Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Peer Kröger
Michael Fromm, Florian Richter

München, 3.11.2018

Einführung in die Programmierung

WS 2018/19

Übungsblatt 2: Boolesche Algebra, Vollständige Induktion

Besprechung: 05.11.2018 - 09.11.2018

Aufgabe 2-1 *Boolsche Operatoren*

Aus der Vorlesung kennen Sie die boolschen Operatoren $\wedge : \mathbb{B}^2 \to \mathbb{B}$, $\vee : \mathbb{B}^2 \to \mathbb{B}$ sowie $\neg : \mathbb{B} \to \mathbb{B}$. Mittels funktionaler Komposition lassen sich weitere Operatoren definieren, zum Beispiel gilt:

$$\Longrightarrow : \mathbb{B}^2 \to \mathbb{B}, (x, y) \mapsto \neg x \vee y$$

Außerdem lassen sich dadurch Operatoren höherer Stelligkeit definieren, z.B. Negation der ersten und zweiten Komponente:

$$\neg_1 : \mathbb{B}^2 \to \mathbb{B}, (x, y) \mapsto \neg x$$

$$\neg_2 : \mathbb{B}^2 \to \mathbb{B}, (x, y) \mapsto \neg y$$

Sei nun $\mathfrak{B}_2 = \{f \in \mathbb{B}^2 \to \mathbb{B}\}$ die Menge aller 2-stelligen totalen boolschen Funktionen.

(a) Benutzen Sie die Junktorenmenge $\{\neg_1, \wedge\} \subset \mathfrak{B}_2$ und konstruieren Sie daraus $\vee \in \mathfrak{B}_2$ durch Komposition.
Hinweis: DeMorgansche Regeln helfen.

Lösungsvorschlag:

DeMorgansche Regeln werden benötigt. In „klassischer“ Schreibweise (Infix+Präfix gemischt):

$$x \vee y = \neg\neg(x \vee y) = \neg(\neg x \wedge \neg y)$$

Alternativ in reiner Präfixschreibweise:

$$\vee(x, y) = \neg_1(\neg_1(x \vee y, *), *) = \neg_1(\wedge(\neg_1(x, *), \neg_1(y, *)), *)$$

Punkte: bei Fehler $-\frac{1}{2}$

(b) $\uparrow$ (NAND) ist wie folgt definiert:

$$x \uparrow y = \neg(x \wedge y)$$

Zeigen Sie, dass sich aus $\{\uparrow\}$ die Junktorenmenge $\{\neg_1, \wedge, \vee\}$ konstruieren lässt.
Hinweis: Zeigen Sie die Herleitungen in der angegebenen Reihenfolge $\neg_1, \wedge, \vee$ *und nutzen Sie Ergebnisse aus der vorherigen Aufgabe.*

Lösungsvorschlag:

In Reihenfolge ist (vermutlich) am einfachsten. Die Menge der Junktoren wächst dann und es bleibt übersichtlich.

$$x \uparrow x = \neg(x \wedge x) = \neg x$$

Nun haben wir $\{\uparrow, \neg\}$

$$\neg(x \uparrow y) = \neg(\neg(x \wedge y)) = x \wedge y$$

Und nun haben wir $\{\uparrow, \neg, \wedge\}$. Mit a) wissen wir auch, dass dann $\vee$ erzeugt werden kann.

Punkte: „$\uparrow$" : $+\frac{1}{2}$; „$\wedge$" : $+1$; „$\vee$" : $+\frac{1}{2}$

(c) Geben Sie die Mächtigkeit von $\mathfrak{B}_2$ an. Wieviele 3-stellige boolsche Funktionen $|\mathfrak{B}_3|$ gibt es? *Hinweis: Wieviele Wahrheitstabellen kann es für zwei Inputs x, y geben?*

Lösungsvorschlag:

Für die zweistelligen boolschen Funktionen gibt es 2×2 Eingaben. Jede Wahrheitstafel besteht also aus 4 Zeilen:

x	y	f(x,y)
0	0	f(0,0)
0	1	f(0,1)
1	0	f(1,0)
1	1	f(1,1)

Für jede Position in der dritten Spalte gilt: $f(x, y) \in \mathbb{B}$. Da jede boolsche Funktion über eine solche Tabelle eindeutig definiert werden kann, reicht es, die Anzahl der Möglichkeiten zu bestimmen, die dritte Spalte zu füllen. Mit anderen Worten: Wieviele Vektoren $f = (f(0,0), f(0,1), f(1,0), f(1,1)) \in \mathbb{B}^4$ gibt es? Genau $|\mathbb{B}^4| = 2 * 2 * 2 * 2$.
Damit ist $|\mathfrak{B}_2| = 2 * 2 * 2 * 2 = 2^{2^2}$. $\mathfrak{B}_3$ nutzt drei Argumente x, y, z. Damit gibt es 8 Zeilen in der Tabelle und $|\mathfrak{B}_3| = |\mathbb{B}^8| = 2^{2^3}$. Generell gilt: $|\mathfrak{B}_n| = 2^{2^n}$.

