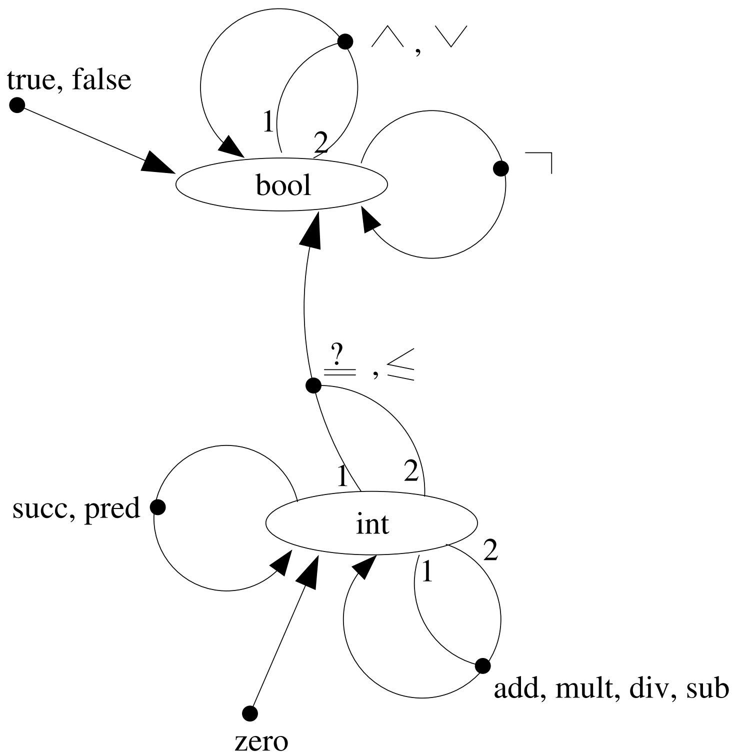
Technische Universität München
Fakultät für Informatik
M. Bauer, D. Golias, S. Riesner,
Dr. M. Schneider

WS 2002/2003
Lösungsvorschläge zu Blatt 9
19. Dezember 2002

Übungen zu Einführung in die Informatik I

Aufgabe 31 Die Signatur der Rechenstruktur der ganzen Zahlen (Lösungsvorschlag)

Der Signaturgraph der ganzen Zahlen mit den Sorten $\mathbf{S} = \{\mathbf{bool}, \mathbf{int}\}$ und den Funktionssymbolen $\mathtt{F} = \{\mathtt{true}, \mathtt{false}, \wedge, \vee, \neg, \mathtt{zero}, \mathtt{succ}, \mathtt{pred}, \mathtt{add}, \mathtt{mult}, \mathtt{sub}, \mathtt{div}, \stackrel{?}{=}, \leq\}$ hat die Form:



Durch Umbenennung der Sorte **int** in **nat** ergibt sich der Signaturgraph der Rechenstruktur der natürlichen Zahlen.

Um zu zeigen, dass die Spezifikationen der beiden Rechenstrukturen unterschiedlich sind, betrachten wir die Spezifikation des Funktionssymboles `pred`:Für die ganzen Zahlen gilt

$$\mathtt{pred}^{\mathbf{INT}}(z) = z - 1$$

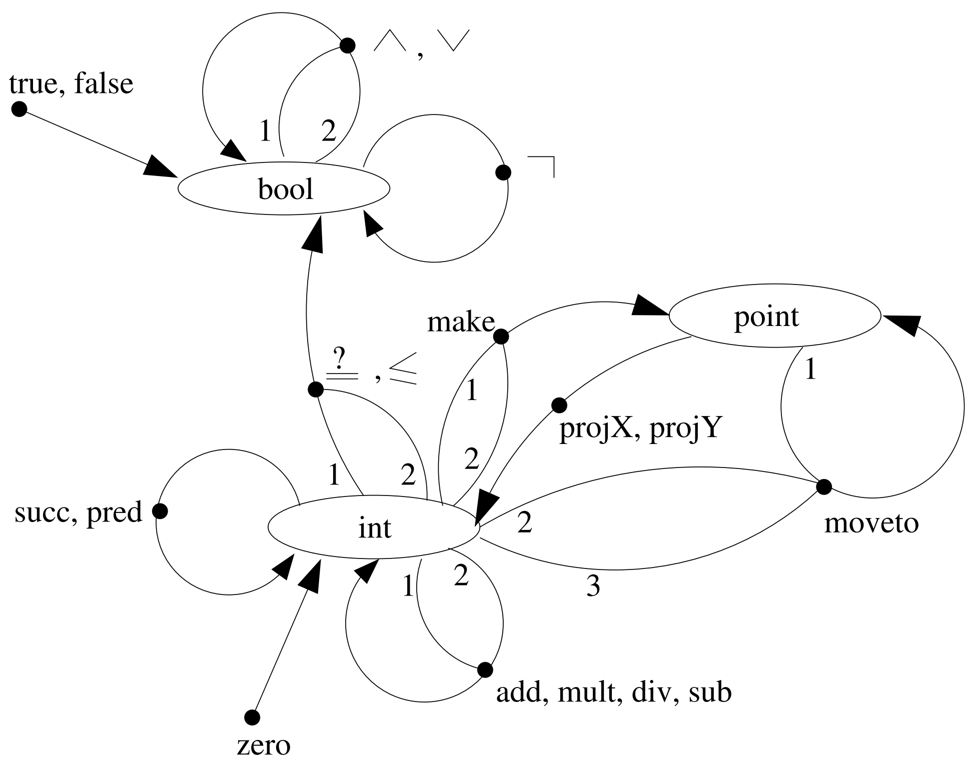
wohingegen bei der Rechenstruktur **NAT** diese Funktion durch

