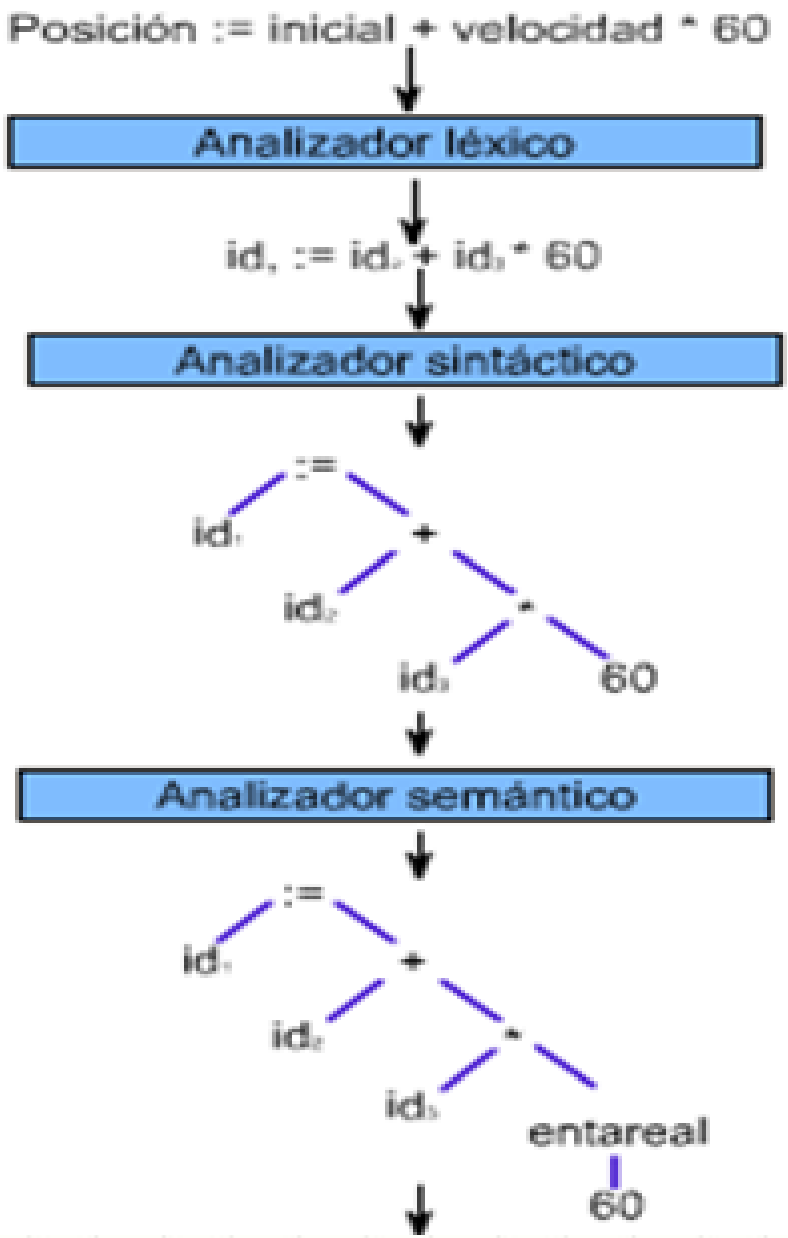
COMPILADORES
TEMA 5. GENERADORES DE CÓDIGO

