

## Fase 3 del Proyecto de Compiladores – Análisis Semántico

Materia: Compiladores II

Docente: Dra. Blanca Guadalupe Estrada Rentería

Semestre: 9.<sup>º</sup> ISC

Fechas de entrega: 30 y 31 de octubre

### Objetivo General

El objetivo de esta fase es diseñar e implementar el analizador semántico del compilador, tomando como base el árbol sintáctico abstracto (AST) generado en la Fase 2. El analizador debe asignar, propagar y verificar atributos de tipo y de valor, construir la tabla de símbolos y detectar errores semánticos, tales como variables no declaradas, tipos incompatibles y conversiones no válidas.

### Gramática Base Extendida

La gramática de la Fase 2 debe extenderse con atributos semánticos heredados y sintetizados. Ejemplo:

```
declaracion_variable → tipo id ;
    tipo.t = tipo
    id.tipo = tipo.t
    insertarTabla(id.lexema, id.tipo)
```

```
asignacion → id = expresion ;
    si no declarado(id.lexema) → error('Variable no declarada')
    si tipo(id) ≠ tipo(expresion) → error('Tipos incompatibles')
```

### Requisitos Técnicos

- Árbol Sintáctico y Árbol Anotado: Tomar el AST de la Fase 2 y añadir anotaciones semánticas que incluyan atributos de tipo y valor. Mostrar cómo se propagan los atributos entre los nodos.
- Tabla de Símbolos: Registrar todos los identificadores del programa con nombre, tipo de dato, ámbito y dirección o desplazamiento de memoria.
- Verificación e Inferencia de Tipos: Implementar las reglas de equivalencia y promoción de tipos, verificando compatibilidad entre operandos y resultados.
- Manejo de Errores Semánticos: Detectar y reportar variables no declaradas, duplicidad de identificadores, incompatibilidad o conversión de tipos inválida, y uso indebido de operadores.

## **Entregables**

- Código fuente del analizador semántico.
- Árbol sintáctico base y árbol con anotaciones semánticas.
- Tabla de símbolos completa en formato de tabla o archivo de texto.
- Archivo de errores semánticos detectados.
- Documento PDF con explicación técnica, gramática extendida con atributos, ejemplos y capturas de ejecución.
- Block de notas con nombres y matrículas de los integrantes.

## **Validación Esperada**

El analizador semántico deberá validarse utilizando el archivo de prueba TestSemantica.txt, disponible en la plataforma. Dicho archivo contiene casos representativos del lenguaje que permiten comprobar la correcta construcción de la tabla de símbolos, la evaluación de atributos y la detección de errores semánticos.

El programa deberá:

- Leer el archivo TestSemantica.txt y procesar los tokens generados previamente.
- Construir el árbol sintáctico base y el árbol con anotaciones semánticas.
- Mostrar la tabla de símbolos
- Reporte de errores semánticos.

El análisis será exitoso si:

1. Se detectan correctamente los errores incluidos en el archivo de prueba.
2. Los programas válidos muestran resultados coherentes en tipos y valores.
3. El analizador permite continuar la ejecución tras errores no fatales.

## **Nota Importante sobre la Entrega y Penalización**

- Calificación máxima de 10 si se entrega el día asignado (30 o 31 de octubre).
- Por cada día de retraso se descuenta 1 punto.
- La entrega es obligatoria aunque la calificación llegue a cero.
- El proyecto debe estar 100% funcional para ser evaluado.