Prof. Dr. Andreas Podelski
Dr. Matthias Heizmann
Alexander Nutz

Abgabe: 6. Februar 2019

14. Übungsblatt zur Vorlesung
Informatik III

Die hier enthaltenen Lösungen werden am 8. Februar in der Vorlesung vorgestellt. Wenn keine Fragen gestellt werden, wird die Präsentation der Lösungen zügig vonstattengehen. Alle Vorlesungsteilnehmer sind aber herzlichen eingeladen Fragen zu stellen und sich Details genauer erklären zu lassen. Genauere Erklärungen werden in die Lösung einfließen und diese Lösungen werden (zumindest bis zur Klausur) dauerhaft online bleiben.

Aufgabe 1: Eigenschaften der Reduktionsrelation 2 Punkte

Beweisen Sie die folgende Aussage.

> Die Reduktionsrelation $\preceq_p$ ist reflexiv und transitiv.

Lösung

Anmerkung: Die Lösung ist nahezu identisch zum analogen Beweis für $\preceq$.
Seien $L_i \subseteq \Sigma_i^*, i \in \{1, 2, 3\}$.

reflexiv Setze $f := \text{id}$. Die Identität ist offensichtlich total und polynomiell berechenbar und es gilt

$$w \in L_1 \Leftrightarrow \underbrace{f(w)}_{w} \in L_1.$$

Damit gilt $L_1 \preceq_p L_1$ mittels f.

transitiv Gelte $L_1 \preceq_p L_2$ mittels f_1 und $L_2 \preceq_p L_3$ mittels f_2. Setze $f := f_2 \circ f_1$. Die Komposition zweier totaler, polynomiell berechenbarer Funktionen ist offensichtlich total und es gilt

$$w \in L_1 \Leftrightarrow f_1(w) \in L_2 \Leftrightarrow \underbrace{f(w)}_{f_2(f_1(w))} \in L_3.$$

Damit gilt $L_1 \preceq_p L_3$ mittels f.

f ist ebenfalls polynomiell berechenbar, da Polynome unter Komposition abgeschlossen sind. Seien $k_1, k_2 \in \mathbb{N}$ zwei Konstanten, sodass f_i in $O(n^{k_i})$ liegt. Dann liegt f in $O(n^{k_1 \cdot k_2})$.

Aufgabe 2: $SAT \in P$ **?** 1 Punkt

Während der Erstellung der Aufgaben ist uns folgender Algorithmus eingefallen, der SAT in quadratischer Zeit löst und damit zeigt, dass $P = NP$ gilt. Gegeben sei eine boolesche Formel F mit den Variablen $x_1, \ldots, x_k$. Offensichtlich ist $k \leq |F|$.

1. Wir reduzieren das Problem, ob F erfüllbar ist, auf ein einfacheres Problem mit der Formel $F^{(1)}$, die $k-1$ Variablen enthält. Die Reduktion ersetzt jedes Vorkommen von x_k in F einmal durch 0 und einmal durch 1:

   $$F^{(1)} = F[x_k := 0] \vee F[x_k := 1]$$

   Der Algorithmus lässt sich auf einer Zweibandturingmaschine in $3 \cdot |F| + c$ Schritten durchführen, wobei c eine kleine Konstante ist. Insgesamt ist die Reduktion in $O(n)$. Außerdem ist $F^{(1)}$ genau dann erfüllbar, wenn F erfüllbar ist.

2. Wir wiederholen den ersten Schritt k Mal, bis wir eine Formel $F^{(k)}$ erhalten, die keine Variablen mehr enthält. Der Algorithmus hat Laufzeit $k \cdot O(n)$, also $O(n^2)$.

3. Im letzten Schritt berechnen wir den Wahrheitswert von $F^{(k)}$. Weil die Formel keine Variablen mehr enthält, ist das in linearer Zeit möglich. Der gesamte Algorithmus hat also Zeitkomplexität $O(n^2 + n) = O(n^2)$.

Erklären Sie *kurz*, wo der Fehler liegt.

.....Lösung .....