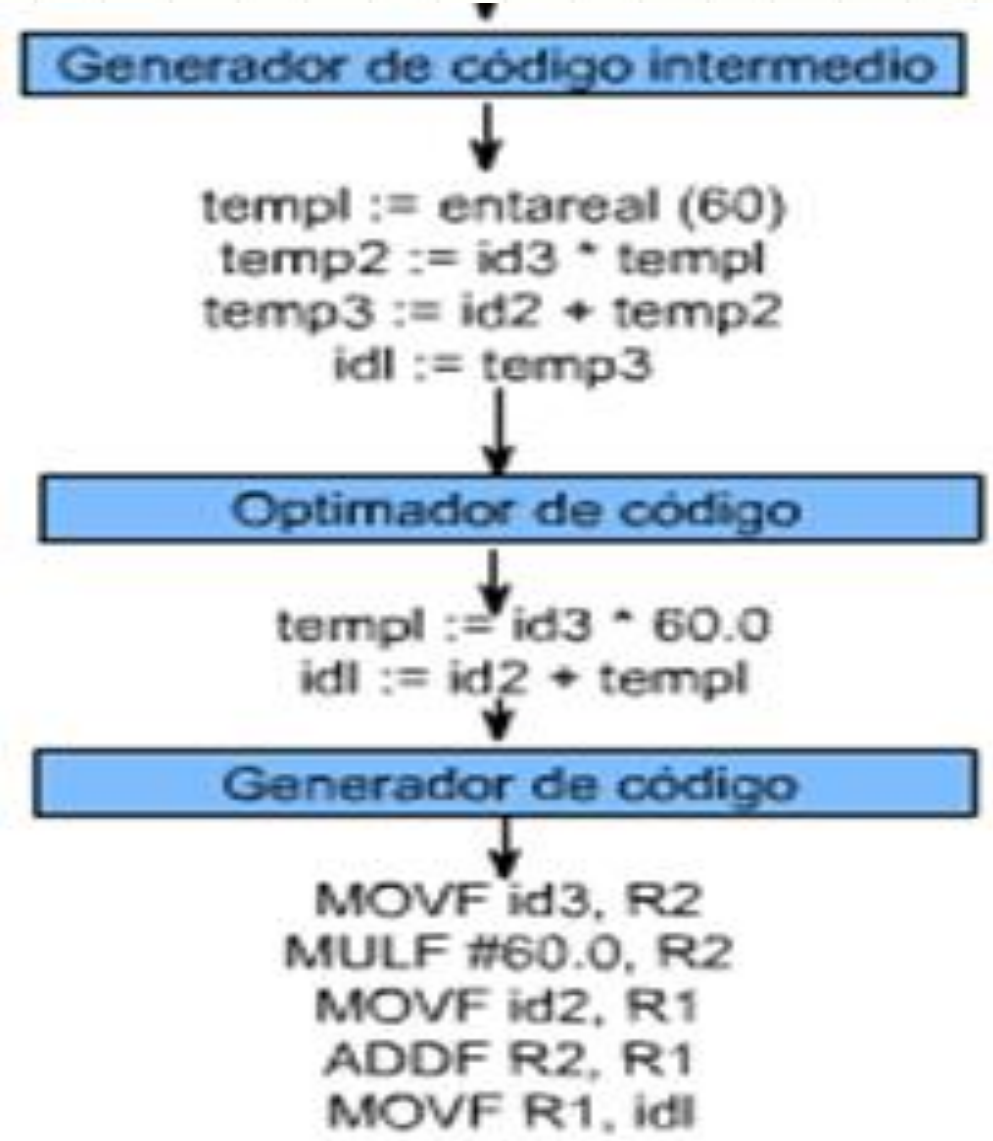
Presenta: Mtro. David Martínez Torres
Universidad Tecnológica de la Mixteca
Instituto de Computación
Oficina No. 37
dtorres@gs.utm.mx

Contenido

1. Aspectos del diseño de un generador de código
2. Lenguajes intermedios
3. La máquina objeto
4. Bloques básicos y diagramas de flujo
5. Un generador de código simple
6. Generadores de generadores de código

Recordemos fases de un compilador con un ejemplo.





