

Instituto Tecnológico de Costa Rica

Análisis contextual

Curso:

Compiladores e Intérpretes

Estudiantes:

Andrés Fernández Calderón

Jurguen Romero Rodríguez

06 de Mayo del 2018

Introducción

El proyecto consiste en realizar la fase de análisis contextual. Esta fase consiste en validaciones para la existencia de métodos invocados, igualdades de tipos en las expresiones. La etapa de este proyecto requiere la construcción del visitor, proporcionado por la herramienta ANTLR4, con el fin de elaborar la inspección de tipos y alcances de los identificadores del lenguaje. El proyecto contendrá una tabla de identificadores la cual servirá como ayuda para llevar la información relevante de cada identificador.

Análisis del problema

El proyecto pretende realizar un análisis contextual para un lenguaje a partir de la gramática Monkey. El lenguaje tiene una estructura de bloque anidados por lo cual el uso de una tabla para guardar la información como el tipo de dato (INTEGER, STRING...) y los niveles de los identificadores es necesaria. esta tabla debe contener los métodos pre-establecidos que el lenguaje dicta los cuales son LEN, FIRST, LAST, REST y PUSH.

El uso de la tabla permitirá el chequeo de tipos de los datos. Los cuales se deben validar mediante las funciones del visitor. El lenguaje permite 4 tipos de operadores aritméticos multiplicación(*), resta(-), división(/) y suma(+). El lenguaje también contiene operadores relacionales los cuales son: <, <=, ==, >= y >. Los cuales solo acepta números enteros, a excepción del == que puede aceptar operadores de cualquier tipo básico como booleano, String e Integer.

Soluciones

La solución del proyecto para el análisis contextual del lenguaje Monkey se da mediante la realización del visitor que se genera gracias a la herramienta antlr.runtime-4.7.1 de Java. Las funciones generadas en el visitor serán editadas para poder hacer las inserciones a la tabla y los chequeos de tipos. En estos se planea realizar un checker el cual a partir de los visitor se irá moviendo por el árbol y retornando los valores Integer, ArrayList, String, Hash... los cuales se insertarán en la tabla.

Se generará una tabla especial para las funciones, esta solo guardará el nombre de la función, el tipo de retorno, parámetros y el context con el cual fue creado.

Para cada tipo se diferencia por un valor especial ejemplo:

Tipo de dato	Valor que lo identifica
Integer	2
String	3
Booleanos	1
ArrayLiteral	5
Neutro	0
Error	-1
Identificador	4

A partir de estos tipos se busca en la tabla el tipo y se compara para al hacer algún tipo de operación aritmética o alguna operación de relaciones.

En el caso de las funciones la tabla funciones guardará el nombre, los parámetros, el context y el tipo de retorno de la función. Esto permitirá comparar a la hora de llamar las funciones y evaluar si los parámetros son correctos. De ser los parámetros y el nombre incorrecto regresará un error especificando lo que ha ocurrido. Existen funciones por defecto como len, first, last, push y rest.

Los ArrayList es un estilo Array y permitirán guardar cualquier tipo de datos dentro de ellos así que no será necesario un chequeo de tipos dentro de estas.

El hash es estilo un diccionario o Json, este permite guardar como llave un tipo Integer o identificadores. El valor puede ser cualquier tipo de dato.

En el caso de los returns dentro de las funciones retornan valores neutros. Y no será posible realizar un return si no está dentro de una función.

Para validar los identificadores se debe saber qué estas no se pueden declarar dos veces dentro del mismo ámbito es decir si se declara a global se podrá volver a declarar a dentro de una función ya que esta posee un ámbito distinto.

Sí los identificadores no se han declarados con let no podrán ser utilizados.

Las indentaciones con elementAccess tanto de arreglos/listas como de hashLiterals, serán con valores enteros solamente (o neutros según sea el caso).

El if contendrá como expresión a solo aquellas que regresan valores booleanos.

Y por último al declarar una función es posible declarar funciones dentro de un arreglo, sin que esto sea un error.

Pruebas.

Se genera un archivo adjunto txt en el cual vendrán las pruebas realizadas para el proyecto.

Resultados obtenidos.

Se presenta la tabla con las tareas realizadas y el porcentaje al cual se encuentra cada una.

Tarea	Porcentaje %	Comentario
Tabla de símbolos	100	
Chequeo de alcances	80	
Chequeos de tipos	80	
Documentación	100	

Conclusiones del trabajo.

ANTLR4_complete es una herramienta muy simple a la hora de crear compiladores, esta fue de gran ayuda para la generación del visitor y a partir del visitor crear el checker que permite la validación de los datos del proyecto.

El análisis contextual para el interpreter creado a partir de la gramática Monkey es una manera de ayuda para entender los movimientos que realiza el compilador. A la hora de validación de tipos la ayuda del árbol proporcionado por la herramienta ANTLR4_complete también llegó a ser de ayuda.