Info III Tutorium

Letzte Sitzung

Thomas Pajor

Fakultät für Informatik

ITI Sanders

13. Februar 2007

Nicht behandelte Themen

Es gibt ein paar Themen, die ich im Tutorium nicht gemacht habe:

- Kapitel „Graphenalgorithmen“
- WHILE–Programme, Quantencomputing, ...
  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Ackermann Funktion
  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Approximationsalgorithmen für $\mathcal{NP}$–vollständige Probleme
  $\rightsquigarrow$ Kapitel „Komplexistätstheorie“

Nicht vergessen beim Lernen!

Nicht behandelte Themen

Es gibt ein paar Themen, die ich im Tutorium nicht gemacht habe:

- Kapitel „Graphenalgorithmen“
- While–Programme, Quantencomputing, ...
  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Ackermann Funktion
  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Approximationsalgorithmen für $\mathcal{NP}$–vollständige Probleme
  $\rightsquigarrow$ Kapitel „Komplexistätstheorie“

Nicht vergessen beim Lernen!

Nicht behandelte Themen

Es gibt ein paar Themen, die ich im Tutorium nicht gemacht habe:

- Kapitel „Graphenalgorithmen“
- WHILE–Programme, Quantencomputing, ...
  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Ackermann Funktion
  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Approximationsalgorithmen für $\mathcal{NP}$–vollständige Probleme
  $\rightsquigarrow$ Kapitel „Komplexistätstheorie“

Nicht vergessen beim Lernen!

Nicht behandelte Themen

Es gibt ein paar Themen, die ich im Tutorium nicht gemacht habe:

- Kapitel „Graphenalgorithmen“
- WHILE–Programme, Quantencomputing, ...

  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Ackermann Funktion

  $\rightsquigarrow$ Kapitel „Berechenbarkeit“
- Approximationsalgorithmen für $\mathcal{NP}$–vollständige Probleme

  $\rightsquigarrow$ Kapitel „Komplexistätstheorie“

Nicht vergessen beim Lernen!

Nicht behandelte Themen

Es gibt ein paar Themen, die ich im Tutorium nicht gemacht habe:

- Kapitel „Graphenalgorithmen“
- WHILE–Programme, Quantencomputing, ...

  $\leadsto$ Kapitel „Berechenbarkeit“
- Ackermann Funktion

  $\leadsto$ Kapitel „Berechenbarkeit“
- Approximationsalgorithmen für $\mathcal{NP}$–vollständige Probleme

  $\leadsto$ Kapitel „Komplexistätstheorie“

Nicht vergessen beim Lernen!

Klausur

Ein paar Angaben zur Klausur:

- 26. Februar 9:00 Uhr
- Anmeldung ist bis zum 21. Februar in Raum 218 (Sekretariat von Prof. Sanders)
- Hilfsmittel: 1 DIN/A4 Blatt mit beliebigem Inhalt (Doppelseitig)

Klausur

Ein paar Angaben zur Klausur:

- 26. Februar 9:00 Uhr
- Anmeldung ist bis zum 21. Februar in Raum 218 (Sekretariat von Prof. Sanders)
- Hilfsmittel: 1 DIN/A4 Blatt mit beliebigem Inhalt (Doppelseitig)

Klausur

Ein paar Angaben zur Klausur:

- 26. Februar 9:00 Uhr
- Anmeldung ist bis zum 21. Februar in Raum 218 (Sekretariat von Prof. Sanders)
- Hilfsmittel: 1 DIN/A4 Blatt mit beliebigem Inhalt (Doppelseitig)

Wiederholung: Chomsky–Hierarchie & Maschinenmodelle

Typ	Maschine	Beispiel
3 (regulär)	NEAs und DEAs Es gilt NEA $\equiv$ DEA	$\{a^n \mid n \geq 0\}$
„2,5“ (det.–kontextfrei)	DKAs	$\{a^n b^n \mid n \geq 0\}$
2 (kontextfrei)	NKAs (Ein Zustand reicht aus)	$\{ww^R \mid w \in \Sigma^*\}$
1 (kontextsensitiv)	Linear platzbeschr. TMs	$\{a^n b^n c^n \mid n \geq 0\}$
0 (rek.aufz. Sprachen)	DTMs und NTMs DTM kann NTM in exp. Zeit simulieren!	Halteproblem
Der Rest	-	Diagonalsprache

Wiederholung: Pumping Lemma

Satz

Das Pumping Lemma liefert folgende Implikation:

$$
\begin{aligned}
L \text{ regulär} \quad \Rightarrow \quad & \exists n \in \mathbb{N} : \\
& \forall w \in L \text{ mit } |w| > n : \\
& \exists \text{ Zerlegung } w = uvx \text{ mit } |uv| \leq n \text{ und } |v| > 0 : \\
& \forall i \in \mathbb{N}_0 : uv^i x \in L
\end{aligned}
$$

Tipp: Benutze die *Kontraposition* des PLs um die Nichtregularität einer Sprache nachzuweisen.