Die Konstruktion in Schritt 1 verdoppelt die größe der Formel F. Die Zeitkomplexität für eine Ausführung ist also linear. Allerdings ist die Ausgabe einer Ausführung von Schritt 1 die Ausgabe der folgenden Ausführung von Schritt 1. In k Schritten wächst die Eingabe somit um Faktor 2^k, also exponentiell.
Betrachtet man Schritt 3, ist die Laufzeit hier nur linear zum exponentiell gewachsenen Input. Der Algorithmus löst das SAT-Problem also nur in exponentieller Zeit und seine Existenz sagt nichts über das Verhältnis der Klassen P und NP aus.

Aufgabe 3: Reduktion 2 Punkte

Welche der folgenden Reduktionen gelten? Beweisen Sie Ihre Behauptungen.

(a) $SAT \preceq H$

(b) $H \preceq SAT$

(c) $SAT \preceq_p H$

(d) $H \preceq_p SAT$

Dabei ist H das allgemeine Halteproblem für Turingmaschinen und SAT das Erfüllbarkeitsproblem für Boolesche Ausdrücke in konjunktiver Normalform.

.....Lösung .....

(a,c) Es gilt $SAT \preceq_p H$, also auch $SAT \preceq H$.

Da SAT entscheidbar ist, existiert eine Turingmaschine $\mathcal{M}$, die SAT entscheidet. Wir konstruieren eine neue Turingmaschine $\mathcal{M}'$, die wie $\mathcal{M}$ funktioniert, aber anstatt zu verwerfen in eine Endlosschleife geht.

Definiere $f(w) := \ulcorner \mathcal{M}' \urcorner \# w$. f ist total und mit Zeitaufwand von $O(|w|)$ berechenbar ($\ulcorner \mathcal{M}' \urcorner$ ist konstant) (es gibt sogar eine Implementierung in $O(|w|)$, da die TM einfach nach links laufen kann).

$w \in SAT \Leftrightarrow \mathcal{M}$ akzeptiert $w \Leftrightarrow \mathcal{M}'$ hält auf $w \Leftrightarrow f(w) \in H$

(b,d) Es gilt nicht $H \preceq SAT$, also auch nicht $H \preceq_p SAT$.

Da SAT entscheidbar ist, würde folgen, dass H entscheidbar ist – ein Widerspruch.

Aufgabe 4: Polynomielle Reduktion 3 Punkte

Gegeben sei ein Graph $G = (V, E)$. Ein Hamiltonpfad in G ist ein Pfad, der jeden Knoten in V genau einmal besucht.

Das Problem GHP (gerichteter Hamiltonpfad) ist wie folgt definiert:

> *Gegeben:* Ein gerichteter Graph $G = (V, E)$.
>
> *Frage:* Besitzt G einen gerichteten Hamiltonpfad?

Das Problem UHP (ungerichteter Hamiltonpfad) ist wie folgt definiert:

> *Gegeben:* Ein ungerichteter Graph $G = (V, E)$.
>
> *Frage:* Besitzt G einen ungerichteten Hamiltonpfad?

Zeigen Sie: Das Hamiltonpfadproblem für ungerichtete Graphen lässt sich polynomiell auf das Hamiltonpfadproblem für gerichtete Graphen reduzieren, also

$$\text{UHP} \preceq_p \text{GHP}.$$

Es genügt, wenn Sie Ihre Laufzeitabschätzung grob begründen. Sie müssen weder Pseudocode noch Turingmaschinen explizit angeben.

.....Lösung .....

- Machen Sie sich mit der Aufgabenstellung vertraut.

  Meinung des Autors: Um eine Definition zu verstehen sollte man sich Beispiele machen. Überlegen Sie sich einen Graphen der einen gerichteten Hamiltonpfad hat, überlegen Sie sich einen Graphen der keinen gerichteten Hamiltonpfad hat, machen Sie selbiges für den ungerichteten Fall. Besprechen Sie Ihr Beispiel mit mindestens einem anderen Menschen. (Zustandsdiagramme von Automaten, Flußdiagramme, Bäume) Alternative/Ergänzung: Überlegen Sie sich welche Graphen Ihnen in letzer Zeit begegneten, fragen Sie sich ob diese Hamiltonpfade haben.

