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

2. Übungsblatt — *Abgabetermin: 10. Mai 2004, 12 Uhr*

Aufgabe 1: *5 Punkte*

Gib Übersetzungsschemata an für:

a) die Prä- und Postfix-Inkrementierung in C: `++`e und e`++`,

b) die Ausdrücke e_1`+=`e_2 und e_1`-=`e_2.

Beachte dabei, dass die obigen Operatoren auch auf komplexere Ausdrücke angewendet werden können, wie z.B.: `a[f(1+x)]++`. Welche Ausdrücke können für e, e_1 und e_2 verwendet werden?

Aufgabe 2: *7 Punkte*

Übersetze das folgende Programm:

```
int result;

int fibo (int n){
  int result;

  if (n<0) return -1;
  switch (n){
    case 0 : return 0;
    case 1 : return 1;
    default: result = fibo(n-1) + fibo(n-2);
             return result;}
}

main(){
  int n;

  n = 5;
  result = fibo(n);
}
```

Abgabe des lauffähigen, kommentierten CMa-Programms per E-mail.

Aufgabe 3: *8 Punkte*

Zum Setzen des Extreme-Pointers EP während der Ausführung benötigt der Übersetzer den maximalen Zuwachs des Stacks. Dieser Wert kann während der Übersetzung (statisch) berechnet werden.

Sei $t(e)$ die Anzahl der Stackzellen, die für die Auswertug eines Ausdrucks e benötigt werden. Gib eine rekursive Definition für die Berechnung von $t(e)$ an!

a) Für arithmetische Ausdrücke. Betrachte insbesondere die beiden Extremfälle $a_1 + (a_2 + (\cdots + (a_{n-1} + a_n) \ldots))$ und $(\ldots((a_1 + a_2) + a_3) + \ldots) + a_n$ (a_i sind Konstanten oder einfache Variablen);

b) für Ausdrücke im allgemeinen, insbesondere auch Funktionsaufrufe;

c) für C-Anweisungen (nicht CMa).