# Theoretische Grundlagen der Informatik

Tutorium 7

Moritz Klammler

Wintersemester 2014/15

9. Dezember 2014

# Anmerkungen zum 3. Übungsblatt

- Something is wrong, perhaps a missing `\item` ...

# Tagesthemen

- Wiederholung Problemarten
- Wiederholung Kodierungsschemata
- $\mathcal{NP}$-Vollständigkeit (Reduktionen) üben
- Die Komplementklassen co-$\mathcal{P}$ und co-$\mathcal{NP}$
- Turing-Reduzierbarkeit

Wiederholung

# Probleme

- Ein **Suchproblem** ist eine Frage nach einer Lösung.
- Ein **Optimierungsproblem** ist eine Frage nach einer (in einem gewissen Sinne) optimalen Lösung.
- Ein **Optimalwertproblem** ist eine Frage nach dem Wert einer (in einem gewissen Sinne) optimalen Lösung.
- Ein **Entscheidungsproblem** ist eine binäre Frage.

Wiederholung

# Kodierungsschema

Sei $\Pi$ ein Problem und $\Sigma$ ein Alphabet. Ein **Kodierungsschema** ist eine Abbildung

$$\begin{array}{rccc} s: & \Pi & \to & \Sigma^* \\ & I & \mapsto & \langle I \rangle \end{array},$$

die jeder Instanz $I$ des Problems eine Kodierung $\langle I \rangle$ zuordnet.

Die *Größe* einer Probleminstanz $I$ im Kodierungsschema $s$ ist die Länge der Codierung $|s(I)|$.

Die **Entscheidungssprache** $L[\Pi, s]$ enthält alle in $s$ kodierten Ja-Instanzen von $\Pi$.

Wiederholung

# $\mathcal{NP}$-Vollständigkeit

Eine Sprache $L$ ist **$\mathcal{NP}$-vollständig** genau dann wenn

- $L \in \mathcal{NP}$
- $\forall L' \in \mathcal{NP} : L' \leq_{\mathrm{p}} L$

Die Menge aller $\mathcal{NP}$-vollständigen Sprachen heißt $\mathcal{NPC}$.

Wichtig: SAT $\in \mathcal{NPC}$ und $\leq_{\mathrm{p}}$ ist transitiv.

Wiederholung

# Polynomielle Transformation

Sei $\Sigma$ ein Alphabet und $B \subseteq \Sigma^*$ **in Polynomialzeit entscheidbar**, sowie $A \subseteq \Sigma^*$.

Wenn es eine **in Polynomialzeit berechenbare** Funktion $f : \Sigma^* \to_{\mathrm{p}} \Sigma^*$ gibt, sodass für alle $w \in \Sigma^*$ gilt

$$w \in A \quad \Leftrightarrow \quad f(w) \in B$$

dann ist auch $A$ in Polynomialzeit entscheidbar.

Man schreibt diesfalls $A \leq_{\mathrm{p}} B$ oder $A \propto B$.

## Partition

Gegeben $M = \{m_1, \ldots, m_n\}$ für $n \in \mathbb{N}$ und $\omega : M \to \mathbb{N}$, existieren $\Pi_1, \Pi_2 \subset M$, sodass

$$\Pi_1 \cup \Pi_2 = M \ ,$$

$$\Pi_1 \cap \Pi_2 = \emptyset \text{ und}$$

$$\sum_{m \in \Pi_1} \omega(m) = \sum_{m \in \Pi_2} \omega(m) \ ?$$

## BinPacking

Gegeben $(a_1, \ldots, a_n) \in \mathbb{N}^n$ für $n \in \mathbb{N}$ und $b, k \in \mathbb{N}$, existiert $\pi : \{1, \ldots, n\} \to \{1, \ldots, k\}$ sodass

$$\forall i \in \{1, \ldots, k\} : \sum_{j=1}^{n} \delta_{i,\pi(j)} a_j \leq b \ ?$$

Zu zeigen:

$$\textsc{Partition} \in \mathcal{NPC} \Rightarrow \textsc{BinPacking} \in \mathcal{NPC}$$

## HamiltonCycle

Gegeben Graphen $G = (V, E)$ mit $|V| = n$, existiert eine Permutation $v_1, \ldots, v_n \equiv v_0$ von $V$ sodass

$$\forall i \in \{0, \ldots, n\} : \{v_i, v_{i+1}\} \in E \ ?$$

## TSP

Gegeben vollständigen Graphen $G = (V, E)$ mit $|V| = n$ und Kantengewichtungsfunktion $\omega : E \to \mathbb{N}$ und $k \in \mathbb{N}$, existiert eine Permutation $v_1, \ldots, v_n \equiv v_0$ von $V$ sodass

$$\sum_{i=0}^{n} \omega(\{v_i, v_{i+1}\}) \leq k \ ?$$

Zu zeigen:

$$\text{HamiltonCycle} \in \mathcal{NPC} \Rightarrow \text{TSP} \in \mathcal{NPC}$$

# Die Komplementklassen co-$\mathcal{P}$ und co-$\mathcal{NP}$

Die Klasse co-$\mathcal{P}$ enthält alle Sprachen $L^c$ für die $L \in \mathcal{P}$.

Die Klasse co-$\mathcal{NP}$ enthält alle Sprachen $L^c$ für die $L \in \mathcal{NP}$.

- Es gilt $\mathcal{P} =$ co-$\mathcal{P}$ und damit $\mathcal{P} \subseteq \mathcal{NP} \cap$ co-$\mathcal{NP}$.
- Es ist unklar, ob $\mathcal{NP} =$ co-$\mathcal{NP}$ gilt.
- Aus $\mathcal{P} = \mathcal{NP}$ folgt (trivialerweise) auch $\mathcal{NP} =$ co-$\mathcal{NP}$.

# Turing-Reduzierbarkeit

Sei $\Sigma$ ein Alphabet, $L \subset \Sigma^*$ und $O$ ein magisches Gerät (*Orakel*), das für jedes Wort $w \in \Sigma^*$ in konstanter Zeit entscheiden kann, ob $w \in L$. Eine Turingmaschine, deren Zustandsübergangsfunktion mitilfe von $O$ berechnet wird, heißt **Orakel-Turingmaschine**.

Seien $A$ und $B$ Sprachen. $A$ ist **Turing-reduzierbar** auf $B$ genau dann wenn $A$ von einer Orakel-Turingmaschine entschieden werden kann, die ein Orakel für $B$ benutzt.

Man schreibt diesfalls $A \leq_\mathrm{t} B$.

Es ist unklar, ob $\{l \subseteq \Sigma^* : L[\mathrm{SAT}, s] \leq_\mathrm{t} l\} = \mathcal{NPC}$ (für ein sinnvolles Kodierungsschema $s$) gilt.