Lehrstuhl für Informatik 1
Prof. Dr. Berthold Vöcking
Alexander Skopalik
Thomas Kesselheim

WS 2009/10
02.02.2010

Lösungen zur Vorlesung

Berechenbarkeit und Komplexität

2. Zulassungsklausur

Aufgabe 1:

(a) Worin besteht der Unterschied zwischen der Definition von NTM und TM? **(1 Punkt)**

Die Zustandsübergangsfunktion δ der TM wird zu einer Relation bei der NTM.

(b) Wie ist das Akzeptanzverhalten einer NTM definiert? **(1 Punkt)**

Eine NTM akzeptiert, genau dann wenn ein Berechnungspfad existiert, auf dem die NTM eine akzeptierende Endkonfiguration erreicht.

(c) Wie ist die Laufzeit einer NTM bei Eingabe x im Falle einer akzeptierenden Rechnung definiert? **(2 Punkte)**

Die Laufzeit der NTM ist die Länge eines kürzesten akzeptierenden Rechenweges, von q_0 bis zur akzeptierenden Konfiguration.

(d) Definieren Sie die Komplexitätsklasse P. **(1 Punkt)**

P ist die Klasse der Probleme, für die es einen Polynomialzeitalgorithmus gibt.

(e) Definieren Sie die Klasse NP unter Verwendung von NTMs. **(1 Punkt)**

$L \in \mathrm{NP} \Leftrightarrow \exists$ NTM M, die L erkennt und polynomielle Laufzeit hat.

(Name / Matrikelnummer)

Aufgabe 1:

(f) Definieren Sie die Klasse NP unter Verwendung von Zertifikaten und Verifizierern. **(2 Punkte)**

$L \in \text{NP} \Leftrightarrow \exists$ (det.) Verfizierer V, der bei Eingabe x und einem Zertifikat y (y polynomiell beschränkt in $|x|$) genau dann akzeptiert, falls $x \in L$.

(g) Seien L_1 und L_2 Sprachen. Definieren Sie die polynomielle Reduktion $L_1 \leq_p L_2$. **(2 Punkte)**

Seien L_1 und L_2 zwei Sprachen über Σ_1 bzw. Σ_2. L_1 ist polynomiell reduzierbar auf L_2, wenn es eine Reduktion, d.h. eine berechenbare Abbildung $f : \Sigma_1^* \to \Sigma_2^*$ mit der Eigenschaft: $\forall x \in \Sigma_1^* : x \in L_1 \Leftrightarrow f(x) \in L_2$, von L_1 nach L_2 gibt, die in polynomieller Zeit berechenbar ist.

(h) Sei A NP-hart. Um zu zeigen, dass B NP-hart ist, zeigt man (bitte ankreuzen) **(1 Punkt)**

◯ $A \leq_p B$ oder ◯ $B \leq_p A$

(i) Was folgt aus $A \leq_p SAT$ und $SAT \leq_p A$ (mehrere Antworten sind möglich)? **(1 Punkt)**

◯ $A \in$ P ◯ $A \in$ NP ◯ A ist NP-hart ◯ nichts von all dem

(j) Was ist die Aussage des Satzes von Cook und Levin? **(1 Punkt)**

SAT ist NP-vollständig.

(Name / Matrikelnummer)

Aufgabe 1:

(k) Für ein Entscheidungsproblem A gibt es einen Algorithmus mit Laufzeitschranke $O(n^4)$. Die polynomielle Reduktion $B \leq_p A$ benötige Laufzeit $O(n^5)$. Dabei bezeichne n jeweils die Eingabelänge. Welche Laufzeit kann man daraus für einen Algorithmus für B folgern?

(1 Punkt)

Laut Aufgabenstellung kann man mit Hilfe der Reduktion f eine Probleminstanz des Entscheidungsproblems B in Zeit $O(n^5)$ in eine Instanz des Entscheidungsproblems A transformieren. Dadurch entsteht eine Ausgabe der Länge m (höchstens $O(n^5)$). Anschließend kann die transformierte Instanz in Zeit $O(m^4)$ gelöst werden. Somit folgt für die Lösung der ursprünglichen Probleminstanz des Entscheidungsproblems B eine obere Schranke von $O((n^5)^4) = O(n^{20})$.

