Theorie der Informatik

G. Röger
Frühjahrssemester 2019

Universität Basel
Fachbereich Informatik

Präsenzaufgaben 9 — Lösungen

Aufgabe 9.1

Betrachten Sie die aussagenlogische Formel $\varphi = \neg(A \lor (\neg B \land C))$.

(a) Geben Sie die Formel χ_{all} an, wie sie in der polynomiellen Reduktion von SAT auf 3SAT verwendet wird.

Lösung:

Formel χ_{all} über $\{X_A, X_B, X_C, X_{\neg B}, X_{(\neg B \land C)}, X_{(A \lor (\neg B \land C))}, X_{\neg(A \lor (\neg B \land C))}\}$ ist aus folgenden Formeln aufgebaut:

$$\begin{aligned}
\chi_A &= (X_A \leftrightarrow A)\\
\chi_B &= (X_B \leftrightarrow B)\\
\chi_C &= (X_C \leftrightarrow C)\\
\chi_{\neg B} &= (X_{\neg B} \leftrightarrow \neg X_B)\\
\chi_{(\neg B \land C)} &= (X_{(\neg B \land C)} \leftrightarrow (X_{\neg B} \land X_C))\\
\chi_{(A \lor (\neg B \land C))} &= (X_{(A \lor (\neg B \land C))} \leftrightarrow (X_A \lor X_{(\neg B \land C)}))\\
\chi_{\neg(A \lor (\neg B \land C))} &= (X_{\neg(A \lor (\neg B \land C))} \leftrightarrow \neg X_{(A \lor (\neg B \land C))})
\end{aligned}$$

Dann ist

$$\chi_{\text{all}} = \chi_A \land \chi_B \land \chi_B \land \chi_{\neg B} \land \chi_{(\neg B \land C)} \land \chi_{(A \lor (\neg B \land C))} \land \chi_{\neg(A \lor (\neg B \land C))},$$

wobei jeweils die logisch äquivalente KNF-Formel verwendet wird, die durch die Auflösung der Abkürzung $\leftrightarrow$ entsteht. Z.B. $\chi_{(A \lor (\neg B \land C))} \equiv (X_{(A \lor (\neg B \land C))} \rightarrow (X_A \lor X_{(\neg B \land C)})) \land ((X_A \lor X_{(\neg B \land C)}) \rightarrow X_{(A \lor (\neg B \land C))}) \equiv (\neg X_{(A \lor (\neg B \land C))} \lor (X_A \lor X_{(\neg B \land C)})) \land (\neg(X_A \lor X_{(\neg B \land C)}) \lor X_{(A \lor (\neg B \land C))})$.

(b) $\mathcal{I} = \{A \mapsto F, B \mapsto T, C \mapsto T\}$ ist ein Modell von φ. Geben Sie das entsprechende Modell von χ_{all} an.

Lösung:

$$\begin{aligned}
\mathcal{I}' = \{&X_A \mapsto F, X_B \mapsto T, X_C \mapsto T, X_{\neg B} \mapsto F, X_{(\neg B \land C)} \mapsto F,\\
&X_{(A \lor (\neg B \land C))} \mapsto F, X_{\neg(A \lor (\neg B \land C))} \mapsto T\}
\end{aligned}$$

Aufgabe 9.2

Das Entscheidungsproblem SAT(Erfüllbarkeit) ist wie folgt definiert:

Gegeben: eine aussagenlogische Formel φ

Gefragt: Ist φ erfüllbar?

Das generelle Problem GenSAT(Modellerzeugung) ist wie folgt definiert:

Gegeben: eine aussagenlogische Formel φ

Ausgabe: ein Modell für φ, oder eine Meldung, dass kein Modell existiert.

Zeigen Sie, dass ein polynomieller Algorithmus für GenSAT existiert, falls ein polynomieller Algorithmus für SAT existiert.

Lösung:

Wir definieren einen Algorithmus, der GenSAT löst. Wir benutzen Transformationen $\psi[v \mapsto T]$ und $\psi[v \mapsto F]$, die alle Vorkommen von v mit einer kleinen allgemeingültigen (z.B. $v \vee \neg v$) respektive unerfüllbaren (z.B. $v \wedge \neg v$) Formel ersetzen. Dann gilt:

- $\mathcal{I}'$ ist ein Modell von $\psi[v \mapsto T]$ g.d.w. $\mathcal{I}$ mit $\mathcal{I}(v) = T$ und $\mathcal{I}(v') = \mathcal{I}'(v')$ für $v' \neq v$ ein Modell von ψ ist, und
- $\mathcal{I}'$ ist ein Modell von $\psi[v \mapsto F]$ g.d.w. $\mathcal{I}$ mit $\mathcal{I}(v) = F$ und $\mathcal{I}(v') = \mathcal{I}'(v')$ für $v' \neq v$ ein Modellv on ψ ist.

Zudem gilt, dass $\psi[v \mapsto F]$ erfüllbar ist, wenn ψ erfüllbar und $\psi[v \mapsto T]$ unerfüllbar ist.

Der Algorithmus geht wie folgt vor:

Wir rufen den Algorithmus für SAT mit Eingabe φ auf. Falls φ unerfüllbar ist, geben wir aus, dass kein Modell existiert.

Ansonsten konstruieren wir folgendermassen eine Interpretation $\mathcal{I}$ mit $\mathcal{I} \models \varphi$:

Initialisiere $\varphi' := \varphi$.

Solange es eine noch nicht zugewiesene Variable v gibt, rufen wir den SATAlgorithmus für $\varphi[v \mapsto T]$ auf. Falls die Antwort Ja lautet, setzten wir $\mathcal{I}(v) = T$ und fahren mit $\varphi' := \varphi'[v \mapsto T]$ fort, ansonsten setzten wir $\mathcal{I}(v) = F$ und fahren mit $\varphi' := \varphi'[v \mapsto F]$ fort.

Da die Anzahl Variablen durch die Grösse von φ beschränkt ist, gibt es nur eine polynomielle Anzahl an Iterationen. Die Grösse der letzten Formel φ' ist höchstens k mal grösser als φ ($k = 2$, falls wir nur die vorkommenden Variablen zählen, um die Grösse der Formel zu bestimmen). Wenn also jeder Aufruf des SAT-Algorithmus in polynomieller Zeit möglich ist, dann ist die gesamte Laufzeit auch polynomiell.