

Compilerbau - Wintersemester 2021/22

Theoretisches Übungsblatt 11

Besprechung der Aufgaben am 21.01.22 ab 14:30 Uhr in 25.12.02.55 und gleichzeitig online per BBB
Fragen an lukas.lang@hhu.de
Die Bearbeitung ist freiwillig

Aufgabe 11.1

Gegeben seien folgende Ausdrücke:

- (1) $x = a * b + c * d$
- (2) $x[i] = y[j] + z[k]$
- (3) $x = x + 1$

Hierbei sind i, j, k lokale Variablen einer Funktion und alle anderen Variablen globale Variablen. Arrayelemente sind 8 Bit groß.

- (a) Geben Sie die IR-Bäume für alle Ausdrücke an. Der Knoten `ind` gibt hierbei an, dass der Wert an einer Adresse genutzt wird.
- (b) Erweitern Sie die Baumersetzungsregeln aus der Vorlesung um eine Regel für Multiplikation und ein einzelnes Register an einem `ind` Knoten.
- (c) Nutzen Sie die Baumersetzungsregeln um RISC Code zu erzeugen.
- (d) Geben Sie für jeden IR-Baum eine Präfixdarstellung an und nutzen Sie die syntaxgerichtete Definition um RISC Code zu erzeugen.

Hinweis: Links einer Zuweisung befindet sich per Konvention ein `ind` Knoten.

Aufgabe 11.2

Gegeben seien folgende Ausdrücke:

- (1) $a / (b + c) - d * (e + f)$
- (2) $a + b * (c * (d + e))$
- (3) $(-a + *p) * ((b * q) / -c + *r)$

- (a) Erzeugen Sie hierzu Zwischencode in Baumdarstellung
- (b) Berechnen Sie für alle Bäume die Ershov-Zahlen.
- (c) Spilling: Erzeugen Sie optimalen Code für alle Ausdrücke. Nutzen Sie nicht mehr als drei Register.

- (d) Spilling: Erzeugen Sie optimalen Code für alle Ausdrücke. Nutzen Sie nicht mehr als zwei Register.

Hinweis: * ist eine Multiplikation oder eine Dereferenzierung

Aufgabe 11.3

Erzeugen Sie Drei-Adress RISC Pseudocode (analog zur Vorlesung) für die folgenden Anweisungen unter der Voraussetzung, dass alle Variablen im Arbeitsspeicher abgelegt sind und binäre Operationen nur zwischen Registern möglich sind:

- (a) $x = 1$
- (b) $x = a$
- (c) $x = a + 1$
- (d) $x = a + b$
- (e) Die beiden folgenden Anweisungen:
 $x = b * c$
 $y = a + x$

Aufgabe 11.4

Erzeugen Sie Drei-Adress RISC Pseudocode (analog zur Vorlesung) für die folgenden Anweisungen unter der Voraussetzung, dass a und b Arrays mit 4-Byte-Werten als Elementen und binäre Operationen nur zwischen Registern möglich sind:

- (a) Die Folge aus den vier Anweisungen:
 $x = a[i]$
 $y = b[j]$
 $a[i] = y$
 $b[j] = x$
- (b) Die Folge aus den drei Anweisungen:
 $x = a[i]$
 $y = b[i]$
 $z = x * y$
- (c) Die Folge aus den folgenden drei Anweisungen:
 $x = a[i]$
 $y = b[x]$
 $a[i] = y$

Aufgabe 11.5

Gegeben sei folgender Teil eines Javaprogramms:

```

int n = 10;
int s = 1;
while (n>0) {
    s = s + n * n;
    n = n - 1;
}

```

Erzeugen Sie dafür Jasmin Assembler Code (Schriftlich. Keine Java Programmierung).

Aufgabe 11.6

Gegeben sei folgender Teil eines Javaprogramms:

```

if (a<b) {
    c = 7 * 2 + 1;
} else {
    c = c + 1;
}
d = c+1;

```

Erzeugen Sie dafür Jasmin Assembler Code (Schriftlich. Keine Java Programmierung).