Skript Komplexitätstheorie

Andreas Abel

2006-10-31

3.2 Vollständigkeit

Definition 1

1. L ist P-*reduzierbar* auf L', geschrieben $L \leq_{\mathsf{P}} L'$, wenn es ein $f \in \mathsf{FP}$ gibt, so dass $x \in L$ gdw. $f(x) \in L'$ für alle x.
2. L heißt NP-*schwer* (engl. NP-*hard*), wenn $L' \leq_{\mathsf{P}} L$ für alle $L' \in \mathsf{NP}$.
3. L heißt NP-*vollständig* (engl. NP-*complete*), wenn L NP-schwer und $L \in \mathsf{NP}$.
4. $\mathsf{NPC} = \{L \in \mathsf{NP} \mid L \text{ } \mathsf{NP}\text{-schwer}\}$ bezeichnet die Klasse der NP-vollständigen Probleme.

Bemerkung 2 In der ersten Def. wird nicht gefordert, dass f injektiv ist.

Lemma 3 (Eigenschaften)

1. $\leq_{\mathsf{P}}$ *is reflexiv (mit* f *Identitätsfunktion).*
2. $\leq_{\mathsf{P}}$ *is transitiv (da* P *abgeschlossen unter Komposition).*
3. *Wenn* L NP*-schwer und* $L \leq_{\mathsf{P}} L'$*, dann auch* L' NP*-schwer.*
4. *Wenn* $\mathsf{NPC} \cap \mathsf{P} \neq \emptyset$*, dann* $\mathsf{P} = \mathsf{NP}$*.*

Wiederholung 4 (Das Problem 3SAT) Gegeben: Variablenmenge $\{x_1, \ldots, x_n\}$ und

$$\begin{array}{lcll} F & = & \bigwedge_{i \leq m} C_i & \textit{Klauselmenge, wobei} \\ C_i & = & a_{i1} \vee a_{i2} \vee a_{i3} & \textit{Klausel, und} \\ a_{ij} & = & x_l \text{ oder } \overline{x}_l & \textit{Literal.} \end{array}$$

Gesucht: Belegung der Variablen, die F erfüllt.

Wiederholung 5 (Das Problem VC "vertex cover") Gegeben ist ein ungerichteter Graph $G = (V, E)$ und ein $k \in \mathbb{N}$. Gesucht ist eine Teilmenge $U \subseteq V$ von Knoten der Größe $|U| \leq k$, die alle Kanten überdeckt, d.h. $e \cap U \neq \emptyset$ für alle $e \in E$.

Lemma 6 $\mathsf{3SAT} \leq_{\mathsf{P}} \mathsf{VC}$.

Beweis. Aus einem 3SAT-Problem $(\{x_i \mid 1 \leq i \leq n\}, F = \{C_i \mid 1 \leq i \leq m\})$ konstruieren wir in polynomieller Zeit ein VC-Problem $(G = (V, E), k = n+2m)$, so dass es eine erfüllende Belegung für F gibt gdw. dieses VC-Problem lösbar ist.

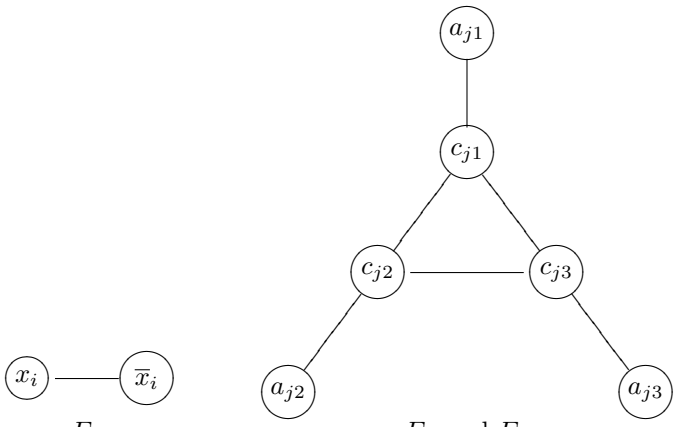
1. Knotenmenge

$$\begin{array}{rcl} V & = & \{x_i, \overline{x}_i \mid 1 \leq i \leq n\} \\ & \cup & \{c_{j1}, c_{j2}, c_{j3} \mid 1 \leq j \leq m\}. \end{array}$$

   Größe $|V| = 2n + 3m$.

2. Kantenmenge

$$\begin{array}{rcll} E & = & E_1 \cup E_2 \cup E_3 & \\ E_1 & = & \{\{x_i, \overline{x}_i\} \mid 1 \leq i \leq n\} & \\ E_2 & = & \{\{c_{j1}, c_{j2}\}, \{c_{j1}, c_{j3}\}, \{c_{j2}, c_{j3}\} \mid 1 \leq j \leq m\} & \text{Dreiecke} \\ E_3 & = & \{\{c_{ji}, a_{ji}\} \mid a_{ji} \in C_j\} & \end{array}$$



E_1 $\qquad\qquad$ E_2 und E_3

Eine Knotenüberdeckung U hat mindestens $n + 2m$ Knoten: jeweils einen aus $\{x_i, \overline{x}_i\}$ und jeweils 2 pro Dreieck.

