

Instituto Tecnológico de Costa Rica
Escuela de Computación

Ingeniería en computación

Compiladores e Intérpretes
Grupo 2

Profesor: Erika Marín

Análisis Semántico y Generación de código

Josué Arrieta Salas, 2014008153
Adrián López Quesada, 2014081634

Cartago

Miércoles 22 de junio

Índice

Análisis de resultados	3
¿Cómo compilar y correr el compilador de MYPY?	5

Análisis de resultados

Objetivo	Porcentaje de éxito (100%)
Programa recibe código fuente escrito en Python y analiza el archivo.	100%
Se listan errores léxicos encontrados (por línea y el error).	100%: objetivo del proyecto pasado.
El programa ante un error léxico se recupera de este y no despliega errores en cascada.	100%: objetivo del proyecto pasado.
Se listan errores sintácticos encontrados (por línea y el error).	100%: objetivo del proyecto pasado.
El programa ante un error sintáctico se recupera de este y no despliega errores en cascada.	100%: objetivo del proyecto pasado, con mensajes muy significativos. También recupera en gran medida; sin embargo se ha de mencionar que en algunos casos es simplemente imposible recuperarse del error y continuar parseando.
Ante un error sintáctico, el análisis semántico se detiene.	100%
Se listan los errores semánticos encontrados por línea.	100%: con errores de mensaje específicos. Ante un error semántico se siga con el análisis semántico, sintáctico y léxico. No se despliegan errores en cascada ni se detiene.
Análisis semántico: variables no definidas.	100%
Análisis semántico: variables doblemente definidas.	100%
Análisis semántico: funciones no definidas	100%
Análisis semántico: corroboración de tipos en asignaciones y tipo de operadores con los operadores	
Análisis semántico: corroboración de cantidad y tipo de parámetros.	100%
Análisis semántico: break y continue dentro de los bloques permitidos.	100%
Desplegar los contenidos de la tabla de símbolos una vez concluido el análisis.	100%: se incluye nombre, tipo y ámbito de variables globales, funciones, variables locales y parámetros.
Se genera un archivo con código ensamblador (traducción de código).	100%: se ha de mencionar que para la generación de código y análisis semántico se creó una pila semántica y tabla de símbolos para realizar Traducción dirigida por sintaxis.
Ante un error semántico o sintáctico, la traducción de código se detiene.	100%
Traducción de código: declaración de variables	100%

globales.	
Traducción de código: declaración de funciones	100%
Traducción de código: expresiones binarias aritméticas.	100%: se generan donde estén, ya sea el programa principal o dentro de una función.
Traducción de código: if-else	100%: se genera donde estén, ya sea en el programa principal o dentro de una función.

¿Cómo compilar y correr el compilador de MYPY?

Debido a problemas para enviar el archivo por Gmail, se tuvieron que modificar las extensiones .jar a .jara; por lo tanto se debe eliminar esta “a” extra en los siguientes archivos:

Si se desea correr el ejecutable:

- Compilador-mypy\Ejecutable\Compilador.jara
- Compilador-mypy\Ejecutable\lib\ java-cup-11b.jara
- Compilador-mypy\Ejecutable\lib\ jflex-1.6.1.jara

Si se desea compilar el proyecto:

- Compilador-mypy\CodigoFuente\Tools\ java-cup-11b.jara
- Compilador-mypy\CodigoFuente\Tools\ jflex-1.6.1.jara

Para correr el parser se debe usar el archivo Prueba.mypy en la ubicación:

- Compilador-mypy\Ejecutable\
- Compilador-mypy\CodigoFuente\

Según se ejecute el ejecutable o se quiera recompilar el proyecto, de manera respectiva.

Los archivos con el código ensamblador “Prueba.asm” se encuentran según el ejecutable o recompilación del proyecto en:

- Compilador-mypy\Ejecutable\
- Compilador-mypy\CodigoFuente\

Notas extras: Se desarrolló el proyecto en el IDE Netbeans y se utilizaron las librerías Java Cup y JFlex para su desarrollo. Se implementó una interfaz gráfica para facilitar la ejecución del ejecutable, ya que en consola solo se puede correr desde línea de comando. La gramática quedó de manera extensa debido a problemas encontrados con la recuperación de errores de la librería Java Cup., por lo que se intentó ser lo más específico posible. El código ensamblador que se genera es para TASM 1.4.