Einführung in die Modallogik

J.-G. Smaus, S. Wölfl
R. Mattmüller
Sommersemester 2011

Universität Freiburg
Institut für Informatik

Übungsblatt 2

Abgabe: 18. Mai 2011

Aufgabe 2.1 (Präsenzaufgabe: Äquivalenzrelationen und Homomorphismen)

(a) Seien $\mathcal{S} = \langle S, \sim_S \rangle$ und $\mathcal{T} = \langle T, \sim_T \rangle$ zwei relationale Strukturen, wobei $\sim_S$ und $\sim_T$ Äquivalenzrelationen sind, und sei $f : \mathcal{S} \to \mathcal{T}$ ein Homomorphismus. Definieren Sie eine Abbildung $\hat{f} : S/{\sim_S} \to T/{\sim_T}$, so dass das folgende Diagramm kommutiert, und zeigen Sie, dass die Abbildung wohldefiniert ist.

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "{[\cdot]_{\sim_S}}"'] & T \arrow[d, "{[\cdot]_{\sim_T}}"] \\ S/{\sim_S} \arrow[r, "\hat{f}"'] & T/{\sim_T} \end{tikzcd}$$

(b) Sei $\mathcal{S} = \langle S, \sim, R \rangle$ eine relationale Struktur, wobei $\sim$ eine Äquivalenzrelation und R eine mit $\sim$ *verträgliche* binäre Relation ist, d. h. für alle $s, s', t, t' \in S$ mit $s \sim t$ und $s' \sim t'$ gilt, dass $s\,R\,t$ gdw. $s'\,R\,t'$.
Zeigen Sie, dass in der relationalen Struktur $\mathcal{S}' = \langle S/{\sim}, R' \rangle$ die Relation $R' := \{([s]_\sim, [t]_\sim) \mid sRt\}$ wohldefiniert ist.
Angenommen, S und damit R ist endlich. Geben Sie eine möglichst genaue obere Abschätzung der Kardinalität $|R'|$ von R' in Abhängigkeit von R und $\sim$ an.

(c) Sei $\langle S, < \rangle$ eine strikte schwache Ordnung, d. h. eine Menge S zusammen mit einer transitiven und asymmetrischen Relation $<$, so dass die *Unvergleichbarkeitsrelation* $\perp$ transitiv auf S ist ($x \perp y := x \not< y \wedge y \not< x$ für $x, y \in S$). Zeigen Sie, dass dadurch auf der Menge der Äquivalenzklassen bzgl. $\perp$ eine (strikte) totale Ordnung gegeben ist.

Aufgabe 2.2 (Gültigkeitstest für Bewertungssemantik)

In dieser Aufgabe implementieren wir den Gültigkeitstest bzgl. der Bewertungssemantik aus Theorem 1.3.

Die Wahl der Programmiersprache steht Ihnen frei, bei sehr exotischen Sprachen sollten Sie die Wahl aber kurz mit uns absprechen. Abgabe bitte per Mail an `mattmuel@informatik.uni-freiburg.de`.
Die Aufgabe ist relativ umfangreich. Wenn Sie keine Zeit haben, die gesamte Aufgabe zu bearbeiten, empfehlen wir Ihnen, sich auf die Teilaufgaben (a), (b), (c), (g), (h) und (i) zu konzentrieren, da wir den Code für diese Teilaufgaben voraussichtlich in späteren Übungen wiederverwenden werden.
Ausnahmsweise erlauben und begrüßen wir bei dieser Aufgabe auch die Zusammenarbeit mehrerer Studenten.

(a) Implementieren Sie eine interne Repräsentation von $\mathcal{L}_\Box(P)$-Formeln der Gestalt

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi \mid \Box\varphi \mid \Diamond\varphi \ .$$

(b) Implementieren Sie eine Prozedur, die in einer gegebenen $\mathcal{L}_\Box(P)$-Formel alle Vorkommen von $\rightarrow$ und $\leftrightarrow$ eliminiert, indem $(\varphi \rightarrow \psi)$ durch $(\neg\varphi \vee \psi)$ und $(\varphi \leftrightarrow \psi)$ durch $((\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi))$ ersetzt werden (vgl. Schritt 1 im Beweis von Theorem 1.1).

(c) Implementieren Sie eine Prozedur, die in einer gegebenen $\mathcal{L}_\Box(P)$-Formel, in der keine $\rightarrow$ oder $\leftrightarrow$ mehr vorkommen, alle Negationen möglichst weit nach innen schiebt, indem $\neg(\varphi \vee \psi)$ durch $(\neg\varphi \wedge \neg\psi)$ ersetzt wird, $\neg(\varphi \wedge \psi)$ durch $(\neg\varphi \vee \neg\psi)$, sowie $\neg\neg\varphi$ durch φ.

(d) Erweitern Sie die Prozedur aus der letzten Teilaufgabe so, dass zusätzlich noch die Regeln (R0), (R0*), (R1), (R1*), (R2) und (R2*) aus Lemma 1.4 systematisch angewendet werden. Die Prozedur sollte also genau Schritt 2 aus dem Beweis von Theorem 1.1 implementieren.

(e) Schreiben Sie eine Prozedur, die Schritt 3 aus dem Beweis von Theorem 1.1 implementiert. Sie dürfen davon ausgehen, dass die Eingabe schon die Form hat, die von Schritt 2 erzeugt wird.

(f) Schreiben Sie eine Prozedur, die eine Formel, wie sie von der Prozedur aus der vorigen Teilaufgabe ausgegeben wird, in geordnete MCNF transformiert (vgl. Beweis zu Theorem 1.2).

(g) Schreiben Sie eine Prozedur, die eine aussagenlogische Formel der Form

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi$$

in CNF transformiert. Es reicht aus, wenn die resultierende Formel zur Eingabe *erfüllbarkeitsäquivalent* ist, d. h. Sie dürfen Hilfvariablen einführen (vgl. `http://de.wikipedia.org/wiki/Tseitin-Transformation`).

(h) Schreiben Sie eine Prozedur, die eine aussagenlogische Formel in CNF im DIMACS-Eingabeformat ausgibt (`http://www.satlib.org/Benchmarks/SAT/satformat.ps`, Kap. 2.1).

(i) Schreiben Sie eine Prozedur, die eine aussagenlogische Formel im DIMACS-Format an einen SAT-Solver Ihrer Wahl (z. B. `zchaff`, `minisat`, ...) übergibt und die Ausgabe des SAT-Solvers (erfüllbar oder nicht, ggf. erfüllende Belegung) einliest.

(j) Schreiben Sie eine Prozedur, die für eine Formel in geordneter MCNF über die Konjunktionsglieder iteriert, für jedes Konjunktionsglied der Form $\beta \vee \Box\gamma_1 \vee \cdots \vee \Box\gamma_n \vee \Diamond\delta$ die (aussagenlogischen) Formeln $\beta \vee \delta$, $\gamma_1 \vee \delta$, ..., $\gamma_n \vee \delta$ erzeugt und entsprechende Anfragen an einen SAT-Solver stellt, um die Gültigkeit der Formeln zu testen. Die Prozedur soll beantworten, ob die gegebene Formel in geordneter MCNF den Disjunktionstest besteht oder nicht.

(k) Integrieren Sie die Prozeduren aus den obigen Teilaufgaben so, dass Sie einen Gültigkeitstest bzgl. der Bewertungssemantik aus Theorem 1.3 erhalten.