"$\Longrightarrow$" Sei das 3SAT-Problem erfüllbar. Dann sei U_1 die Menge aller Knoten, die den erfüllten Literalen entsprechen. U_1 deckt E_1 ab. Da jede Klausel C_j erfüllt ist, also mindestens eines ihrer Literale, deckt U_1 ausserdem für jedes C_j mindestens eine Kante aus E_3 ab. Für jede Klausel brauchen wir noch 2 weitere Knoten, um die Dreieckskanten (E_2) und die restlichen zwei Kanten aus E_3 abzudecken. Nehmen wir diese Knoten zu U_1 hinzu, dann erhalten wir eine Knotenüberdeckung U mit $|U| = n + 2m$.

"$\Longleftarrow$" Sei U eine Überdeckung mit $n + 2m$ Knoten. Diese muss sicher n Knoten aufwenden, um E_1 zu überdecken. Nenne diese Knotenmenge U_1; sie kodiert genau eine Variablenbelegung. Von den verbleibenden $2m$ müssen je

2 auf ein Dreieck verwendet werden, um alle Dreiecke in E_2 zu überdecken. Nenne diese Knotenmenge U_2. Mit U_2 sind pro Klausel auch zwei Kanten aus E_3 überdeckt. Die jeweils letzte Kante muss schon durch U_1 überdeckt sein. Dies bedeutet, aber dass das zugehörige Literal erfüllt ist, damit auch diese Klausel. Insgesamt sind also alle Klauseln erfüllt.

□

3.3

Wie sieht die Welt aus, wenn $\mathsf{P} \neq \mathsf{NP}$?

Definition 7 (Symmetrische Differenz)

$$A \triangle B := (A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B)$$

Lemma 8 *Ist $L \triangle L'$ endlich, dann ist $L \in \mathsf{P}$ gdw. $L' \in \mathsf{P}$.*

Beweis. Zugehörigkeit zu einer endlichen Menge kann man in konstanter Zeit entscheiden. Unterscheiden sich L und L' in endlich Wörtern, so kann man diese Wörter in konstanter Zeit einer Sonderbehandlung unterziehen, die hat auf das asymptotische Verhalten keinen Einfluss. □

Satz 9 *Falls $\mathsf{P} \neq \mathsf{NP}$, dann gibt es eine Menge $L^* \in \mathsf{NP} \setminus \mathsf{P}$, aber $L^* \notin \mathsf{NPC}$.*

Beweis. Idee: Nehme ein NPC-Problem und "schlage Löcher" hinein. Dies muss so geschehen, dass die entstehende Sprache L^* nicht mehr in NPC, aber noch nicht in P ist.

Sei $L \in \mathsf{NPC}$. Definiere "gap"-Menge $G \subseteq \mathbb{N}$ mit $G \in \mathsf{P}$ und setze $L^* := \{x \in L \mid |x| \in G\}$. Sei $w_0 \notin L$ fest. Dann ist $L^* \leq_{\mathsf{P}} L$ vermöge der Reduktion

$$f(x) = \begin{cases} x & \text{wenn } |x| \in G \\ w_0 & \text{sonst.} \end{cases}$$

Sei $\{L_1, L_2, \dots\}$ eine Aufzählung[1] aller Sprachen in P und $\{L'_1, L'_2, \dots\}$ eine Aufzählung aller Sprachen in NPC. Wir definieren:

$$\begin{array}{lcl} g_i(n) & := & \text{kürzestes } w \text{ mit } |w| \geq n \text{ und } w \in L \triangle L_i \\ b_i(n) & := & \text{kürzestes } w \text{ mit } |w| \geq n \text{ und } w \in L'_i \\ r(n) & := & \max\{|g_i(n)|, |b_i(n)| \mid i \leq n\} + 1 \end{array}$$

Das w in der Definition von $g_i(n)$ existiert immer, weil sonst $L \triangle L_i$ endlich wäre, also nach dem Lemma $L \in \mathsf{P}$ im Widerspruch zu $\mathsf{P} \neq \mathsf{NP}$. Ebenso existiert das w in der Definition von $b_i(n)$, weil sonst L'_i endlich wäre, also in P im Widerspruch zu der Annahme.

[1] Diese Aufzählungen werden durch Aufzählung der entsprechenden Turing-Maschinen implementiert.

Die Funktion r hat die folgende Eigenschaft. Für jedes $i \leq n$ gibt es ein $z \in L \triangle L_i$ mit $n \leq |z| < r(n)$, und ein $z' \in L_i'$ mit $n \leq |z| < r(n)$.

Mit $r_0 := 0$ und $r_{i+1} := r(r_i)$ definieren wir

$$G := \{n \mid r_i \leq n < r_{i+1} \text{ für gerades } i\}.$$

Wir zeigen nun $L^* \notin \mathsf{P}$. Sei $L^* \in \mathsf{P}$, also $L^* = L_i$ für ein i. Es gibt sicher ein gerades n mit $r_n \geq i$. Nun existiert ein z mit $z \in L \triangle L_i$ und $n \leq |z| < r(n)$, also $|z| \in G$, was impliziert, dass $z \in L \iff z \in L^*$. Damit folgt $z \in L^* \triangle L_i = \emptyset$, Widerspruch!

Analog zeigen wir, dass $L^* \notin \mathsf{NPC}$. Nehmen wir an, dass $L^* \in \mathsf{NPC}$, dann ist $L^* = L_i$ für ein i. Es gibt sicher ein un gerades n mit $r_n \geq i$. Nun existiert ein z mit $z \in L_i'$ und $n \leq |z| < r(n)$, also $|z| \notin G$. Damit aber folgt $z \notin L^*$, Widerspruch!

Es bleibt noch zu zeigen, dass $G \in \mathsf{P}$.

□