8. Übungsblatt zu Informatik 1 WS 98/99

Andreas Rossberg, Prof. Gert Smolka

Abgabe: 18. Dezember 1998 vor der Vorlesung

Hinweis: Die Programme aus den Aufgaben 8.1, 8.4, 8.5 und 8.6 bitte zusätzlich per Email.

Aufgabe 8.1: Produkt (3 Punkte)

Ähnlich wie für die Summe kennen die Mathematiker auch eine kompakte Schreibweise für Produkte:

$$\prod_{i=1}^{n} f(i) = f(1) \cdot \ldots \cdot f(n)$$

Dabei ist $\prod_{i=1}^{0} f(i) = 1$ definiert. Schreiben Sie eine SML-Prozedur `prod`, die die Funktion

$$\begin{aligned} &prod \in (\mathbf{N} \to \mathbf{Z}) \times \mathbf{N} \to \mathbf{Z} \\ &prod(f,n) = \prod_{i=1}^{n} f(i) \end{aligned}$$

berechnet.

Aufgabe 8.2: Typinferenz (4 Punkte)

Werfen Sie einen Blick auf folgende SML-Deklaration:

```
fun f(a,b,c,d) = if a = b then c+1
                 else if a > b then d
                 else b+c
```

(a) Erläutern Sie, warum der SML-Interpreter herleiten kann, dass `a`, `b`, `c` und `d` alle den Typ `int` haben.

(b) Was ist der Typ von `f`?

Aufgabe 8.3: Konditional (4 Punkte)

Folgende Prozedur sei deklariert:

```
fun cond(b,a1,a2) = if b then a1 else a2
```

(a) Was ist der Typ von `cond`?

(b) Stellen Sie Sich vor, Sie realisieren die Fakultät mithilfe dieser Prozedur:

```
fun fac(n) = cond(n=0, 1, n*fac(n-1))
```

Warum wird diese Implementierung nicht funktionieren?

Aufgabe 8.4: Strings (3 Punkte)

Schreiben Sie eine Prozedur `bar: int -> string`, die für nicht-negative Argumente einen entsprechend langen Balken aus Sternchen erzeugt. So sei beispielsweise das Ergebnis von `bar(5)` der String `"*****"`. Ihre Prozedur sollte endrekursiv sein, deklarieren Sie etwaige Hilfsprozeduren lokal mit Let-Ausdrücken.

Aufgabe 8.5: Kubikwurzel (6 Punkte)

Um eine Näherung für die Kubikwurzel einer reellen Zahl x zu finden, kann man die Tatsache ausnutzen, dass die Folge

$$a_1 = 1 \qquad\qquad a_{i+1} = \frac{1}{2}\left(a_i + \frac{x}{a_i^2}\right)$$

gegen $\sqrt[3]{x}$ konvergiert.

(a) Nutzen Sie diese Erkenntnis aus, indem Sie eine SML-Prozedur `cbrt: real * real -> real` (*cubic root*) schreiben, so dass `cbrt(`e`,`x`)` für nicht-negative e und x einen Wert y liefert mit $|x - y^3| \leq e$. Deklarieren Sie Hilfsprozeduren lokal mit Let-Ausdrücken.

(b) Modifizieren Sie Ihre Prozedur so, dass Sie ermitteln können, wieviele Rekursionsschritte nötig sind, um `cbrt(0.0001,73.0)` zu berechnen.

Aufgabe 8.6: Tupel (4 Punkte)

Schreiben Sie folgende Prozeduren mithilfe von Pattern-Matching und ohne Konditionale:

(a) `fst: 'a * 'b -> 'a` und `snd: 'a * 'b -> 'b` liefern die erste bzw. die zweite Komponente eines Paares.

(b) `f: ('a * 'b) * ('c * 'd) -> 'a * 'b * 'c * 'd` macht aus den Argumentpaaren ein 4-Tupel.

(c) `g: int * int * int -> int` liefert

- 0, falls alle Argumente 0 sind, oder eines der Argumente 1 ist,
- sonst den Wert des zweiten Arguments, falls das erste Argument 3 ist,
- sonst den Wert des ersten Arguments, falls eines der anderen 2 ist,
- sonst die Summe der drei Argumente.

Aufgabe 8.7: Pattern-Matching (4 Punkte)

Welche Bezeichner werden durch folgende Deklarationen eingeführt und welche Typen und Werte werden ihnen zugeordnet?

```
val  a        = (false, (), "ML", (3.5,0.2))
val (b,c)     = ("$", a)
val (d,u,_,e) = c
val  f        = if d then ((op+,op<),(3,4)) else ((op div,op=),(0,1))
val ((g,h),_) = f
val ((_,(i,j)),k) = (f,fn x=>(x,x))
```

Aufgabe 8.8: Weihnachtsfeier (0 Punkte)

Betrachten Sie am Donnerstag den 17.12.98 um 14 Uhr die Rotunde (der runde Raum nahe Eingang zu Bau 45). Alle Angehörigen des Fachbereichs seien herzlich eingeladen. Was veranstaltet die Fachschaft Informatik dort?