$\rightsquigarrow$ www.logn.de/tut/ws0506/mat/pumpsatz_2.pdf

Wiederholung: Pumping Lemma

Satz

Das Pumping Lemma liefert folgende Implikation:

$$L \text{ regulär} \;\Rightarrow\; \begin{array}{l} \exists n \in \mathbb{N}: \\ \forall w \in L \text{ mit } |w| > n: \\ \exists \text{ Zerlegung } w = uvx \text{ mit } |uv| \leq n \text{ und } |v| > 0: \\ \forall i \in \mathbb{N}_0 : uv^i x \in L \end{array}$$

Tipp: Benutze die *Kontraposition* des PLs um die Nichtregularität einer Sprache nachzuweisen.

⇝ www.logn.de/tut/ws0506/mat/pumpsatz_2.pdf

Wiederholung: Pumping Lemma

Satz

Das Pumping Lemma liefert folgende Implikation:

$$
\begin{array}{lcl}
L \text{ regulär} & \Rightarrow & \exists n \in \mathbb{N} : \\
 & & \forall w \in L \text{ mit } |w| > n : \\
 & & \exists \text{ Zerlegung } w = uvx \text{ mit } |uv| \leq n \text{ und } |v| > 0 : \\
 & & \forall i \in \mathbb{N}_0 : uv^ix \in L
\end{array}
$$

Tipp: Benutze die *Kontraposition* des PLs um die Nichtregularität einer Sprache nachzuweisen.
⇝ www.logn.de/tut/ws0506/mat/pumpsatz_2.pdf

Wiederholung: Die Nerode Relation

Definition

Die Nerode Relation R_L zu einer Sprache $L \subseteq \Sigma^*$ ist eine *rechtsinvariante Äquivalenzrelation*. Für $x, y \in \Sigma^*$ gilt $x \; R_L \; y$ gdw.:

$$xz \in L \quad \Leftrightarrow \quad yz \in L \qquad \forall z \in \Sigma^*$$

Anzahl Äquivalenzklassen über R_L ist $\text{index}(R_L)$.

Theorem

Sei $L \subseteq \Sigma^*$ eine Sprache. Es gilt:

$$\text{index}(R_L) < \infty \quad \Leftrightarrow \quad L \text{ ist regulär}$$

Wiederholung: Die Nerode Relation

Definition

Die Nerode Relation R_L zu einer Sprache $L \subseteq \Sigma^*$ ist eine *rechtsinvariante Äquivalenzrelation*. Für $x, y \in \Sigma^*$ gilt $x \; R_L \; y$ gdw.:

$$xz \in L \quad \Leftrightarrow \quad yz \in L \qquad \forall z \in \Sigma^*$$

Anzahl Äquivalenzklassen über R_L ist $\text{index}(R_L)$.

Theorem

Sei $L \subseteq \Sigma^*$ eine Sprache. Es gilt:

$$\text{index}(R_L) < \infty \quad \Leftrightarrow \quad L \textit{ ist regulär}$$

Aufgabe 1.

Sei $\Sigma := \{a, b, c\}$ und

$$L := \{a^j b^k c^l \mid j, k, l \geq 0 \text{ und } k = l \text{ falls } j = 1\}$$

(a) Zeigen Sie, dass L alle Bedingungen des Pumping Lemma erfüllt.

(b) Zeigen Sie, dass L trotzdem nicht regulär ist.

