Technische Universität Braunschweig

TCS

Dr. Jürgen Koslowski

Theoretische Informatik 2, Klausur, alte P.O.

2019-07-23

- Bitte schreiben Sie mit einem dokumentenechten Stift, nicht in Rot und verwenden Sie separate Blätter für jede bearbeitete Aufgabe.
- Markieren Sie jede bearbeitete Aufgabe in der untenstehenden Tabelle, indem Sie die **Aufgabennummer einkreisen**. Das erleichtert die Korrektur ungemein.
- Legen Sie bitte Ihren Studenten- und Personalausweis bereit.
- Schalten Sie Handys aus und nehmen Sie sie während der Klausur nicht zur Hand!
- Bearbeitungseit: 180 Minuten; mit 40 von 100 Punkten ist die Klausur sicher bestanden.

Aufgabe 1 [10 PUNKTE]

Aus der Theoretischen Informatik 1 sollte Ihnen bekannt sein, dass die Sprache

$$L = \{\, a^n b^m c^n d^m \,:\, m, n > 0\} \subseteq \{a, b, c, d\}^+$$

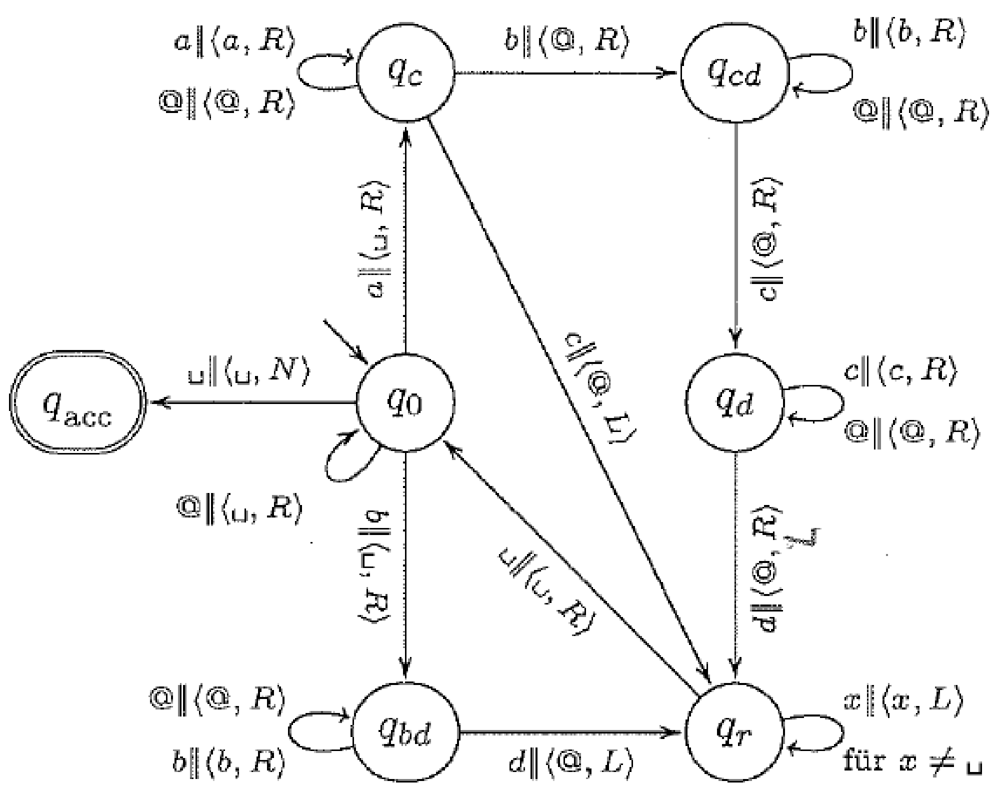
nicht kontextfrei ist. Konstruieren Sie eine deterministische(!) 2-Band-Maschine M_1 mit $\mathcal{L}(M_1) = L$, indem Sie

- [3 PUNKTE] zunächst eine grobe Spezifikation der Arbeitsweise angeben, so dass die Maschine i.A. schneller laufen sollte als eine 1-Band-Maschine;
- [7 PUNKTE] dann einen Zustandsgraphen für die Maschine zeichnen.

Zur Verbesserung der Übersichtlichkeit können Sie den abweisenden Zustand q_{rej} weglassen. [Hinweis: es reichen weniger als 10 Zustände.]

Aufgabe 2 [10 PUNKTE]

Gegeben sei die folgende deterministische * 1-Band Turingmaschine M .