(l) Wann wird eine Programmiersprache als *Turing-mächtig* bezeichnet? **(1 Punkt)**

Wenn jede Funktion, die von einer TM berechnet werden kann, auch von einer Funktion dieser Programmiersprache berechnet werden kann.

(m) Geben Sie zwei Beispiele für Programmiersprachen, die Turing-mächtig sind, an. **(2 Punkte)**

WHILE-Programme, Java

(n) Geben Sie ein Beispiel für eine Programmiersprache, die nicht Turing-mächtig ist, an. **(1 Punkt)**

LOOP-Programme

(Name / Matrikelnummer)

Aufgabe 1:

(o) Ist das Entscheidungsproblem, ob ein gegebenes WHILE-Programm eine gegebene Eingabe akzeptiert, rekursiv? — Begründen Sie Ihre Antwort. **(1 Punkt)**

Nein, denn WHILE-Programme sind Turing-mächtig. Man könnte sonst das spezielle Halteproblem entscheiden.

(p) Ist das Entscheidungsproblem, ob ein gegebenes LOOP-Programm eine gegebene Eingabe akzeptiert, rekursiv? — Begründen Sie Ihre Antwort. **(1 Punkt)**

Ja, denn LOOP-Programme terminieren immer.

(Name / Matrikelnummer)

Aufgabe 2:

(a) Definieren Sie das Problem 3SAT. **(2 Punkte)**

Eingabe: Formel φ in 3KNF

Frage: Ist φ erfüllbar.

(b) Definieren Sie das Independent-Set-Problem (IS). **(2 Punkte)**

Eingabe: Graph $G = (V, E)'$ und Zahl k

Frage: Existiert eine Teilmenge I der Größe k von V, so daß keine Kante zwischen zwei Knoten aus I existiert.

(c) Beschreiben Sie eine polynomielle Reduktion 3SAT $\leq_p$ IS und beweisen Sie ihre Korrektheit. **(13 Punkte)**

Für eine Formel $\varphi = \bigwedge_{1 \leq j \leq m} C_j$ mit $C_j = (l_{j,1} \vee l_{j,2} \vee l_{j,3})$ und $l_{j,i} \in \{X_i, \neg X_i \mid 1 \leq i \leq n\}$ konstruieren wir einen Graphen $G_\varphi = (V, E)$, der ein Independent Set der Größe m genau dann enthält, wenn φ erfüllbar ist.

Setze $V = \bigcup_{1 \leq j \leq m} C_j$ (also ein Knoten für jedes Literal, insgesamt also $3m$ Knoten) und $E = \{\{l_{j,1}, l_{j,2}\}, \{l_{j,2}, l_{j,3}\}, \{l_{j,3}, l_{j,1}\} \mid 1 \leq j \leq m\} \cup \{\{l, l'\} \mid l = \neg l'\}$ (Eine Kante zwischen Literalen in derselben Klausel und zwischen zwei Literalen, von denen je eines negiert und eines nicht negiert ist).

Laufzeit: Offensichtlich polynomiell

Korrektheit: Wenn G ein IS I der Größe m enthält, dann:

(a) enthält I höchsten einen Knoten pro Klausel.

(b) Für $1 \leq j \leq m$ enthält I genau ein Knoten aus der Menge $\{l_{j,1}, l_{j,2}, l_{j,3}\}$.

(c) Es gibt keine zwei Knoten $l, l' \in I$ so dass das Literal l eine Negation von l' ist.

(d) Wähle nun eine Variablenbelegung für φ so, dass alle Literale erfüllt werden.

Sein nun T eine Variablenbelegung die φ erfüllt.

(a) Wähle für jede Klausel genau ein Literal, dass unter T wahr ist.

(b) Nach Konstruktion die die korrespondierende Knotenmenge unabhängig.

