Diskrete Modellierung

Wintersemester 2018/19

Mario Holldack, M. Sc.
Prof. Dr. Georg Schnitger
Hannes Seiwert, M. Sc.

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 5

Ausgabe: 13.11.18
Abgabe: 20.11.18

Aufgabe 5.1 *Quantoren, Implikationen, Negationen* $(6 + 8 + 12 = 26$ Punkte)

a) Sei $n \in \mathbb{N}$. Auf dem Grill liegen nebeneinander mehrere Steaks $s_1, \dots, s_n$, welche jeweils auf einer Seite mit einem Buchstaben aus der Menge $\{\text{A, B}, \dots, \text{Z}\}$ und auf der anderen Seite mit einer natürlichen Zahl mithilfe eines Brandeisens beschriftet wurden. Wir können auf dem Grill jeweils nur eine der beiden Beschriftungen sehen und wollen testen, ob folgende Aussage wahr ist:

Steht auf einer Seite des Steaks eine gerade Zahl, dann steht auf der anderen Seite ein Vokal.

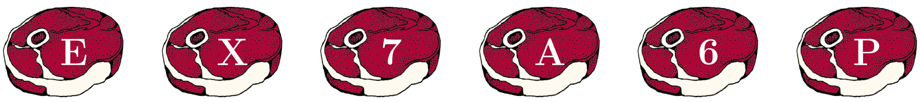

Abbildung 1: Zu Aufgabe 5.1 a) i)

Natürlich könnten wir **alle** Steaks umdrehen, um den Wahrheitsgehalt der Aussage zu überprüfen, aber es geht oft auch einfacher.

i) Ein konkretes Beispiel: Die Steaks liegen wie in Abbildung 1 auf dem Grill. Welche Steaks **müssen** wir umdrehen?

ii) Verallgemeinern Sie Ihre Antwort aus i) für beliebige Folgen $s_1, \dots, s_n$ von Steaks. Nach welcher Regel drehen Sie die Steaks um?

b) Betrachten Sie eine Färbung $f : \mathbb{N} \to \{\text{rot}, \text{blau}\}$ der natürlichen Zahlen. Folgende Aussage über f sei wahr: *Für jede rot gefärbte Zahl gibt es eine* ***größere*** *blau gefärbte Zahl.*

Welche der folgenden Aussagen kann man folgern? Begründungen sind nicht erforderlich.

i) Die Menge $f^{-1}(\{\text{blau}\})$ ist unendlich.

ii) Es gibt eine rote Zahl.

iii) Für jede blau gefärbte Zahl gibt es eine kleinere rot gefärbte.

iv) Wenn $f(1) = \text{rot}$ gilt, dann gibt es ein $n \in \mathbb{N}$, sodass $f(n) = \text{rot}$ und $f(n+1) = \text{blau}$ gilt.

c) Formulieren Sie die Negation der folgenden Aussagen jeweils umgangssprachlich. Begründungen sind nicht erforderlich.

i) Keine meiner Antworten ist falsch.

ii) Wenn ein Dominostein umfällt, fallen alle Dominosteine um.

iii) Zwei Geraden sind entweder parallel oder haben genau einen Schnittpunkt.

iv) *Nur* wenn ich diese Aufgabe korrekt löse, erhalte ich drei Punkte.

Bitte wenden!

Aufgabe 5.2 *Resolution* (8 + 10 + 4 = 22 Punkte)

a) Leiten Sie mittels Resolution den leeren Disjunktionsterm ϵ aus der Menge K her.

$$K := \Big\{\{C, B\}, \{A, \neg C\}, \{C, \neg B\}, \{\neg C, B\}, \{\neg A, \neg B\}\Big\}$$

b) Zeigen Sie mit Resolution, dass die KNF-Formel

$$\psi := \big(X \vee Y \vee Z\ \big) \wedge \neg Y \wedge \big(W \vee \neg X \vee Y \vee Z\big) \wedge \big(Y \vee \neg Z\big) \wedge (W \vee \neg Y) \wedge \big(\neg W \vee \neg X \vee Y \vee Z\big)$$

unerfüllbar ist.

c) Erläutern Sie, warum der folgende „Resolutionsschritt“ **falsch** ist:

$$\frac{\{A, B\}, \{\neg A, \neg B\}}{\epsilon}$$

Aufgabe 5.3 *Beweise führen* (9 + 9 + 18 = 36 Punkte)

a) Sei $n \in \mathbb{N}$ und seien φ und ψ aussagenlogische Formeln mit $\mathrm{Var}(\varphi) = \mathrm{Var}(\psi) = \{V_1, \ldots, V_n\}$. Die Anzahl der Belegungen $\mathcal{B} : \{V_1, \ldots, V_n\} \to \{0, 1\}$, die φ bzw. ψ erfüllen, seien mit $\#\mathrm{Sat}(\varphi)$ bzw. $\#\mathrm{Sat}(\psi)$ bezeichnet.

Zeigen Sie: Wenn $\#\mathrm{Sat}(\varphi) > \#\mathrm{Sat}(\psi)$, dann gilt $\varphi \not\models \psi$.

b) Seien $a, b, c \in \mathbb{Z}$ und gelte $a^2 + b^2 = c^2$.

Zeigen Sie: a ist gerade oder b ist gerade.

