

Universidad del Valle de Guatemala

Construcción de compiladores

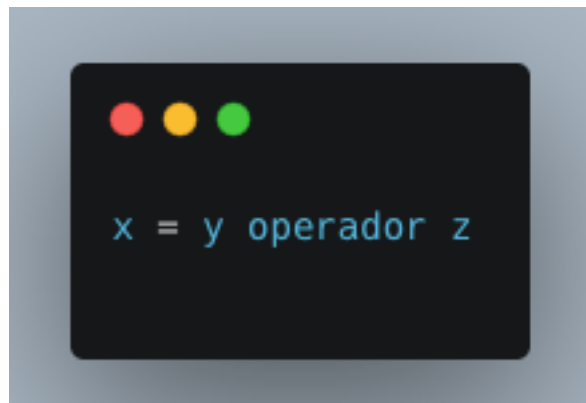
2023

Fredy Velásquez

Jorge Caballeros

Diseño de Código Intermedio

El diseño del código de tres direcciones es una forma esencial en la representación intermedia dentro de los compiladores. Su propósito principal es facilitar la evaluación de operaciones aritméticas y se estructura de la siguiente manera:



Tipos de Propositiones en Código de Tres Direcciones

Las proposiciones que pueden manifestarse en nuestro diseño de código de tres direcciones son:

1. Asignaciones Binarias
 - Ejemplo: $a = b \text{ operador } c$
2. Asignaciones Unarias
 - Ejemplo: $a = -b$
3. Instrucciones de Copia por Valor
 - Ejemplo: $a = b$
4. Instrucciones de Copia por Referencia
 - Ejemplo 1: $a = c[b]$
 - Ejemplo 2: $c[b] = a$
5. Saltos Incondicionales

- Ejemplo: goto B1

6. Saltos Condicionales

- Ejemplos:
 - if(a) goto B1
 - if(a < b) goto B1
 - if(a < b or c > d) goto B1
- Nota: Este esquema también se aplica para estructuras de control como bucles while y for.

7. Llamadas a Funciones

- Ejemplo 1: call función(a, b, ...n)
- Ejemplo 2: $a = \text{función}(a, b, \dots n)$

Estructura de Datos Utilizada

Generalmente, se prefiere usar la estructura denominada "cuarteto" para la implementación del código de tres direcciones. Sin embargo, debido a su mayor ocupación de espacio y la necesidad de numerosas variables temporales para cálculos intermedios, utilizamos un "terceto".

Mientras que el cuarteto tiene una estructura $\backslash (op, a, b, res) \backslash$ que consta de:

- op: operador
- a: primer argumento
- b: segundo argumento
- res: resultado

El terceto se simplifica a (op, a, b) . En esta estructura, el resultado es implícito y los tercetos referencian otros tercetos donde se encuentra el resultado requerido.

Veamos un ejemplo para una expresión $x = y + z - (a * b)$:

No.	Trans.	Tercet
1	temp1 = a * b	(*, a, b)
2	temp2 = z - temp1	(-, z, (1))
3	temp3 = y + temp2	(+, y, (2))
4	x = temp3	(=, (3), x)

Tamaños para Tipos de Datos Básicos

Tipo	Size (bytes)
Int	8
Bool	1
String	1 per carácter

Este diseño de código intermedio busca ser una representación más eficiente y legible de las operaciones aritméticas, proporcionando una estructura clara y concisa para su interpretación y posterior traducción en lenguaje de máquina.