- Zwei Literale X und $\neg X$ können nicht gleichzeitig wahr sein.
- Wir haben keine zwei Literale aus der selben Klausel gewählt.

(Name / Matrikelnummer)

Aufgabe 2:

(d) Wenden Sie Ihre Reduktion auf die Formel $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ an. Zeichnen Sie den resultierenden Graphen. **(3 Punkte)**

- x_1
- x_2
- $\neg x_3$
- $\neg x_1$
- x_2
- x_3

(Name / Matrikelnummer)

Aufgabe 3:

(a) Das Entscheidungsproblem DoubleSAT sei wie folgt definiert:

Eingabe: Eine KNF-Formel ϕ

Ausgabe: Ja, gdw. ϕ mindestens zwei erfüllende Belegungen besitzt.

Beschreiben Sie eine polynomielle Reduktion SAT $\leq_p$ DoubleSAT und beweisen Sie ihre Korrektheit. **(8 Punkte)**

Berechne zu einer Eingabe ϕ für SAT eine Eingabe ϕ' für DoubleSAT wie folgt:

Sei X eine Variable, die nicht in ϕ vorkommt. Setze $\phi' = \phi \wedge (X \vee \neg X)$.

ϕ' ist in KNF, da nur eine Klausel hinzugefügt wurde. Ferner kann ϕ' in Polynomialzeit berechnet werden.

Es gilt:

ϕ ist erfüllbar
$\Leftrightarrow$ ex. Belegung, die ϕ erfüllt
$\Leftrightarrow$ ex. Belegung mit X = 0, die ϕ' erfüllt, und ex. Belegung mit X = 1, die ϕ' erfüllt
$\Leftrightarrow$ ϕ' hat zwei erfüllende Belegungen

(Name / Matrikelnummer)

Aufgabe 3:

(b) Das Entscheidungsproblem HalfIndependentSet sei wie folgt definiert:

Eingabe: Ein ungerichteter Graph G mit $2n$ Knoten.

Ausgabe: Ja, gdw. ein Independent Set der Größe n existiert.

Zeigen Sie, dass HalfIndependentSet NP-vollständig ist.
(Hinweis: Nutzen Sie eine Reduktion von IndependentSet.)

(12 Punkte)

Zu zeigen sind:

1. HalfIndependentSet $\in$ NP
2. HalfIndependentSet ist NP-hart. Hierzu zeige: IndependentSet $\leq_p$ HalfIndependentSet.

Es gilt HalfIndependentSet $\in$ NP, da HalfIndependentSet ein Speziallfall von IndependentSet ist.

Aus einer Eingabe $G = (V, E)$, b mit $|V| = N$ für IndependentSet berechne eine Eingabe $G' = (V', E')$ für HalfIndependentSet wie folgt:

- Falls $b \geq \frac{N}{2}$, füge $2b - N$ Knoten ein. Jeder eingefügte Knoten wird mit allen anderen Knoten verbunden.
- Falls $b < \frac{N}{2}$, füge $N - 2b$ Knoten ein. Jeder eingefügte Knoten hat keine Kanten.

Laufzeit Das Hinzufügen der Kanten und Knoten kann in Polynomzeit durchgeführt werden, da ihre Anzahl auch polynomiell beschränkt ist.

Behauptung G hat Independent Set der Größe $b \Leftrightarrow G'$ hat Independent Set der Größe $\frac{|V'|}{2}$.
Falls $b \geq \frac{N}{2}$ gilt:

$$G \text{ hat IS der Größe } b \quad \Leftrightarrow \quad G' \text{ hat IS der Größe } b = \frac{|V'|}{2}$$

denn keiner der hinzugefügten Knoten kann im IS vorkommen (es sei denn $b = 1$)
Falls $b < \frac{N}{2}$ gilt:

$$G \text{ hat IS der Größe } b \quad \Leftrightarrow \quad G' \text{ hat IS der Größe } b + (N - 2b) = N - b = \frac{2N - 2b}{2} = \frac{|V'|}{2}$$

denn alle hinzugefügten Knoten können dem IS hinzugefügt werden.