$$\mathtt{pred}^{\mathbf{NAT}}(x) = \begin{cases} x-1 & x \geq 1 \\ \perp & x = 0 \end{cases}$$

gegeben ist.

Aufgabe 32 Kartesisches Koordinatensystem (Lösungsvorschlag)

a) Der Signaturgraph der Rechenstruktur **K** besitzt die Form:



b) (i) syntaktisch korrekt

(ii) syntaktisch inkorrekt

(iii) syntaktisch inkorrekt

c) Die gesuchte formale Spezifikation lautet:

$$
\begin{array}{lcl}
\mathbf{bool}^{\mathbf{K}} & = & \mathbb{B}^{\perp} \\
\mathbf{int}^{\mathbf{K}} & = & \mathbb{Z}^{\perp} \\
\mathbf{point}^{\mathbf{K}} & = & \mathbb{Z}^{\perp} \times \mathbb{Z}^{\perp} \\
\text{make}^{\mathbf{K}}(x,y) & = & (x,y) \\
\text{projX}^{\mathbf{K}}(\text{make}^{\mathbf{K}}(x,y)) & = & x \\
\text{projY}^{\mathbf{K}}(\text{make}^{\mathbf{K}}(x,y)) & = & y \\
\text{moveto}^{\mathbf{K}}(p,dx,dy) & = & \text{make}^{\mathbf{K}}(\text{projX}^{\mathbf{K}}(p)+dx, \text{projY}^{\mathbf{K}}(P)+dy))
\end{array}
$$

d) Eine Implementierung der oben spezifizierten Sorten und Funktionen könnte lauten:

```
type point = Make of (int * int);;

let projX p = match p  with
| Make (x, y) -> x;;

let projY p = match p  with
| Make (x, y) -> y;;

let moveto p dx dy = Make ((projX p + dx), (projY p + dy));;
```

Aufgabe 33 Termersetzung: Sequenzen (Lösungsvorschlag)

Das Termersetzungssystem ist ziemlich einfach aufgebaut: zuerst stellen wir Regeln auf die im Falle zweier Unterschiedlich langer Listen dann *false* ergeben, bzw. *true* für zwei leere Listen. Anschließend reduzieren wir beide Listen indem wir den Kopf vergleichen und den Rest der Liste unberührt lassen.

Beachten Sie hier die Ähnlichkeit zum Pattern-Matching das bei der Implementierung in OCaml verwendet werden würde.

1. $\texttt{kompl(Empty, x)} \to \texttt{x} \overset{?}{=} \texttt{Empty}$
2. $\texttt{kompl(Elem(a, x), Empty)} \to \texttt{false}$
3. $\texttt{kompl(Elem(a, x), Elem(b, y))} \to \texttt{(a} \overset{?}{=} \neg\texttt{b)} \wedge \texttt{kompl(x, y)}$

Für die Vereinfachung von Booleschen Termen mit der Vergleichsoperation $\overset{?}{=}$ auf Sequenzen bzw. auf Booleschen Termen werden Vereinfachungsregeln vorausgesetzt:

4. $\neg\texttt{true} \to \texttt{false}$
5. $\neg\texttt{false} \to \texttt{true}$
6. $\texttt{true} \overset{?}{=} \texttt{true} \to \texttt{true}$
7. $\texttt{false} \overset{?}{=} \texttt{false} \to \texttt{true}$
8. $\texttt{true} \overset{?}{=} \texttt{false} \to \texttt{false}$
9. $\texttt{false} \overset{?}{=} \texttt{true} \to \texttt{false}$
10. $\texttt{true} \wedge \texttt{true} \to \texttt{true}$
11. $\texttt{false} \wedge \texttt{true} \to \texttt{false}$
12. $\texttt{true} \wedge \texttt{false} \to \texttt{false}$
13. $\texttt{false} \wedge \texttt{false} \to \texttt{false}$

Im Beispiel ergibt das dann:

