

Übungen zur Vorlesung  
Bäume, Ordnungen und Anwendungen  
Blatt 6

Juniorprof. Dr. Roland Meyer

Abgabe bis 03.12.2013 um 14h

**Aufgabe 6.1** (Transferverhalten)Betrachten Sie die folgende Prozedur `foo()`:

```

proc [foo()]1
  if [x2]2 then
    [assert x2 = 1]3
    [x1 := ¬x1]4
    [x2 := 0]5
    [foo()]6
  else
    [assert x2 = 0]8
    if [x3]9 then
      [assert x3 = 1]10
      [x2 := ¬x2]11
      [x3 := 0]12
      [foo()]13
    else
      [assert x3 = 0]15

```

Berechnen Sie das Transferverhalten von `foo()` auf der Wertemenge  $\{0, 1\}^3$ . Ist die berechnete Transferfunktion präzise?

*Hinweis:* Sie müssen die Fixpunktiteration nicht für alle Variablen durchführen. Vereinfachen Sie das Summary-Gleichungssystem, indem Sie Funktionswerte vorherberechnen.

**Aufgabe 6.2** (Parameter in Prozeduren)

Bei der Definition von rekursiven Programmen gibt es bisher keine Parameter und keine Rückgabewerte in Prozeduren. Erweitern Sie rekursive Programme um Prozeduraufrufe der Form  $[x_0 := q(x_1 \dots x_n)]_{ret}^{call}$ . Betrachten Sie dabei den Aufruf als Makro. Geben Sie ein Code-Fragment *ohne Parameterübergabe und ohne Rückgabewert* an, das den Aufruf simuliert.

**Aufgabe 6.3** (Parameter in Prozeduren)

Wenden Sie Ihre Methode aus Aufgabe 6.2 an, um das Transferverhalten des Aufrufs  $[x_0 := \text{or}(x_1, x_2)]_{ret}^{call}$  auf der Wertemenge  $\{0, 1\}^3$  zu bestimmen. Die Prozedur  $\text{or}(y_1, y_2)$  ist dabei folgendermaßen definiert:

```

proc [or(y1, y2)]1
  [y3 := y1 ∨ y2]2
  return [y3]3

```

Abgabe bis 03.12.2013 um 14h im Kasten neben Raum 34-401.4