

### Proyecto 3 Analizador Semántico

#### Reglas de inferencia de tipos

$$1. \frac{i \text{ es una literal entera}}{\vdash i:int}$$

$$2. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 + e_2:int}$$

$$3. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 - e_2:int}$$

$$4. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 * e_2:int}$$

$$5. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1/e_2:int}$$

$$6. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 < e_2:int}$$

$$7. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 \leq e_2:int}$$

$$8. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 > e_2:int}$$

$$9. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 \geq e_2:int}$$

$$10. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1 == e_2:int}$$

$$11. \frac{\vdash e_1:int \quad \vdash e_2:int}{\vdash e_1! = e_2:int}$$

$$12. \frac{\vdash e_1:int \quad e_1 \text{ tiene parametros}}{\vdash e_1:int \text{ funcion}}$$

$$13. \frac{\vdash e_1:void \quad e_1 \text{ no tiene parametros}}{\vdash e_1:int \text{ variable}}$$

$$14. \frac{\vdash e_1:void \quad e_1 \text{ tiene parametros}}{\vdash e_1:void \text{ funcion}}$$

Nota: Para determinar si un nombre de tipo 'int' de la tabla de símbolos es función o variable se revisa el diccionario de sus propiedades, si tiene una entrada llamada 'params' se determina que es función, de lo contrario se determina que es variable.

## Estructura de tabla de símbolos

Las tablas de símbolos se almacenan en estructura de árbol en donde la raíz del árbol representa el scope global, cada tabla de símbolos puede tener de 0 a varios hijos.

Se entra a un nuevo scope en C- al entrar a un *"compound statement"* (declaración encerrada por llaves {} ).

Cada uno de los nombres introducidos en las tablas de símbolos se relaciona con dos propiedades:

1. Tipo : int, int[] o void

2. Diccionario: propiedades adicionales de la variable o función.

Ejemplos:

{}	-	Diccionario vacío utilizado en int
{'size' : 10}	-	Propiedad size utilizada en arreglos
{'params': void}	-	Propiedad utilizada en funciones sin parámetros
{'params': ['int[]', 'int']}	-	Propiedad utilizada en funciones con parámetros

A continuación se muestra el ejemplo de un programa en C- junto con su respectiva tabla de símbolos:

## Programa

---

```
int var;
int arr[12];

int funcionuno(int paramuno, int paramdos[])
{
    {
        int a;
    }

    if(paramuno)
    {
        int b;
    }
    else
    {
        int b;
    }
    return paramuno;
}

void main(void)
{
    int varlocal;
    int arrlocal[10];

    while(varlocal)
    {
        int a;
        if(a)
        {
            int b;
        }
    }
}
```

Tablas de símbolos

