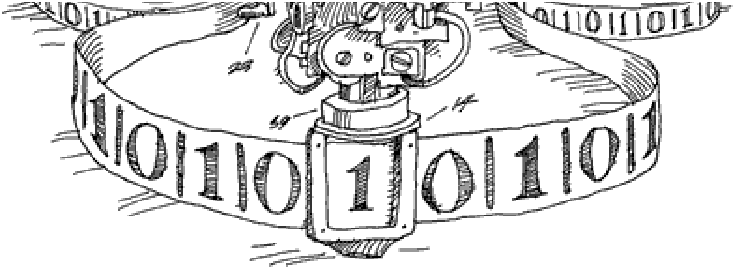
Karlsruher Institut für Technologie

Tutorium Theoretische Grundlagen der Informatik

Simon Bischof

Institut für Kryptographie und Sicherheit



Zum Übungsblatt

- Beim PKP ist eine leere Puzzlestückfolge KEINE gültige Lösung
- Turingreduktion noch mal anschauen!
- Wenn ihr Entscheidbarkeit bewiesen habt, folgt daraus automatisch Semientscheidbarkeit
- $K(x)$: immer hinschreiben, dass c konstante Größe der TM ist
- $\text{Th}(\mathbb{N}, +)$: eigentlich nur Gleichungen der Form $x + y = z$ erlaubt

$\mathcal{NP}$-Vollständigkeit

polynomielle many-one-Reduktion

- Sei f in polynomieller Zeit berechenbare Funktion und A und B Sprachen
- Sei $\forall w \in \Sigma^* : w \in A \Leftrightarrow f(w) \in B$
- Dann ist A polynomiell many-one-reduzierbar auf B ($A \leq_p B$)

polynomielle many-one-Reduktion

- Sei f in polynomieller Zeit berechenbare Funktion und A und B Sprachen
- Sei $\forall w \in \Sigma^* : w \in A \Leftrightarrow f(w) \in B$
- Dann ist A polynomiell many-one-reduzierbar auf B ($A \leq_p B$)
- $A \leq_p B$ und $B \in \mathcal{P} \Rightarrow A \in \mathcal{P}$
- poly many-one-Reduzierbarkeit ist transitiv

polynomielle many-one-Reduktion

- Sei f in polynomieller Zeit berechenbare Funktion und A und B Sprachen
- Sei $\forall w \in \Sigma^* : w \in A \Leftrightarrow f(w) \in B$
- Dann ist A polynomiell many-one-reduzierbar auf B ($A \leq_p B$)
- $A \leq_p B$ und $B \in \mathcal{P} \Rightarrow A \in \mathcal{P}$
- poly many-one-Reduzierbarkeit ist transitiv
- poly many-one-reduzierbar ist nicht dasselbe wie poly turingreduzierbar

$\mathcal{NP}$-Schwere und -Vollständigkeit

- A ist $\mathcal{NP}$-schwer, falls $\forall B \in \mathcal{NP} : B \leq_p A$

$\mathcal{NP}$-Schwere und -Vollständigkeit

- A ist $\mathcal{NP}$-schwer, falls $\forall B \in \mathcal{NP} : B \leq_p A$
- A ist NP-vollständig ($A \in \mathcal{NP} - \mathcal{C}$), falls $A \in \mathcal{NP}$ und A $\mathcal{NP}$-schwer ist

$\mathcal{NP}$-Schwere und -Vollständigkeit

- A ist $\mathcal{NP}$-schwer, falls $\forall B \in \mathcal{NP} : B \leq_p A$
- A ist *NP*-vollständig ($A \in \mathcal{NP} - \mathcal{C}$), falls $A \in \mathcal{NP}$ und A $\mathcal{NP}$-schwer ist
- $B \in \mathcal{NP} - \mathcal{C}$, $A \in \mathcal{NP}$, $B \leq_p A \Rightarrow A \in \mathcal{NP} - \mathcal{C}$

$\mathcal{NP}$-Schwere und -Vollständigkeit

Karlsruher Institut für Technologie

- A ist $\mathcal{NP}$-schwer, falls $\forall B \in \mathcal{NP} : B \leq_p A$
- A ist *NP*-vollständig ($A \in \mathcal{NP} - \mathcal{C}$), falls $A \in \mathcal{NP}$ und A $\mathcal{NP}$-schwer ist
- $B \in \mathcal{NP} - \mathcal{C}$, $A \in \mathcal{NP}$, $B \leq_p A \Rightarrow A \in \mathcal{NP} - \mathcal{C}$
- falls $\mathcal{P} \cap \mathcal{NP} - \mathcal{C} \neq \emptyset$, ist $\mathcal{P} = \mathcal{NP}$

SAT (Erfüllbarkeit)

