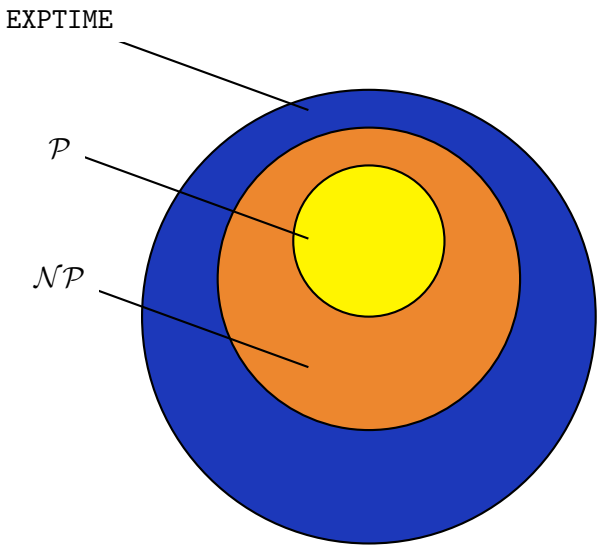
Vorlesungsmitschrift

Algorithmen und Berechenbarkeit

Vorlesung 19

Letztes Update: 2018/01/27 - 11:06 Uhr

$\mathcal{P}$ vs. $\mathcal{NP}$



Eine der bekanntesten, wichtigsten und offenen Fragen der theoretischen Informatik beschäftigt sich ebenfalls mit $\mathcal{P}$ und $\mathcal{NP}$:

$$\mathcal{NP} = \mathcal{P} \ ?$$

Wie der Grafik entnommen werden kann, gilt offensichtlich $\mathcal{P} \subseteq \mathcal{NP}$ und außerdem $\mathcal{NP} \subseteq$ EXPTIME.

Idee: Man zeigt äquivalente Schwere einiger Probleme in $\mathcal{NP}$ $\rightarrow$ „$\mathcal{NP}$-vollständige Probleme“. Dies hätte zur Folge, dass wenn man von einem dieser Probleme zeigen könnte

- $\in \mathcal{P}$, dann gilt $\Rightarrow \mathcal{NP} = \mathcal{P}$
- $\notin \mathcal{P}$, dann gilt $\Rightarrow \mathcal{NP} \neq \mathcal{P}$

Zunächst wird eine weitere Form der Reduktion eingeführt.

Definition polynomieller Reduktion: L_1 und L_2 seien zwei Sprachen über Σ_1 bzw. Σ_2. L_1 ist polynomiell reduzierbar auf L_2, wenn es ein

$$f : \Sigma_1^* \rightarrow \Sigma_2^*$$

gibt, das in polynomieller Zeit berechnet werden kann mit $x \in L_1 \Leftrightarrow f(x) \in L_2$.
Geschrieben $L_1 \leq_p L_2$.
Lemma: $L_1 \leq_p L_2$, $L_2 \in \mathcal{P} \Rightarrow L_1 \in \mathcal{P}$

Beweis: Klar.

Beispiel polynomieller Reduktion: Coloring $\leq_p$ SAT

Definition: Coloring

Gegeben sei $G(V, E)$ (ungerichtet) und $k \in \{1, \ldots, |V|\}$.

Frage: Gibt es eine Färbung

$$c : V \to \{1, \ldots, K\}$$

der Knoten in G mit k-Farben, sodass benachbarte Knoten nie dieselben Farben haben?

Definition: SAT (Satisfiability)

Gegeben sei eine aussagenlogische Formel ϕ in KNF.

Frage: Ist ϕ erfüllbar?

Satz: Coloring $\leq_p$ SAT.

Beweis: Man beschreibt eine Reduktionsfunktion $f(G, K) = \phi$, sodass gilt:

$$G \text{ hat } k\text{-Färbung} \Leftrightarrow \phi \text{ erfüllbar}$$

Für jeden Knoten $v \in V$ und jede Farbe $i \in \{1, \ldots, K\}$ führt man eine Variable x_v^i ein
($x_v^i =$`true` $\Rightarrow v$ bekommt die Farbe i).

$$\phi = \underbrace{\bigwedge_{v \in V} (x_v^1 \vee x_v^2 \vee \cdots \vee x_v^k)}_{\text{Knotenbedingung}} \quad \underbrace{\bigwedge_{\{u,v\} \in E} \bigwedge_{i \in \{1,\ldots,k\}} (\overline{x_u^i} \vee \overline{x_v^i})}_{\text{Kantenbedingung}}$$

ϕ hat die Größe $\mathcal{O}(k \cdot n + k \cdot n) \Rightarrow \mathcal{O}(n^3)$.

Nun muss noch gezeigt werden, dass die Reduktionsfunktion gültig ist.

- $\Rightarrow$ Sei c eine k-Farbe für G: Nun setzt man $x_v^i =$`true` für v mit $c(v) = i$, sonst $x_v^i =$`false`. Die Knotenbedingung ist offensichtlich erfüllt. Auch die Kantenbedingung ist erfüllt, da immer $\overline{x}_u^i \vee \overline{x}_v^i$ gilt, weil sonst v und u dieselbe Farbe haben müssten.
- $\Leftarrow$ Angenommen, man hat eine erfüllende Belegung für ϕ. Für jeden Knoten gibt es also mindestens ein $x_v^i =$`true`. Nun wählt man für jeden dadurch eine Farbe aus. Sei $\{u, v\} \in E$: Angenommen, $c(u) = c(v) = i$. Dann wäre jedoch $x_v^i = x_u^i =$`true` und $\overline{x}_v^i \vee \overline{x}_u^i =$`false`. Daraus folgt, ϕ ist nicht erfüllt, woraus gilt $c(u) \neq c(v)$. □

Korollar:

a) Wenn SAT in Polynomzeit deterministisch lösbar ist, ist Coloring auch in Polynomzeit lösbar.

b) Wenn SAT nicht in Polynomzeit deterministisch lösbar ist, ist Coloring auch nicht in Polynomzeit lösbar.

Definition $\mathcal{NP}$-hart/$\mathcal{NP}$-schwer: Ein Problem L heißt $\mathcal{NP}$-hart falls gilt:

$$\forall L' \in \mathcal{NP} \ : \ L' \leq_p L$$

Satz: L ist $\mathcal{NP}$-hart und $L \in \mathcal{P}$, dann gilt $\mathcal{P} = \mathcal{NP}$.

Definition $\mathcal{NP}$-vollständig: Ein Problem heißt $\mathcal{NP}$ vollständig, falls

1. $L \in \mathcal{NP}$ (meistens einfacher zu zeigen)
2. L ist $\mathcal{NP} -$ hart (meistens schwieriger zu zeigen)

NPC ist die Klasse der $\mathcal{NP}$-vollständigen Probleme.

Satz: SAT ist $\mathcal{NP}$-vollständig.

Beweisidee:

a) SAT $\in \mathcal{NP}$ ist einfach.

b) $\forall L' \in \mathcal{NP} \ : \ L' \leq_p L$
$L' \in \mathcal{NP}$ heißt, es gibt eine NTM $\mathcal{M}$, die L' in polynomieller Zeit erkennt. Nun kodiert man das Verhalten von $\mathcal{M}$ in eine aussagenlogische Formel, die genau dann erfüllbar ist, falls $\mathcal{M}$ die Eingabe in polynomieller Zeit akzeptiert *(mehr Informationen im Skript)*.

Lemma: SAT $\leq_p$ 3SAT

Satz: 3SAT ist $\mathcal{NP}$-vollständig.