Prof. Dr. M. Grohe

E. Fluck, A. Riazsadri, J. Feith

# Übungsblatt 9 mit Lösungen

Abgabetermin: Mittwoch, der 21. Dezember 2022 um 14:30

**Aufgabe 4** **4 Punkte**

Eine aussagenlogische Formel ist in *disjunktiver Normalform* (DNF), falls sie eine Disjunktion von Konjunktionen von Literalen ist.

Wir betrachten das Erfüllbarkeitsproblem für DNF-Formeln:

> DNF-SAT
>
> **Eingabe:** Eine aussagenlogische Formel $\varphi$ in DNF.
>
> **Frage:** Existiert eine erfüllende Belegung $\alpha$ für $\varphi$?

Zeigen Sie, dass DNF-SAT in $\mathsf{P}$ liegt.

**Lösung:**

Sei $\varphi$ eine aussagenlogische Formel in DNF mit Klauseln $C_1, \ldots, C_m$ auf den Variablen $X_1, \ldots, X_n$. Zunächst bemerken wir, dass $\varphi$ erfüllbar ist, wenn eine der "Klauseln" erfüllbar ist.

Wir betrachten nun nacheinander die Klauseln $C_1, \ldots, C_m$ von $\varphi$. Eine solche Klausel $C_i$ ist eine Konjunktion von Literalen und somit erfüllbar, solange es keine Variable $X_j$ gibt, sodass $C_i$ sowohl $X_j$ als auch $\overline{X_j}$ enthält. In diesem Fall ist $C_i$ unerfüllbar; ansonsten betrachten wir die Belegung $\alpha\colon \{X_1, \ldots, X_n\} \to \{0, 1\}$ mit

$$\alpha(X_j) = \begin{cases} 1, & \text{falls } X_j \text{ in } C_i \text{ enthalten ist} \\ 0, & \text{andernfalls.} \end{cases}$$

Dann erfüllt $\alpha$ die Klausel $C_i$ und somit $\varphi$.

Folglich reicht es, eine solche erfüllbare Klausel von $\varphi$ zu finden (oder festzustellen, dass alle Klauseln von $\varphi$ unerfüllbar sind), was in $O(m \cdot n^2)$ Schritten machbar ist, indem wir für jede Klausel und jede Variable den obrigen Test durchführen. Somit haben wir DNF-SAT $\in \mathsf{P}$.

### Aufgabe 5 — 5 Punkte

Wir betrachten die folgende Variante von SAT:

> MAX-3-VORKOMMNISSE-SAT
>
> **Eingabe:** Eine aussagenlogische Formel $\varphi$ in KNF, wobei jede Variable von $\varphi$ höchstens 3 mal vorkommt.
>
> **Frage:** Existiert eine erfüllende Belegung $\alpha$ für $\varphi$?

Zeigen Sie, dass MAX-3-VORKOMMNISSE-SAT NP-schwer ist.

**Lösung:**

Wir zeigen, dass SAT $\leq_p$ MAX-3-VORKOMMNISSE-SAT gilt.

Zunächst stellen wir fest, dass MAX-3-VORKOMMNISSE-SAT $\subseteq$ SAT gilt. Somit können wir "syntaktisch inkorrekte" Eingaben (welche keine aussagenlogischen Formeln in KNF kodieren) identisch abbilden.

Betrachten wir also nun eine aussagenlogische Formel $\varphi$ in KNF, welche ohne Einschränkung der Allgemeinheit nur die Variablen $X_1, \ldots, X_k$ nutzt. Wir konstruieren eine MAX-3-VORKOMMNISSE-SAT-Instanz $\varphi'$ aus $\varphi$ wie folgt:

- Für jede Variable $X_i$ von $\varphi$:
  1. Ersetze das $j$-te Vorkommnis von $X_i$ durch die neue Variable $X_i^j$ (unter Berücksichtigung der Negation).
  2. Füge die Klauseln $(\overline{X_i^1} \vee X_i^2), (\overline{X_i^2} \vee X_i^3), \ldots, (\overline{X_i^{\ell-1}} \vee X_i^\ell), (\overline{X_i^\ell} \vee X_i^1)$ hinzu, wobei $\ell$ die Anzahl der Vorkommnisse von $X_i$ in $\varphi$ ist.

Offensichtlich kann $\varphi'$ in Polynomialzeit aus $\varphi$ konstruiert werden. Weiterhin enthält $\varphi'$ jede Variable genau 3 mal (in einer Klausel aus $\varphi$ und in zwei der neu hinzugefügten Klauseln) und ist in KNF, womit $\varphi'$ eine MAX-3-VORKOMMNISSE-SAT-Instanz ist.

