Übungsblatt 4

Vorlesung Theoretische Grundlagen der Informatik im WS 19/20

Ausgabe: 3. Dezember 2019
Abgabe: 17. Dezember 2019, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1 (3 Punkte)

Die in der Vorlesung definierte nichtdeterministische Turing-Maschine wird auch als RV-NTM bezeichnet (Raten/Verifizieren). Eine andere Möglichkeit ist es, NTMs analog zu NEAs zu definieren (vgl. Wegener und Übung):

Eine solche alternative nichtdeterministische Turing-Maschine (A-NTM) ist definiert wie eine TM, nur ist die Übergangsfunktion δ durch eine zweistellige Relation $\subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R, N\})$ ersetzt, die wir ebenfalls δ nennen.

Die Arbeitsweise einer A-NTM ist die folgende. Wenn die A-NTM im Zustand $q \in Q$ das Zeichen $a \in \Gamma$ liest, ist für jedes $((q,a),(q',a',d)) \in \delta$ der Rechenschritt möglich, den eine DTM für $\delta(q,a) = (q',a',d)$ durchführt. Falls kein Rechenschritt möglich ist, stoppt die A-NTM. Für eine feste Eingabe x können offensichtlich viele Rechenwege möglich sein.

Eine A-NTM $\mathcal{M}$ akzeptiert eine Eingabe x, falls es mindestens einen Rechenweg von $\mathcal{M}$ gibt, der in einen akzeptierenden Endzustand führt. Die von $\mathcal{M}$ erkannte Sprache $L = L(\mathcal{M})$ besteht aus allen Worten, die $\mathcal{M}$ akzeptiert.

Die Rechenzeit für eine Eingabe $x \in L(\mathcal{M})$ ist gleich der Anzahl der Rechenschritte auf einem kürzesten akzeptierenden Rechenweg. Die Zeitkomplexität $T_{\mathcal{M}}(n)$ ist das Maximum der Rechenzeiten für alle Eingaben aus $L(\mathcal{M})$ der Länge n, und 1, falls es keine solche Eingabe gibt.

Die Klasse ANP ist die Klasse aller Probleme, die von einer solchen A-NTM in polynomialer Zeit entschieden werden können. In der Übung wurde gezeigt, dass $\mathsf{ANP} \subseteq \mathsf{NP}$ gilt. Zeigen Sie: $\mathsf{NP} \subseteq \mathsf{ANP}$.

Aufgabe 2 (5 Punkte)

In der Vorlesung wurde (ohne Beweis) behauptet, dass eine universelle Turing-Maschine existiert, die als Eingabe eine Gödelnummer w und ein Wort v bekommt und dann die Ausführung der Turing-Maschine T_w mit der Eingabe v simuliert. In dieser Aufgabe sollen Sie konkret die Arbeitsweise einer solchen universellen Turing-Maschine T_U beschreiben. Ihre universelle Turing-Maschine darf mehrere Bänder verwenden. Der Einfachheit halber können Sie davon ausgehen, dass ein festes Bandalphabet Γ vorgegeben ist, dass sowohl T_U als auch alle von T_U simulierten Turing-Maschinen verwenden. Die Laufzeit Ihrer Simulation ist unerheblich.

Ihre Beschreibung von T_U sollte mindestens folgende Fragen beantworten:

- Wieviele Bänder hat T_U und wofür werden sie benutzt?
- Wie repräsentiert T_U die aktuelle Bandbeschriftung von T_w?
- Wie repräsentiert T_U den aktuellen Zustand der endlichen Kontrolle von T_w?
- Wie repräsentiert T_U die aktuelle Kopfposition von T_w?
- Wie identifiziert T_U den passenden Übergang von T_w für den aktuellen Zustand und die aktuelle Bandbeschriftung?
- Wie führt T_U diesen Übergang aus, d.h. wie aktualisiert sie Bandbeschriftung, Kopfposition und Zustand?

Aufgabe 3 (3 Punkte)

Das Entscheidungsproblem ZWEIFACH-SAT ist wie folgt definiert: Gegeben ist eine Menge U von Variablen und eine Menge C von Klauseln (mit beliebig vielen Literalen) über U. Gefragt ist, ob es *mindestens zwei* erfüllende Wahrheitsbelegungen für C gibt. Zeigen Sie, dass ZWEIFACH-SAT NP-vollständig ist.

