Komplexitätstheorie

Kapitel 6: Mehr Ressourcen, mehr Möglichkeiten?

Einleitung

Bisher haben wir viele Komplexitätsklassen gesehen,

die wahrscheinlich unterschiedlich sind.

Einziges Gegenbeispiel bisher: $\mathrm{AC}_0 \subsetneq \mathrm{NC}_1$

Intuitiv aber doch: mehr Ressourcen, mehr Probleme lösbar!!

In diesem Kapitel:

- Für natürliche Fälle ist das meist auch richtig (Hierarchie-Theoreme)
- Es gibt eklatante Ausnahmen (Gap-Theoreme)

Kapitel 5

Hierarchie-Theoreme

Hierarchietheoreme

Hierarchietheoreme:

- Klasse von Resultaten, die Komplexitätsklassen separieren, mit denen also die Echtheit von Inklusionen nachgewiesen werden kann
- Betrachten Klassen, die auf demselben Maschinenmodell und derselben Ressource beruhen
- Basieren auf Diagonalisierungsbeweisen, ähnlich zu Beweisen von Unentscheidbarkeit (z.B. Halteproblem)

Zum warm werden: ein kurzer Exkurs zur Unentscheidbarkeit

Unentscheidbarkeit

Theorem.

Es gibt unentscheidbare Sprachen.

Diagonalisierungsbeweis:

- Wähle Kodierung μ von 1-Band DTMs als Worte, mit μ Bijektion (o.B.d.A.: alle TMs arbeiten über Alphabet $\{0,1\}$)
- Betrachte Tabelle für Akzeptanz von Worten durch TMs
- Deren Diagonale besteht aus Positionen M, w mit $\mu(M) = w$
- Sprache L_D: Komplementiere die Werte auf der Diagonalen
- Führe Widerspruchsbeweis für Unentscheidbarkeit von L_D

Hierarchietheoreme

Wir beweisen nur ein sehr spezielles Hierarchietheorem

Zur Erinnerung: ExpTime $:= \bigcup_{i \geq 1} \text{DTIME}(2^{\mathcal{O}(n^i)})$.

Theorem.

$\text{P} \subsetneq \text{ExpTime}$

Diagonalisierung soll Problem in $\text{ExpTime} \setminus \text{P}$ liefern:

- in ExpTime: betrachte nur Akzeptanz in exponentiell vielen Schritten
- Intuitiv sollte das Problem immernoch nicht in P sein

Beweis basiert auf geschickter Kodierung von DTMs als Worte

P und ExpTime

Sei μ Abbildung, die jedem Wort $w \in \Sigma^*$ (1-Band) DTM $\mu(w)$ zuordnet mit

- $\mu(w)$ ist für jedes Wort w definiert

  (wähle sinnvolle Kodierung, setze $\mu(w)$ auf Default-DTM, wenn w unter Kodierung nicht sinnvoll ist)

- Wenn $\mu(w) = M$, dann gibt es w' mit $|w'| > |w|$ und $\mu(w') = M$

  (Stelle z.B. sicher, dass $\mu(0^*w) = \mu(w)$ für alle $w \in \Sigma^*$)

- Basierend auf w kann $\mu(w)$ "effizient" simuliert werden

  (gegeben w und Konfiguration uqv von $\mu(w)$ kann DTM in $\mathsf{poly}(|w|+|uqv|)$ Schritten Konfiguration $u'q'v'$ berechnen mit $uqv \vdash_{\mu(w)} u'q'v'$)

Es ist nicht schwierig, so ein μ zu entwerfen.

P und ExpTime

Sprache L_D implementiert Diagonalisierung und Komplementierung:

Lemma.

$L_D := \{w \in \Sigma^* \mid \mu(w)$ akzeptiert w nicht in $\leq 2^{|w|}$ Schritten $\}$
ist in ExpTime $\setminus$ P

$L_D \notin$ P: Widerspruchsbeweis

- Annahme: es gibt p-zeitbeschr. (1-Band) DTM M mit $L(M) = L_D$.
- Wähle $w_M \in \Sigma^*$ mit $\mu(w_M) = M$ und $2^{|w_M|} \geq p(|w_M|)$
- $w_M \in L(M)$ und $w_M \notin L(M)$ führt beides zu Widerspruch

P und ExpTime

Sprache L_D implementiert Diagonalisierung und Komplementierung:

Lemma.

$L_D := \{w \in \Sigma^* \mid \mu(w)$ akzeptiert w nicht in $\leq 2^{|w|}$ Schritten $\}$
ist in ExpTime \ P

$L_D \in$ ExpTime: DTM M

- simuliert $\mu(w)$ auf Eingabe w
- zählt die Schritte, die schon simuliert wurden
- verwirft, wenn $\mu(w)$ in $\leq 2^{|w|}$ Schritten akzeptiert
- akzeptiert, wenn $\mu(w)$ in $\leq 2^{|w|}$ Schritten verwirft
- akzeptiert, wenn $\mu(w)$ mehr als $2^{|w|}$ Schritte macht

P und ExpTime

L_D ist unnatürliches Problem. Natürliche $L \in \text{P} \setminus \text{ExpTime}$ via Vollständigkeit:

Definition ExpTime-Härte, ExpTime-Vollständigkeit

Problem L ist

- *ExpTime-hart* wenn $L' \leq_p L$ für alle $L' \in \text{ExpTime}$;
- *ExpTime-vollständig* wenn L ExpTime-hart und in ExpTime.

