13. Aufgabenblatt

(Besprechung in den Tutorien 29.01.2024–02.02.2024)

Aufgabe 1. Schnitt von NP und coNP (Schriftlicher Test WS 20/21)

Wir erinnern: Für eine Sprache $L \subseteq \Sigma^*$ ist $\bar{L} := \Sigma^* \setminus L$ das *Komplement* von L. Die Komplexitätsklasse coNP enthält alle Sprachen, deren Komplement in NP ist, also $\text{coNP} = \{L \subseteq \Sigma^* \mid \bar{L} \in \text{NP}\}$.

Das Problem 2-Coloring ist wie folgt definiert:

2-Coloring
Eingabe: Ein ungerichteter Graph $G = (V, E)$.
Frage: Lassen sich die Knoten von G mit zwei Farben so färben, dass keine zwei mit einer Kante verbundenen Knoten die gleiche Farbe haben?

Zeigen Sie, dass 2-Coloring in $\text{NP} \cap \text{coNP}$ liegt.

————Lösungsskizze————

Zwei mögliche Argumentationen:

1. Klar: 2-Coloring ist in P (siehe Aufgabenblatt 10). Also ist 2-Coloring laut VL auch in $\text{NP} \cap \text{coNP}$.

2. 2-Coloring hat sowohl "kurze" (polynomielle) Zertifikate (gültige Knotenfärbung) als auch kurze Gegenbeispiele (Kreis ungerader Länge in G). Denn G ist 2-färbbar gdw. G bipartit ist gdw. G keinen Kreis ungerader Länge enthält.

————————

Aufgabe 2. Verschiedenes zu P, NP und coNP

(a) Begründen oder widerlegen Sie die folgenden Behauptungen.

(i) Falls $\text{NP} \neq \text{coNP}$, dann gilt $\text{P} \neq \text{coNP}$.

————Lösungsskizze————

Die Aussage stimmt. Wir stellen zunächst fest, dass die Menge $\overline{\text{SAT}}$ der unerfüllbaren aussagenlogischen Formeln coNP-vollständig ist. Dies folgt sofort aus dem Satz von Cook und Levin, da jede Reduktion von A auf B auch eine Reduktion von $\overline{A}$ auf $\overline{B}$ darstellt. Wir zeigen im Folgenden die Kontraposition der ursprünglichen Aussage, also

$$\text{P} = \text{coNP} \implies \text{NP} = \text{coNP}.$$

Angenommen $\text{P} = \text{coNP}$. Dann ist $\overline{\text{SAT}} \in \text{coNP} = \text{P}$, das heißt $\overline{\text{SAT}}$ lässt sich in polynomieller Zeit entscheiden. Da eine aussagenlogische Formel genau dann unerfüllbar ist, wenn sie nicht erfüllbar ist, können wir aus einem Polynomzeitentscheider für $\overline{\text{SAT}}$ per Negation der Entscheidung auch einen Polynomzeitentscheider für SAT konstruieren. Da SAT NP-vollständig ist, folgt $\text{NP} = \text{P} = \text{coNP}$.

————————

(ii) Falls $A \leq_m^p B$ für zwei Sprachen A, B, dann gilt $A \in \text{P} \Leftrightarrow B \in \text{NP}$.

————Lösungsskizze————

Die Aussage ist falsch. Sei $A := \emptyset \subseteq \Sigma^*$ und sei $B := K$ das spezielle Halteproblem. Dann ist $A \in \text{P}$ aber $B \notin \text{NP}$, da B unentscheidbar ist. Die Abbildung, die jedes Wort über Σ auf eine feste niemals haltende Turing-Maschine abbildet, ist eine Polynomzeitreduktion von A auf B, denn

- sie kann als konstante Funktion in konstanter und somit polynomieller Zeit berechnet werden und
- jedes Wort ist eine „Nein"-Instanz für A und die konstruierte Turing-Maschine ist eine „Nein"-Instanz für B.

Also gilt wie gefordert $A \leq_m^p B$.

(iii) Jedes Problem in NP $\setminus$ P lässt sich auf jedes NP-vollständige Problem mittels einer Polynomzeitreduktion reduzieren.

——————Lösungsskizze——————

Die Aussage stimmt. Nach der Definition der NP-Schwere lässt sich jede Sprache in NP auf jedes NP-schwere Problem und somit auch auf jedes NP-vollständige Problem in polynomieller Zeit reduzieren.