Achtung: ZWEIFACH-SAT ist nicht zu verwechseln mit 2SAT! Bei ZWEIFACH-SAT wird nach zwei erfüllenden Belegungen gefragt. Bei 2SAT ist die Klausellänge auf 2 *beschränkt.*

Aufgabe 4 (3 + 2 = 5 Punkte)

In dieser Aufgabe betrachten wir das Problem 2COLOR: Gegeben ist ein Graph $G = (V, E)$. Gefragt ist, ob es eine Färbung der Knoten mit zwei Farben gibt, sodass je zwei benachbarte Knoten nicht dieselbe Farbe haben.

(a) Es gilt 2COLOR $\in$ P. Geben Sie einen Polynomialzeitalgorithmus an, der entscheidet, ob ein gegebener Graph zweifärbbar ist.

(b) Gegeben ist folgender *falscher* Beweis für die NP-Schwere von 2COLOR:

> Reduziere von 3COLOR. Gegeben sei ein Graph $G = (V, E)$. Konstruiere daraus einen Graph G' mit einem zusätzlichen Knoten v, der zu allen anderen verbunden ist. Dies geht offensichtlich in polynomialer Zeit. Zeige: G besitzt genau dann eine Zweifärbung, wenn G' eine Dreifärbung besitzt.
> „$\Rightarrow$": Betrachte eine gültige Zweifärbung von G. Weise v die dritte, noch unbenutzte Farbe zu. Das ergibt offensichtlich eine gültige Dreifärbung.
> „$\Leftarrow$": Betrachte eine gültige Dreifärbung von G'. Da v zu allen anderen Knoten verbunden ist, kann die Farbe von v für keinen Knoten aus V benutzt worden sein. Also wurden für V nur zwei Farben benutzt und wir haben eine gültige Zweifärbung von G.

Wenn dieser Beweis richtig wäre, würde daraus P = NP folgen. Wo liegt der Fehler in dem Beweis?

Aufgabe 5

(2 + 1 (+ 2) = 3 (5) Punkte)

Für die Definition von NP-Schwere haben wir den Begriff der polynomialen Transformation verwendet. Ein Problem Π ist NP-schwer, wenn für jedes Problem $\Pi' \in \mathsf{NP}$ eine polynomiale Transformation $\Pi' \propto \Pi$ existiert.

Allgemeiner ist eine *Transformation* eines Entscheidungsproblems Π_1 in ein Entscheidungsproblem Π_2 definiert als eine Funktion $f\colon D_{\Pi_1} \to D_{\Pi_2}$, sodass $f(I)$ genau dann eine Ja-Instanz von Π_2 ist, wenn I eine Ja-Instanz von Π_1 ist. Für die *polynomiale* Transformation haben wir zusätzlich gefordert, dass f von einer DTM in Polynomialzeit berechnet werden kann.

In dieser Aufgabe sollen Sie untersuchen, warum diese zusätzlichen Einschränkungen notwendig sind. Sei dazu Π_1 ein beliebiges entscheidbares Problem und Π_2 ein beliebiges *nichttriviales* Entscheidungsproblem, d.h. Π_2 besitzt sowohl Ja-Instanzen als auch Nein-Instanzen.

(a) Zeigen Sie: Es gibt eine Transformation von Π_1 zu Π_2, die von einer DTM (in beliebiger Zeit) berechnet werden kann.

(b) ~~Zeigen Sie: Wenn $\Pi_1 \in \mathsf{NP}$ gilt, dann gibt es eine Transformation von Π_1 zu Π_2, die von einer NTM in Polynomialzeit berechnet werden kann.~~

(c) Welche Konsequenzen hätte es jeweils für die Klasse der NP-schweren Probleme, wenn wir statt einer polynomialen Transformation nur

  (i) eine Transformation, die von einer DTM berechenbar ist, oder

  (ii) ~~eine Transformation, die von einer NTM in Polynomialzeit berechenbar ist,~~

  fordern würden?

