11. Übungsblatt zu Programmierung (WS 1999/2000)

Andreas Rossberg, Prof. Gert Smolka

Abgabe: Freitag, 21. Januar 2000 vor der Vorlesung

Hinweis: Bitte alle SML-Programme zusätzlich per Email abgeben.

Aufgabe 11.1: Produkt (6 Punkte)

Gegeben sei die Prozedurdeklaration

```
fun prod(m,n) = if n = 0 then 0 else m + prod(m,n-1)
```

die eine partielle Funktion $prod \in \mathbb{Z} \times \mathbb{Z} \xrightarrow{\text{par}} \mathbb{Z}$ definiert.

(a) Zeigen Sie, dass `prod` für $(m,n) \in \mathbb{Z} \times \mathbb{N}$ terminiert, indem sie die Terminationsbedingungen und eine geeignete wohlfundierte Relation angeben.

(b) Zeigen Sie die Termination, indem Sie die Eigenschaft

$$\forall m \in \mathbb{Z}\, \forall n \in \mathbb{N}: \quad prod(m,n) = m \cdot n$$

durch Induktion über n beweisen.

(c) Was ist die Laufzeit von `prod` (mit Θ)?

(d) Schreiben Sie eine iterative Version `prodi` von `prod`.

Aufgabe 11.2: Iteration (5 Punkte)

Betrachten Sie die Deklaration

```
fun f n = if n < 3 then n else f(n-1) + 2*f(n-2) + 3*f(n-3)
```

die eine Funktion $f \in \mathbb{Z} \to \mathbb{Z}$ definiert.

(a) Zeichnen Sie den Rekursionsbaum für `f 4`.

(b) Schreiben Sie eine iterative Prozedur `fi`, die f berechnet. Tipp: Denken Sie an `fib`.

(c) Zeichnen Sie den Rekursionsbaum für `fi 4`.

Aufgabe 11.3: Reverses Map (3 Punkte)

Schreiben Sie eine iterative Prozedur

```
rmap: ('a -> 'b) -> 'a list -> 'b list
```

die die gleiche Funktion berechnet wie

```
fun f g xs = rev(map g xs)
```

Sie sollen keine vordefinierten Prozeduren benutzen.

Aufgabe 11.4: Asymptotische Laufzeit (6 Punkte)

Gegeben sei eine Prozedur `l: int -> unit` mit linearer Laufzeit für Argumente in $\mathbb{N}$. Bestimmen Sie für die folgenden Prozeduren `f: int -> unit` jeweils die asymptotische Laufzeit (mit Θ) für Argumente $n \in \mathbb{N}$.

(a) `fun f n = if n < 10 then () else f(n div 2)`

(b) `fun f n = if n < 10 then () else f(n mod 2)`

(c) `fun f n = if n = 0 then () else ( l(n) ; f(n-1) )`

(d) `fun f n = if n = 0 then () else ( f(n div 2) ; f(n div 2) )`

(e) `fun f n = if n = 0 then () else ( f(n div 2) ; l(n) ; f(n div 2) )`

(f) `fun f n = if n = 0 then () else f( f(n div 2) ; n div 2 )`

Aufgabe 11.5: Ganzzahlige Kubikwurzel (8 Punkte)

Die Funktion

$$r \in \mathbb{N} \to \mathbb{N}$$
$$r(n) = \max\{k \in \mathbb{N} \mid k^3 \leq n\}$$

heißt *ganzzahlige Kubikwurzel*. Offensichtlich gilt:

$$\forall m,n \in \mathbb{N}: \quad m^3 \leq n \Leftrightarrow m \leq r(n)$$

Betrachten Sie die Prozeduren

```
fun f(m,n) = if m*m*m > n then m-1 else f(m+1,n)
fun g(n)   = f(0,n)
```

die zwei partielle Funktionen $f \in \mathbb{N} \times \mathbb{Z} \xrightarrow{\text{par}} \mathbb{Z}$ und $g \in \mathbb{Z} \xrightarrow{\text{par}} \mathbb{Z}$ definieren.

