

Informe: Análisis Semantico

Yampaul Chaux

Juan Marulanda

Kevin Niño

Faculta de Ingeniería, Corporación Universitaria Remington

Sede, Tuluá

Compiladores

Ing. Jorge Leonardo Gamarra

22 de abril del 2025

Introducción

El análisis semántico es la fase del compilador encargada de verificar el significado del programa, asegurando que las construcciones sintácticamente correctas también sean válidas desde el punto de vista de tipos y contexto.

Tabla de Símbolos

La tabla de símbolos se implementa como un diccionario en Python, mapeando identificadores a sus tipos de datos. Ejemplo de estructura:

```
symbol_table = {  
    'x': 'int',  
    'y': 'float'  
}
```

Reglas de Validación

Se definen las siguientes reglas semánticas:

- Declaración previa: Una variable debe declararse antes de usarse.
- Compatibilidad de tipos: No se puede asignar un valor `float` a una variable `int`.
- Reduplicación: No se permite re declarar una variable con el mismo nombre en el mismo ámbito.

Manejo de Errores Semánticos

Los errores semánticos se detectan y reportan con mensajes claros:

- Error: variable 'x' no declarada

- Error: no se puede asignar float a int en 'y'
- Error: variable 'z' ya declarada

Casos de Prueba

A continuación, algunos ejemplos de casos de prueba:

- Caso válido:

```
int a = 10;
```

```
float b = a + 2.5;
```

```
print(b);
```

Salida:

- Declaración válida: a de tipo int
- Declaración válida: b de tipo float
- Impresión válida: b

- Caso inválido (no declarada):

```
print(c);
```

Salida:

- Error: variable 'c' no declarada

Conclusiones

El análisis semántico asegura la coherencia interna del programa al verificar tipos y declaraciones, evitando errores en tiempo de ejecución e incrementando la fiabilidad del compilador.