Theoretische Grundlagen der Informatik

Tutorium 7 — Wintersemester 2014/15

Moritz Klammler — 16. Dezember 2014

Tagesthemen

- Turing-Reduzierbarkeit
- Pseudopolynomielle Algorithmen
- Approximationsalgorithmen

Turing-Reduzierbarkeit

Sei Σ 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 (polynomiell) **Turing-reduzierbar** auf B genau dann wenn A (in polynomiell vielen Schritten) von einer Orakel-Turingmaschine entschieden werden kann, die ein Orakel für B benutzt.

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

$\mathcal{NP}$-Härte

Eine Sprache L ist **$\mathcal{NP}$-hart**, wenn $L' \leq_\mathrm{t} L$ (polynomiell) für alle $L' \in \mathcal{NP}$.

- Jede $\mathcal{NP}$-vollständige Sprache ist $\mathcal{NP}$-hart.
- Es ist unklar, ob die Menge der $\mathcal{NP}$-harten gleich der Menge der $\mathcal{NP}$-vollständigen Sprachen ist.

Pseudopolynomielle Algorithmen

Vorlesung

- Kodiert man vorkommende Zahlen nicht binär sondern unär, gehen diese nicht logarithmisch, sondern linear in die Inputlänge ein.
- Es gibt $\mathcal{NP}$-vollständige Probleme, die für solche Kodierungen polynomiale Algorithmen besitzen.
- Solche Algorithmen nennt man **pseudopolynomielle Algorithmen**.

Sei Π ein Optimierungsproblem. Ein Algorithmus, der Problem Π löst, heißt pseudopolynomiell, falls seine Laufzeit durch ein Polynom der beiden Variablen *Eingabegröße* und *Größe der größten in der Eingabe vorkommenden Zahl* beschränkt ist.

Starke $\mathcal{NP}$-Vollständigkeit

Informell: Falls die Entscheidungssprache trotz unärer Kodierung $\mathcal{NP}$-vollständig bleibt, heißt sie **stark $\mathcal{NP}$-vollständig**.

Falls $\mathcal{P} \neq \mathcal{NP}$ so gibt es für die zu den stark $\mathcal{NP}$-vollständigen Entscheidungssprachen gehörenden Entscheidungsprobleme keine pseudopolynomiellen Algorithmen.

Approximationsalgorithmen

- Für ein Optimierungsproblem ist die optimale Lösung möglicherweise schwer zu finden.
- Vielleicht ist eine **korrekte aber suboptimale Lösung** wesentlich einfacher zu finden.
- Wie viel schlechter ist die gefundene Lösung (schlimmstenfalls) im Vergleich zur optimalen Lösung?
- Uns interessieren vor allem Approximationsalgorithmen für $\mathcal{NP}$-vollständig Probleme mit polynomieller Laufzeit.

Absolute Gütegarantie

Ein Approximationsalgorithmus $\mathcal{A}$ für ein Problem Π hat eine **absolute Gütegarantie**, genau dann wenn

$$\exists k \in \mathbb{N}_0 : \forall I \in \Pi : |\mathcal{A}(I) - \mathrm{Opt}(I)| \leq k \quad .$$

Relative Güte

Sei Π ein Problem uns $\mathcal{A}$ ein Approximationsalgorithmus für Π. Für eine Instanz I von Π bezeichnet

$$R_{\mathcal{A}}(I) = \max \left\{ \underbrace{\frac{\mathcal{A}(I)}{\mathrm{Opt}(I)}}_{\text{Minimierung}} , \underbrace{\frac{\mathrm{Opt}(I)}{\mathcal{A}(I)}}_{\text{Maximierung}} \right\} \geq 1$$

die **relative Güte** der von $\mathcal{A}$ berechneten Lösung.

Relative Gütegarantie

Ein Approximationsalgorithmus $\mathcal{A}$ für ein Problem Π heißt **ϵ-approximativ** für $\epsilon \geq 0$, genau dann wenn

$$\forall I \in \Pi : R_{\mathcal{A}}(I) \leq 1 + \epsilon \quad .$$

Vorsicht: Alternative Definitionen!

Relative asymptotische Güte

Sei Π ein Problem und $\mathcal{A}$ ein Approximationsalgorithmus für Π.

Die **asymptotischische relative Güte** von $\mathcal{A}$ ist definiert als

$$R_{\mathcal{A}}^{\infty} = \inf\{r \in \mathbb{R} : \exists n \in \mathbb{N} : \forall I \in \Pi : \mathrm{Opt}(I) \geq n \Rightarrow R_{\mathcal{A}}(I) \leq r\} \geq 1 \quad .$$