Technische Universität München
Fakultät für Informatik
Prof. Dr. H. Seidl

A. Berlea, P.Ziewer
{berlea,ziewer}@in.tum.de
Raum: MI 02.07.053/055

Compilerbau

und

Abstrakte Maschinen

Sommersemester 2004

1. Übungsblatt *Abgabetermin: 3. Mai 2004, 12 Uhr*

Aufgabe 1: *8 Punkte*

Erzeuge CMa-Code für die folgende Anweisungsfolge!

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

Verwende die Adressumgebung $\rho = \{n \mapsto 1, j \mapsto 2, y \mapsto 3, x \mapsto 4, z \mapsto 5\}$!

Aufgabe 2: *8 Punkte*

Seien b, e_1 und e_2 drei beliebige Ausdrücke.

a) 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 $\mathit{code}_R \; (b \; ? \; e_1 \; : \; e_2) \; \rho$ an!

b) Unter *short circuit evaluation* für boolsche Ausdrücke versteht man, dass das zweite Argument eines `&&`- bzw. `||`-Operators (*und* bzw. *oder* in C) nicht mehr ausgewertet wird, wenn die Auswertung des ersten Arguments 0 bzw. einen Wert ungleich 0 ergibt. Das erspart die überflüssige Auswertung des zweiten Arguments.
Gib die CMa-Übersetzungsschemata für $\mathit{code}_R \; (e_1 \; \&\& \; e_2) \; \rho$ und $\mathit{code}_R \; (e_1 \; || \; e_2) \; \rho$ an, um *short circuit evaluation* zu realisieren!

Aufgabe 3: *4 Punkte*

Modifiziere das Schema zur Übersetzung von Schleifen, so dass diese *break* 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!