* Der Übersichtlichkeit halber haben wir darauf verzichtet, den Zustand q_{rej} und die Schleifen am Zustand q_{acc} einzuzeichnen..

(1) [6 PUNKTE] Bestimmen Sie die von M akzeptierte Sprache mit detaillierter Begründung.

(2) [2 PUNKTE] Schätzen Sie die Laufzeit für eine Eingabe der Länge k ab.

(3) [2 PUNKTE] Zeigen oder widerlegen Sie: ein Zustand lässt sich einsparen, indem man q_{bd} und q_d zusammenfasst.

Aufgabe 3 [10 PUNKTE]

Untersuchen Sie die folgenden Sprachen darauf, ob der Satz von Rice anwendbar ist, und ermitteln Sie ggf. das Ergebnis: $L_i \subseteq \{0,1\}^*$ bestehe aus allen TM-Codes w , so dass

(1) $\mathcal{L}(M_w)$ eine gerade endliche Anzahl von Elementen hat;

(2) M_w eine gerade Anzahl von Zuständen hat;

(3) $\mathcal{L}(M_w)$ von einer TM mit einer geraden Anzahl von Zuständen erzeugt werden kann;

(4) jedes Wort aus $\mathcal{L}(M_w)$ eine Berechnung gerader Länge hat;

(5) jedes Wort aus $\mathcal{L}(M_w)$ von einer geraden Anzahl von Berechnung akzeptiert wird.

Aufgabe 4 [10 PUNKTE]

[10 PUNKTE] Das E-Problem 3FACH-SAT hat eine Boole'sche Formel φ in KNF als Eingabe. Zu entscheiden ist, ob φ auf mindestens drei verschiedene Weisen erfüllt werden kann. Weisen Sie die $\boldsymbol{NP}$-Vollständigkeit von 3FACH-SAT nach.

Aufgabe 5 [10 PUNKTE]

In *Multi-Graphen* sind Schleifen und mehr als eine Kante zwischen je zwei Knoten erlaubt.

<u>Euler-Kreis</u> (EULER-KREIS) **Gegeben:** ein gerichteter Multi-Graph $G = \langle V, E \rangle$ **zu entscheiden:** ob es in G einen Euler-Kreis gibt, d.h., einen Rundweg, der alle Knoten berührt (evtl. auch mehrfach) und jede Kante genau einmal verwendet.

(1) [4 PUNKTE] Zeigen Sie, dass G genau dann einen Eulerschen Kreis hat, G zusammenhängend ist und jeder Knoten dieselbe Anzahl von eingehenden wie ausgehenden Kanten hat. [Hinweis: mindestens eine Beweisrichtung sollte eine Induktion verwenden.]

(2) [6 PUNKTE] Entwerfen Sie einen deterministischen Polynomialzeitalgorithmus, der Codierungen gerichteter Graphen darauf testet, ob sie Eulersche Kreise haben. Verwenden Sie zur Codierung quadratische Adjazenzmatrizen mit natürlichen Zahlen als Einträgen.

Aufgabe 6 [10 PUNKTE]

Richtig oder falsch: geben Sie jeweils eine kurze Begründung:

(1) Bzgl. der Reduktions-Quasiordnung $\leq^R$ für eine Klasse R von Funktionen zwischen freien Monoiden, die alle Identitätsabbildungen enthält und unter Komposition abgeschlossen ist, sind alle Sprachen der Form Σ^* äquivalent (Σ eine endliche Menge).

(2) Aus $\boldsymbol{P} = \boldsymbol{NP}$ folgt $\boldsymbol{NP} = \mathrm{co}\boldsymbol{NP}$.

(3) Es existieren $\boldsymbol{NP}$-harte Probleme außerhalb von $\boldsymbol{P}$. (Falls ja, nennen Sie eins.)

(4) Es existieren Probleme in $\boldsymbol{L}$, die nicht $\boldsymbol{NL}$-hart sind. (Falls ja, nennen Sie eins.)

(5) Für jedes $\boldsymbol{NL}$-harte Problem A und jede endliche Teilmenge $B \subseteq A$ ist das Problem $A - B$ ebenfalls $\boldsymbol{NL}$-hart.

