Dr. Theo Lettmann

Paderborn, den 07. November
Abgabe 17. November 2002

Übungen zur Vorlesung

Modellierung

WS 2003/2004

Blatt 4

AUFGABE 25 :
Beweise durch vollständige Induktion:

$$\sum_{k=1}^{n}(2k-1) = n^2$$

AUFGABE 26 :
Beantworte die folgenden Fragen möglichst ausführlich:

1. Was sind die Unterschiede zwischen konkreten und abstrakten Algebren?
2. Was ist die Normalform eines Terms?
3. Beschreibe die Funktionsweise eines Kellers (Stack) und einer Warteschlange (Queue).

AUFGABE 27 :
Die Datenstruktur *Schlange* verwaltet Elemente in der Reihenfolge ihres Eintreffens. Operationen auf einer Schlange erlauben es, Elemente hinten anzufügen oder vorne die am längsten wartenden Elemente zu entnehmen. Schlangen arbeiten nach dem FIFO-Prinzip (First-In-First-Out). Sie sind z.B. an der Kasse im Supermarkt als Warteschlange zu finden.
Die folgende abstrakte Algebra beschreibt eine *Schlange* $= (T, \Sigma, Q)$ von natürlichen Zahlen (als Elemente) mit der **Signatur** $\Sigma = (S, F)$:

$$\begin{array}{lllll}
S = \{ & \textit{Queue}, Nat, \textit{Bool} \quad \} & & & (S_1) \\
F = \{ & \textit{createQueue}: & & \rightarrow \textit{Queue}, & (F_1) \\
 & \textit{enqueue}: & \textit{Queue} \times Nat & \rightarrow \textit{Queue}, & (F_2) \\
 & \textit{dequeue}: & \textit{Queue} & \rightarrow \textit{Queue}, & (F_3) \\
 & \textit{front}: & \textit{Queue} & \rightarrow Nat, & (F_4) \\
 & \textit{empty}: & \textit{Queue} & \rightarrow \textit{Bool} \quad \} & (F_5)
\end{array}$$

Die Sorte *Queue* stellt eine Schlange dar, die Sorte Nat beschreibt Elemente der Schlange. Für die Sorte *Bool* sind die Konstanten *true* und *false* definiert.
createQueue bezeichnet eine 0-stellige Operation und ist damit eine Konstante. Sie steht für die leere Schlange. Die Operation *enqueue* fügt ein Element am Ende der Schlange an. Die Operation *front* liefert das erste Element vorne in der Schlange, *dequeue* entfernt dieses Element aus der Schlange. Ob die Schlange leer ist oder nicht, zeigt die Operation *empty* an.
Für die Menge der **Axiome** Q seien x, y Variablen der Sorte *Element* und q eine Variable der Sorte *Queue*:

$$
\begin{array}{llll}
Q = \{ & \textit{dequeue}(\textit{enqueue}(\textit{createQueue}, x)) & \equiv \textit{createQueue}, & (Q_1)\\
& \textit{dequeue}(\textit{enqueue}(\textit{enqueue}(q, y), x)) & \equiv \textit{enqueue}(\textit{dequeue}(\textit{enqueue}(q, y)), x), & (Q_2)\\
& \textit{front}(\textit{enqueue}(\textit{createQueue}, x)) & \equiv x, & (Q_3)\\
& \textit{front}(\textit{enqueue}(\textit{enqueue}(q, x))) & \equiv \textit{front}(\textit{enqueue}(q, x)), & (Q_4)\\
& \textit{empty}(\textit{createQueue}) & \equiv \textit{true}, & (Q_5)\\
& \textit{empty}(\textit{enqueue}(q, x)) & \equiv \textit{false}, \quad \} & (Q_6)
\end{array}
$$

Hinweis: Hinter den einzelnen Spezifikationen sind Beschriftungen in Klammern angegeben. Beziehen Sie sich auf diese in Ihrer Lösung. Im folgenden seien *x, y, z* Variable der Sorte *Element*.

(a) Axiome anwenden

Formen Sie die folgenden Terme mit Hilfe der Axiome so um, dass Sie als Ergebnis eine einzelne Konstante oder Variable erhalten. Notieren Sie bei jeder Umformung das benutzte Axiom.