(d) *Bonusaufgabe:* In der Übung wurden die Komplexitätsklassen L und NL vorgestellt. Diese enthalten genau die Probleme, die sich in deterministisch bzw. nichtdeterministisch logarithmischem Platz lösen lassen.

  Wir wollen nun analog zur NP-Schwere den Begriff der NL-Schwere einführen. Wir führen dazu eine neue Transformation $\leq_{\mathsf{NL}}$ ein, deren Eigenschaften noch zu bestimmen sind. Wir nennen ein Problem Π_1 NL-schwer, wenn für jedes Problem $\Pi' \in \mathsf{NL}$ eine solche Transformation $\Pi' \leq_{\mathsf{NL}} \Pi$ existiert.

  In der Vorlesung wurde gezeigt: Wenn es ein NP-schweres Problem gibt, das in P liegt, dann gilt $\mathsf{P} = \mathsf{NP}$. Eine analoge Eigenschaft wollen wir auch für NL-Schwere haben: Wenn es ein NL-schweres Problem gibt, das in L liegt, dann gilt $\mathsf{L} = \mathsf{NL}$. Welche Eigenschaften muss die Transformation $\leq_{\mathsf{NL}}$ besitzen, damit dies gilt?

Anmerkung: Die Aufgabenstellungen von 5b und dem zweiten Teil von 5c sind fehlerhaft. Diese Aufgaben wurden dementsprechend gestrichen.

Aufgabe 6

(5 Punkte)

Das Entscheidungsproblem $\mathsf{LERNGRUPPE}$ ist wie folgt definiert: Gegeben sind eine Menge S von n Studenten, die sich auf die TGI-Klausur vorbereiten wollen, und eine Menge T von k klausurrelevanten Themen. Die Studenten wollen sich in 3 Lerngruppen aufteilen.

Zwar haben sich alle Studenten auf alle Themen vorbereitet, aber manche Studenten haben zu manchen Themen fehlerhaftes Wissen. Zum Beispiel sagt Alice: „Ein endlicher Automat ist genau dann NP-vollständig, wenn er einen Fehlerzustand hat.“ Bob sagt: „Der Äquivalenzklassenautomat entscheidet, ob P und NP äquivalent ist.“ Alice und Bob haben also fehlerhaftes Wissen zum Thema „endliche Automaten“. Wenn sie zusammen in eine Lerngruppe eingeteilt würden, könnten sie die anderen Teilnehmer von ihrem falschen Wissen überzeugen. Deswegen darf es innerhalb jeder Lerngruppe zu jedem Thema nur einen Studenten mit fehlerhaftem Wissen geben.

Formal stellen wir das fehlerhafte Wissen als Abbildung $f\colon S \times T \to \{0,1\}$ dar. Dabei gilt $f(s,t) = 1$ genau dann, wenn Student s zum Thema t fehlerhaftes Wissen hat. Beim Entscheidungsproblem LERNGRUPPE sind also S und T sowie die Abbildung f gegeben. Gesucht ist eine Einteilung von S in drei disjunkte Teilmengen G_1, G_2, G_3, sodass für jedes G_i und jedes Thema t gilt:

$$\sum_{s \in G_i} f(s,t) \leq 1$$

Zeigen Sie, dass LERNGRUPPE NP-vollständig ist. Reduzieren Sie von 3COLOR.

Aufgabe 7 (4 Punkte)

Die Komplexitätsklasse EXPTIME ist definiert als die Menge aller Sprachen, die von einer DTM in Exponentialzeit entschieden werden können, d.h. in Zeit $\mathcal{O}(2^{n^c})$ für eine Konstante c. Analog dazu ist NEXPTIME die Menge aller Sprachen, die von einer NTM in Exponentialzeit entschieden werden können.

Zeigen Sie: Wenn $\mathsf{P} = \mathsf{NP}$ gilt, dann gilt auch $\mathsf{EXPTIME} = \mathsf{NEXPTIME}$.

Hinweis: Gegeben eine Sprache $L \in$ NEXPTIME, konstruieren Sie eine Sprache L', die genau dann in polynomialer Zeit entscheidbar ist, wenn L in exponentieller Zeit entscheidbar ist. Versuchen Sie, die Wörter aus L geeignet auf exponentielle Länge zu „strecken“.