

---

## Einführung in die Theoretische Informatik

---

Abgabetermin: 26. Juli 2010 vor 12.00 Uhr in die THEO Briefkästen

### 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  $\chi_A$   $\mu$ -rekursiv.
3. Das Halteproblem auf leerem Band  $H_0$  ist NP-hart.

### 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  $P = NP$ .

### 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  $MZ \leq_p SAT$ . 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  $B \in P$ , dann gilt  $B \leq_p A$ .

## 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  $i \in \{1 \dots n\}$ .