
Einführung in die Theoretische Informatik

Hausaufgabe 1 (4 Punkte)

Beweisen oder widerlegen Sie:

1. Jedes Problem ist entweder in P oder in NP .
2. Wenn A NP-vollständig ist, dann ist χ_A μ -rekursiv.
3. Das Halteproblem auf leerem Band H_0 ist NP-hart.

Lösungsvorschlag

1. Falsch! unentscheidbare Mengen liegen weder in P noch in NP .
2. Wahr! $A \in NP \implies A$ entscheidbar $\implies \chi_A$ berechenbar $\implies \chi_A$ μ -rekursiv
3. Wahr! Es gibt eine Turingmaschine M , die das SAT Problem entscheidet und insbesondere stets hält. Falls M in einem nichtakzeptierenden Zustand hält, dann kann man eine nicht haltende Turingmaschine starten. Somit wird SAT polynomiell auf H_0 reduziert.

Hausaufgabe 2 (4 Punkte)

Seien $d \in \Sigma = \{0, 1\}^*$ die Darstellungen natürlicher Zahlen und $POT = \{m\#p\#n \in (\Sigma \cup \{\#\})^* \mid m^p = n\}$, wobei m^p die Potenzierung natürlicher Zahlen bedeute. Zeigen Sie: Wenn POT NP-vollständig ist, dann gilt $P = NP$.

Lösungsvorschlag

Das Problem POT beschreibt ja nur die Potenzierung von Zahlen, was selbstverständlich in Polynomialzeit lösbar ist. Wäre POT NP-vollständig, dann wäre nach Lemma 5.19 $P = NP$.

Hausaufgabe 3 (6 Punkte)

Das Mengenzerteilungsproblem (MZ) ist wie folgt definiert:

Gegeben: Eine endliche Menge U , und Teilmengen $S_1, \dots, S_k \subseteq U$.

Problem: Kann man die Elemente von U so einfärben (jeweils in Rot oder Blau), dass jedes S_i mindestens ein rotes und ein blaues Element enthält?

Formal: Gibt es eine Teilmenge $R \subseteq U$, so dass für alle $i \in \{1, \dots, k\}$ gilt:

$$\emptyset \subsetneq S_i \cap R \subsetneq S_i$$

Intuition: R enthält genau die roten Elemente.

1. Geben Sie einen deterministischen Algorithmus an, der das Mengenzerteilungsproblem für $k = 2$ in Polynomialzeit löst.
2. Zeigen Sie, dass $MZ \leq_p \text{SAT}$. Erläutern Sie Ihre Konstruktion auch informell. Hinweis: Verwenden Sie die Variablenmenge $\{r_x \mid x \in U\}$.
3. Seien nun $A, B \subseteq \Sigma^*$ beliebige Probleme.
Zeigen Sie: Wenn $\emptyset \subsetneq A \subsetneq \Sigma^*$ und $B \in \text{P}$, dann gilt $B \leq_p A$.

Lösungsvorschlag

1. Wenn es nur zwei Mengen gibt, hat eine Probleminstanz genau dann eine Lösung, wenn S_1 und S_2 mindestens zwei Elemente haben. Denn dann gibt es die Fälle:
 - (a) $S_1 \subseteq S_2$. Dann legt man für die Elemente von S_1 eine geeignete Färbung fest, die dann auch zu S_2 passt.
 - (b) $S_2 \subseteq S_1$. Analog
 - (c) Andernfalls gibt es ein $x \in S_1 \setminus S_2$ und ein $y \in S_2 \setminus S_1$. Dann kann man x und y rot färben und alle anderen Elemente blau.

Man muss also nur die Größe der Mengen anschauen.

2. Wir wählen die Variablenmenge $\{r_x \mid x \in U\}$ und konstruieren die Formel

$$F = \bigwedge_{1 \leq i \leq k} \left(\left(\neg \bigwedge_{x \in S_i} r_x \right) \wedge \bigvee_{x \in S_i} r_x \right)$$

Das ist offensichtlich in Polynomialzeit möglich, und die Belegungen kodieren genau die möglichen Lösungen des Problems.