Aufgabe 7 [10 PUNKTE]

In Analogie zu den $\mathcal{C}$-harten Problemen bzgl. $\leq^{\log}$ bzw. $\leq^{\text{poly}}$ (oder allgemeiner $\leq^R$) für eine Klasse $\mathcal{C}$ entscheidbarer Probleme kann man $\mathcal{C}$*-leichte* Probleme definieren: diese können auf jedes Problem in $\mathcal{C}$ reduziert werden:

$$A \text{ is } \mathcal{C}\text{-}\textit{leicht}, \text{ wenn } A \leq^R C \text{ für alle } c \in \mathcal{C}$$

Weiter möge die jeweilige *Hülle* $\mathcal{C}^\star$ von $\mathcal{C}$ aus allen Problemen bestehen, die leicht bzgl. der Klasse aller $\mathcal{C}$-harten Probleme sind, also

$$A \in \mathcal{C}^* \text{ sofern für alle E-Probleme } B \text{ gilt: wenn } B \ \mathcal{C}\text{-hart ist, dann } A \leq^R B\,.$$

Zeigen oder widerlegen Sie:

(1) [3 PUNKTE] $(-)^\star$ ist *extensiv*: $\mathcal{C}$ ist immer Teilmenge der Hülle $\mathcal{C}^\star$.

(2) [3 PUNKTE] $(-)^\star$ ist *monoton*: Aus $\mathcal{C} \subseteq \mathcal{D}$ folgt $\mathcal{C}^\star \subseteq \mathcal{D}^\star$.

(3) [4 PUNKTE] $(-)^\star$ ist *idempotent*: $\mathcal{C}^{\star\star} = \mathcal{C}^\star$.

Wenn Sie die Hüllen von $\emptyset$ bzgl. $\leq^{\log}$ bzw. $\leq^{\text{poly}}$ identifizieren können, gibt es [5. SONDERPUNKTE]

Aufgabe 8 [10 PUNKTE]

Betrachte das E-Problem

Cycle (CYCLE)
Gegeben: eingerichteter Graph $G = \langle V, E\rangle$ und ein Knoten $v \in V$
zu entscheiden: Liegt v auf einem Kreis in G?

Zeigen Sie die ***NL***-Vollständigkeit von CYCLE.

Aufgabe 9 [10 PUNKTE]

Betrachte das E-Problem

Inzidenzsystem (IS)
Gegeben: eine endliche Menge X, eine Teilmenge $\mathcal{S}$ der Potenzmenge von X und eine Zahl $k \in \mathbb{N}$.
zu entscheiden: Existiert eine Teilmenge T von X mit $\lvert T\rvert \leq k$ und $T \cap S \neq \emptyset$ für alle $S \in \mathcal{S}$?

(1) [3 PUNKTE] überlegen Sie sich zunächst eine vernünftige quasi-binäre Codierung (über dem Alphabet $\{0, 1, \#\}$) von Instanzen dieses Problems als Eingabe für eine Turingmaschine.

(2) [7 PUNKTE] Beweisen Sie dann detailliert die ***NP***-Vollständigkeit dieses Problems.

Aufgabe 10 [10 PUNKTE]

Wo steckt der Fehler in folgenden Argumenten?

(1) [5 PUNKTE]

- ▷ Annahme $\boldsymbol{P} = \boldsymbol{NP}$.
- ▷ Dann existiert ein k mit SAT $\in \boldsymbol{DTIME}(n^k)$.
- ▷ Da jede Sprace in $\boldsymbol{NP}$ auf SAT reduzierbar ist, gilt $\boldsymbol{NP} \subseteq \boldsymbol{DTIME}(n^k)$.
- ▷ Nach Voraussetzung gilt damit $\boldsymbol{P} \subseteq \boldsymbol{DTIME}(n^k)$.
- ▷ Damit kann $\boldsymbol{DTIME}(n^k)$ keine echte Teilmenge von $\boldsymbol{DTIME}(n^{k+1})$ sein, Widerspruch.
- ▷ Also folgt $\boldsymbol{P} \neq \boldsymbol{NP}$.

(2) [5 PUNKTE]

- ▷ Das E-Problem 3-FÄRBBARKEIT ist $\boldsymbol{NP}$-vollständig.
- ▷ Jeder Graph, der einen K_4 (= Clique mit 4 Knoten) als Untergraphen enthält, benötigt mindestens vier Farben zu einer legitimen (Knoten-)Färbung.
- ▷ Benötigt man umgekehrt vier oder mehr Farben, so muß spätestens dann eine vierten Farbe verwendet werden, wenn ein Knoten drei verschiedenfarbige Nachbarn hat, also Teil einer 4-Clique ist.
- ▷ Da die Anzahl der möglichen 4-Cliquen eines Graphen polynmial in der Knotenzahl ist, liegt 3-FÄRBBARKEIT in $\boldsymbol{P}$.
- ▷ Somit gilt $\boldsymbol{P} = \boldsymbol{NP}$.