Gedankenprotokoll zur Fragestunde

July 25, 2016

1 Aufbau der Klausur

- Vier Aufgaben - je eine zu den schwerpunktmäßig behandelten Themenbereichen. Alle Aufgaben haben in etwa gleich viele Punkte.
- Jede Aufgabe ist in mehrere Teilaufgaben untergliedert. Die Teilaufgaben unterscheiden sich hinsichtlich Bepunktung und Schwierigkeit.

2 Erwartungshorizont

Wichtig sind zur Klausurvorbereitung die Übungsaufgaben, die in den Tutorien besprochen wurden. Dabei sind die Übungsaufgaben zu $\mathcal{F}_{\text{Loop}}, \mathcal{F}_{\text{While}}, \mathcal{F}_{\text{Goto}}$-Programmen und zu den fleißigen Bibern (s. Übungsaufgabe 26) nicht relevant, da die Inhalte nicht in der Vorlesung behandelt wurden. Allerdings können die Modelle (sofern nicht in der Aufgabenstellung ausgeschlossen) für Begründungen/Beweise herangezogen werden.

Weiter sind zu den Inhalten der Vorlesung sind folgende (an Schwierigkeit zunehmende) Aufgabentypen denkbar:

1. Beispiele erbringen (z.B. für ein $f \in \mathcal{F}_{\mu}^{tot}$, etc.),
2. die Aussage einer Definition, eines Theorems bzw. Lemmas wiedergeben,
3. Beweisideen (kurz) beschreiben,
4. Beweise vollständig formulieren.

Grundwissen aus Vorsemestern (z.B. über Grammatiken oder endliche Automaten) wird nicht explizit abgefragt, kann aber bei Bedarf verwendet werden.

Folgende Einschränkungen bzw. Hinweise sind zu beachten:

- Zu Theorem 4 ($\mathcal{F}_{\mu} \subseteq \mathcal{F}_{\text{TM}}$) genügt die Beweisidee, wie sie im Tutorium zu Übungsaufgaben 40 besprochen wurde. Die Programme der Turingmaschinen sind nicht wiederzugeben, ggf. jedoch die Idee.
- Zu Theorem 5 ($\mathcal{F}_{\mu} \supseteq \mathcal{F}_{\text{TM}}$) genügt die Beweisidee, wie sie im Tutorium zu Übungsaufgabe 41 besprochen wurde. Die Beweise der Lemma 9 - 11 sind nicht wiederzugeben, ggf. jedoch die Aussagen.
- Bei Theorem 6 und 7 sind neben dem Beweis insbesondere wichtig, wie die Aussagen der Theoreme in Vorlesung und Übung verwendet wurden. Gleiches gilt für Lemma 13.

- Theorem 8 (Satz von Rice) wird in der Klausur nicht abgefragt und muss nicht angewendet werden. Sie dürfen ihn allerdings verwenden, falls dies nicht explizit in der Aufgabenstellung ausgeschlossen wurde.

- Die Charakterisierung der Klasse *NP*durch Zertifikate steht zwar im Skript, wurde in der Vorlesung jedoch nicht behandelt. Entsprechende Definitionen, Theoreme und Lemmata in diesem Abschnitt sind nicht für die Prüfung relevant, außer sie wurden in der Vorlesung behandelt. Relevant sind jedoch Aufgaben des Typs Übungsaufgabe 47, siehe dazu weiter unten.

- Zu Theorem 5 (Satz von Cook) genügt im Wesentlichen die Beweisidee, wie in der Fragestunde (s.u.) an der Tafel skizziert wurde.

3 Behandelte Fragen in der Fragestunde

Laufzeiten und Eingabegrößen

Prof. G. Witzt behauptet, $P = NP$ gezeigt zu haben. Dazu hat er einen Algorithmus mit worst-case Laufzeit $O\left((n^2) \cdot M\right)$ für das Rucksack-Problem entwickelt, wobei n die Anzahl der Objekte und M das Maximum über alle Nutzenwerte der Objekte ist. Kann es einen solchen Algorithmus geben? Hat er wirklich $P = NP$ gezeigt?

Nein, das hat er nicht. Zu beachten ist, dass die Laufzeit des Algorithmus von M abhängt. M wird in der Eingabe der Turingmaschine durch $\log(M)$ Bits dargestellt. Der Algorithmus ist damit nicht polynomiell in der Eingabelänge. Siehe hierzu auch Skript S. 51/52 zum Rucksackproblem.

Folgerung aus Reduktion

Seien $L_1 \leq_{n \log n} L_2$ Sprachen. Kann (möglicherweise) L_2 in Zeit $O(n^2)$ entschieden werden, wenn man weiß, dass

- L_1 nicht in Zeit $O(n)$ entschieden werden kann? Ja, ggf. schon.