3. Die charakteristische Funktion $\chi_B : \Sigma^* \rightarrow \mathbb{N}$ ist in polynomieller Zeit durch eine DTM M berechenbar.

Da $\emptyset \subsetneq A \subsetneq \Sigma^*$, gibt es ein $x_1 \in A$ und ein $x_2 \notin A$.

Wir definieren $f : \Sigma^* \rightarrow \Sigma^*$ durch

$$f(w) := \begin{cases} x_1 & \text{falls } \chi_B(w) = 1, \\ x_2 & \text{falls } \chi_B(w) = 0. \end{cases}$$

f ist total und in polynomieller Zeit berechenbar, und es gilt:
 $w \in B \Leftrightarrow f(w) \in A$.

Hausaufgabe 4 (6 Punkte)

Wir betrachten einen PC-Konfigurator, der PCs aus verschiedenen Komponenten zusammenstellt. Dafür sei A eine Vereinigung von paarweise disjunkten, endlichen Mengen A_1, \dots, A_n von möglichen Komponenten. Intuitiv enthalten die A_i Komponenten gleichen Typs (z.B. "Prozessoren", "Grafikkarten", "Festplatten").

Eine *Konfiguration* ist eine Menge $K \subseteq A$, die aus jedem A_i genau ein Element enthält (damit ist $|K| = n$). Einige Komponentenkombinationen sind aber inkompatibel. Das beschreibt man durch Inkompatibilitätsmengen $I_1, \dots, I_k \subseteq A$. Eine Konfiguration K heißt *lauffähig*, wenn es kein $j \in \{1 \dots k\}$ gibt mit $I_j \subseteq K$.

1. Wir betrachten folgendes Problem:

KONFIGURATOR:

Gegeben: Konstanten n und k , endliche Mengen A, A_1, \dots, A_n und $I_1, \dots, I_k \subseteq A$, so dass $A = A_1 \cup \dots \cup A_n$ und $A_i \cap A_j = \emptyset$ für $i \neq j$.

Problem: Gibt es eine lauffähige Konfiguration $K \subseteq A$?

Geben Sie eine polynomielle Reduktion von KONFIGURATOR auf SAT an.

2. Das folgende modifizierte Problem beachtet auch Kundenwünsche:

KONFIGURATOR+ (Änderungen unterstrichen):

Gegeben: Konstanten n und k , endliche Mengen A, A_1, \dots, A_n und $I_1, \dots, I_k \subseteq A$ und $W \subseteq A$, so dass $A = A_1 \cup \dots \cup A_n$ und $A_i \cap A_j = \emptyset$ für $i \neq j$.

Problem: Gibt es eine lauffähige Konfiguration $K \subseteq A$ mit $W \subseteq K$?

Intuitiv bezeichnet die Wunschmenge W dabei die Menge der Komponenten, die der Kunde auf jeden Fall in seinem System haben möchte.

Zeigen Sie:

- (a) KONFIGURATOR \leq_p KONFIGURATOR+
- (b) KONFIGURATOR+ \leq_p KONFIGURATOR

Hinweis: Beachten Sie, dass eine Wunschmenge W unerfüllbar sein kann, wenn z.B. $|W \cap A_i| > 1$ für ein $i \in \{1 \dots n\}$.

Lösungsvorschlag

1. Wir verwenden die Variablen v^a , die besagt ob $a \in A$ in der Konfiguration vorkommt. Es wird mindestens eine Komponente ausgewählt:

$$\bigwedge_{1 \leq i \leq n} \bigvee_{a \in A_i} v^a$$

Es wird maximal eine Komponente ausgewählt:

$$\bigwedge_{1 \leq i \leq n} \bigwedge_{\substack{a, a' \in A_i \\ a \neq a'}} \neg v^a \vee \neg v^{a'}$$

Es werden keine inkompatiblen Komponenten ausgewählt:

$$\bigwedge_{1 \leq j \leq k} \bigvee_{a \in I_j} \neg v^a$$

2. (a) Mit leerer Wunschmenge.
- (b) Die Wünsche $w \in W$ werden durch neue, einelementige Inkompatibilitätsmengen $J_{i,w'} = \{w'\}$ für alle $w, w' \in A_i$, $w' \neq w$, $i \in [n]$ modelliert.