Compiladores: Generación de Código Intermedio

Pontificia Universidad Javeriana Cali
Ingeniería de Sistemas y Computación
Prof. Gloria Inés Alvarez V.

Código Intermedio

- En el modelo de compilación análisis-sintesis, el front-end traduce el programa fuente en una representación de código intermedio, y el back-end traduce esta representación en código final.
  - Permite crear facilmente un compilador para diferentes maquinas
  - La representación intermedia puede ser optimizada por un optimizador independiente del código final
- Lenguajes Intermedios:
  - Arboles de Sintaxis
  - Código de tres direcciones

Código de tres direcciones

- Se compone de instrucciones que contienen tres (o menos) direcciones, dos para los operandos y una para el resultado
- Los operandos y el resultado pueden ser nombres,o variables temporales generadas por el compilador. Los operandos pueden tambien ser constantes.
- Las instrucciones pueden estar marcadas por etiquetas símbólicas (representan un índice dentro del conjunto de instrucciones)
- Tiene instrucciones de control de flujo

Código de tres direcciones: Instrucciones comunes

- Instrucciones de Asignación:

  x := y op z (op es un operador binario, aritmético o lógico)

  x := op y (op es un operador unario: menos, conversión)

  x := y

  Asignaciones Indexadas: x := y[i]

  x[i] := y

  Apuntadores: x := &y (asigna la dirección de y)

  x := *y (y es un apuntador)

  *x := y (x es un apuntador)

Código de tres direcciones: Instrucciones comunes

- Saltos:

  Incondicional: goto L

  Condicional: if x rel-op y goto L

- Llamado a procedimientos y retorno:

  param x1

  param x2

  ....

  param xn

  call p n

  return y

Ejemplo

```
a := 2 * x + 10;
while a <= p + 2 do
        j := p[i*2+4]
end
j := j + 1
```

Ejemplo:

```
a := 2 * x + 10;
while a <= p + 2 do
        a := p[4+i*2]
end
a := a + 1
```

```
        t1 := 2 * x
        t2 := t1 + 10
        a := t2
w1:     t1 := p + 2
        if a <= t1 goto w2
        goto w3
w2:     t1 := i * 2
        t2 := 4 + t1
        a := p[t2]
        goto w1
w3:     t1 := a + 1
        a := t1
```

Implementación de Código de tres direcciones

- Cuadruplas.
  - Estructura con 4 campos: operador, operando1, operando2, resultado.

	operador	**op1**	**op2**	**res**
(0)	*	2	x	t1
(1)	+	t1	10	t2

Implementación de Código de tres direcciones

- Triplas.
  - Estructura con 3 campos: operador, operando1, operando2. Los resultados temporales se referecian por la posición en que fueron calculados

	operador	**op1**	**op2**
(0)	*	2	x
(1)	+	(0)	10

Ejemplo: Definición Dirigida por Sintaxis para generar codigo de tres direcciones para expresiones y asignación