

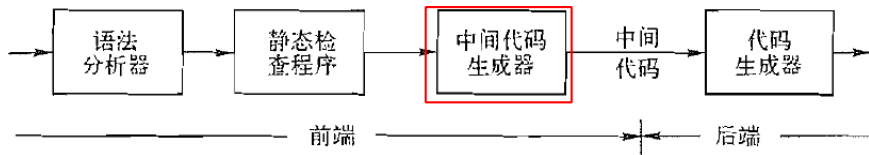
中间代码生成

魏恒峰

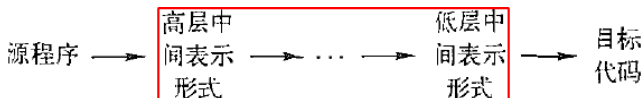
hfwei@nju.edu.cn

2020 年 12 月 21 日





Intermediate Representation (IR)



精确: 不能丢失源程序的信息

独立: 不依赖特定的源语言与目标语言
(如, 没有复杂的寻址方式)

Intermediate Representation (IR)

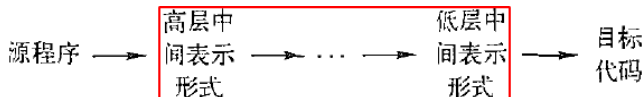
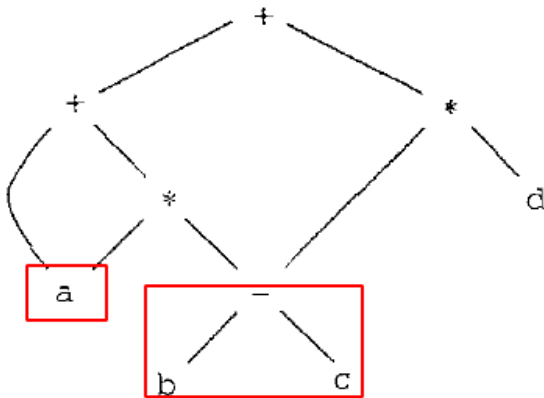


图 (抽象语法树)、三地址代码、C 语言

表达式的有向无环图

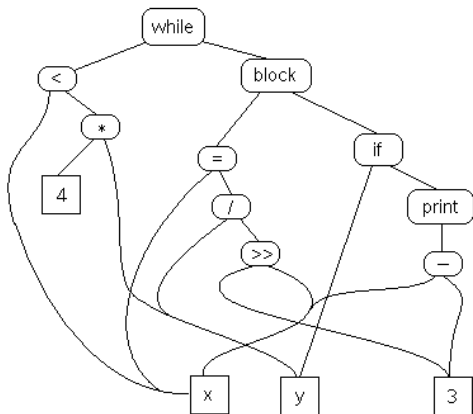


$$a + a * (b - c) + (b - c) * d$$

产生式	语义规则
1) $E \rightarrow E_1 + T$	$E.node = \text{new Node}('+', E_1.node, T.node)$
2) $E \rightarrow E_1 - T$	$E.node = \text{new Node}('-', E_1.node, T.node)$
3) $E \rightarrow T$	$E.node = T.node$
$T \rightarrow T_1 * F$	$T.node = \text{new Node}('*', T_1.node, F.node)$
4) $T \rightarrow (E)$	$T.node = E.node$
5) $T \rightarrow \text{id}$	$T.node = \text{new Leaf}(\text{id}, \text{id.entry})$
6) $T \rightarrow \text{num}$	$T.node = \text{new Leaf}(\text{num}, \text{num.val})$

在创建节点之前, 先判断是否已存在 (哈希表)

```
while (x < 4 * y) {
    x = y / 3 >> x;
    if (y) print x - 3;
}
```



Definition (三地址代码 (Three-Address Code (TAC; 3AC)))

每个 **TAC** 指令**最多**包含三个操作数。

$x = y \text{ op } z$ (1)

$x = \text{op } y$ (2)

$x = y$ (3)

goto L (4)

if x **goto** L (5)

if False x **goto** L (6)

if x **relop** y **goto** L (7)

Definition (三地址代码 (Three-Address Code (TAC; 3AC)))

每个 **TAC** 指令**最多**包含三个操作数。

`param x` (8)
`call p, n` (9)
`y = call p, n` (10)
`return y` (11)

`param x1`
`param x2`
`...`

`param xn`
`call p, n`

$p(x_1, x_2, \dots, x_n)$

Definition (三地址代码 (Three-Address Code (TAC; 3AC)))

每个 **TAC** 指令**最多**包含三个操作数。

$$x = y[i] \quad (12)$$

$$x[i] = y \quad (13)$$

$$x = \&y \quad (14)$$

$$x = *y \quad (15)$$

$$*x = y \quad (16)$$

距离位置 y 处 i 个内存单元

do i = i + 1; while (a[i] < v);

```
L:  t1 = i + 1  
    i = t1  
    t2 = i * 8  
    t3 = a [ t2 ]  
    if t3 < v goto L
```

```
100: t1 = i + 1  
101: i = t1  
102: t2 = i * 8  
103: t3 = a [ t2 ]  
104: if t3 < v goto 100
```

三地址代码的**四元式**表示

Definition (四元式 (Quadruple))

一个四元式包含四个字段, 分别为 op 、 arg_1 、 arg_2 与 $result$ 。

$$a + a * (b - c) + (b - c) * d$$

$t_1 = \text{minus } c$

$t_2 = b * t_1$

$t_3 = \text{minus } c$

$t_4 = b * t_3$

$t_5 = t_2 + t_4$

$a = t_5$

	op	arg_1	arg_2	$result$
0	minus	c		t_1
1	*	b	t_1	t_2
2	minus	c		t_3
3	*	b	t_3	t_4
4	+	t_2	t_4	t_5
5	=	t_5		a
		...		

$x = y[i]$	$=[$	y	i	x
$x[i] = y$	$[$	i	y	x

$x = \&y$	$=\&$	y	x
$x = *y$	$=*$	y	x
$*x = y$	$*=$	y	x

Thank
You!



Office 926

hfwei@nju.edu.cn