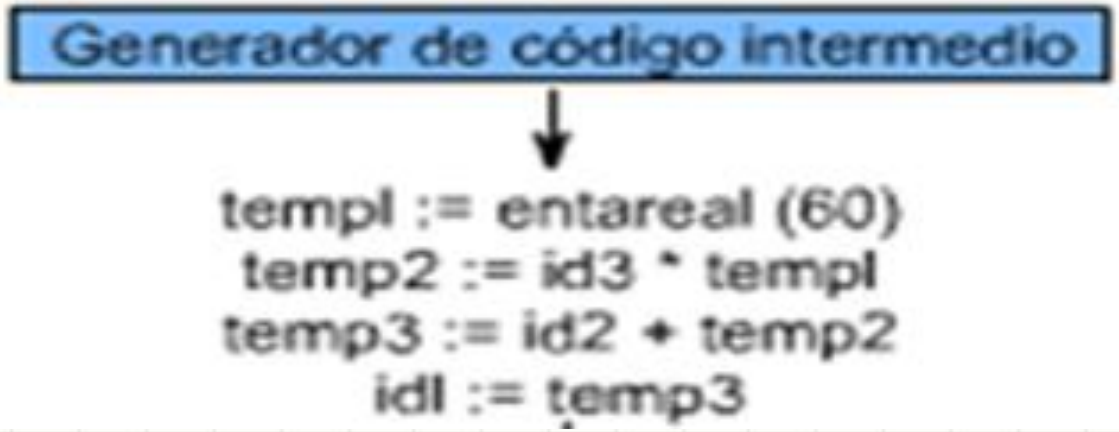
1. Aspectos del diseño de un generador de código

Los detalles de diseño dependen de la máquina objeto y del sistema operativo, aspectos como el manejo de memoria, la selección de instrucciones, la asignación de registros y el orden de evaluación.

1. Aspectos del diseño de un generador de código

Entrada al generador de código

- **Representación intermedia** del programa fuente producida por la etapa inicial, junto con **la tabla de símbolos** que se utiliza para determinar las direcciones durante la ejecución de los objetos de datos denotados por los nombres de la representación intermedia.
- **Código de tres direcciones, notación postfija, etc**.



1. Aspectos del diseño de un generador de código: La salida es el **programa objeto**, puede representarse **de 3 maneras**:

Programa en lenguaje máquina absoluto: la ventaja que se puede colocar en una posición fija de memoria y ejecutarse inmediatamente.

Programa en lenguaje maquina relocalizable: permite que los subprogramas se compilen por separado. Al final se pueden enlazar y cargar para su ejecución.

Programa en lenguaje ensamblador. Facilita el proceso de generación de código. Se pueden generar instrucciones simbólicas y utilizar las macros del ensamblador para ayudar a generar código.

1. Aspectos del diseño de un generador de código

Otros aspectos:

- Administración de la memoria
- Selección de instrucciones
- Asignación de registros

2. Lenguajes intermedios

- **Lenguajes intermedios**: árboles sintácticos, notación postfija y código de tres direcciones.
- El **código de tres direcciones** es una secuencia de proposiciones de la forma general:

  x:= y op z

  - Donde x, y, y z son nombres, constantes o variables temporales generadas por el compilador; op representa cualquier operador.
  - No se permite ninguna expresión compuesta, **solo hay un operador del lado derecho de una proposición.**

2. Lenguajes intermedios

Por tanto, una expresión x+y*z se puede traducir en código de tres direcciones de la siguiente manera:

t1:=y*z

t2:=x+t1

Donde t1 y t2 son nombres de variables temporales generados por el compilador. La generación de este código hace deseable la fase de optimización de código.

2. Lenguajes intermedios

- Otro ejemplo de código de tres direcciones.

Código en C	Código de 3 direcciones
v=a[n];	t1:=4*n v:=a[t1]

Nota: esto es, si se considera que cada elemento de la matriz ocupa cuatro bytes.

3. La máquina objeto

- Una **máquina objeto común** es utilizar una máquina de registros representativa de varios minicomputadores.
- La **máquina objeto** (o computador objeto) puede ser una máquina direccionable por bytes, con palabra de cuatro bytes y n registros generales, R0, R1, …., Rn-1; con instrucciones de dos direcciones direcciones de la forma:

  op fuente, destino

3. La máquina objeto

La dirección

op fuente, destino

Donde **op** es un código de operador, **fuente** y **destino** son campos de datos.

Ejemplos de códigos de operaciones:

MOV (mueve fuente a destino)

ADD (suma fuente a destino)

SUB (resta fuente a destino)

3. La máquina objeto

- Ejemplo: La siguiente instrucción guarda el contenido del registro R0 en la posición de memoria M.

  MOV R0, M

- Las **proposiciones de tres direcciones son análogas al código ensamblador**. Las proposiciones pueden tener etiquetas simbólicas y existen proposiciones para el flujo del control.

