DEPENDABLE SYSTEMS AND SOFTWARE
Fachrichtung 6.2 — Informatik
Prof. Dr.-Ing. Holger Hermanns
Dipl.-Inform. Lijun Zhang

Übungsblatt 11 (Programmierung I)

Lesen Sie im Skript: Kapitel 10

Aufgabe 11.1: (Fibonacci-Zahlen)
Die Fibonacci-Zahlen lassen sich auch ohne die höherstufige Prozedur *iter* endrekursiv bestimmen. Dazu spezialisiert man die Prozedur *iter* gemäß der für die Fibonacci-Zahlen verwendeten Schrittfunktion f.

(a) Konstruieren Sie eine Prozedur $\mathit{fibi} : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N}$, für die $\mathit{fib}\ n = \mathit{fibi}\ 0\ 1\ (n-1)$ für alle Argumente $n \in \mathbb{N}^+$ gilt. Die ersten beiden Argumente sollen als Akkumulatorargumente dienen.

(b) Schreiben Sie in Standard ML eine Prozedur fib', die Fibonacci-Zahlen mithilfe der endrekursiven Prozedur *fibi* berechnet.

Aufgabe 11.2: (Ergebnisfunktion von gcd mit first berechnen)
Zeigen Sie, wie man die Ergebnisfunktion der Prozedur *gcd* aus Abbildung 9.1 mit der Prozedur *first* aus Kapitel 10.2 berechnen kann. Beweisen Sie die Korrektheit Ihrer Konstruktion.

Aufgabe 11.3: (rev)
Geben Sie für die Prozedur *rev* die Rekursionsfunktion, die Rekursionsrelation sowie eine natürliche Terminierungsfunktion an.

Aufgabe 11.4: (Endrekursion und Korrektheit)
Die Länge von Listen lässt sich mit einer endrekursiven Prozedur bestimmen, die die folgende Funktion berechnet:

$$f : \mathbb{N} \to \mathcal{L}(X) \to \mathbb{N}$$
$$f\ a\ xs = a + |xs|$$

(a) Konstruieren Sie eine endrekursive Prozedur *leni*, die die Funktion f berechnet und beweisen Sie die Korrektheit Ihrer Prozedur (Korrektheit per Konstruktion).

(b) Schreiben Sie in Standard ML eine Prozedur *len*, die die Länge von Listen mithilfe einer endrekursiven Prozedur *leni* berechnet.

Aufgabe 11.5: (Strukturelle Induktion)
Sei X eine Menge. Beweisen Sie mit struktureller Induktion, dass für alle Listen $xs, ys \in \mathcal{L}(X)$ gilt:

(a) $|xs@ys| = |xs| + |ys|$

(b) $|rev\ xs| = |xs|$

Aufgabe 11.6: (strukturelle Induktion)
Sei X eine Menge. Beweisen Sie mit struktureller Induktion, dass für alle Listen $xs, ys \in \mathcal{L}(X)$ gilt:

(a) $xs@nil = xs$

(b) $rev(xs@ys) = rev\ ys\ @\ rev\ xs$

(c) $rev(rev\ xs) = xs$

Aufgabe 11.7: (strukturelle Induktion über Listen)
Seien X, Y Mengen und f eine Funktion $X \times Y \to Y$. Beweisen Sie durch strukturelle Induktion über xs, dass gilt:

$$\forall xs \in \mathcal{L}(X)\ \forall xs' \in \mathcal{L}(X)\ \forall y \in Y : \mathit{foldl}\ f\ y\ (xs@xs') = \mathit{foldl}\ f\ (\mathit{foldl}\ f\ y\ xs)\ xs'$$

Aufgabe 11.8: (Endrekursives Reversieren)
Listen lassen sich mit einer endrekursiven Prozedur reversieren, die die folgende Funktion berechnet:

$$\begin{aligned}&\mathit{revi} : \mathcal{L}(X) \to \mathcal{L}(X) \to \mathcal{L}(X)\\ &\mathit{revi}\ xs\ ys\ =\ (\mathit{rev}\ ys)\ @\ xs\end{aligned}$$

(a) Konstruieren Sie eine endrekursive Prozedur *revi*, die die Funktion *revi* berechnet und beweisen Sie die Korrektheit Ihrer Prozedur (Korrektheit per Konstruktion). Hinweis: Sie benötigen dafür die Assoziativität der Konkatenation (Proposition 10.5)

(b) Schreiben Sie in Standard ML eine Prozedur rev', die die Länge von Listen mithilfe einer endrekursiven Prozedur *revi* berechnet.

Aufgabe 11.9: (Reversion mit foldl)
In §4.4 haben Sie gelernt, dass Listen mit *foldl* reversiert werden können. Jetzt können Sie die Korrektheit dieses Vorgehens beweisen.
Sei X eine Menge und sei f die Funktion $\lambda(x, xs) \in X \times \mathcal{L}(X).\ x :: xs$. Für die Korrektheit der Reversion mit *foldl* muss die Gültigkeit der Aussage $\forall xs \in \mathcal{L}(X) :\ \mathit{rev}\ xs = \mathit{foldl}\ f\ \mathit{nil}\ xs$ gezeigt werden.
Suchen Sie eine geeignete Verstärkung dieser Korrektheitsaussage und beweisen Sie die Gültigkeit der Verstärkung durch strukturelle Induktion über xs.

Aufgabe 11.10: (Balancierte Binärbäume)

(a) Wieviele Blätter hat ein balancierter Binärbaum der Tiefe 10?

(b) Wieviele Knoten hat ein balancierter Binärbaum der Tiefe 10?

(c) Wieviele innere Knoten hat ein balancierter Binärbaum der Tiefe n?

(d) Hat ein balancierter Binärbaum der Tiefe n mehr Blätter oder mehr innere Knoten? Wieviele Knoten beträgt der Unterschied?

Aufgabe 11.11: (Induktionsloser Beweis)
Beweisen Sie Proposition 10.7 ohne Induktion mithilfe von Proposition 9.8.

Aufgabe 11.12: (Binäre Bäume)
Ein binärer Baum ist ein Baum, bei dem jeder innere Knoten genau zwei Nachfolger hat. Sei X eine Menge.

- Definieren Sie die Menge $\mathcal{M}(X) \subseteq \mathcal{T}(X)$ der binären Bäume über X formal.
- Beweisen Sie: $\forall t \in \mathcal{M}(X) : b(t) \leq 2^{d(t)}$.
- Beweisen Sie: $\forall t \in \mathcal{M}(X) : s(t) \leq 2^{d(t)+1} - 1$.

Aufgabe 11.13: (Balancierte ternäre Bäume)
Unter einem *ternären Baum* wollen wir einen Baum verstehen, bei dem jeder innere Knoten genau drei Nachfolger hat. Sei X eine Menge.

(a) Definieren Sie die Menge $\mathcal{M}(X) \subseteq \mathcal{T}(X)$ der balancierten ternären Bäume über X formal.

(b) Beweisen Sie: $\forall t \in \mathcal{M}(X) :\ b(t) = 3^{d(t)}$

(c) Beweisen Sie: $\forall t \in \mathcal{M}(X) :\ s(t) = \frac{1}{2}(3^{d(t)+1} - 1)$