Punkte: 1,5 für $|\mathfrak{B}_2|$ und 0,5 für $|\mathfrak{B}_3|$

Aufgabe 2-2 *Vollständige Induktion*

(a) Zeigen Sie mittels vollst. Induktion: $1 + 2 + 4 + 8 + \ldots + 2^n = 2^{n+1} - 1$

Lösungsvorschlag:

Vollständige Induktion ganz formal mit Induktionsanfang IA, Induktionsvoraussetzung IV und Induktionsschluss IS.
(IA) $n = 0 : 2^0 = 1 = 2^{0+1} - 1$
(IV) Es gibt ein $n \in \mathbb{N}$, sodass $1 + 2 + 4 + 8 + \ldots + 2^n = 2^{n+1} - 1$ gilt.
(IS) Betrachte $n + 1$:

$$\begin{aligned} 1 + 2 + 4 + 8 + \ldots + 2^n + 2^{n+1} &= (1 + 2 + 4 + 8 + \ldots + 2^n) + 2^{n+1} \\ &\overset{\text{(IV)}}{=} 2^{n+1} - 1 + 2^{n+1} = 2 \cdot 2^{n+1} - 1 = 2^{(n+1)+1} - 1 \end{aligned}$$

Punkte: $+0,5$ für IA, $+0,5$ für IV, und $+1,0$ für IS

(b) Sei a eine Folge, die folgendermaßen rekursiv definiert ist:

$$a_1 = 2$$

$$a_{n+1} = 2 - \frac{1}{a_n}$$

Zeigen Sie mittels vollst. Induktion: $a_n = \frac{n+1}{n}$.

Lösungsvorschlag:

(IA) $n = 1 : a_1 = 2 = \frac{1+1}{1}$
(IV) Es gibt ein $n \in \mathbb{N}, n > 1$, sodass $a_n = \frac{n+1}{n}$ gilt.
(IS) Betrachte $n + 1$:

$$\begin{aligned} a_{n+1} &= 2 - \frac{1}{a_n} \\ &\overset{\text{(IV)}}{=} 2 - \frac{1}{\frac{n+1}{n}} = 2 - \frac{n}{n+1} = \frac{2(n+1)}{n+1} - \frac{n}{n+1} \\ &= \frac{2(n+1) - n}{n+1} = \frac{2n+2-n}{n+1} = \frac{(n+1)+1}{n+1} \end{aligned}$$

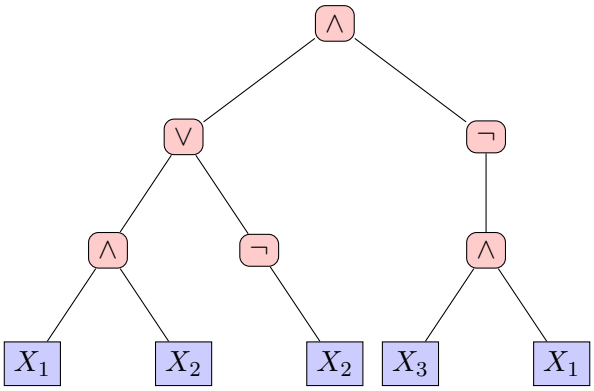
Punkte: $+0,5$ für IA, $+0,5$ für IV, und $+1,0$ für IS

Aufgabe 2-3 *Bäume*

Sei $B = \{\neg, \wedge, \vee\} \subset \mathfrak{B}$. Außerdem sei $\Sigma = \{X_1, \dots X_n\}$ eine endliche Menge von logischen Literalen. Diese können einen beliebigen Wahrheitswert aus $\{\text{True}, \text{False}\}$ annehmen. Wir definieren darüber die Menge der Wahrheitsbäume T_B induktiv:

$$\begin{aligned} \Sigma &\subseteq T_B \\ \neg(t) &\in T_B \text{ für } t \in T_B \\ \wedge(t_1, t_2) &\in T_B \text{ für } t_1, t_2 \in T_B : \\ \vee(t_1, t_2) &\in T_B \text{ für } t_1, t_2 \in T_B : \end{aligned}$$

