

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Seminario de Solución de Problemas de Traductores de Lenguajes II

Reporte de practica

Nombre del alumno:	Casillas Núñez Fernando
Profesor:	López Franco Michel Emanuel
Titulo de la practica:	Analizador Semantico
Fecha:	26/05/2020

METODOLOGÍA

La implementación realizada para este analizador semántico, consta de una pila de elementos no terminales, los cuales son las reglas introducidas a la pila sintáctica durante el análisis sintático, esta pila manda a llamar desde su tope a un método que permite validar los tipos de los elementos en cada nodo de la pila, que a fin de cuentas es un árbol sintático.

Para realizar esto, modifiquemos el analizador sintático y realicemos un modelado en clases de cada una de las reglas gramaticales, así como los tokens. Así es más fácil diseñar un método específico para cada regla/clase y así mandar llamar al método correspondiente de cada tipo de elemento. Un ejemplo en general del método en pseudocódigo:

```
void validaTipos(lista (objeto) tablasimbolos, lista(string)
listaerrores)
{
if siguiente != null;
siguiente.validaTipos(tablasimbolos, listaerrores);
}
```

Si el nodo "siguiente" de cada clase es diferente a nulo, se llamara al método de validación de ese respectivo nodo. Para algunas clases como LlamadaFunc, DefFunc, DefVar, etc... se realizaron modificaciones extras a este método, donde se manda a llamar a los métodos homónimos de los nodos de cada elemento que lo requiera.

Por ejemplo, esta es la implementación del método para la clase Expresion2 (que corresponde a la Operacion1 - R42 $iExpresion2 ::= opSuma \ iExpresion2$):

```
void validaTipos(lista (objeto) tablasimbolos, lista(string)
listaerrores)
{
Nodo auxiliar = expresionderecha.siguiente.siguiente;
expresionderecha = aux;
```

```

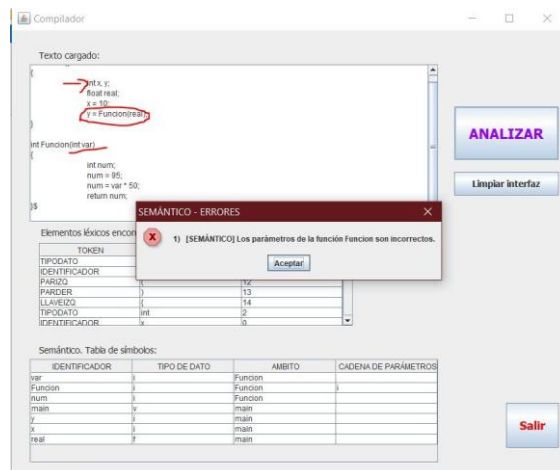
if der != null
der.validaTipos(tablasimbolos, listaerrores);
this.tipodato = der.tipodato; if siguiente != null;
siguiente.validaTipos(tablasimbolos, listaerrores);
}

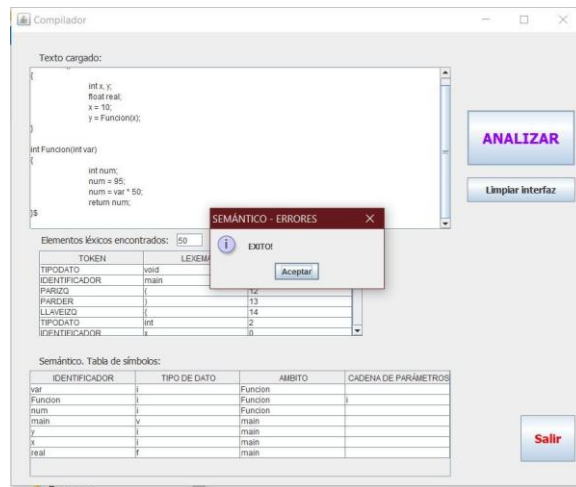
```

Para ver la completa implementación de cada una de las clases y ver cómo se diferencian, además de los métodos extras añadidos para la validación de ciertos elementos, se puede consultar el código fuente de este entregable, anexo al reporte.

Resultados

Se muestran los siguientes ejemplos de código con errores escritos a propósito para comprobar el correcto funcionamiento de la validación semántica.





CONCLUSIONES

La implementación de un analizador semántico puede llegar a ser una tarea muy complicada si no se tiene una buena noción de como debe quedar el árbol sintáctico producido por el análisis del mismo nombre. Lo comento porque en lo personal ese fue mi error, ya que el árbol no estaba bien formado, no había el correcto llamado al método de "validaTipos()" que permitiera evaluar la semántica de cada nodo de la estructura. El manejo de la tabla de símbolos es fundamental, aunque nunca lo veamos tangiblemente en algunos otros compiladores convencionales de diferentes lenguajes, ya que es la estructura que organiza las declaraciones y llamadas a ciertos elementos que se usan en la codificación de un lenguaje.

REFERENCIAS

- 1.- Capítulo 5. Análisis semántico. UAM. Sitio web:
<http://arantxa.ii.uam.es/~alfonsec/docs/compila5.htm>