

Programmanalysen und Compilerbau

Wintersemester 2007/8
Prof. Plödereder, Stefan Staiger

7. Übungsblatt

Die Besprechung findet am 29.01. statt.

Aufgabe 7.1

Gegeben sei folgender Zwischencode: ((hx) = Speicher-Inhalt an Adresse hx)

```
h1=adr(A)-40
h2=I*4
h3=h1+h2
h7=h3+12
h8=(h7)*2
(h3)=h8+30
```

Erzeugen Sie für diesen Zwischencode optimalen 2-Adreßcode, wobei Sie sich im wesentlichen am Zwischencode orientieren sollten, ohne aber notwendigerweise eine 1-zu-1 Abbildung anzustreben. Kriterien für die Optimierung sind in absteigender Priorität:

- Die Anzahl der LOAD und STORE Operationen sind zu minimieren
- Es sind möglichst wenige Register zu verwenden
- Multiplikation dauert doppelt so lange wie Addition

Es stehen (ausschließlich) folgende 2-Adreß 32-Bit-Instruktionen zur Verfügung:

- LOAD op₁ op₂ -- lade Wert in Adresse op₁ in das Register op₂
- STORE op₁ op₂ -- speichere Wert aus Register op₂ nach Adresse op₁
- MOVE op₁ op₂ -- kopiere Wert von op₁ nach op₂
- sonst: op op₁ op₂ -- op₂ := op₂ op op₁
Hierbei als op möglich: MULT(ultiply), ADD, SUB(tract)
- <op₁> = Registerwert oder literaler Wert (z.B. R1, 17, A); (globale Adressen sind literale Werte)
<op₂> = Register; kann immer auch als op₁ verwendet werden, wobei op₁ den alten Wert von op₂ annimmt.

Beispiele:

```
MOVE A R1      -- lade Adresse von A nach R1, aber
LOAD A R1      -- lade Inhalt der Adresse von A nach R1;
STORE R2 R1    -- speichere Wert von R1 nach Adresse in R2
SUB R2 R2      -- setzt Register 2 auf 0
```

Aufgabe 7.2

Für die Schleife

```
for (j = 0; j < 100; ++j)
{
  A [j] = 0
  i = 4*j
  B [i] = 1
  i = 8*j
  C [i] = 2
}
```

ist folgender Zwischencode entstanden:

```
      j = 0
L1:  if (j >= 100) goto L2
      h1 = 4*j + &A
      *h1 = 0
      i = 4*j
      h2 = 4*i + &B
      *h2 = 1
      i = 8*j
      h3 = 4*i + &C
      *h3 = 2
      j = j + 1
      goto L1
L2:
```

- Bestimmen Sie grundlegende und abhängige Induktionsvariablen.
- Führen Sie nun strength reduction durch.
- Eliminieren Sie unnötige Zuweisungen und propagieren Sie Konstanten. Sie können davon ausgehen, dass i, j und die Hilfsvariablen nach der Schleife nicht mehr benutzt werden.

Aufgabe 7.3

Gegeben sei ein Prozessor mit den folgenden Befehlen und Ausführungszeiten:

Befehlsmuster	Prozessorzyklen
$R_i := M_j$	5
$R_i := R_i \text{ op } R_j$	2
$R_i := R_i \text{ op } M_j$	6
$R_i := R_j$	1
$M_i := R_j$	5

Verwenden Sie den DP-Algorithmus, um für die Anweisung

$$a := (a+b) * (c-d) - b/c + d$$

die schnellste Instruktionsfolge unter Benutzung von höchstens drei Registern zu berechnen.