

Compilerbau & Abstrakte Maschinen

Sommersemester 2006

1. Übungsblatt Abgabetermin: 2. Mai 2006, 12 Uhr, Raum 02.07.52

Aufgabe 1: Code-Erzeugung

8 Punkte

Gegeben sei folgende Anweisungsfolge:

```
z = 1;
while (n > 0) {
    j = 1;
    y = x;
    while (2 * j <= n) {
        y = y * y;
        j = j * 2;
    }
    z = y * z;
    n = n - j;
}
```

- Was berechnet die Anweisungsfolge?
- Erzeuge CMa-Code für die Anweisungsfolge!
Verwende die Adressumgebung $\rho = \{n \mapsto 0, j \mapsto 1, x \mapsto 2, y \mapsto 3, z \mapsto 4\}$!

Aufgabe 2: Short circuit evaluation

8 Punkte

Seien b , e_1 und e_2 drei beliebige Ausdrücke.

- Ein $?$ -Ausdruck in C hat die Form $b ? e_1 : e_2$ und sein Wert ist e_1 falls $b \neq 0$ und e_2 falls $b = 0$. Gib das CMa-Übersetzungsschema für $code_R(b ? e_1 : e_2) \rho$ an!
- Unter *short circuit evaluation* für boolesche Ausdrücke versteht man, dass das zweite Argument eines $\&\&$ -Operators (*und* in C) nicht mehr ausgewertet wird, wenn die Auswertung des ersten Arguments 0 ergibt. Das erspart die überflüssige Auswertung des zweiten Arguments.
Gib die CMa-Übersetzungsschemata für $code_R(e_1 \&\& e_2) \rho$ an, um *short circuit evaluation* zu realisieren!

Aufgabe 3: Break

4 Punkte

Modifiziere das Schema zur Übersetzung von Schleifen, so dass diese *break*-Anweisungen zum sofortigen Beenden der Schleife enthalten dürfen! Erweitere dazu die Code-Erzeugungsfunktion um ein weiteres Argument l , welches das Sprungziel (label) beschreibt, zu dem bei einem *break* gesprungen werden soll!