

Programmanalysen und Compilerbau

Wintersemester 2007/8
Prof. Plödereder, Stefan Staiger

3. Übungsblatt

Dieses Blatt wird am 20.11. besprochen.

Aufgabe 3.1

Gegeben sei das folgende Programm in einer Pascal-ähnlichen Sprache:

```
PROGRAM Zagadka;
  b, c : Integer;

  PROCEDURE a1 (c : Integer);
  BEGIN
    print(c*b);          (* Punkt 1 *)
  END;

  PROCEDURE a2 (c : Integer);
  b : Integer;
  BEGIN
    b := c;              (* Punkt 2 *)
    a1(c);
  END;

BEGIN
  b := 6; c := 8;       (* Punkt 3 *)
  a1(c); a2(c);
END.
```

- Bestimmen Sie die Ausgabe dieses Programms bei statischer Namensbindung und bei dynamischer Namensbindung.
- Zu welchen Problemen führt die dynamische Namensbindung bei der semantischen Analyse im Compiler und bezüglich der Lesbarkeit von Programmtexten (denken Sie an lokale Variablen)?
- Nehmen Sie an, dass die dynamische Namensbindung mittels Shallow Binding implementiert ist. Stellen Sie den Zustand der hierfür verwendeten Datenstruktur in folgenden Programmzuständen dar:
 - Punkt 3
 - Punkt 1 (a1 vom Hauptprogramm aus aufgerufen)
 - Punkt 2
 - Punkt 1 (a1 von a2 aus aufgerufen)

Aufgabe 3.2

Stellen Sie den arithmetischen Ausdruck

$$-(a+b)*(c+d)+(a+b+c)$$

als Syntaxbaum und als gerichteten azyklischen Graphen (DAG) dar. Übersetzen Sie beide Darstellungen in 3-Adresscode.

Beachten Sie, dass es verschiedene Lösungen gibt, die jeweils von der Semantik der Programmiersprache abhängen. Entscheiden Sie sich für eine.

Aufgabe 3.3

```
X, Y : Integer;  
B    : Boolean;
```

```
while (0 < Y and 10 >= Y) or not (X < 5 and B) loop  
    X := X + 1;  
end loop;
```

- a. Stellen Sie die Anweisung als Syntaxbaum dar und leiten Sie daraus 3-Adresscode ab. (Die booleschen Operationen dürfen explizit verwendet werden.)
- b. Eine mögliche Optimierung besteht darin, boolesche Operationen durch Verzweigungen darzustellen. Wie sieht in diesem Fall der zugehörige 3-Adresscode aus?

Aufgabe 3.4

- a. Imperative Programmiersprachen verfügen im Allgemeinen über
 - arithmetische Ausdrücke,
 - boolesche Ausdrücke sowie
 - Unterprogramme und
 - bedingte Anweisungen (IF).

Welche Optimierungen könnte man für diese Klassen von Sprachkonstrukten während der Übersetzung durchführen? Geben Sie dazu Beispiele an.

- b. Welche dieser Optimierungen sind von der Zielarchitektur unabhängig?
- c. Bestehen Wechselwirkungen zwischen den verschiedenen Optimierungen?