Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey und Eric Nöth

Komplexitätstheorie
WS 2013/2014

Übungsblatt 10

Aufgabe 1 (P-Vollständig: Lineare Ungleichungen). Zeigen Sie, dass das folgende Problem **P**−schwer ist.
Gegeben. Eine Matrix $A \in \mathbb{Z}^{n \times d}$ und ein Vektor $\mathbf{b} \in \mathbb{Z}^n$.
Gesucht. Gibt es ein $\mathbf{x} \in \mathbb{Q}^d$ mit $\mathbf{x} > 0$, so dass $A\mathbf{x} \leq \mathbf{b}$?
Dabei bedeutet $\mathbf{x} > 0$, dass alle Komponenten nichtnegativ sind und mindestens eine Komponente positiv ist.
Hinweis. Führen Sie eine Reduktion via **CVP** durch. Erstellen Sie dabei, ähnlich wie bei HORNSAT, Ungleichungen für Gatter.
Die Fragestellung ist **P**-vollständig; einen Algorithmus, der *Lineare Ungleichungen* löst, finden Sie in [Kac79].

Aufgabe 2 (NP-Vollständig: Integer Programming). Zeigen Sie, dass Aufgabe 1 **NP**-vollständig wird, falls stattdessen gefordert wird, dass die Einträge von $\mathbf{x}$ nur 0 oder 1 sein dürfen (der Nullvektor ist dann also auch zugelassen) und $A\mathbf{x} = \mathbf{b}$ gesucht wird. (dieses Problem stammt aus [Kar72]).
Hinweis. Führen Sie die Reduktion via **SAT**. Setzen Sie dafür

$$a_{i,j} = \begin{cases} 1 & \text{falls } x_j \in C_i \\ -1 & \text{falls } \bar{x}_j \in C_i \\ 0 & \text{sonst} \end{cases}$$

für die j-te Variable und die i-te Klausel. Wie definiert sich $\mathbf{b}$?

Literatur

[Kac79] L.G. Kachian. A polynomial time algorithm for linear programming. In *Soviet Math. Dokl. 20*, page 191–194, 1979.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.