TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Lehrstuhl II *Sommersemester 2007*
Prof. Helmut Seidl *Andrea Flexeder*

Compilerbau & Virtuelle Maschinen

2. Übungsblatt

Abgabe: 30. April 2007, bis 16 Uhr in der Vorlesung oder Raum 02.07.59

Aufgabe 4: *Structs und Felder* *4 Punkte*

```
int j;

struct list{
  int entry;
  struct list* next;
};

struct list a[3];

for(j=0;j<3;j++){
  a[j].entry=j+1;
  a[j].next=0;
}
```

a) Berechnen Sie eine Adressumgebung für die Variablen.
b) Erzeugen Sie Code für dieses Programmfragment.

Aufgabe 5: *Blöcke und Initialisierung* *7+3 Punkte*

1. Erweiteren Sie die Codeerzeugungsfunktion für Anweisungsfolgen um Blöcke. Berücksichtigen Sie dabei, daß Variablendeklarationen an beliebigen Stellen innerhalb von Blöcken vorkommen dürfen. In Blöcken deklarierte Variablen sollen von dem Punkt, an dem sie eingeführt werden, bis zum zugehörigen Blockende sichtbar sein.

   Bei der Ausführung des folgenden Beispielprogramms wird erst `2` und anschließend `1` als Wert der Variablen `x` ausgegeben.

```
int x;
x = 1;
{
    int x;
    x = 2;
    write(x);
}
write(x);
```

   Tipp: Verwalten Sie zusätzlich zur Adressumgebung die erste vergebbare Relativadresse.

2. Modifizieren Sie zusätzlich die Codeerzeugungsfunktion so, dass auch Variablen intialisiert werden können, beispielsweise `int x = 1;`

Aufgabe 6: *Extreme Pointer* *6 Punkte*

Um den Extreme-Pointer **EP** während der Ausführung zu setzen, benötigt der Übersetzer den maximalen Zuwachs des Stacks. Dieser Wert kann während der Übersetzung (statisch) berechnet werden.

Definieren Sie eine Funktion t, die einem Ausdruck e eine möglichst präzise obere Schranke $t(e)$ für die Anzahl der Stackzellen zuordnet, die für die Auswertung des Ausdrucks e benötigt werden.

a) Berechnen Sie zunächst $t(e)$ für die beiden Fälle:
   $e = a_1 + (a_2 + (\cdots + (a_{n-1} + a_n)\ldots))$ und $e = (\ldots((a_1 + a_2) + a_3) + \ldots) + a_n$
   (a_i seien Konstanten oder einfache Variablen).

b) Berechnen Sie $t(e)$ für die folgenden Ausdrücke:
   $\{x \quad e_1 = e_2 \quad b\,?\,e_1 : e_2 \quad e_1[e_2] \quad e.a \quad malloc(e) \quad \&e \quad *e \quad e \to a \quad e_1 \diamond e_2\}$.

c) Erweiteren Sie den Definitionsbereich der Funktion t auf die **while**-Anweisung.