Bevor wir die Korrektheit der Reduktion zeigen, bemerken wir, dass jede erfüllende Belegung von $\varphi'$ den aus $X_i$ erzeugten Variablen den gleichen Wert zuweisen muss: Die Klausel $(\overline{A} \vee B)$ ist äquivalent zu $(A \to B)$, weswegen die Konjunktion der für $X_i$ hinzugefügten Klauseln äquivalent zu einem Implikationskreis

$$X_i^1 \to X_i^2 \to \ldots \to X_i^\ell \to X_i^1$$

sind.

Nehmen wir nun an, dass $\varphi$ erfüllbar ist, d.h. es existiert eine Belegung $\alpha\colon \{X_1, \ldots, X_k\} \to \{0,1\}$, welche $\varphi$ erfüllt. Wir konstruieren nun eine Belegung $\alpha'$, welche $\varphi'$ erfüllt, indem wir $\alpha'(X_i^j) = \alpha(X_i)$ für alle $j$ setzen. Aus unserer Vorüberlegung folgt, dass $\alpha'$ alle neu hinzugefügten Klauseln von $\varphi'$ erfüllt. Betrachten wir also im Folgenden eine Klausel $C'$ von $\varphi'$, welche durch das obrige Verfahren aus einer Klausel $C$ von $\varphi$ hervorgegangen ist. Da $\varphi$ $C$ erfüllt, gibt es also ein $i$ so, dass $X_i$ (bzw. $\overline{X_i}$) in $C$ vorkommt und von $\alpha$ erfüllt wird. Entsprechend existiert dann ein $j$ so, dass $X_i^j$ (bzw. $\overline{X_i^j}$) in $C'$ vorkommt und von $\alpha'$ erfüllt wird, da $\alpha'(X_i^j) = \alpha(X_i)$ gilt. Folglich erfüllt $\alpha'$ also die Klausel $C'$, und somit alle Klauseln von $\varphi'$ Insgesamt folgt also, dass $\varphi'$ ebenfalls erfüllbar ist.

Sei nun $\varphi'$ erfüllbar, d.h. es existiert eine erfüllende Belegung $\alpha'$ für $\varphi'$, welche nach unserer Vorüberlegung für alle $i \in \{1, \dots, k\}$ allen aus $X_i$ erzeugten Variablen den gleichen Wert zuweist. Wir definieren eine Belegung $\alpha\colon \{X_1, \dots, X_k\} \to \{0, 1\}$ via $\alpha(X_i) = \alpha'(X_i^1)$. Betrachten wir nun eine beliebige Klausel $C$ von $\varphi$, welche durch das obrige Verfahren zu einer Klausel $C'$ von $\varphi'$ transformiert wird. Da $\alpha'$ insbesondere $C'$ erfüllt, existiert also $i, j$ so, dass $X_i^j$ (bzw. $\overline{X_i^j}$) in $C'$ liegt und von $\alpha'$ erfüllt wird. Entsprechend gilt $\alpha'(X_i^j) = \alpha'(X_i^1) = \alpha(X_i)$, und somit erfüllt $\alpha$ die Klausel $C$, welche $X_i$ (bzw. $\overline{X_i}$) enthält. Insgesamt folgt also, dass $\varphi$ erfüllbar ist.

Folglich ist $\varphi$ genau dann erfüllbar, wenn $\varphi'$ erfüllbar ist, und wir erhalten SAT $\leq_p$ Max-3-Vorkommnisse-SAT, womit Max-3-Vorkommnisse-SAT NP-schwer ist.

## Aufgabe 6 6 Punkte

Wir betrachten das folgende Problem (siehe Übungsblatt 8):

> $\{-1,0,1\}$-Restricted Integer Programming
>
> **Eingabe:** Eine Matrix $A \in \{-1,0,1\}^{m\times n}$ und ein Vektor $b \in \{-1,0,1\}^m$.
>
> **Frage:** Gibt es einen Vektor $x \in \{0,1\}^n$ mit $Ax \geq b$?

Zeigen Sie, dass $\{-1,0,1\}$-Restricted Integer Programming NP-vollständig ist.

**Lösung:**

$\{-1,0,1\}$-Restricted Integer Programming $\in$ NP wurde bereits auf Übungsblatt 8 gezeigt.

Um zu zeigen, dass $\{-1,0,1\}$-Restricted Integer Programming NP-schwer ist, zeigen wir 3SAT $\leq_p$ $\{-1,0,1\}$-Restricted Integer Programming.

Dafür bilden wir zunächst "syntaktisch inkorrekte" Eingaben auf eine beliebige Nein-Instanz von $\{-1,0,1\}$-Restricted Integer Programming ab.