- SAT $:= \{$boolsche Formel $b|$
  es existiert eine Variablenbelegung, so dass b wahr wird$\}$ (Version 1)
- meist eingeschränkt auf konjunktive Form: z.B.
  $$b = (\underbrace{x_1}_{\text{Literal}} \vee x_3 \vee \underbrace{\bar{x}_4}_{\text{Literal}} \vee x_5) \wedge \underbrace{(\bar{x}_1 \vee x_2)}_{\text{Klausel}} \wedge (x_4)$$
  $x_1, \ldots, x_n$ nennt man Variablen

SAT (Erfüllbarkeit)

- SAT $:= \{$boolsche Formel $b|$
  es existiert eine Variablenbelegung, so dass b wahr wird$\}$ (Version 1)
- meist eingeschränkt auf konjunktive Form: z.B.
  $b = (\underbrace{x_1}_{\text{Literal}} \vee x_3 \vee \underbrace{\bar{x}_4}_{\text{Literal}} \vee x_5) \wedge \underbrace{(\bar{x}_1 \vee x_2)}_{\text{Klausel}} \wedge (x_4)$
  $x_1, \ldots, x_n$ nennt man Variablen
- Satz von Cook: $SAT \in \mathcal{NP} - \mathcal{C}$

SAT (Erfüllbarkeit)

- SAT $:= \{$boolsche Formel $b|$
  es existiert eine Variablenbelegung, so dass b wahr wird$\}$ (Version 1)
- meist eingeschränkt auf konjunktive Form: z.B.
  $b = (\underbrace{x_1}_{\text{Literal}} \vee x_3 \vee \underbrace{\bar{x}_4}_{\text{Literal}} \vee x_5) \wedge \underbrace{(\bar{x}_1 \vee x_2)}_{\text{Klausel}} \wedge (x_4)$
  $x_1, \ldots, x_n$ nennt man Variablen
- Satz von Cook: $SAT \in \mathcal{NP} - \mathcal{C}$
- Erfüllbarkeit von disjunktiven Formen ist aber in $\mathcal{P}$

weitere wichtige $\mathcal{NP}$-vollständigen Probleme

- siehe Tutblatt
- siehe Übungsblätter
- siehe Vorlesungsfolien und Skript von Wagner (http://i11www.iti.uni-karlsruhe.de/teaching/winter2011/tgi/index) [für diese VL natürlich inoffiziell...]

Zero-Knowledge-Beweise

Ziel von ZK-Beweisen

Karlsruher Institut für Technologie

Einen anderen überzeugen, eine Lösung zu kennen, ohne

- diese zu verraten
- dass andere von einer Mitschrift des ”Gesagten” überzeugt werden

Wie kann so etwas gehen?

- mithilfe von Zufall: falsche Lösungen werden ”ziemlich sicher” erkannt

Wie kann so etwas gehen?

- mithilfe von Zufall: falsche Lösungen werden ”ziemlich sicher” erkannt
- verwende eine Art ”Zeuge”, der aber nur teilweise abgefragt wird

Wie kann so etwas gehen?

- mithilfe von Zufall: falsche Lösungen werden ”ziemlich sicher” erkannt
- verwende eine Art ”Zeuge”, der aber nur teilweise abgefragt wird
- zuerst wählt der Beweiser den ”Zeugen”, dann sagt der Prüfer, welchen Teil er wissen will

Wie kann so etwas gehen?

- mithilfe von Zufall: falsche Lösungen werden ”ziemlich sicher” erkannt
- verwende eine Art ”Zeuge”, der aber nur teilweise abgefragt wird
- zuerst wählt der Beweiser den ”Zeugen”, dann sagt der Prüfer, welchen Teil er wissen will
- Wiederholung des obigen Schritts bis die Fehlerwahrscheinlichkeit ”gering genug”

Wie kann so etwas gehen?

- mithilfe von Zufall: falsche Lösungen werden ”ziemlich sicher” erkannt
- verwende eine Art ”Zeuge”, der aber nur teilweise abgefragt wird
- zuerst wählt der Beweiser den ”Zeugen”, dann sagt der Prüfer, welchen Teil er wissen will
- Wiederholung des obigen Schritts bis die Fehlerwahrscheinlichkeit ”gering genug”
- wichtig dabei: falls der Beweiser wüsste, welchen Teil der Prüfer sehen will, würde er den Zeugen entsprechend ”fälschen” können (sieht daher für außenstehende wie Absprache aus)

ZK-Beweise und $\mathcal{NP}$

- 3-COLOR $\in \mathcal{NP} - \mathcal{C}$
- 3-COLOR lässt sich ZK-beweisen
- alle Probleme aus $\mathcal{NP}$ lassen sich ZK-beweisen