

Theorie der Programmierung Wintersemester 2006/07

Übungsblatt 8

Korrektur: Die folgende small step Regel wurde in der Vorlesung falsch angegeben.

$$(\text{SEND-ATTR}) \quad (\mathbf{val} \ id = v; r) \# m \rightarrow (r[v/id]) \# m$$

Intuition: $\mathbf{val} \ id = v; r$ wird ähnlich behandelt wie $\mathbf{let} \ id = v \ \mathbf{in} \ e$.

Aufgabe 1

Implementieren Sie die folgenden Funktionen auf Listen und überlegen Sie sich jeweils, wie der allgemeinste Typ der Funktion aussieht.

- a. eine Funktion *length*, die die Länge einer Liste liefert,
- b. eine Funktion *append*, die zwei Listen aneinanderhängt,
- c. eine Funktion *sum*, die die Summe der Elemente einer Liste liefert,
- d. eine Funktion *map* mit der Eigenschaft: $\mathit{map} \ f \ [a_1, \dots, a_n]$ liefert $[f \ a_1, \dots, f \ a_n]$.

Aufgabe 2

Bestimmen Sie—in groben Zügen—die small step Semantik der folgenden Programme.

a. **let** o = **object**

val $a = 1 + 2$

method $f = a + 3$

val $a = 4 + 5$

method $g = a + 6$

val $a = 7$

method $h = a + \text{self}\#f + \text{self}\#g$

end

in $o\#h$

b. **let** o = **object**

method $g = 1 + 2$

method $f = \text{self}\#g + 3$

method $g = 4 + 5$

end

in $o\#f$

c. **let** $\text{point} = \lambda x. \lambda y. \text{object}$

val $x = x$

val $y = y$

method $x_coord = x$

method $y_coord = y$

method $\text{equal} = \lambda q. x = q\#x_coord \ \&\& \ y = q\#y_coord$

end

in **let** $p = \text{point } 2 \ 3$ **in** **let** $q = \text{point } 4 \ 5$ **in** $p\#\text{equal } q$