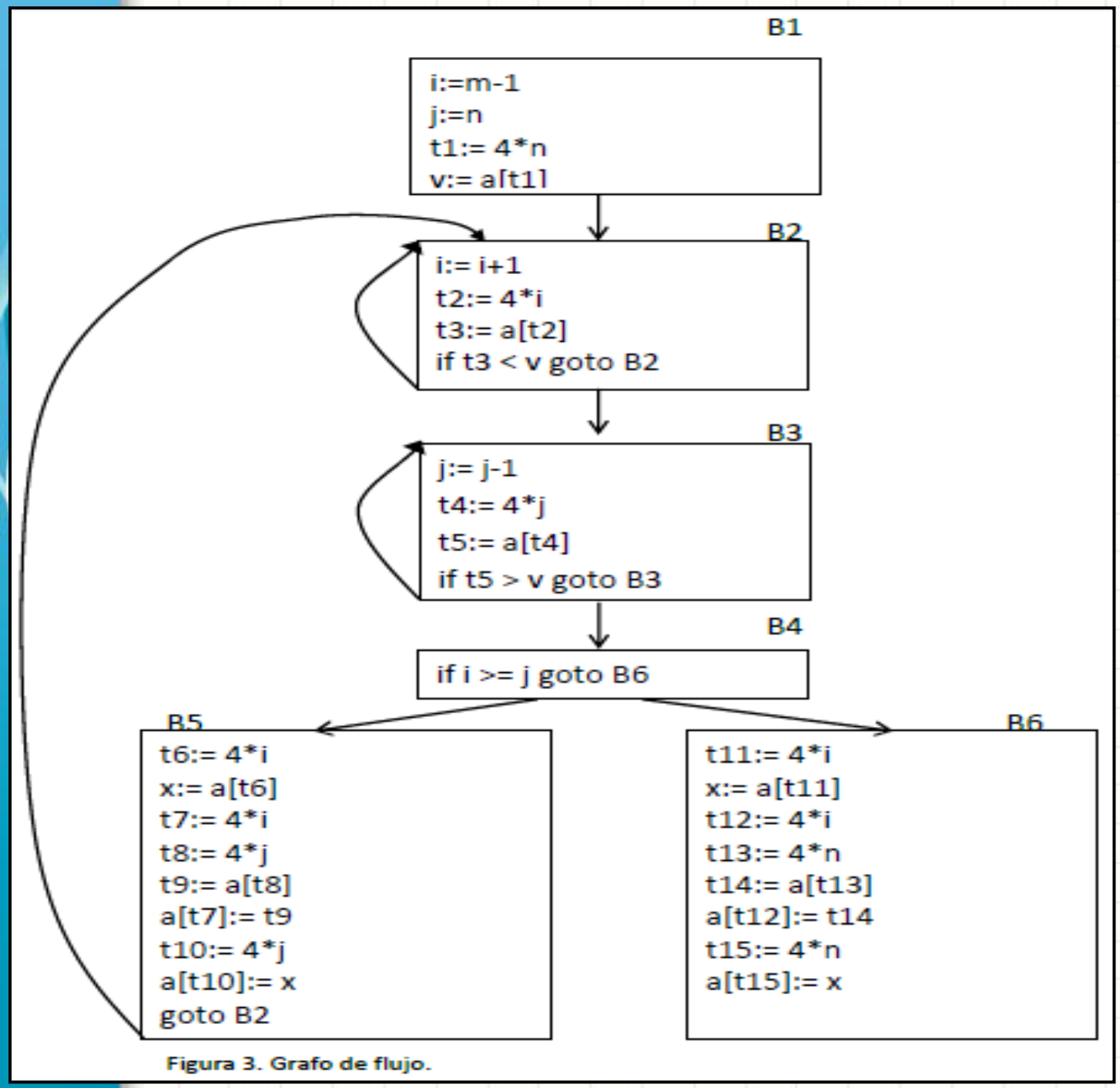


Figura 3. Grafo de flujo.

