Tecnológico de Monterrey

Juan Pablo Cruz Rodriguez | A01783208

Analizador Semántico

Desarrollo de aplicaciones avanzadas ciencias computacionales

Reglas de Inferencia

Operaciones Aritméticas

Para operadores aritméticos: +, -, *, /

$\vdash E_1 : int \quad \vdash E_2 : int$

$\vdash E_1 \ op \ E_2 : int$

$(op \in \{+, -, *, /\})$

Operaciones Lógicas

Para operadores relacionales: ==, !=, <, <=, >, >=

$\vdash E_1 : int \quad \vdash E_2 : int$

$\vdash E_1 \ opl \ E_2 : int$

$(opl \in \{==, !=, <, <=, >, >=\})$

Nota: El resultado es tipo int porque en C- no existe bool; 0 representa falso.

Asignación de Variables

$\vdash a : int \quad \vdash E : int$

$\vdash a = E : int$

Acceso a Arreglo

$\vdash a : int[] \quad \vdash i : int$

$\vdash a[i] : int$

Llamada a Función

$\vdash f : (\tau_1, ..., \tau\square) \rightarrow \tau$

$\vdash e_1 : \tau_1 \quad ... \quad \vdash e\square : \tau\square$

$\vdash f(e_1, ..., e\square) : \tau$

Declaración de Variables

Si a no está en el ámbito actual, entonces: $\Gamma \cup \{a : \tau\}$

Errores Semánticos

• Usar variable no declarada:

$\Gamma \nvdash a$

• Redefinir función o variable en el mismo ámbito:

$\Gamma \vdash a \Rightarrow$ Error de redefinición

Estructura de Tabla de Símbolos

La tabla de símbolos se implementa como una lista de objetos Symbol, donde cada símbolo representa un identificador del programa. Cada símbolo está representado de la siguiente manera:

Campo	Descripción
name	Nombre del identificador
var_type	Tipo de símbolo (variable, funcion, parametro)
type	Tipo de dato (int, void, etc.)
scope	Ámbito en el que fue declarado

Búsqueda en la tabla

El método find_symbol(name, scope) busca un identificador en el ámbito actual. Si no se encuentra, se permite la búsqueda en el ámbito global.

Esto permite:

- Uso de variables locales o globales según el contexto
- Llamadas a funciones globales desde cualquier punto

Stack

Aunque no se implementa una estructura explícita de **stack de símbolos por bloque**, el campo scope permite agrupar los identificadores por función o bloque. Esto **simula el comportamiento de un stack** sin necesidad de estructuras adicionales.

Por ejemplo:

- Todos los símbolos con scope = mínimo pertenecen a esa función.

- Al entrar a una función, el analizador actualiza current_scope para que las nuevas declaraciones se registren correctamente.