

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.

- 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 &&- 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 $code_R(e_1 \ \&\& \ e_2) \rho$ und $code_R(e_1 \ || \ e_2) \rho$ 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!