Offensichtlich: wenn ExpTime-hartes $L \in \text{P}$, dann $L_D \in \text{P}$; also:

Lemma.

Wenn L ExpTime-hart, dann $L \notin \text{P}$.

P und ExpTime

ExpTime-vollständige Probleme:

- das Wortproblem für polyplatzbeschränkte alternierende TMs
  typisch für ExpTime, ähnlich wie SAT für NP, QBF für PSpace

- manche Spielprobleme
  z.B. Existenz von Gewinnstrategien in Dame Spielen
  (bei beliebig grossem Brett)

- manche Probleme in der Logik
  z.B.: Erfüllbarkeit von Formeln der Spezifikationslogik CTL

- manche Datenbankprobleme
  z.B. Inklusion zwischen XPath-Ausdrücken, für manche Fragmente von XPath 1.0 und 2.0

Zeithierarchiesatz

Das gerade gezeigte Resultat ist Spezialfall eines viel generelleren Satzes

Definition Zeitkonstruierbar

Funktion $t : \mathbb{N} \to \mathbb{N}$ heißt *zeitkonstruierbar* wenn es eine DTM M gibt mit $\mathsf{time}_M(w) = t(|w|)$ für alle $w \in \Sigma^*$.

Zeitkonstruierbarkeit im Prinzip "technische Annahme", wie platz-konstruierbarkeit bei Savitch's Theorem

Natürliche Funktionen sind i.d.R. zeitkonstruierbar, z.B.:

- n^i für alle $i \geq 1$
- 2^n

Zeithierarchiesatz

Theorem (Zeithierarchie).

Für jede zeitkonstruierbare Funktion t_2 und jede Funktion t_1 mit

1. $t_2 \in \omega(t_1 \cdot \log(t_1))$ und
2. $t_1(n) \geq n$ für alle $n \in \mathbb{N}$

gilt: $\mathsf{DTime}(t_1) \subsetneq \mathsf{DTime}(t_2)$.

Konsequenzen dieses Resultats z.B.:

- $\mathsf{DTime}(n^i) \subsetneq \mathsf{DTime}(n^{i+1})$ für alle $i \geq 1$

  (aber keine natürlichen Probleme in $\mathsf{DTime}(n^{i+1}) \setminus \mathsf{DTime}(n^i)$ bekannt)

- $\mathsf{DTime}(n^i) \subsetneq \mathsf{P}$ für alle $i \geq 0$

- $2\text{-ExpTime} := \bigcup_{i \geq 1} \mathsf{DTime}(2^{2^{\mathcal{O}(n^i)}})$, $3\text{-ExpTime} := \bigcup_{i \geq 1} \mathsf{DTime}(2^{2^{2^{\mathcal{O}(n^i)}}})$, etc.

  Dann k-ExpTime $\subsetneq k+1$-ExpTime für alle $k \geq 1$

Zeithierarchiesatz

Theorem (Zeithierarchie).

Für jede zeitkonstruierbare Funktion t_2 und jede Funktion t_1 mit

1. $t_2 \in \omega(t_1 \cdot \log(t_1))$ und
2. $t_1(n) \geq n$ für alle $n \in \mathbb{N}$

gilt: $\mathsf{DTime}(t_1) \subsetneq \mathsf{DTime}(t_2)$.

Unterschiede im Beweis:

- Konstruiere TM M, einige Schritte von $\mu(w)$ simuliert,

  komplementär akzeptiert und dabei selbst exakt $t_2(n)$ Schritte macht
- Dazu wird Zeitkonstruierbarkeit verwendet: gleichzeitiges Simulieren einer TM, die exakt $t_2(|w|)$ Schritte macht
- Verwende $L(M)$ als L_D, dann trivial: $L(M) \in \mathsf{DTime}(t_2)$
- $L(M) \notin \mathsf{DTime}(t_1)$: gleiche Idee, vorsichtigere Analyse Zeitverbrauch

Hierarchiesätze

Weitere Hierarchiesätze existieren, z.B. nicht-deterministische Zeit:

Beweis anders aber Konsequenzen wie in det. Fall, z.B.

- $\mathsf{NTime}(n^i) \subsetneq \mathsf{NTime}(n^{i+1})$ für alle $i \geq 1$;
- NP $\subsetneq$ NExpTime $:= \bigcup_{i\geq 1} \mathsf{NTime}(2^{\mathcal{O}(n^i)})$

Auch ein Satz für Platzhierarchie existiert:

etwas leichter zu zeigen, etwas mildere Annahmen reichen

Hierarchiesätze

Theorem (Platzhierarchie).

Für jede platzkonstruierbare Funktion s_2 und jede Funktion s_1 mit

1. $s_2 \in \omega(s_1)$ und
2. $s_1(n) > \log(n)$ für alle $n \in \mathbb{N}$

gilt: $\mathsf{DSpace}(s_1) \subsetneq \mathsf{DSpace}(s_2)$.

Konsequenzen z.B.:

- $\mathsf{DSpace}(n^i) \subsetneq \mathsf{DSpace}(n^{i+1})$ für alle $i \geq 0$
- $\mathsf{LogSpace} \subsetneq \mathsf{PSpace} \subsetneq \mathsf{ExpSpace} := \bigcup_{i \geq 1} \mathsf{DSpace}(2^{\mathcal{O}(n^i)})$
- $\mathsf{LogSpace} \subsetneq \mathsf{DSpace}(\log(n)^2)$