una estructura de datos, Tabla de Símbolos, que no es mas que una tabla de Hash que mapea nombres de variables (robots y variables dentro de los comportamientos) a estructuras que contengan informacion sobre estas, como la lista de comportamientos, valor y tipo.

La tabla de símbolos posee metodos para insertar una lista de variables, buscarlas en todo el entorno del programa y chequear la lista de comportamientos agregando la variable “*me*” necesaria en cada alcance de comportamientos de robot.

Se procede a llenar la tabla de símbolos con las declaraciones de un nuevo programa, cuidando las redeclaraciones. Nos dimos cuenta que introducir un nuevo alcance requeriría un modelo jerárquico de organización de las tablas, lo cual nos llevo a introducir un campo padre en el cual cada tabla apuntara a la tabla directamente superior en el orden de anidamiento, así, la primera variable que se encuentre en la subida sera la que esté tomando parte de la expresión que se esté chequeando, permitiendo la coexistencia de dos variables con el mismo nombre en distintos niveles de anidamineto. Cada clase dentro de nuestro ASA contiene su metodo check, aquí es donde realizamos la mayoría de los chequeos de tipos que se requieren para esta entrega, destacando los chequeos de las listas de comportamientos de robot, que requirieron la previa inserción de la variable “*me*” para cada uno de sus alcances. Tambien se cubrierion los chequeos de las *expresiones binarias* y *unarias* en las cuales aplicamos recorrido DFS para obtener el tipo de los operandos, con las hojas del ASA (literales de bot) devolviendo su tipo, ya sea inmutable o buscado en la tabla y comparado con el operador oportuno. Reportamos el primer error semántico que encontramos como una excepción que es atrapada por el programa principal con un mensaje que ayuda a su localización.