

Proyecto #2

Análisis Semántico y Generación Código Intermedio

Introducción

Un grupo de desarrolladores desea crear un nuevo lenguaje imperativo, ligero, que le permita realizar operaciones básicas para la configuración de chips, ya que esta es una industria que sigue creciendo constantemente, y cada vez estos chips necesitan ser configurados por lenguajes más ligeros y potentes. Es por esto que este grupo de desarrolladores requiere desarrollar su propio lenguaje para el desarrollo de sistemas empujados.

Proyecto a desarrollar

Este proyecto comprende la fase de análisis semántico y la generación de código intermedio para la gramática descrita en el Proyecto I, el Lexer y Parser generado. Se debe desarrollar el Analizador Semántico y el Generador de Código Intermedio.

Un programa escrito para este lenguaje está compuesto por una secuencia de declaraciones de funciones, que contienen diferentes expresiones; todo programa debe contener exactamente un método main.

Para comprobar el desarrollo de ambas fases del programa a presentar deberá tomar un archivo fuente y realizar lo siguiente:

- a) Preservar y corregir los alcances del Proyecto I
- b) Indicar si el archivo fuente puede o no ser generado por la gramática. Tomando en cuenta la gramática, sintaxis y **Semántica (tipado explícito y fuerte)**.
- c) Manejar los errores léxicos, sintácticos y semánticos encontrados. Debe utilizar la técnica de Recuperación en Modo Pánico.
- d) Escribir en un archivo el código intermedio para el archivo fuente con el mismo sentido semántico.

Gramática, Scanner y Parser

La gramática BNF que reconocerá el lenguaje será la descrita en el Proyecto I. Deberá tomarse como base para el Proyecto II los archivos jflex y cup del proyecto generado en el Proyecto I.

Analizador Semántico

El desarrollo del Analizador Semántico se puede realizar utilizando la herramienta Cup. Incluye la comprobación semántica y recuperación de errores. Debe reportar los errores y seguir procesando el código fuente. Tiene un valor de 47.5 puntos

Generación de Código Intermedio

El desarrollo del Generador de Código Intermedio debe escribir en un archivo el código intermedio (tres direcciones) para el archivo fuente. Tiene un valor de 32.5 puntos.

Puntos Extra

Se darán 3 puntos extra al implementar técnicas de recuperación de errores del tipo Recuperación a nivel de frase, Producción de Errores o Corrección Global.

Se darán 10 puntos si incluye el manejo de clases –estructura y uso- (deberá realizar análisis léxico, sintáctico, semántico y generación de código intermedio). Además, estará condicionado alcanzar un 90% de los requerimientos funcionales del proyecto.

Se darán **2.5 puntos adicionales** al entregar a más tardar el miércoles 29 de octubre a las 11:55:55 PM el Documento de Requerimientos, ver plantilla suministrada en el Tec Digital. Debe subirse en la documentación llamada “Proyecto II (archivos adicionales)” debajo de la carpeta de “Proyectos”.

Aspectos técnicos

El proyecto deberá correr en java y utilizar las herramientas JFlex y Cup (Utilizadas en el Proyecto I). En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación. Se debe incluir el archivo Flex y Cup, y el proyecto en Java que utilice los archivos generados por JFlex y Cup.

Deberán utilizar el sistema de control de versiones GitHub, el repositorio deberá ser público o incluir al profesor en el control de acceso de este.

Se valorará el aporte generado por cada estudiante, considerando, entre otras cosas, los commit generados por cada uno. Por lo que el puntaje obtenido por cada uno de los estudiantes puede ser diferenciado.

Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada parte, con sus entradas, salidas, restricciones y **objetivo**.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema.
5. Diseño del programa: decisiones de diseño, algoritmos usados.
6. Librerías usadas: creación de archivos, etc.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. Justificación de toma de decisiones: explicar la forma en que se realizó la validación semántica y sus estructuras.
9. Bitácora (autogenerada en git, commit por usuario incluyendo comentario).

Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación.

El proyecto programado tiene un valor de **20%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

1. Documentación interna 2 ptos.
2. Documentación externa 8 ptos.
3. Funcionalidad 80 ptos (ver detalle en Proyecto a Desarrollar)
4. Revisión del proyecto (gestión del tiempo) 5 ptos.
5. Hora de Entrega 5 ptos.

Forma de trabajo

El trabajo se debe realizar en parejas.

Aspectos administrativos

Debe crear un archivo **.zip** ("PP2_Integrante1_Integrante2.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **programa**, en la primera deberá incluir el documento de *word* o pdf solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de este proyecto programado. El archivo **info.txt** debe contener la siguiente información (cualidades):

- a. Nombre del curso
- b. Número de semestre y año lectivo
- c. Nombre de los Estudiantes
- d. Número de carnet de los estudiantes
- e. Número de proyecto programado
- f. Fecha de entrega
- g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de COMPILADORES E INTERPRETES GR 60, en la asignación llamada "P2" debajo del rubro de "Proyectos". En la evaluación del Proyecto el rubro de "Hora de Entrega" valdrá por 5 puntos de la nota total del proyecto, según la siguiente escala:

- a. Si se entrega antes de las 11:55:55 **PM** del lunes 10 de noviembre de 2025, 5 puntos.
- b. Si se entrega antes de las 11:55:55 **AM** del martes 11 de noviembre de 2025, 2.5 puntos.
- c. Si se entrega antes de las 11:55:55 **PM** del martes 11 de noviembre de 2025, 0 puntos.

Después de este punto, **NO SE ACEPTARÁN** más trabajos.

Dentro de la carpeta de **programa**, deberá incluir un archivo **.txt**

(**PrimerNombreMiembro1.PrimerNombreMiembro2.txt**) que contenga todo el texto de la solución del o de los archivos **.flex** y **.cup** presentados. Estos archivos pueden ser revisados en el sistema de Control de Plagio del TEC Digital.

Todo el contenido de cada proyecto debe ser 100% original y en caso de plagio todos los integrantes del grupo tendrán nota cero. Todos los miembros del grupo deberán participar en el desarrollo del proyecto y en la revisión, ya que de lo contrario se les penalizará con puntos.