(b) Das Tautologieproblem ist wie folgt definiert:

TAUT
Eingabe: Aussagenlogische Formel F.
Frage: Ist F eine Tautologie, d.h. wird F für alle $\{0,1\}$-wertigen Belegungen der in F verwendeten Booleschen Variablen zu wahr (d.h. 1) ausgewertet?

Wo liegt der Fehler im folgenden „Beweis", dass TAUT $\in$ NP?

„Wir zeigen, dass es eine nichtdeterministische Turing-Maschine gibt, die für eine gegebene aussagenlogische Formel F entscheidet, ob diese eine Tautologie ist. Dazu wird zunächst die Formel $\neg F$ erstellt und anschließend auf der Eingabe $\neg F$ eine NTM simuliert, die SAT in Polynomzeit löst, d.h., die entscheidet, ob $\neg F$ erfüllbar ist (da SAT in NP liegt, gibt es eine solche NTM). Die Formel F ist genau dann eine Tautologie, wenn $\neg F$ nicht erfüllbar ist."

——————Lösungsskizze——————

Sei M eine NTM, die SAT in Polynomzeit entscheidet. Die Idee im obigen „Beweis" besteht darin, eine NTM M' für TAUT zu konstruieren, die für eine gegebene Formel F zunächst deterministisch in polynomieller Zeit $\neg F$ generiert und dann M auf $\neg F$ simuliert. Die NTM M' soll F genau dann akzeptieren, wenn $\neg F$ *nicht* erfüllbar ist, also wenn M die Eingabe $\neg F$ ablehnt. Es ist aber nicht klar, wie eine solche (Komplement-)NTM M' konstruiert werden kann. Es muss gelten, dass M' auf Eingabe F genau dann einen akzeptierenden Berechnungspfad polynomieller Länge besitzt, wenn *alle* Berechnungspfade polynomieller Länge von M auf $\neg F$ ablehnen. Ein naheliegender Ansatz besteht darin, akzeptierende und ablehnende Berechnungen von M „zu vertauschen", also alle Übergänge von einem Nicht-Endzustand in einen Endzustand in M zu entfernen und stattdessen alle nichtdefinierten Übergänge aus einem Nicht-Endzustand nun in einen Endzustand zu definieren. Dieser Ansatz funktioniert nicht, wie folgende Überlegung zeigt: Sei F eine erfüllbare Formel, die aber keine Tautologie ist, also $F \in$ SAT und $\neg F \in$ SAT (und damit $F \notin$ TAUT). Da $\neg F \in$ SAT, existiert also ein akzeptierender Berechnungspfad polynomieller Länge von M auf $\neg F$. Nun kann M für $\neg F$ aber auch ablehnende Berechnungspfade polynomieller Länge besitzen (nach Definition von nichtdeterministischen Turing-Maschinen, kann dies bisher nicht ausgeschlossen werden). In diesem Fall besitzt M' dann eine akzeptierende Berechnung polynomieller Länge für F (nämlich die entsprechenden ablehnenden Berechnungen von M auf $\neg F$) und akzeptiert somit nach Definition fälschlicherweise die Eingabe F.

(c) Welches der folgenden Probleme ist NP-schwer, welches coNP-schwer?

(i) **Eingabe:** Eine aussagenlogische Formel F.
Frage: Ist $\neg F$ erfüllbar?

——————Lösungsskizze——————

Dieses Problem $L_{(\text{ci})}$ ist NP-schwer, denn es gilt SAT $\leq_m^p L_{\text{ci}}$ vermöge $f(\langle F\rangle) := \langle \neg F\rangle$, da $\neg\neg F \equiv F$.

(ii) **Eingabe:** Ein Graph G und eine positive ganze Zahl k.
Frage: Existiert nach dem Löschen von k beliebigen Knoten stets noch mindestens eine Kante im verbleibenden Graph?

——————Lösungsskizze——————

Dieses Problem $L_{(\text{cii})}$ ist coNP-schwer. Vertex Cover ist bekanntlich NP-vollständig und es gilt $L_{\text{cii}} = \overline{\text{Vertex Cover}}$. (Jede Knotenteilmenge V' ist genau dann ein Vertex Cover für G, wenn durch Löschen aller Knoten $v \in V'$ aus G sämtliche Kanten entfernt werden.)