- L_1 nicht in Zeit $O(n^2)$ entschieden werden kann? Nein, denn sonst könnte L_1 (nach Anwenden der Reduktionsfunktion) doch in $O(n^2)$ entschieden werden.

- L_1 nicht in Zeit $O(n^3)$ entschieden werden kann? Nein. (Begründung wie bisher)

Begründung warum Reduktionsfunktion polynomiell berechenbar

In Übungsaufgabe 44 haben wir $\textsc{Partition} \leq_p \textsc{SubsetSum}$ mit einer Reduktionsfunktion f gezeigt. Die Eingabe von f war eine Menge M aus $|M| = m$ Zahlen. Wie kann begründet werden, dass f in polynomieller Zeit auf einer DTM berechenbar ist?

Dazu können wir das Modell der RAM verwenden. In der Vorlesung wurde gezeigt, dass sich eine RAM mit polynomieller Laufzeit im logarithmischen Kostenmaß durch eine 1-Band DTM in polynomieller Zeit simulieren lässt. (Überlegen Sie sich, warum das im Einheitskostenmaß nicht funktioniert!) Im logarithmischen Kostenmaß kann die RAM die m Additionen zur Berechnung von f in polynomieller Zeit durchführen. Aus diesem Grund trifft dies auch für die DTM zu.

Aufgaben zu μ-rekursiven Funktionen

Durch Rückführung auf die Definition soll gezeigt werden, dass eine gegebene Funktion primitiv bzw. μ-rekursiv ist. Verwenden Sie dazu nur die Funktionen aus der Definition und die eventuell zusätzlich in der Aufgabenstellung als primitiv rekursiv gegebenen Funktionen. Sollten Sie in der finalen Darstellung auf Projektionen des Eingabevektors verzichten oder Vertauschungen der Variablen vornehmen, so ist zusätzlich kurz zu erläutern und zu begründen. Eine Beispielaufgabe ist:

Zeigen Sie durch Herleitung über die Definition, dass

- $f(x, y) = \lceil \log_x(y) \rceil$
- $g(x, y) = \begin{cases} \log_x(y), & \text{falls } \log_x(y) \in \mathbb{N}_0 \\ \textit{undefiniert}, & \text{sonst.} \end{cases}$

in $\mathcal{F}_\mu$ liegen. Sie dürfen dazu verwenden, dass Addition, modifizierte Differenz und die Multiplikation primitiv rekursiv sind. (Siehe Übungsaufgabe 10, bzw. Beispiel 2 im Skript S. 5.)

Aufgabe 47, Blatt 12

Folgende Hilfestellungen helfen beim Beweis. Es muss jedoch noch gezeigt werden, dass die geratene Lösung in polynomieller Zeit überprüft werden kann.

L_1 Rate binäre Kodierung $y \in \{0, 1\}^{|w|}$ eines Teilers mit $1 < y < w$. Teile anschließend mit Rest und prüfe, ob Rest 0 ist.

L_2 Rate einen Vektor $y \in \{0, 1\}^{|M|}$, wobei die i-te Stelle 1 ist, falls das i-te Element aus M in der Teilmenge T enthalten ist und 0 sonst. Addiere alle Elemente aus T, multipliziere mit 2 (Bitshift) und prüfe, ob dies der Summe der Elemente aus M entspricht.

Aufgabe 48, Blatt 12 (Satz von Cook und Levin)

Eine Skizze des Beweises sollte insbesondere die folgenden Punkte umfassen:

1. Warum SAT $\in$ *NP* ist, d.h. eine kurze Beschreibung des Programms der NTM. (siehe Lemma 15)
2. Warum SAT *NP*-vollständig ist. Dazu:
   (a) Wie aus der Turingmaschine der aussagenlogische Ausdruck
   $$A(x) = S \wedge R \wedge U \wedge q_{p(n),u}$$
   konstruiert werden kann. Was $A(x)$ insgesamt und was die Bestandteile ausdrücken, d.h. jeweils kurze Beschreibung (1-2 Sätze). Welche Variablen insgesamt eingeführt wurden, um die obigen Bestandteile zu beschreiben.
   (b) Beweisskizze, warum $x \in L \Leftrightarrow A(x) \in \text{SAT}$.
   (c) $A(x)$ besteht aus $O(p^3(n))$ Literalen. Die Größe von N ist konstant. Daher kann $A(x)$ mit einer DTM in polynomieller Zeit (d.h. polynomiell in x) konstruiert werden.