$\texttt{kompl(Elem(true, Elem(false, Elem(false, Elem(true, Elem(true, Empty))))), Elem(false, Elem(true, Elem(true, Elem(false, Elem(false, Empty))))));;}$

$\xrightarrow{3} \texttt{true} \overset{?}{=} \neg\texttt{false} \wedge \texttt{kompl(Elem(false, Elem(false, Elem(true, Elem(true, Empty))))), Elem(true, Elem(true, Elem(false, Elem(false, Empty))))));;}$

$\xrightarrow{3} \texttt{true} \overset{?}{=} \neg\texttt{false} \wedge \texttt{false} \overset{?}{=} \neg\texttt{true} \wedge \texttt{kompl(Elem(false, Elem(true, Elem(true, Empty))))), Elem(true, Elem(false, Elem(false, Empty))))));;}$

$\xrightarrow{3} \texttt{true} \overset{?}{=} \neg\texttt{false} \wedge \texttt{false} \overset{?}{=} \neg\texttt{true} \wedge \texttt{false} \overset{?}{=} \neg\texttt{true} \wedge \texttt{kompl(Elem(true, Elem(true, Empty))))), Elem(false, Elem(false, Empty))))));;}$

$\xrightarrow{3} \ldots$

Aufgabe 34 Induktion über den Termaufbau: Negationsnormalform (Lösungsvorschlag)

Vorbemerkung:
Aufgrund der Gesetze

$$\begin{aligned}
(t_1 \Rightarrow t_2) &= (\neg t_1 \vee t_2)\\
(t_1 \Leftrightarrow t_2) &= (t_1 \wedge t_2) \vee (\neg t_1 \wedge \neg t_2)\\
true &= (x \vee \neg x)\\
false &= \neg true
\end{aligned}$$

für beliebige Boolesche Terme (BT) t_1, t_2 und Identifikator x, können wir $\Rightarrow$, $\Leftrightarrow$, $true$, $false$ auf $\vee$, $\wedge$, $\neg$ zurückführen. Wir betrachten daher nur BT gebildet mit den Operatoren $\vee$, $\wedge$ und $\neg$ über Identifikatoren aus ID und elementaren Aussagen in E.

Induktion über den Termaufbau:
Aufgrund der induktiven Definition von BT gilt folgendes Induktionsprinzip für BT:
Sei $P(t)$ ein Prädikat über BT t. Wenn gilt

a) $P(t)$ gilt für alle atomaren elementaren Aussagen in E und alle Identifikatoren in ID, d.h. für alle $t \in E \cup ID$.

b) Aus $P(t)$ folgt $P(\neg t)$ für alle BT t.

c) Aus $P(t_1), P(t_2)$ folgen $P(t_1 \wedge t_2)$ und $P(t_1 \vee t_2)$ für alle BT t_1, t_2.

so gilt $P(t)$ für alle BT t.

Induktionsbeweis für Existenz der Negationsnormalform (NNF):
Wir definieren als spezielles Prädikat

$$P(t) = \text{“Die BT } t \text{ und } \neg t \text{ lassen sich auf semantisch äquivalente BT in NNF reduzieren”}$$

Induktionsbeginn (IB):
Zu zeigen ist (a) für unser spezielles P. Sei $t \in E \cup ID$. t und $\neg t$ sind bereits in NNF und jeweils zu sich selbst semantisch äquivalent.

Induktionsschritt (IS):
Zu zeigen sind (b) und (c) für unser spezielles P.
zu (b): Als Induktionshypothese (IH) gelte $P(t)$, d.h. t und $\neg t$ lassen sich auf semantisch äquivalente BT in NNF reduzieren. Zu zeigen ist $P(\neg t)$, d.h. $\neg t$ und $\neg(\neg t)$ lassen sich auf semantisch äquivalente BT in NNF reduzieren. Wegen $\neg(\neg t) = t$ folgt dies aus der IH.
zu (c): Als IH gelte $P(t_1)$ und $P(t_2)$. Äquivalente BT in NNF für t_1, t_2 seien s_1, s_2 und für $\neg t_1, \neg t_2$ seien dies r_1, r_2. Dann gilt

$$\begin{aligned}
(t_1 \wedge t_2) &= (s_1 \wedge s_2) \text{ ist in NNF} \\
(t_1 \vee t_2) &= (s_1 \vee s_2) \text{ ist in NNF} \\
\neg(t_1 \wedge t_2) &= ((\neg t_1) \vee (\neg t_2)) = (r_1 \vee r_2) \text{ ist in NNF} \\
\neg(t_1 \vee t_2) &= ((\neg t_1) \wedge (\neg t_2)) = (r_1 \wedge r_2) \text{ ist in NNF}
\end{aligned}$$

Somit gilt $P(t)$ für alle BT t nach dem obigen Induktionsprinzip.