Betrachten wir nun eine Formel $\varphi$ in 3-KNF mit $m$ Klauseln $C_1,\dots,C_m$. Ohne Einschränkung der Allgemeinheit nutzt $\varphi$ nur die Variablen $X_1,\dots,X_n$.

Wir konstruieren nun aus $\varphi$ eine $((2n+m)\times 2n)$-Matrix $A$, welche folgendes (Un)-Gleichungssystem über den $2n$ Literalen $X_1,\dots,X_n,\overline{X_1},\dots,\overline{X_n}$ von $\varphi$ beschreibt:

- Für jede Variable $X_j$ fügen wir die Ungleichungen $X_j + \overline{X_j} \geq 1$ (Typ Ia) und $-X_j - \overline{X_j} \geq -1$ (Typ Ib) ein. Somit setzt jede Lösung entweder $X_j$ oder $\overline{X_j}$ auf 1, während das andere Literal auf 0 gesetzt werden muss.
- Für jede Klausel $C_i = \lambda_{i,1} \vee \lambda_{i,2} \vee \lambda_{i,3}$ fügen wir die Ungleichung $\lambda_{i,1}+\lambda_{i,2}+\lambda_{i,3} \geq 1$ (Typ II) hinzu. Somit muss jede Lösung mindestens ein Literal jeder Klausel auf 1 setzen.

Schreiben wir diese Ungleichungen nun der Reihe nach in die Zeilen von $A$ (d.h. für $1 \leq j \leq n$ gibt die Spalte $j$ die Koeffizienten von $X_j$ an; für $n+1 \leq n+j \leq 2n$ gibt die Spalte $n+j$ die Koeffizienten von $\overline{X_j}$ an), so erhalten wir

$$A_{i,j} = \begin{cases} 1, & \text{falls } j = i \text{ oder } j = n+i \text{ und } i \leq n \text{ (Typ Ia)},\\ -1, & \text{falls } j = i \text{ oder } j = n+i \text{ und } n+1 \leq i \leq 2n \text{ (Typ Ib)},\\ 1, & \text{falls } X_j \in C_i \text{ und } 1 \leq j \leq n \text{ sowie } 2n+1 \leq i \text{ (Typ II)}\\ 1, & \text{falls } \overline{X_j} \in C_i \text{ und } n+1 \leq n+j \leq 2n \text{ sowie } 2n+1 \leq i \text{ (Typ II)}\\ 0, & \text{sonst} \end{cases}$$

und

$$b_i = \begin{cases} 1, & \text{falls } i \leq n \text{ (Typ Ia)}, \\ -1, & \text{falls } n+1 \leq i \leq 2n \text{ (Typ Ib)}, \\ 1, & \text{falls } 2n+1 \leq i \text{ (Typ II)}. \end{cases}$$

Offensichtlich können $A$ und $b$ in Polynomialzeit aus $\varphi$ konstruiert werden.

Nehmen wir zunächst an, dass $\varphi$ erfüllbar ist. Dann existiert eine erfüllende Belegung $\alpha$ von $\varphi$, und wir betrachten $x \in \{0,1\}^{2n}$ mit

$$x_j = \begin{cases} \alpha(X_j), & \text{falls } 1 \leq j \leq n, \\ 1 - \alpha(X_j), & \text{falls } n+1 \leq n+j \leq 2n, \end{cases}$$

also den Vektor der Belegungen von $X_1, \dots, X_n, \overline{X_1}, \dots, \overline{X_n}$ unter $\alpha$. Dann gilt auch $Ax \geq b$, da $\alpha$ mindestens ein Literal jeder Klausel erfüllt (also die Ungleichungen vom Typ II erfüllt); die Ungleichungen der Typen Ia und Ib werden von jedem “Belegungsvektor” erfüllt.

Sei nun andersherum $x \in \{0,1\}^{2n}$ eine Lösung von $Ax \geq b$. Wir betrachten die Belegung $\alpha\colon \{X_1, \dots, X_n\} \to \{0,1\}$ mit

$$\alpha(X_j) = \begin{cases} 1, & \text{falls } x_j = 1, \\ 0, & \text{falls } x_j = 0. \end{cases}$$

Dann erfüllt $\alpha$ genau die Literale, welche in $x$ den Wert 1 haben, weil $x$ die Ungleichungen der Typen Ia und Ib erfüllt. Da $x$ die Ungleichungen vom Typ II erfüllt, muss auch $\alpha$ somit mindestens ein Literal einer jeden Klausel von $\varphi$ erfüllen, womit $\varphi$ erfüllbar ist.

Es folgt 3SAT $\leq_p$ $\{-1,0,1\}$-Restricted Integer Programming und da 3SAT bereits NP-vollständig ist, ist auch $\{-1,0,1\}$-Restricted Integer Programming NP-vollständig.