1. *empty(enqueue(createQueue, y))*
2. *front(dequeue(enqueue(enqueue(createQueue, z), x)))*
3. *empty(enqueue(dequeue(enqueue(createQueue, x)), z))*
4. *empty(dequeue(enqueue(createQueue, x)))*
5. *dequeue(enqueue(createQueue, front(enqueue(enqueue(createQueue, x), z))))*

(b) Algebra erweitern

Erweitern Sie die abstrakte Algebra „Schlange“ um die Operation *sum*, die die Summe aller Elemente der Schlage ausgibt, sowie die hierfür benötigten Operationen +, 1 und 0. Dazu ist es notwendig auch die Menge der Axiome zu erweitern.

AUFGABE 28 :
Unifikation: Es seien x,y,z Variablen und a,b,c,0 Konstanten. Ermitteln Sie jeweils einen allgemeinsten Unifikator, falls ein solcher existiert.

1. plus(x,0,y) und plus(a,y,y)
2. f(x,y,z) und f(g(y),g(z),a)
3. f(x,x) und f(y,g(y))
4. f(x, g(x)) und f(g(g(y)), g(z))
5. R(g(x), a, g(z) ,c) und R(y, x, g(a), g(x))

AUFGABE 29 :
Erstelle eine abstrakte Algebra zum Datentyp Binärbaum, dessen Knoten Zahlen enthalten.
Ein Binärbaumknoten bzw. -blatt hat also also immer maximal 2 Nachfolger. Diese Nachfolger sind dann entweder Blätter oder weitere Binärbäume.
In dieser Aufgabe sind Teile bereit gelöst. Schreiben Sie jedoch in der Lösung die komplette Definition der Algebra. Ihre Lösung muss also enthalten:

1. Sorten
2. Operationen
3. Axiome (mind. 3)

Geben Sie zum Schluss eine Klassifikation der Operationen an (Konstruktoren, Hilfskonstruktoren, Projektionen).

Die abstrakte Algebra *Binärbaum* $= (T, \Sigma, Q)$ mit der **Signatur** $\Sigma = (S, F)$ wird beschrieben durch:

$$
\begin{array}{rllllr}
S = \{ & BinTree, Nat, \textit{Bool} \quad \} & & & & (S_1) \\
F = \{ & createBinTree: & & \to & BinTree, & (F_1) \\
 & bin: & BinTree \times Nat \times BinTree & \to & BinTree, & (F_2) \\
 & left: & & \to & & (F_3) \\
 & right: & & \to & & (F_4) \\
 & value: & & \to & & (F_5) \\
 & empty: & & \to & & (F_6)
\end{array}
$$

Die Sorte *BinTree* stellt einen Binärbaum dar, die Sorte *Nat* beschreibt ein Blatt.

Funktionsbeschreibung:

create: Erzeugt einen leeren Binärbaum
bin: Erzeugt einen nicht leeren Binärbaum
left: Beschreibt den linken Teilbaum
right: Beschreibt den rechten Teilbaum
value: Beschreibt den Wert im Knoten (aus Nat)
empty: Gibt an, ob der Baum leer ist

AUFGABE 30 :
Benenne die folgenden Gesetze und zeige die Gültigkeit mit Hilfe von Wahrheitstafeln:

1. $\neg(X \wedge Y) = \neg X \vee \neg Y$
2. $X \wedge Y = Y \wedge X$
3. $(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$

KORREKTURAUFGABE 31 :
Gebe die Wahrheitstafeln zu folgenden Formeln an. Als Spalten müssen hier die Variablen, sinnvolle und wichtige Einzelaussagen sowie die Gesamtaussage angegeben sein.
Beispiel: Für die Funktion $\neg(X \to (Y \wedge \neg Z)) \to (\neg X)$ sollten ca. 7 Spalten angeben werden $(X, Y, Z, \neg X, Y \wedge \neg Z, \neg(X \to (Y \wedge \neg Z)), \neg(X \to (Y \wedge \neg Z)) \to (\neg X))$.
Überprüfe überdies die 4 Eigenschaften der Formeln (erfüllbar, falsifizierbar, unerfüllbar, tautologisch) und forme die Formeln in eine äquivalente NNF um.

(1) $\neg(\neg Y \vee (Z \to (\neg X \vee Z)) \wedge \neg(X \wedge Y))$

(2) $X \to (Y \leftrightarrow Z)$

(3) $\neg((\neg X \vee \neg Y \vee \neg Z) \vee (\neg Y \to Z) \vee Y)$