Damit sind Blattknoten stets Literale und innere Knoten logische Junktoren. Als Beispiel soll folgende Darstellung dienen:



Der gezeigte Baum kann sequentiell dargestellt werden als

$$t' = \wedge(\vee(\wedge(X_1, X_2), \neg(X_2)), \neg(\wedge(X_3, X_1)))$$

Wir können nun eine rekursive Funktion definieren, die die induktive Konstruktion eines Baumes

nutzt, um seinen Wahrheitswert zu bestimmen:

$$\begin{aligned}
\text{evaluate} : T_B &\to \mathbb{B} \\
\text{evaluate}(X \in \Sigma) &= X \\
\text{evaluate}(\neg(t)) &= \neg\text{evaluate}(t) \\
\text{evaluate}(\wedge(t_1, t_2)) &= \text{evaluate}(t_1) \wedge \text{evaluate}(t_2) \\
\text{evaluate}(\vee(t_1, t_2)) &= \text{evaluate}(t_1) \vee \text{evaluate}(t_2)
\end{aligned}$$

(a) Ist die Funktion evaluate(t) injektiv/surjektiv/bijektiv?

Lösungsvorschlag:

Injektiv: (Jedes Bild hat maximal ein Urbild.) gilt nicht
Surjektiv: (Jedes Bild hat mindestens ein Urbild.) gilt
Bijektiv: (Jedes Bild hat genau ein Urbild.) gilt nicht
evaluate(t) bildet auf $\mathbb{B}$ ab. Es gibt unendlich viele Bäume, die als True ausgewertet werden. Genauso gibt es unendlich viele, die zu False abgebildet werden. Allein die Mächtigkeit der Mengen reicht also schon, um die Surjektivität zu begründen.

Punkte: $+1$*, wenn alles richtig, sonst* 0*.*

(b) Definieren Sie eine rekursive Funktion, die die Höhe eines Baumes zurückgibt:

$$\text{height} : T_B \to \mathbb{N}$$

Zur Vereinheitlichung: $\text{height}(X \in \Sigma) = 0$

Lösungsvorschlag:

$$\begin{aligned}
\text{height}(X \in \Sigma) &= 0 \\
\text{height}(\neg(t)) &= 1 + \text{height}(t) \\
\text{height}(\wedge(t_1, t_2)) &= 1 + \max(\text{height}(t_1), \text{height}(t_2)) \\
\text{height}(\vee(t_1, t_2)) &= 1 + \max(\text{height}(t_1), \text{height}(t_2))
\end{aligned}$$

Punkte: $+1,0$ *für Idee verstanden,* $+1,0$ *für Rest auch richtig, und* $-0,5$ *für kleine Fehler*

(c) Eine Formel ist in Negationsnormalform (NNF), wenn Negationen nur noch direkt vor den atomaren Aussagen auftauchen. In unserem Fall bedeutet dies, dass Negationen nur unmittelbar vor den Blattknoten auftreten dürfen. Der oben gezeigte Baum t' erfüllt dies nicht. Es gilt aber

$$\text{NNF}(t') = \wedge(\vee(\wedge(X_1, X_2), \neg(X_2)), \vee(\neg(X_3), \neg(X_1))).$$

Definieren Sie nun eine rekursive Funktion NNF : $T_B \to T_B$, die einen Baum so transformiert, dass das Resultat in NNF vorliegt.

Lösungsvorschlag:

$$
\begin{aligned}
\mathrm{NNF}(X \in \Sigma) &= X \\
\mathrm{NNF}(\neg(X)) &= \neg(X) \\
\mathrm{NNF}(\neg(\neg(t))) &= \mathrm{NNF}(t) \\
\mathrm{NNF}(\wedge(t_1, t_2)) &= \wedge(\mathrm{NNF}(t_1), \mathrm{NNF}(t_2)) \\
\mathrm{NNF}(\neg(\wedge(t_1, t_2))) &= \vee(\mathrm{NNF}(\neg(t_1)), \mathrm{NNF}(\neg(t_2))) \\
\mathrm{NNF}(\vee(t_1, t_2)) &= \vee(\mathrm{NNF}(t_1), \mathrm{NNF}(t_2)) \\
\mathrm{NNF}(\neg(\vee(t_1, t_2))) &= \wedge(\mathrm{NNF}(\neg(t_1)), \mathrm{NNF}(\neg(t_2)))
\end{aligned}
$$

Punkte: $+1,0$ *für Idee verstanden,* $+1,0$ *für Rest auch richtig, und* $-0,5$ *für kleine Fehler*