Wiederholung: Komplexitätsklassen

„Deterministisch Polynomiell":

$$\mathcal{P} := \{L \subseteq \Sigma^* \mid \exists \text{ DTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{time}_{\mathcal{T}}(w) < p(|w|)\}$$

„Nichtdeterministisch Polynomiell":

$$\mathcal{NP} := \{L \subseteq \Sigma^* \mid \exists \text{ NTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{ntime}_{\mathcal{T}}(w) < p(|w|)\}$$

„$\mathcal{NP}$–hart":

$$\mathcal{NP}\# := \{L \subseteq \Sigma^* \mid \forall L' \in \mathcal{NP} : L' \leq_p L\}$$

„$\mathcal{NP}$–vollständig":

$$\mathcal{NPV} := \{L \subseteq \Sigma^* \mid L \in \mathcal{NP} \text{ und } L \in \mathcal{NP}\#\}$$

Wiederholung: Komplexitätsklassen

„Deterministisch Polynomiell“:

$$\mathcal{P} := \{L \subseteq \Sigma^* \mid \exists \text{ DTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{time}_{\mathcal{T}}(w) < p(|w|)\}$$

„Nichtdeterministisch Polynomiell“:

$$\mathcal{NP} := \{L \subseteq \Sigma^* \mid \exists \text{ NTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{ntime}_{\mathcal{T}}(w) < p(|w|)\}$$

„$\mathcal{NP}$–hart“:

$$\mathcal{NP}\# := \{L \subseteq \Sigma^* \mid \forall L' \in \mathcal{NP} : L' \leq_p L\}$$

„$\mathcal{NP}$–vollständig“:

$$\mathcal{NPV} := \{L \subseteq \Sigma^* \mid L \in \mathcal{NP} \text{ und } L \in \mathcal{NP}\#\}$$

Wiederholung: Komplexitätsklassen

„Deterministisch Polynomiell“:

$$\mathcal{P} := \{L \subseteq \Sigma^* \mid \exists \text{ DTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{time}_{\mathcal{T}}(w) < p(|w|)\}$$

„Nichtdeterministisch Polynomiell“:

$$\mathcal{NP} := \{L \subseteq \Sigma^* \mid \exists \text{ NTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{ntime}_{\mathcal{T}}(w) < p(|w|)\}$$

„$\mathcal{NP}$–hart“:

$$\mathcal{NP}\# := \{L \subseteq \Sigma^* \mid \forall L' \in \mathcal{NP} : L' \leq_p L\}$$

„$\mathcal{NP}$–vollständig“:

$$\mathcal{NPV} := \{L \subseteq \Sigma^* \mid L \in \mathcal{NP} \text{ und } L \in \mathcal{NP}\#\}$$

Wiederholung: Komplexitätsklassen

„Deterministisch Polynomiell“:

$$\mathcal{P} := \{L \subseteq \Sigma^* \mid \exists \text{ DTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{time}_{\mathcal{T}}(w) < p(|w|)\}$$

„Nichtdeterministisch Polynomiell“:

$$\mathcal{NP} := \{L \subseteq \Sigma^* \mid \exists \text{ NTM } \mathcal{T}_L : \forall w \in \Sigma^* : \text{ntime}_{\mathcal{T}}(w) < p(|w|)\}$$

„$\mathcal{NP}$–hart“:

$$\mathcal{NP}\# := \{L \subseteq \Sigma^* \mid \forall L' \in \mathcal{NP} : L' \leq_p L\}$$

„$\mathcal{NP}$–vollständig“:

$$\mathcal{NPV} := \{L \subseteq \Sigma^* \mid L \in \mathcal{NP} \text{ und } L \in \mathcal{NP}\#\}$$

Aufgabe (Wiederholung)

Aufgabe 2.

Zeigen Sie: Das Halteproblem ist $\mathcal{NP}$–hart. Warum ist es nicht $\mathcal{NP}$–vollständig?

Wiederholung: $\mathcal{NP}$–Vollständigkeitsbeweis

Wie zeigt man für ein Problem L, dass $L \in \mathcal{NPV}$?

Lösungsweg...

- Zeige $L \in \mathcal{NP}$
- Zeige L ist $\mathcal{NP}$–hart.
  Benutze ein Problem L', von dem bekannt ist, dass es $\mathcal{NP}$–*vollständig* ist, und reduziere es auf L. Dazu ist zu zeigen:
  - Die Reduktion ist polynomiell
  - Die Reduktion funktioniert:

$$I \in L \text{ hat Lösung } \Leftrightarrow I' \in L' \text{ hat Lösung}$$

Wiederholung: $\mathcal{NP}$–Vollständigkeitsbeweis

Wie zeigt man für ein Problem L, dass $L \in \mathcal{NPV}$?

Lösungsweg...

- Zeige $L \in \mathcal{NP}$
- Zeige L ist $\mathcal{NP}$–hart.
  Benutze ein Problem L', von dem bekannt ist, dass es $\mathcal{NP}$–*vollständig* ist, und reduziere es auf L. Dazu ist zu zeigen:
  - Die Reduktion ist polynomiell
  - Die Reduktion funktioniert:

$$I \in L \text{ hat Lösung } \Leftrightarrow I' \in L' \text{ hat Lösung}$$

Wiederholung: $\mathcal{NP}$–Vollständigkeitsbeweis

Wie zeigt man für ein Problem L, dass $L \in \mathcal{NPV}$?

