Komplexitätstheorie

Sommersemester 2019

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

GOETHE UNIVERSITÄT
FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 4

Ausgabe: 13.05.2019
Abgabe: 20.05.2019

In Aufgabe 4.4. können Sie 8 Extrapunkte ergattern.

Aufgabe 4.1 *Randomisierte Komplexitätsklassen* (3 + 3 + 2 = 8 Punkte)

Zeigen Sie:

a) $\texttt{ZPP} = \texttt{RP} \cap \texttt{coRP}$

b) $\texttt{NP} \subseteq \texttt{PP}$

c) $\texttt{PP} \subseteq \texttt{PSPACE}$

Aufgabe 4.2 *Gleichheitstest für Polynome* (4 + 4 = 8 Punkte)

a) Sei $p = p(x_1, \ldots, x_n)$ ein n-stelliges Polynom über einem Körper $\mathbb{F}$ vom Grad[1] d, das sich vom Nullpolynom unterscheide. Sei S eine endliche Teilmenge von $\mathbb{F}$.

   Zeigen Sie: $\text{prob}_{a \in S^n}\big[\, p(a) = 0 \,\big] \leq \frac{d}{|S|}$

   Hinweis: Zeigen Sie per Induktion über n, dass p höchstens $d|S|^{n-1}$ Nullstellen in S^n besitzt.

b) Ein unbekanntes n-stelliges Polynom p vom Grad d über dem Körper $\mathbb{R}$ sei gegeben, wobei n und d bekannt seien.

   Entwerfen Sie einen coRP-Algorithmus, der entscheidet, ob $p \equiv 0$ gilt, d. h. ob p das Nullpolynom ist. Der Algorithmus hat dabei nur Blackbox-Zugriff auf p, d. h. für eine Anfrage der Form $\alpha \in \mathbb{R}^n$ liefert ein Orakel die Antwort $p(\alpha)$ in Zeit $\text{poly}(n)$.

Aufgabe 4.3 *Branching-Programme* (8 Punkte)

Wir definieren das parallele Rechnermodell der *Branching-Programme* als Verallgemeinerung von Entscheidungsbäumen auf „Entscheidungs-DAGs“.

Sei $n \in \mathbb{N}$. Ein *Branching-Programm* (BP) ist ein gerichteter azyklischer Graph (DAG) mit genau einer Quelle und genau zwei Senken. Die beiden Senken haben die Beschriftungen 0 bzw. 1, alle anderen Knoten sind mit einer Variable $V_i \in \{V_1, \ldots, V_n\}$ beschriftet. Bis auf die Senken haben alle Knoten genau zwei ausgehende Kanten, von denen eine mit 0 und die andere mit 1 beschriftet ist.

Ein Branching-Programm B berechnet eine boolesche Funktion $f_B : \{0,1\}^n \to \{0,1\}$ wie folgt: Für eine Eingabe $y = (y_1, \ldots, y_n) \in \{0,1\}^n$ beschreibe den folgenden Weg: Beginne in der Quelle von B und verlasse den aktuellen Knoten mit Variablenbeschriftung V_i stets durch seine ausgehende Kante mit Beschriftung y_i, bis eine Senke erreicht wird. Die *Ausgabe* $f_B(y)$ von B ist die Beschriftung der erreichten Senke.

[1] Der Grad eines Polynoms p ist der höchste Grad eines Monoms in p. Der Grad eines Monoms $m = c\prod_{i=1}^{n} x_i^{k_i}$ ist $\sum_{i=1}^{n} k_i$.

Ein BP heißt *Read-Once-Branching-Programm* (ROBP), wenn auf jedem Weg von der Quelle zu einer Senke jede Variable V_i höchstens einmal als Knotenbeschriftung auftaucht.

Zwei Branching-Programme B_1 und B_2 sind *äquivalent* (kurz $B_1 \equiv B_2$), wenn ihre Funktionen f_{B_1} und f_{B_2} übereinstimmen, d. h. wenn $f_{B_1}(y) = f_{B_2}(y)$ für alle $y \in \{0,1\}^n$ gilt.

Sei $EQ_{\text{ROBP}} := \{(B_1, B_2) : B_1, B_2 \text{ sind ROBPs und } B_1 \equiv B_2\}$.

Zeigen Sie: $EQ_{\text{ROBP}} \in$ `BPP`.

Hinweis: Wenden Sie Aufgabe 4.2 an. Wie können Sie aus einem Branching-Programm B ein geeignetes Polynom p_B über $\mathbb{R}$ erzeugen, das auf booleschen Eingaben mit f_B übereinstimmt? Sie müssen auch zeigen, dass die Polynome zweier *äquivalenter* BPs auch auf $\mathbb{R}$ übereinstimmen.

Kommentar: Die Sprache $EQ_{\text{BP}} := \{(B_1, B_2) : B_1, B_2 \text{ sind BPs und } B_1 \equiv B_2\}$ ist `coNP`-hart. Die Read-Once-Einschränkung ist also wesentlich!

Aufgabe 4.4 *Bipartites Perfektes Matching liegt in* `RNC` (8 Extrapunkte)

Ein *randomisierter* Schaltkreis besitzt zusätzlich zu den Eingabegattern Gatter mit Zufallsbits. Die Klasse `RNC` (Randomized Nick's Class) enthält alle Sprachen, die durch uniforme randomisierte Schaltkreisfamilien der Größe poly(n) und Tiefe poly-log(n) mit höchstens poly(n) vielen Zufallsbits und beidseitigem Fehler höchstens 1/3 entschieden werden können.

Sei $n \in \mathbb{N}$. Wir betrachten bipartite Graphen $G = (L, R, E)$ mit Knotenmengen $L = R = \{1, \ldots, n\}$ und Kantenmenge[2] $E \subseteq L \times R$. Ein solcher Graph G sei durch seine „bipartite Adjazenzmatrix“ A gegeben, wobei $A = \left(a_{i,j}\right)_{i \in L, j \in R}$ und $a_{i,j} = 1$, wenn $(i,j) \in E$, und $a_{i,j} = 0$, sonst.
Wir definieren die Sprache

$$\text{BPM} = \{G = (L, R, E) : \text{ G besitzt ein perfektes Matching}\}.$$

Zeigen Sie: BPM $\in$ `RNC`.

Hinweis: Wenden Sie Aufgabe 4.2 an. Bringen Sie die Determinante

$$\det(A) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

mit der Existenz perfekter Matchings in G in Verbindung. Wie verhält sich $\det(A)$, wenn G gar kein, genau ein bzw. mehrere perfekte Matchings besitzt? Konstruieren Sie ausgehend von $\det(A)$ ein n^2-stelliges Polynom $p = p(x_{1,1}, x_{1,2}, \ldots, x_{n,n})$ über $\mathbb{R}$, sodass $p \equiv 0$ genau dann gilt, wenn G kein perfektes Matching besitzt.
Sie dürfen ohne Beweis verwenden, dass die Berechnung der Determinante $\det(M)$ einer beliebigen reellwertigen[3] $n \times n$-Matrix M in `NC` liegt.

[2] Der Übersichtlichkeit zuliebe stellen wir die Kanten durch Tupel statt durch Mengen dar.

[3] Natürlich vorausgesetzt, die reellen Komponenten der Matrix besitzen nicht zu viele Bits. Diese Einschränkung darf im Rahmen dieser Aufgabe aber vernachlässigt werden.