

	<b>Guía No. 7</b>	Código Intermedio (Segunda Parte).	Departamento de informática.
	Universidad Don Bosco <b>Facultad de Ingeniería</b> Es cuela de Computación		
	Asignatura	: Compiladores	
	Ciclo	: 01 – 2004	
	Lugar de ejecución:	: Centro de Computo	

## I. OBJETIVOS

- ☐ Aplicación de Notación Polaca Inversa.
- ☐ Construir programa para generación de código intermedio, aplicando conceptos de las instrucciones, if, if - else, while.

## II. INTRODUCCIÓN

### Tripletes

No se pone el resultado, se sustituye por referencias a tripletes. Por ejemplo: la expresión  $a*b+c*d$  equivale a:

- (1) (\*,a,b)
- (2) (\*,c,d)
- (3) (+,(1),(2))

mientras que  $a*b+1$  equivale a:

- (1) (\*,a,b)
- (2) (\*,(1),1)

**Tripletes indirectos:** se numeran arbitrariamente los tripletes y se da el orden de ejecución. Ejemplo, sean las instrucciones:

$a = b*c$   
 $b = b*c$

Equivalen a los tripletes

- (1) (\*,b,c)
- (2) (=,(1),a)
- (3) (=,(1),b)

y el orden de ejecución es (1),(2),(1),(3). Esta forma es útil para preparar la optimización de código. Si hay que alterar el orden de las operaciones o eliminar alguna, es más fácil hacerlo ahí.

## NOTACIÓN POLACA / NOTACIÓN POLACA INVERSA

**Notación prefija (polaca):** Se escribe primero el nombre de la función seguida de los operandos de izquierda a derecha. Si un operando es a su vez operación con operandos, se aplican las mismas reglas. Por ejemplo:

$(a+b)*(c-a) == +ab -ca$

No hay ambigüedad alguna ni se necesitan paréntesis para saber como evaluar la expresión.

Debido a que el matemático polaco Lucasiewicz inventó la notación libre de paréntesis, se ha aplicado el término de polaca a esta notación y sus derivados.

**Notación postfija (o polaca inversa):** Es similar a la notación prefija, excepto que el símbolo de operación sigue a la lista de operándos. Por ejemplo:

$a \ b \ + \ c \ a \ - \ *$

**Notación infija:** Es la más adecuada para operaciones binarias. El operador se escribe entre los operandos. Por su uso natural, ha sido adoptada por muchos lenguajes para operaciones aritméticas y lógicas. Cuando la operación no es binaria, se usa de manera algo torpe empleando varios delimitadores; por ejemplo el comando condicional en C:

(?S1:S2)

### III. PROCEDIMIENTO

**Segunda parte de Código intermedio**, haga uso de los conceptos vistos en la clase con las diferentes instrucciones, if, if - else, while.

#### TRIPLETAS INDIRECTAS.

1. Tripletas indirectas. Solucionan el problema de la reordenación mediante indirecciones.

2. Ejemplo:

- Fuente:

$A = B + C * D / E$

$F = C * D$

- OPERACIONES

- |        |               |
|--------|---------------|
| 1. (1) | (1) *, C, D   |
| 2. (2) | (2) /, (1), E |
| 3. (3) | (3) +, B, (2) |
| 4. (4) | (4) =, A, (3) |
| 5. (1) | (5) =, F, (1) |
| 6. (5) |               |

#### **Ejercicios:**

- a) De acuerdo al ejemplo anterior realice un árbol de las operaciones realizadas, haciendo uso de notación polaca inversa (postfija).
- b) Luego diseñe un programa en lenguaje C o C++, para realizar el código intermedio de ambas expresiones aritméticas, haciendo uso de las instrucciones if, if-else, while y for. (Es un programa con cada uno de ellos).

### IV. INVESTIGACIÓN

- Indique cuáles son los pro y los contra de la notación polaca inversa.
- Investigue que es un árbol en CoSeL y para que sirve.

### V. BIBLIOGRAFÍA

<http://homepage.mac.com/eravila/Stop/stop02.html>

<http://www.ii.uam.es/~alfonsec/docs/compila6.htm>

[http://www.spsu.edu/cs/faculty/bbrown/web\\_lectures/postfix/](http://www.spsu.edu/cs/faculty/bbrown/web_lectures/postfix/)