

# Compiladores

## CS3025

*Julio Eduardo Yarasca Moscol*

- Unidad 5: Generación de código
  - 5.1. Llamadas a procedimientos y métodos en envío.
  - 5.2. Compilación separada; vinculación. Selección de instrucciones.
  - 5.3. Asignación de registros. Optimización por rendija (peephole)

## Logro de la sesión:

Discutir por qué la compilación separada limita la optimización debido a efectos de llamadas desconocidas. Discutir oportunidades para la optimización introducida por la traducción y enfoques para alcanzar la optimización, tales como la selección de la instrucción, planificación de instrucción, asignación de registros y optimización de tipo mirilla (peephole optimization).



## Etiquetado de nodos

- El etiquetado de nodos, en el contexto de la optimización de código dentro de compiladores, es el proceso mediante el cual se asignan etiquetas o valores numéricos a los nodos de estructuras intermedias como árboles de sintaxis o grafos acíclicos dirigidos (DAG). Este etiquetado facilita la toma de decisiones durante la generación de código optimizado, ya que permite identificar la mejor secuencia de evaluación, minimizar el uso de registros o detectar subexpresiones reutilizables.



## Etiquetado de nodos

### ► Caso base – Nodo hoja:

- Si  $n$  es el hijo más a la izquierda de su padre, entonces  $\text{etiqueta}(n) = 1$ .
- En caso contrario,  $\text{etiqueta}(n) = 0$ .

### ► Caso recursivo – Nodo interior:

- Sean  $n_1, n_2, \dots, n_k$  los hijos del nodo  $n$ , ordenados en forma no creciente según sus etiquetas:

$$\text{etiqueta}(n_1) \geq \text{etiqueta}(n_2) \geq \dots \geq \text{etiqueta}(n_k)$$

- Entonces, la etiqueta del nodo  $n$  se define como:

$$\text{etiqueta}(n) = \max_{1 \leq i \leq k} (\text{etiqueta}(n_i) + i - 1)$$



## Etiquetado de nodos

- Si  $n$  es un nodo binario, la fórmula para calcular su etiqueta a partir de las etiquetas de sus hijos se puede simplificar de la siguiente manera:

$$\text{etiqueta}(n) = \begin{cases} \text{máx}(l_1, l_2), & \text{si } l_1 \neq l_2 \\ l_1 + 1, & \text{si } l_1 = l_2 \end{cases}$$

donde  $l_1$  y  $l_2$  representan las etiquetas de los hijos izquierdo y derecho de  $n$ , respectivamente.



## Algoritmo de Sethi-Ullman

- ▶ Es un algoritmo que asigna etiquetas a los nodos de un árbol de expresión para indicar el número mínimo de registros necesarios para evaluar la subexpresión correspondiente a ese nodo. Estas etiquetas luego guían la generación de código de forma que se usen la menor cantidad de registros y se evite almacenar resultados intermedios en memoria (stack).



## Algorithm 1 Algoritmo Sethi-Ullman - Adaptación

```

1: procedure GENERARCODIGO(exp)
2:   if  $r = 0$  then
3:     GenerarCodigo(exp.left)
4:     imprimir("movq %rax, %rcx")
5:     GenerarCodigo(exp.right)
6:   else if  $l \geq r$  then
7:     GenerarCodigo(exp.left)
8:     imprimir("pushq %rax")
9:     GenerarCodigo(exp.right)
10:    imprimir("movq %rax, %rcx")
11:    imprimir("popq %rax")
12:   else
13:     GenerarCodigo(exp.right)
14:     imprimir("pushq %rax")
15:     GenerarCodigo(exp.left)
16:     imprimir("movq %rax, %rcx")
17:     imprimir("popq %rax")
18:     imprimir("xchgq %rax, %rcx")
19:   end if
20:   imprimir(".op %rcx, %rax")
21: end procedure

```

Dadas las sentencias:

- ▶  $a + (b * c)$
- ▶  $(a + b) + (c * d)$
- ▶  $(a * b) + [(a + b) + (c * d)]$

1. Elabore el GDA
2. Utilice el algoritmo de etiquetado de nodos (si es posible para optimizar el código)



## Optimización por Mirilla

- ▶ La generación de código instrucción por instrucción suele producir código objeto con instrucciones redundantes y construcciones ineficientes.
- ▶ La calidad de este código puede mejorarse mediante la aplicación de transformaciones optimizadoras directamente sobre el programa objeto.
- ▶ Una técnica simple pero efectiva para mejorar localmente el código generado es la **optimización por mirilla**, que consiste en examinar breves secuencias de instrucciones objeto y reemplazarlas por versiones equivalentes más cortas o más eficientes, cuando sea posible.



# Preguntas y cierre de la sesión

- ▶ ¿En qué consiste la optimización mirilla?

# GRACIAS

*JULIO EDUARDO YARASCA MOSCOL*