Hinweis: Indirekter Beweis. Ist die linke Seite durch 4 teilbar? Ist die rechte Seite durch 4 teilbar?

c) Sei $n \in \mathbb{N}_{>0}$ mit $n \geq 2$. Die Formel χ soll das Konzept von „mindestens zwei Einsen in einer Belegung“ ausdrücken. Also sei χ eine Formel mit $\mathrm{Var}(\chi) = \{V_1, \ldots, V_n\}$ und es gelte für jede Belegung $\mathcal{B} : \mathrm{Var}(\chi) \to \{0, 1\}$:

$$[\![\chi]\!]^{\mathcal{B}} = 1 \iff \big(\mathcal{B}(V_1), \ldots, \mathcal{B}(V_n)\big) \text{ enthält mindestens zwei Einsen}$$

i) Sei D eine DNF für χ. Zeigen Sie, dass (bis auf widersprüchliche Konjunktionsterme) jeder Konjunktionsterm von D mindestens zwei positive Literale enthält.

Hinweis: Indirekter Beweis.

ii) Sei D eine DNF und sei $1 \leq i < j \leq n$. Zeigen Sie: Wenn D keinen Konjunktionsterm besitzt, der nur V_i und V_j als positive Literale enthält, dann ist D keine DNF für χ.

iii) Zeigen Sie: Jede DNF für χ besitzt mindestens $\binom{n}{2}$ Konjunktionsterme.

Hinweis: Die Anzahl der zweielementigen Teilmengen von $\{V_1, \ldots, V_n\}$ ist $\binom{n}{2} = \frac{n(n-1)}{2}$, siehe https://de.wikipedia.org/wiki/Binomialkoeffizient.

iv) Geben Sie eine DNF für χ mit möglichst wenigen Konjunktionstermen an.

Bitte wenden!

Aufgabe 5.4 *Scheduling-Problem und Makespan* (16 Punkte + (4+10+10) Extrapunkte)

Sei $m, n \in \mathbb{N}_{>0}$. Gegeben seien m Prozessoren $p_1, \ldots, p_m$ und n Aufgaben $a_1, \ldots, a_n$ sowie eine Funktion t, die jeder Aufgabe a_i ihre Verarbeitungszeit $t(a_i)$ zuordnet, d. h. ein Prozessor benötigt stets $t(a_i)$ Prozessortakte, um die Aufgabe a_i zu verarbeiten.

Beim sogenannten *Scheduling-Problem* sind alle Aufgaben so auf die Prozessoren zu verteilen, dass der „Makespan", also die für die Verarbeitung[1] aller Aufgaben insgesamt benötigte Zeit, minimal ist. Mit anderen Worten: Sei I_j die Menge aller Aufgaben, die auf Prozessor p_j ausgeführt wird und

$$M_j := \sum_{a_i \in I_j} t(a_i)$$

die Last von p_j. Dann ist der Makespan $M := \max\{M_1, \ldots, M_m\}$ die höchste Last eines Prozessors.

a) Zeigen Sie: Für eine beliebige Verteilung der Aufgaben auf die Prozessoren gilt für den Makespan M:

$$M \geq \max\left\{ t(a_1),\, t(a_2), \ldots, t(a_n),\ \frac{1}{m}\sum_{i=1}^{n} t(a_i) \right\}$$

b*) *Diese Aufgabe ist eine Bonusaufgabe.*

Betrachten Sie den folgenden „Greedy"-Algorithmus[2] für den Spezialfall mit $m = 2$ Prozessoren.

```
1.  Setze I_1 := I_2 := ∅ und M_1 := M_2 := 0.
2.  Für i = 1, 2, ..., n:
    # Ordne Aufgabe a_i dem Prozessor mit der geringeren Last zu.
          Falls M_1 < M_2:
               # Führe Aufgabe a_i auf Prozessor p_1 aus.
               Setze I_1 := I_1 ∪ {a_i} und M_1 := M_1 + t(a_i).
          Sonst:
               # Führe Aufgabe a_i auf Prozessor p_2 aus.
               Setze I_2 := I_2 ∪ {a_i} und M_2 := M_2 + t(a_i).
3.  Gib I_1 und I_2 sowie den Makespan M_G := max{M_1, M_2} aus.
```

i) Führen Sie den Greedy-Algorithmus für die folgenden sechs Aufgaben mit den Verarbeitungszeiten $1, 4, 2, 3, 3$ und 10 aus. Geben Sie für jede Iteration i an, wie groß M_1 und M_2 *vor* der Zuordnung der Aufgabe a_i sind.

ii) Sei M^* der *kleinstmögliche* Makespan für eine Verteilung aller Aufgaben $a_1, a_2, \ldots, a_n$ auf zwei Prozessoren.

Zeigen Sie:

$$M_G \leq 2M^*$$

Fazit: Die vom Greedy-Algorithmus berechnete Lösung benötigt höchstens doppelt so viel Zeit wie die optimale Lösung. In der Veranstaltung GL-1 wird gezeigt, dass ein kleinstmöglicher Makespan vermutlich nicht mit „vernünftigem" Aufwand berechnet werden kann.

iii) Können Sie sogar $M_G \leq \frac{3}{2}M^*$ zeigen?

[1] Beachten Sie, dass die m Prozessoren parallel laufen, d. h. gleichzeitig ihre Aufgaben bearbeiten können. Allerdings wird jede Aufgabe nur von genau einem Prozessor bearbeitet.

[2] Greedy-Algorithmen sind „gierig" (engl. greedy). Sie entscheiden sich immer für die aktuell beste Zwischenlösung.