  Überlegen Sie sich für jedem Begriff in der Aufgabenstellung ob Sie dessen Definition auswendig kennen. Falls nein – kein Problem – schauen Sie sich die Definition im Skript an.

- Überlegen Sie sich eine Reduktionsfunktion.

  Dafür kann es sinnvoll sein sich zu überlegen was Reduzierbarkeit anschaulich bedeutet. Wenn UHP $\preceq_p$ GHP gilt bedeutet dies dass wir jedes UHP Problem mit Hilfe eines GHP Problems lösen können. Unsere Aufgabe ist also eine Übersetzung von gerichteten Graphen auf ungerichtete Graphen zu konstruieren, sodass wir Elemente aus UHP auf Elemente aus GHP abbilden und wir Elemente die nicht in UHP liegen auf Elemente abbilden die nicht in GHP liegen. Das Finden solch einer Abbildung ist ein kreativer Art für den es kein standardisiertes Vorgehen gibt.

Gegeben sei ein ungerichteter Graph G. Als Reduktionsfunktion betrachten wir folgende Konstruktion $f : G \mapsto G'$ eines neuen, gerichteten Graphen G'.

- G' übernimmt die Knoten von G.
- Für jede Kante von G (zwischen Knoten v_i und v_j) werden zwei Kanten in G' (von v_i nach v_j und umgekehrt) eingefügt.

G' ist offensichtlich ein gerichteter Graph, da nur gerichtete Kanten eingefügt wurden. Die Konstruktion von f erfolgt in linearer Zeit zur Anzahl der Knoten und Kanten in G. Falls die Eingabe keinen Graphen codiert, so bildet f ebenfalls auf ein Wort ab, das keinen gerichteten Graphen codiert. Damit ist f auch total.

$$f(w) = \begin{cases} G' = (V', E') & \text{falls } w \text{ einen Graph } G = (V, E) \text{ codiert} \\ \text{ungültige Codierung für } \textsc{Ghp} & \text{sonst} \end{cases}$$

Bei geeigneter Codierung ist f sogar die Identitätsfunktion.

Wir zeigen jetzt die Reduktionseigenschaft von f, d.h.

$$\begin{aligned} w \in \textsc{Uhp} &\iff w \text{ codiert einen ungerichteten Graphen } G \text{ mit einem Hamiltonpfad} \\ &\overset{*}{\iff} f(w) \text{ codiert einen gerichteten Graphen } G' \text{ mit einem Hamiltonpfad} \\ &\iff f(w) \in \textsc{Ghp} \end{aligned}$$

Für den mittleren Schritt $*$ argumentieren wir nochmal genauer:
G hat Hamiltonpfad $\iff$ G' hat gerichteten Hamiltonpfad.

"$\Rightarrow$" Offensichtlich, da für jede Kante zwei gerichtete Kanten in jeweils entgegengesetzte Richtungen konstruiert wurden. Somit ist jede Verbindung in G auch in G' in beide Richtungen benutzbar.

"$\Leftarrow$" Ebenfalls offensichtlich, da keine "neuen Wege" entstehen. Man kann in G' von jedem Knoten aus nur diejenigen Knoten erreichen, die auch in G erreichbar gewesen wären. Das gilt, da keine Kanten in G' zwischen Knoten, die in G nicht verbunden waren, eingefügt wurden.

Aufgabe 5: NP-Vollständigkeit 4 Punkte

Das Problem HSet (Hitting Set) ist wie folgt definiert:

> *Gegeben:* Eine Menge M und eine Menge von Teilmengen $\mathcal{S}$ (d.h. $\mathcal{S} = \{S_1, \ldots, S_n\}$, $S_i \subseteq M$ für $i = 1, \ldots, n$) sowie eine natürliche Zahl $k \leq n$.
>
> *Frage:* Gibt es eine Teilmenge $T \subseteq M$, sodass $|T| \leq k$ und $T \cap S_i \neq \emptyset$ für $i = 1, \ldots, n$? Mit anderen Worten: Gibt es eine höchstens k-elementige Teilmenge T, die mit jedem S_i mindestens ein gemeinsames Element hat?