Lösungsweg...

- Zeige $L \in \mathcal{NP}$
- Zeige L ist $\mathcal{NP}$–hart.
  Benutze ein Problem L', von dem bekannt ist, dass es $\mathcal{NP}$*–vollständig* ist, und reduziere es auf L. Dazu ist zu zeigen:
  - Die Reduktion ist polynomiell
  - Die Reduktion funktioniert:

$$I \in L \text{ hat Lösung } \Leftrightarrow I' \in L' \text{ hat Lösung}$$

Wiederholung: $\mathcal{NP}$–Vollständigkeitsbeweis

Wie zeigt man für ein Problem L, dass $L \in \mathcal{NPV}$?

Lösungsweg...

- Zeige $L \in \mathcal{NP}$
- Zeige L ist $\mathcal{NP}$–hart.
  Benutze ein Problem L', von dem bekannt ist, dass es $\mathcal{NP}$–*vollständig* ist, und reduziere es auf L. Dazu ist zu zeigen:
  - Die Reduktion ist polynomiell
  - Die Reduktion funktioniert:

$$I \in L \text{ hat Lösung } \Leftrightarrow I' \in L' \text{ hat Lösung}$$

Wiederholung: $\mathcal{NP}$–Vollständigkeitsbeweis

Wie zeigt man für ein Problem L, dass $L \in \mathcal{NPV}$?

Lösungsweg...

- Zeige $L \in \mathcal{NP}$
- Zeige L ist $\mathcal{NP}$–hart.
  Benutze ein Problem L', von dem bekannt ist, dass es $\mathcal{NP}$–*vollständig* ist, und reduziere es auf L. Dazu ist zu zeigen:
  - Die Reduktion ist polynomiell
  - Die Reduktion funktioniert:

$$I \in L \text{ hat Lösung} \Leftrightarrow I' \in L' \text{ hat Lösung}$$

Wiederholung: $\mathcal{NP}$–Vollständigkeitsbeweis

Wie zeigt man für ein Problem L, dass $L \in \mathcal{NPV}$?

Lösungsweg...

- Zeige $L \in \mathcal{NP}$
- Zeige L ist $\mathcal{NP}$–hart.
  Benutze ein Problem L', von dem bekannt ist, dass es *$\mathcal{NP}$–vollständig* ist, und reduziere es auf L. Dazu ist zu zeigen:
  - Die Reduktion ist polynomiell
  - Die Reduktion funktioniert:

$$I \in L \text{ hat Lösung } \Leftrightarrow I' \in L' \text{ hat Lösung}$$

Aufgabe (Wiederholung)

Das Problem SUBGRAPHENISOMORPHIE (SI)

Geg.: Graphen $G_1 := (V_1, E_1)$ und $G_2 := (V_2, E_2)$ mit $|V_2| < |V_1|$.
Frage: Gibt es eine Menge $U \subset V_1$ mit $|U| = |V_2|$ und einen Isomorphismus $\Phi : U \to V_2$, also eine bijektive Abbildung so, dass gilt

$$\{x, y\} \in E_1 \quad \Leftrightarrow \quad \{\Phi(x), \Phi(y)\} \in E_2 \qquad \forall x, y \in U$$

Aufgabe 3.

Zeigen Sie: SUBGRAPHENISOMORPHIE ist $\mathcal{NP}$-vollständig.

Aufgabe (Wiederholung)

Das Problem SUBGRAPHENISOMORPHIE (SI)

Geg.: Graphen $G_1 := (V_1, E_1)$ und $G_2 := (V_2, E_2)$ mit $|V_2| < |V_1|$.
Frage: Gibt es eine Menge $U \subset V_1$ mit $|U| = |V_2|$ und einen Isomorphismus $\Phi : U \to V_2$, also eine bijektive Abbildung so, dass gilt

$$\{x, y\} \in E_1 \quad \Leftrightarrow \quad \{\Phi(x), \Phi(y)\} \in E_2 \qquad \forall x, y \in U$$

Aufgabe 3.

Zeigen Sie: SUBGRAPHENISOMORPHIE ist $\mathcal{NP}$-vollständig.

Noch Fragen?

Viel Erfolg in der Klausur!

Noch Fragen?

Viel Erfolg in der Klausur!