4.12.2017

PD Dr. Markus Junker
Dr. Giorgio Laguzzi

Mathematische Logik für Informatiker

WS 2017-2018, Blatt 7
Abgabe bis Montag 11.12.2017, 11:00 Uhr

Name, Vorname: ....
Matrikelnummer: ....
Übungsgruppe: ....

Aufgabe 1: Unter der Annahme $P \neq NP$ zeige man, dass es keinen Algorithmus gibt, der für jede Formel F in polynomieller Zeit eine Formel G in DNF findet, die äquivalent zu F ist.

Aufgabe 2: 2-SAT ist das Erfüllbarkeitsproblem für Formeln in KNF mit maximal zwei Literalen pro Klausel.

Zeigen Sie, dass 2-SAT in P liegt, also polynomiell ist, indem Sie zeigen, dass die Resolutionsmethode „schnell funktioniert". Hierbei soll keine Turingmaschine konstruiert werden, sondern gezeigt werden, dass man mit polynomiellem Aufwand die richtigen Resolutionsschritte finden und damit entscheiden kann, ob die Klauselmenge erfüllbar ist.

Eine Möglichkeit, Aufgabe 2 zu zeigen, besteht darin, über folgende Zwischenschritte zu gehen: Bezeichne L_i ein Literal, und $\neg L_i$ ein Literal mit „umgekehrtem Vorzeichen". Ein Klauselpfad von L_0 nach L_n ist ein Folge von zweielementigen Klauseln der Form

$$\big\{\{\neg L_0, L_1\}, \{\neg L_1, L_2\}, \ldots, \{\neg L_{n-1}, L_n\}\big\}.$$

(a) *Zeigen Sie, dass man einen Klauselpfad von L_0 nach L_n durch Resolutionen zu $\{\neg L_0, L_n\}$ zusammenfassen kann.*

(b) *Zeigen Sie, dass eine Klauselmenge, in der jede Klausel zwei Literale enthält, genau dann nicht erfüllbar ist, wenn es einen Klauselpfad von L_0 nach $\neg L_0$ und einen Klauselpfad von $\neg L_0$ nach L_0 gibt.*

Aufgabe 3: Sei $L = \{a, b, c, d\}$ und $X = L^*$ die Menge der aller endlichen Folgen von Elementen von L (z.B. ist *acbddac* eine Folge in X). Wir verwenden die folgende Notation:

- Für eine Folge $x \in X$ ist $x(1)$ das erste Element der Folge x, $x(2)$ das zweite, $x(3)$ das dritte, ... (z.B. wenn $x = abcb$, dann $x(1) = a$, $x(2) = b$, $x(3) = c$, $x(4) = b$).
- Für eine Folge $x \in X$ nennen wir w eine *Teilfolge von* x, wenn es $n_0 < n_1 < \cdots < n_k$ gibt, so dass $w(k) = x(n_k)$. (z.B. für $x = acbddac$ sind *acd*, *addc*, *cda* Teilfolgen von x, nicht aber *dbc*). Wenn w eine Teilfonge von x ist, dann schreiben wir $w \sqsubseteq x$.
- Für Folgen x, y in X definieren wir $LCS(x, y)$ als die Menge der längsten w, so dass $w \sqsubseteq x$ und $w \sqsubseteq y$. (LCS steht für *Longest Common Subsequence*).

  Bemerkung: solch ein w muss nicht eindeutig bestimmt sein; aber es ist klar, dass es stets mindestens ein w gibt.

Zeigen Sie, dass

$$LCS := \{(x, y, w) \mid x, y \in X, w \in LCS(x, y)\}$$

in NP liegt.

Wie in Aufgabe 2 soll auch hier keine explizite Turing-Maschine konstruiert, sondern argumentiert werden. Sie dürfen Lemma 3.2.2 benutzen.