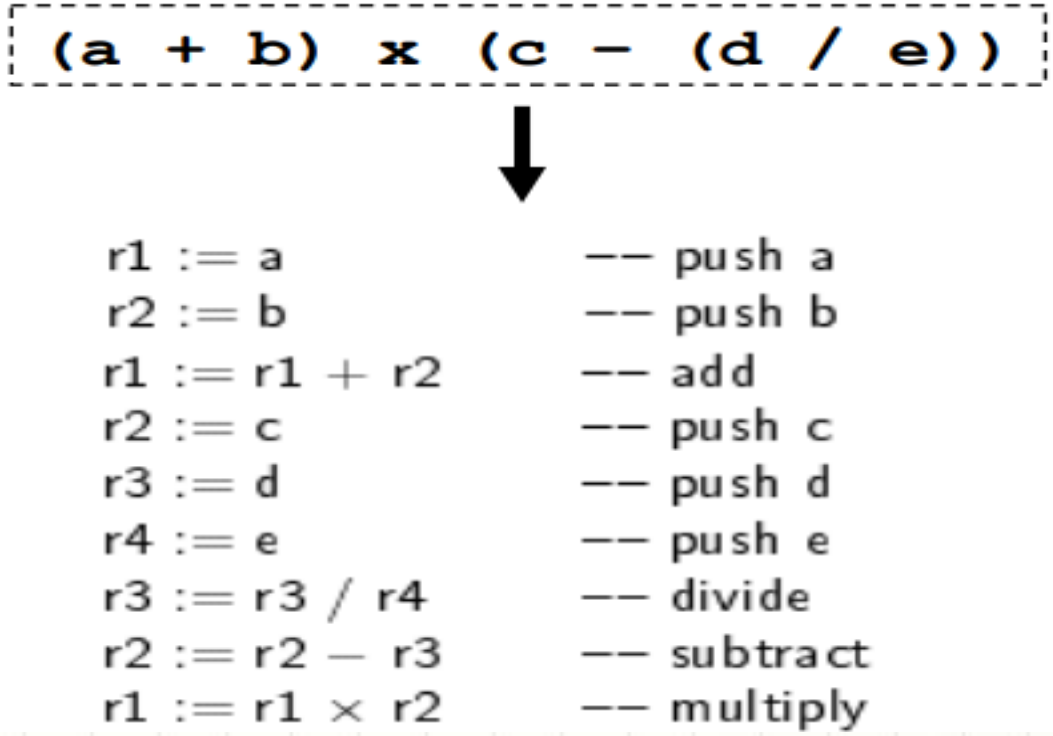
4. Ej. de Bloques básicos y grafos de flujo

- Un **grafo de proposiciones de tres direcciones**, llamada grafo de flujo, es útil para entender los algoritmos de generación de código.
- Los **nodos** representan cálculos y las aristas el flujo de control.
- Un **bloque básico** es una secuencia de proposiciones consecutivas en las que el flujo de control entra al principio y sale al final sin detenerse y sin posibilidad de saltar excepto al final.

5. Un generador de código simple

La generación de código supone elegir un orden de evaluación para las operaciones, asignando valores a los registros, y seleccionando las instrucciones apropiadas en el lenguaje objeto para implantar los operadores en la representación intermedia.

- Ejemplo de generación de código de tres direcciones y generación de código objeto.
- Existen varios algoritmos a seguir como el presentado en el libro de Alfred V. Aho et.al, 2ª edición, pág 542.



6. Generadores de generadores de código

- Programa a reconocer

```
ALFA := n;
FACTORIAL := 1;
WHILE ALFA > 1 DO
 {
 FACTORIAL := FACTORIAL * ALFA;
 ALFA := ALFA - 1;
 };
FIN WHILE;
```

- Código generado

```
        ALFA = n
        tmp1 = 1;
        FACTORIAL = tmp1
label etq1
        tmp2 = 1;
        if ALFA > tmp2 goto etq2
        goto etq3
label etq2
        tmp3 = FACTORIAL * ALFA;
        FACTORIAL = tmp3
        tmp4 = 1;
        tmp5 = ALFA - tmp4;
        ALFA = tmp5
        goto etq1
label etq3
```

Referencias

1. Compiladores. Principios, Técnicas y Herramientas (2a ed.). Aho, A.V. Pearson Educación. 2008.
2. Construcción de Compiladores. Principios y práctica. Louden, K. C. Thomson Editores. 2005