

Theorie der Programmierung (WS 2003/04)

Übungsblatt 7

Aufgabe 1

Bestimmen Sie small und big step Semantik des folgenden Programms:

```
let fun even (x : int) : bool = (x = 0) orelse odd (x - 1)
    and odd (x : int) : bool = not (x = 0) andalso even (x - 1)
in even 1
end
```

Aufgabe 2

Wir hätten die Kernsyntax unserer Programmiersprache kleiner halten können, wenn wir anstelle der **rec**-Ausdrücke einen neuen Operator Y eingeführt hätten so, dass sich **rec** $id:\tau. e$ als syntaktischer Zucker für $Y (\lambda id:\tau. e)$ auffassen lässt. Wie müssen Typregel, small step Regel und big step Regel für Y aussehen, damit man die entsprechenden Regeln für **rec** $id:\tau. e$ aus ihnen ableiten kann.

Aufgabe 3

Zeigen Sie, dass *Preservation* und *Progress* auch für die durch Listen erweiterte Programmiersprache gelten. Wie muss man die ursprünglichen Beweise der beiden Sätze ergänzen?

Aufgabe 4

Definieren Sie die folgenden Funktionen auf integer Listen in unserer Programmiersprache:

- eine Funktion *conc* mit: *conc* $l_1 l_2$ liefert die Konkatenation von l_1 und l_2
- eine Funktion *exists* mit: *exists* $p l$ überprüft, ob mindestens ein Element von l die Eigenschaft p hat
- eine Funktion *all* mit: *all* $p l$ überprüft, ob alle Elemente von l die Eigenschaft p haben
- eine Funktion *filter* mit: *filter* $p l$ liefert die Liste aller Elemente von l , die die Eigenschaft p haben (in der ursprünglichen Reihenfolge)