编译原理

第十一章 代码生成

第十一章 代码生成

- 目标代码形式
- 代码生成着重考虑的问题
- 一个简单代码生成器
  - 待用信息
  - 活跃信息
  - 寄存器描述信息
  - 变量地址描述信息
  - 代码生成算法

代码生成算法

- 对每个四元式 : i: A:=B op C ， 依次执行 :
  1. 以四元式 : i: A:=B op C 为参数，调用函数过程GETREG(i: A:=B op C)，返回一个寄存器 R ，用作存放 A 的寄存器。
  2. 利用 AVALUE[B] 和 AVALUE[C] ，确定 B 和 C 现行值的存放位置 B’ 和 C’ 。如果其现行值在寄存器中，则把寄存器取作 B’ 和 C’

代码生成算法

3. 如果 B’ ≠R ，则生成目标代码：

LD R, B’

op R, C’

否则生成目标代码 op R, C’

如果 B’ 或 C’ 为 R ，则删除 AVALUE[B] 或 AVALUE[C] 中的 R 。

4. 令 AVALUE[A]={R}, RVALUE[R]={A} 。

代码生成算法

5. 若 B 或 C 的现行值在基本块中不再被引用，也不是基本块出口之后的活跃变量，且其现行值在某寄存器 R_k 中，则删除 RVALUE[R_k] 中的 B 或 C 以及 AVALUE[B] 或 AVALUE[C] 中的 R_k ，使得该寄存器不再为 B 或 C 占用。

及时腾空

寄存器分配

- 寄存器分配： GETREG(i: A:=B op C) 返回一个用来存放 A 的值的寄存器

1. 尽可能用 B 独占的寄存器
2. 尽可能用空闲寄存器
3. 抢占非空闲寄存器

- 寄存器分配： GETREG(i: A:=B op C) 返回一个用来存放 A 的值的寄存器

1. 尽可能用 B 独占的寄存器
2. 尽可能用空闲寄存器
3. 抢占非空闲寄存器

1 如果 B 的现行值在某个寄存器 R_i 中， RVALUE[R_i] 中只包含 B ，此外，或者 B 与 A 是同一个标识符，或者 B 的现行值在执行四元式 A:=B op C 之后不会再引用，则选取 R_i 为所需要的寄存器 R ，并转 4 ；

- 寄存器分配： GETREG(i: A:=B op C) 返回一个用来存放 A 的值的寄存器

1. 尽可能用 B 独占的寄存器
2. 尽可能用空闲寄存器
3. 抢占非空闲寄存器

1 如果 B 的现行值在某个寄存器 R_i 中， RVALUE[R_i] 中只包含 B ，此外，或者 B 与 A 是同一个标识符，或者 B 的现行值在执行四元式 A:=B op C 之后不会再引用，则选取 R_i 为所需要的寄存器 R ，并转 4 ；

2 如果有尚未分配的寄存器，则从中选取一个 R_i 为所需要的寄存器 R ，并转 4 ；

1. 尽可能用 B 所在的寄存器
2. 尽可能用空闲寄存器
3. 抢占非空闲寄存器

3　从已分配的寄存器中选取一个 R_i 为所需要的寄存器 R 。最好使得 R_i 满足以下条件：

占用 R_i 的变量的值也同时存放在该变量的贮存单元中，或者在基本块中要在最远的将来才会引用到或不会引用到。

要不要为 R_i 中的变量生成存数指令?

要不要为 R_i 中的变量生成存数指令?

对 RVALUE[R_i] 中每一变量 M，如果 M 不是 A，或者如果 M 是 A 又是 C，但不是 B 并且 B 也不在 RVALUE[R_i] 中，则

3. 如果 B'≠R，则生成目标代码：
   LD R, B'
   op R, C'
   否则生成目标代码 op R, C'
   如果 B' 或 C' 为 R，则删除 AVALUE[B] 或 AVALUE[C] 中的 R

否则令 AVALUE[M]={M}

(3) 删除 RVALUE[R_i] 中的 M

4 给出 R，返回。

例：基本块

1. T:=A-B
2. U:=A-C
3. V:=T+U
4. W:=V+U

设 W 是基本块出口之后的活跃变量，只有 R_0 和 R_1 是可用寄存器，生成的目标代码和相应的 RVALUE 和 AVALUE:

序号	四元式	左值	左操作数	右操作数
(4)	W:=V+U	(^,y)	(^,^)	(^,^)
(3)	V:=T+U	(4,y)	(^,^)	(4,y)
(2)	U:=A-C	(3,y)	(^,^)	(^,^)
(1)	T:=A-B	(3,y)	(2,y)	(^,^)

U:=A – C　LD R_1 , A　　R_0 含有 T　T 在 R_0 中

　　SUB R_1 , C　　R_1 含有 U　U 在 R_1 中

V:=T + U　ADD R_0 , R_1　R_0 含有 V　V 在 R_0 中

　　R_1 含有 U　U 在 R_1 中

W:=V + U ADD R_0 , R_1　R_0 含有 W　W 在 R_0 中

　　ST R_0 , W

本章小结

- 目标代码形式
- 代码生成着重考虑的问题
- 一个简单代码生成器
  - 待用信息
  - 活跃信息
  - 寄存器描述信息
  - 变量地址描述信息
  - 代码生成算法

作业

- P327-1