REACTIVE SYSTEMS GROUP

Universität des Saarlandes

Prof. Bernd Finkbeiner, Ph.D.
Markus Rabe, M.Sc.

Programmierung 1 (SS 2010) - 10. Übungsblatt

http://react.cs.uni-saarland.de/prog1/

Lesen Sie im Buch bis zum Ende von Kapitel 10

Aufgabe 10.1

Beweisen Sie folgende Aussagen durch Induktion.

a) $\forall n \in \mathbb{N} : 3n \leq 3^n$

b) $\forall n \in \mathbb{N}\ \exists k \in \mathbb{N} : n^3 - n = 3k$

Aufgabe 10.2

Zeigen Sie, dass *iter* und die unten definierte Prozedur *iter'* semantisch äquivalent sind.

$$
\begin{aligned}
&\mathit{iter}' : \mathbb{N} \times X \times (X \to X) \to X \\
&\mathit{iter}'\ (0, x, f) = x \\
&\mathit{iter}'\ (n, x, f) = f(\mathit{iter}'\ (n-1, x, f)) \quad \text{für } n > 0
\end{aligned}
$$

Aufgabe 10.3

Seien *fac*, *fib* und *iter* wie in Kapitel 10.2 gegeben. Seien $f = \lambda\,(k, x).\,(k+1,\ k \cdot x)$ und $g = \lambda\,(x, y).\,(y,\ x+y)$. Beweisen Sie:

a) $\forall n \in \mathbb{N} : f(n+1,\ \mathit{fac}\ n) \ = \ (n+2,\ \mathit{fac}\ (n+1))$

b) $\forall n \in \mathbb{N} : (n+1,\ \mathit{fac}\ n) \ = \ \mathit{iter}(n, (1,1), f)$

c) $\forall n \in \mathbb{N}_+ : g(\mathit{fib}(n-1),\ \mathit{fib}\ n) = (\mathit{fib}\ n,\ \mathit{fib}(n+1))$

d) $\forall n \in \mathbb{N}_+ : (\mathit{fib}(n-1),\ \mathit{fib}\ n) = \mathit{iter}\ (n-1, (0,1), g)$

Für den Beweis von (a) bzw. (c) benötigen Sie nur die Definitionen von f und *fac* bzw. g und *fib*. Der Beweis von (b) bzw. (d) gelingt mit Induktion sowie Teil (a) bzw. (c) und Proposition 10.1 auf S. 202. Orientieren Sie sich am Beweis von Proposition 10.2.

Aufgabe 10.4

Schreiben Sie in Standard ML eine Prozedur $\mathit{fib}' : \mathit{int} \to \mathit{int}$, die $fib\ n$ für $n \in \mathbb{N}$ mit Hilfe von *iter* endrekursiv bestimmt. Überzeugen Sie sich mit dem interpreter davon, dass fib' für größere n sehr viel schneller rechnet als *fib*. Beginnen Sie mit $n = 40$.

Aufgabe 10.5

Zeigen Sie, wie man die Ergebnisfunktion der Prozedur *gcd* (aus Abbildung 9.1 auf S. 180 im Buch) mit *first* berechnen kann. Beweisen Sie die Korrektheit Ihrer Konstruktion.

Aufgabe 10.6

Betrachten Sie die Prozedur

$$\begin{aligned}
&p : \mathbb{N} \to \mathbb{N} \\
&p\ 0 = 0 \\
&p\ 1 = 1 \\
&p\ n = p(n-1) + p(p(n-1) - p(n-2)) \quad \text{für } n > 1
\end{aligned}$$

Wegen der geschachtelten Rekursion lässt sich die Terminierung der Prozedur nicht ohne Informationen über die Ergebnisfunktion der Prozedur zeigen. Beweisen Sie durch natürliche Induktion, dass die Prozedur die Identitätsfunktion $\lambda\, n \in \mathbb{N}.\ n$ berechnet.

Aufgabe 10.7

Sei die folgende Prozedur gegeben:

$$\begin{aligned}
&p : \mathbb{Z}^2 \to \mathbb{Z} \\
&p(x, y) = 0 && \text{für } x > y \\
&p(x, y) = x + p(x+1, y) && \text{für } x \leq y
\end{aligned}$$

a) Geben Sie eine natürliche Terminierungsfunktion für p an.

b) Zeigen Sie durch Induktion: $\forall (x, y) \in \mathbb{Z}^2, x \leq y :\ p(x, y) = p(x, y-1) + y$.

Aufgabe 10.8

Geben Sie für die Prozedur *rev* (vgl. Abbildung 10.1 auf Seite 206 im Buch) die Rekursionsfunktion, die Rekursionsrelation sowie eine natürliche Terminierungsfunktion an.

Aufgabe 10.9

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 10.10

Listen lassen sich mit einer endrekursiven Prozedur reversieren, die die folgende Funktion berechnet:

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

Konstruieren Sie eine endrekursive Prozedur $revi'$, die die Funktion *revi* berechnet und beweisen Sie die Korrektheit Ihrer Prozedur. Hinweis: Sie benötigen dafür die Assoziativität der Konkatenation (Proposition 10.5 im Buch).

Aufgabe 10.11 (Buch 10.4)

Zeigen Sie Proposition 10.2 (Seite 202 im Buch) ohne die Benutzung der Vertauschungseigenschaft. Zeigen Sie zunächst die allgemeinere Aussage (die verstärkte Aussage)

$$\forall x \in \mathbb{Z}\ \forall n \in \mathbb{N}\ \forall s \in \mathbb{Z} :\ iter\ (n, s, \lambda a.\ a \cdot x) = s \cdot x^n.$$

Aufgabe 10.12

In Kapitel 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 $\forall xs \in \mathcal{L}(X) : rev\ xs = foldl(f, nil, xs)$ gezeigt werden.
Suchen Sie eine geeignete Verstärkung dieser Korrektheitsaussage und beweisen Sie die Gültigkeit der verstärkten Aussage durch strukturelle Induktion über xs. Orientieren Sie sich an der Funktion *revi* aus Aufgabe 10.10.

Aufgabe 10.13

Unter einem *ternären Baum* wollen wir einen Baum verstehen, bei dem jeder innere Knoten genau drei Nachfolger hat.

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

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

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

Aufgabe 10.14

Geben Sie die Laufzeit und die Rekursionsfolge der Prozedur @ für das Argument $([1,2],\ [3,4,5,6])$ an.

Aufgabe 10.15

Geben Sie für die folgende Prozedur eine Größenfunktion und eine entsprechende Laufzeitfunktion an:

$$\begin{aligned}
&\mathit{revi} : \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X)\\
&\mathit{revi}(\mathit{xs},\ \mathit{nil}) = \mathit{xs}\\
&\mathit{revi}(\mathit{xs},\ y :: \mathit{yr}) = \mathit{revi}(y :: \mathit{xs},\ \mathit{yr})
\end{aligned}$$

Aufgabe 10.16

Geben Sie für die Prozedur *iter* eine Größenfunktion und die entsprechende Laufzeitfunktion an:

$$\begin{aligned}
&\mathit{iter} : (X \to X) \to \mathbb{N} \to X \to X\\
&\mathit{iter}\ f\ 0\ x = x\\
&\mathit{iter}\ f\ n\ x = \mathit{iter}\ f\ (n-1)\ (fx) \quad \text{für } n > 0
\end{aligned}$$