

# Programmanalysen und Compilerbau

Wintersemester 2007/8  
Prof. Plödereder, Stefan Staiger

## 2. Übungsblatt

Dieses Übungsblatt wird am 6. November besprochen.

### Aufgabe 2.1

Gegeben seien folgende Deklarationen:

```
TYPE  Feld1 = ARRAY[1..10] OF BOOLEAN;
      Feld2 = ARRAY[1..10] OF BOOLEAN;

VAR   A      : Feld1;
      B, C    : ARRAY[1..10] OF BOOLEAN;
      D, E    : Feld2;
      F      : Feld2;
      X      : ARRAY[1..10] OF Feld1;
      Z, Y    : ARRAY[1..10] OF Feld2;
```

- Welche Variablen sind strukturäquivalent?
- Welche Variablen sind namensäquivalent?  
(Hier gibt es eine Klasse von Fällen, die in verschiedenen Programmiersprachen unterschiedlich gelöst ist – welche?)
- Welche Vor- und Nachteile haben die beiden Konzepte der Typäquivalenz aus der Sicht eines Programmierers und aus der eines Compilerbauers?

### Aufgabe 2.2

Gegeben sei folgendes Programmfragment in einer Sprache mit statischer Namensbindung:

```
PROGRAM Schach;

  PROCEDURE Bewerte (FUNCTION f : REAL);
  BEGIN
    ... := f();          (* Punkt 4 *)
  END Bewerte;

  PROCEDURE AlphaBeta;
    FUNCTION Eroeffnung : REAL; BEGIN ... END Eroeffnung;
  BEGIN
    ...
    Bewerte (Eroeffnung)  (* Punkt 3 *)
    ...
  END AlphaBeta;

  PROCEDURE Start; BEGIN ... END Start;
```

BEGIN

Start; (\* Punkt 1 \*)

AlphaBeta; (\* Punkt 2 \*)

END Schach;

- a. Wozu werden die statische und die dynamische Verkettung der Aktivierungsblöcke benötigt? Welche Informationen stellen sie dar?
- b. Welche Struktur hat der Laufzeitstapel (inkl. statischer und dynamischer Verweise) jeweils *nach* folgenden Schritten der Programmausführung?
  - i) Schach ruft Start auf (Punkt 1).
  - ii) Start endet.
  - iii) Schach ruft AlphaBeta auf (Punkt 2).
  - iv) AlphaBeta ruft Bewerte auf und übergibt dabei Eroeffnung als Unterprogramm-Parameter (Punkt 3).
  - v) Bewerte ruft seinen Unterprogramm-Parameter f auf (Punkt 4).
  - vi) f und Bewerte enden.
- c. Welche Struktur ergibt sich für die Situationen aus der vorigen Teilaufgabe, wenn anstelle der statischen Verkettung die Display-Technik verwendet wird?
- d. Welche Vor- oder Nachteile haben Verkettung und Display-Verfahren?

### Aufgabe 2.3

Welche Konsequenzen ergeben sich aus den folgenden Spracheigenschaften für die Struktur der Aktivierungsblöcke sowie die Verwaltung der statischen und dynamischen Verweise bzw. der Display-Vektoren?

- a. Es dürfen nur solche Unterprogramme als Parameter übergeben werden, die nicht innerhalb eines anderen Unterprogrammes deklariert sind (z.B. in Modula-2).
- b. Die Sprache verbietet verschachtelte Unterprogramme (wie beispielsweise C oder Fortran).
- c. Die Sprache erlaubt Unterprogramme mit einer variablen Anzahl von Parametern (z.B. printf in C).

### Aufgabe 2.4

Die meisten prozeduralen Programmiersprachen verbieten die Verwendung von GOTO-Befehlen über Unterprogrammgrenzen hinweg.

- a. Nennen Sie mögliche Gründe dafür. Unter welchen Umständen wären solche Sprünge möglich und mit welcher Semantik?
- b. Die Sprache C erlaubt Sprünge zwischen Unterprogrammen mit den Standard-Funktionen `setjmp` (erzeugt eine Momentaufnahme) und `longjmp` (führt das Programm an der Stelle der Momentaufnahme fort).
  - i) Welche Bedingungen müssen an die Anwendung dieser Funktionen gestellt werden?
  - ii) Wie können die Funktionen implementiert werden?