(a) Zeigen Sie, dass `f` für $(m,n) \in \mathbb{N} \times \mathbb{Z}$ terminiert, indem sie die Terminationsbedingungen und eine geeignete wohlfundierte Relation angeben.

(b) Zeigen Sie:

$$\forall n \in \mathbb{N}: \quad g(n) = r(n)$$

Beweisen Sie dazu zunächst:

$$\forall m,n \in \mathbb{N}: \quad m \leq r(n) + 1 \Rightarrow f(m,n) = r(n)$$

bzw. die äquivalente Behauptung

$$\forall k,m,n \in \mathbb{N}: \quad m + k = r(n) + 1 \Rightarrow f(m,n) = r(n)$$

durch Induktion über k.

(c) Was ist die asymptotische Laufzeit von `g` (mit Θ)?

Aufgabe 11.6: Rationale Zahlen (10 Punkte)

Jede rationale Zahl $q \in \mathbb{Q}$ mit $q \neq 0$ lässt sich eindeutig darstellen als

$$q = \frac{m}{n} \quad \text{mit } m \in \mathbb{Z}, n \in \mathbb{N}^+, ggT(m,n) = 1$$

also durch ein Paar $(m,n) \in \mathbb{Z} \times \mathbb{N}^+$. Schreiben Sie darauf basierend eine Struktur `Rational` mit der Signatur

```
signature RATIONAL =
  sig
    eqtype rat
    val rat:     int * int -> rat    (* m/n (n>0)  [Domain] *)
    val num:     rat -> int          (* Zähler (numerator) *)
    val denom:   rat -> int          (* Nenner (denominator) *)
    val neg:     rat -> rat          (* -q *)
    val invert:  rat -> rat          (* 1/q  [Div] *)
    val plus:    rat * rat -> rat    (* p + q *)
    val minus:   rat * rat -> rat    (* p - q *)
    val times:   rat * rat -> rat    (* p * q *)
    val divide:  rat * rat -> rat    (* p / q  [Div]*)
  end
```

die das Rechnen mit rationalen Zahlen ermöglicht. Stellen Sie 0 als $0/1$ dar.

Die Prozedur `rat` soll die vordefinierte Ausnahme `Domain` werfen, falls das zweite Argument nicht positiv ist, `invert` und `divide` sollen entsprechend `Div` werfen, falls $q = 0$.

Achten Sie darauf, die eindeutige Darstellung zu erreichen, indem Sie Ergebnisse kürzen und Vorzeichen korrekt behandeln.

Aufgabe 11.7: Multimengen (12 Punkte)

Multimengen kann man sich als Mengen vorstellen, in denen Elemente mehrfach auftreten können. Formal ist eine Multimenge über X eine Funktion $X \to \mathbb{N}$. Wir betrachten die folgenden Operationen auf Multimengen:

$$MS(X) \;=\; X \to \mathbb{N}$$

$$count \in MS(X) \times X \to \mathbb{N}$$
$$count(m,x) \;=\; m(x)$$

$$empty \in MS(X)$$
$$empty \;=\; \lambda x \in X.0$$

$$insert \in MS(X) \times X \to MS(X)$$
$$insert(m,x') \;=\; \lambda x \in X.\text{if } x = x' \text{ then } m(x)+1 \text{ else } m(x)$$

$$union \in MS(X) \times MS(X) \to MS(X)$$
$$union(m_1,m_2) \;=\; \lambda x \in X.m_1(x)+m_2(x)$$

(a) Deklarieren Sie eine Signatur `MULTISET`, die Multimengen mit einem abstrakten Typ

```
type ''a mset
```

und den obigen Operationen beschreibt.

(b) Deklarieren Sie eine Struktur `Multiset1` mit der Signatur `MULTISET`, die Multimengen mit

```
type ''a mset = ''a -> int
```

implementiert.

(c) Schreiben Sie eine Struktur `Multiset2` mit der Signatur `MULTISET`, die Multimengen mit

```
type ''a mset = (''a * int) list
```

realisiert.