Das *NP*-vollständige Problem Knüb (Knotenüberdeckung bzw. Vertex Cover) ist wie folgt definiert.

> *Gegeben:* Ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl $k \leq |V|$.
>
> *Frage:* Besitzt G eine „überdeckende Knotenmenge“ der Größe höchstens k? Eine überdeckende Knotenmenge ist eine Teilmenge $V' \subseteq V$, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V'$ oder $v \in V'$.

(a) Begründen Sie, warum HSet in *NP* liegt.

(b) Beweisen Sie, dass HSet *NP*-vollständig ist. Sie dürfen dabei verwenden, dass Knüb NP-vollständig ist.

Es genügt, wenn Sie Ihre Laufzeitabschätzung grob begründen. Sie müssen weder Pseudocode noch Turingmaschinen explizit angeben.

Lösung

(a) HSet $\in NP$:
Rate eine k-Teilmenge $T \subseteq M$ sowie je ein Element $s_j \in S_i$ für jedes $S_i \in \mathcal{S}$. Prüfe anschließend für alle $j = 1, \ldots, n$, ob $s_j \in T$. Das ist mit Zeitaufwand $O(k \cdot n)$ möglich.

Anmerkung: Das Raten der s_j ist optional; diese kann man auch in polynomieller Zeit bestimmen.

(b) Wir abstrahieren hier von einer konkreten Codierung und betrachten nur die Komponenten. Die Mengen Knüb bzw. HSet sind hier nochmal übersichtlicher dargestellt:

$$\begin{aligned}
\text{Knüb} := \{(V, E, k) \mid\ & E \subseteq V^2 \text{ und } \overbrace{(\forall (u,v) \in E : (v,u) \in E)}^{\text{ungerichtet}} \text{ und } k \in \mathbb{N} \text{ und} \\
& \exists V' \subseteq V : |V'| \leq k \text{ und } \forall (u,v) \in E : u \in V' \text{ oder } v \in V'\} \\
\text{HSet} := \{(M, \mathcal{S}, k') \mid\ & \mathcal{S} = \{S_1, \ldots, S_n\} \text{ und } \forall S_i \in \mathcal{S} : S_i \subseteq M \text{ und } k' \in \mathbb{N} \text{ und} \\
& \exists T \subseteq M : |T| \leq k' \text{ und } \forall S_i \in \mathcal{S} : T \cap S_i \neq \emptyset\}
\end{aligned}$$

Knüb $\preceq_p$ HSet mittels f:

$$f(w) = \begin{cases} (M, \mathcal{S}, k') \text{ mit } M := V, \mathcal{S} := \underbrace{\{\{u,v\} \mid (u,v) \in E\}}_{\text{aus Kanten werden Mengen}}, k' := k & \text{falls } w = (V, E, k) \\ \text{ungültige Codierung für HSet} & \text{sonst} \end{cases}$$

Offensichtlich gilt $f \in O(|V|^2)$ bzw. je nach Repräsentation sogar $f \in O(1)$.

$$\begin{aligned}
w \in \text{Knüb} &\iff \exists V, E, k : w = (V, E, k) \text{ und} \\
&\quad \exists V' \subseteq V : \ |V'| \leq k \text{ und } \forall (u,v) \in E : u \in V' \vee v \in V' \\
&\iff \exists M, \mathcal{S}, k : f(w) = (M, \mathcal{S}, k) \text{ und} \\
&\quad \exists T \subseteq M : \ |T| \leq k \text{ und } \forall \{u,v\} \in \mathcal{S} : u \in T \vee v \in T \\
&\overset{*}{\iff} \exists M, \mathcal{S}, k : f(w) = (M, \mathcal{S}, k) \text{ und} \\
&\quad \exists T \subseteq M : \ |T| \leq k \text{ und } \forall S_i \in \mathcal{S} : T \cap S_i \neq \emptyset \\
&\iff f(w) \in \text{HSet}
\end{aligned}$$

Die mit $*$ gekennzeichnete Umformung gilt aufgrund der Definition von f, denn $\mathcal{S}$ kann nur zweielementige Mengen enthalten.