

Komplexitätstheorie

WiSe 2008/09 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Komplexitätstheorie Gesamtübersicht

- Organisatorisches / Einführung
Motivation / Erinnerung / Fragestellungen
- Diskussion verschiedener Komplexitätsklassen:
Zeitkomplexität
Platzkomplexität
- zugehörige Reduktionsbegriffe
- vollständige Probleme
- Anpassung von Klassenbegriffen und Reduktionen

Orakel-Turingmaschinen

Eine *Orakel-Turingmaschine* ist eine Turingmaschine M mit einem speziellen *Anfrage-Band* (engl.: query tape) und drei Zuständen

$q?$ q_{yes} q_{no}

Berechnung von M : nutzt Eingabe und zudem eine *Orakel-Menge* A .

Meist: M arbeitet wie eine normale (det. / n.det.) Turingmaschine.

Orakel-Turingmaschinen

Meist: M arbeitet wie eine normale (det. / n.det.) Turingmaschine.

Ausnahme: M erreicht den Zustand $q_?$ bei Orakel A .

—Orakel sei A

— a sei Inhalt des Anfrage-Bandes

\Rightarrow in einem Schritt:

—bei $a \in A$ geht M in q_{yes} über

—bei $a \notin A$ geht M in q_{no} über

Akzeptanz von Worten: wie bisher ...

$L(M^A) :=$ von der Orakel-Turingmaschine M bei Orakel A akzeptierte Sprache.

Orakel-Turingmaschinen

Sei \mathcal{K} Komplexitätsklasse, A ein Orakel:

\mathcal{K}^A := Klasse aller Sprachen, die beim Orakel A in der Komplexitätsklasse \mathcal{K} liegen.

Beispiele:

$$\mathcal{K}^\emptyset = \mathcal{K}$$

CHROMATIC NUMBER $\in \mathbf{P}^{\text{FÄRBBARKEIT}}$

Relativierte Berechenbarkeit

Beobachtung: Viele (alle bisherigen) Simulationen zwischen Sprach- / Komplexitätsklassen / Maschinenmodellen sind *orakelstabil*; sie funktionieren auch *relativiert* (d.h., mit einem Orakel).

Daher müsste gelten: Fall $\mathbf{P} = \mathbf{NP}$, so auch $\mathbf{P}^A = \mathbf{NP}^A$ für “viele” A .

So *hoffte* man, durch Betrachtung geeigneter Orakel der wichtigen Frage $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ näher zu kommen.

The *Millennium Prize Problems* are seven problems in mathematics that were stated by the Clay Mathematics Institute in 2000. Currently, six of the problems remain unsolved. A correct solution to each problem results in a US \$1,000,000 prize (sometimes called a Millennium Prize) being awarded by the institute. Only the Poincaré conjecture has been solved, but the solver, Grigori Perelman, has not pursued the conditions necessary to claim the prize.

Satz 1 *Es gibt ein Orakel A mit $P^A = NP^A$.*

Betrachte **PSPACE**-vollständiges Problem A .

(1) **PSPACE** \subseteq P^A :

Zu $L \in \mathbf{PSPACE}$ verwende Reduktionsfunktion f_L mit $x \in L \iff f_L(x) \in A$ als Orakelanfrage.

(2) $P^A \subseteq NP^A$ (trivial)

(3) $NP^A \subseteq \mathbf{NPSPACE}$:

Ersetze bei Orakel-Maschine M für $L \in NP^A$ die Orakelanfragen $y \in A$ durch Berechnungen einer det. TM für A . Dies führt insgesamt zu einer **NPSPACE**-Maschine, denn bei jeder Eingabe x für M haben die Orakelanfragen y (die zur Akzeptanz führen) eine Länge polynomial in $|x|$.

(4) **NPSPACE** = **PSPACE** (Folgerung aus dem Satz von Savitch)

Satz 2 *Es gibt ein Orakel B mit $\mathbf{P}^B \neq \mathbf{NP}^B$.*

Zu einer Menge $B \subseteq \Sigma^*$ definiere: $L_{[B]} = \{0^n \mid \exists w \in B : |w| = n\}$.

Trivial: $L_{[B]} \in \mathbf{NP}^B$

Bei Eingabe von 0^n rate $w \in \Sigma^n$ und teste mit Orakel B , ob $w \in B$.

Aufgabe: Konstruiere B so, dass $L_{[B]} \notin \mathbf{P}^B$.

—Hierzu $\Sigma = \{0, 1\}$.

— (M_n) : Aufzählung aller deterministischen Orakel-TM mit Orakeln über Σ .

—Konstruiere B (und parallel dazu eine disjunkte Menge C) “stufenweise”, so dass man mit “kurzen” Berechnungen keine Orakelanfragen $w \in B$ stellen kann.

—Wir konstruieren daher aufsteigende Mengenfolgen B_n, C_n mit $B = \bigcup B_n$ und $C = \bigcup C_n$.

Rekursive Stufenkonstruktion

Setze $B_0 = C_0 = \emptyset$.

Für $n > 0$ seien B_{n-1}, C_{n-1} bereits definiert.

Definiere B_n und C_n wie folgt:

—Simuliere M_n mit Orakel B_{n-1} auf 0^n für $2^{\lfloor n/2 \rfloor}$ Schritte. S_n sei die Menge der Orakelanfragen, die dabei gemacht wurde.

—Setze $C_n := C_{n-1} \cup S_n$.

—Falls M_n in $2^{\lfloor n/2 \rfloor}$ Schritten die Eingabe 0^n verwirft, wähle $y_n \in \{0, 1\}^n \setminus C_n$ beliebig. Setze $B_n := B_{n-1} \cup \{y_n\}$.

—Falls M_n nicht hält oder akzeptiert, setze $B_n := B_{n-1}$.

Beobachte: Stufenkonstruktion läuft über alle Orakel-TM.

Eigenschaften der Stufenkonstruktion

— $|S_n| \leq 2^{n/2}$. Die Orakelanfragen müssen aufs Band geschrieben werden.

$\leadsto |C_n| = |\bigcup_{i=1}^n S_i| \leq \sum_{i=1}^n 2^{i/2} < 2^n$.

\leadsto Es kann stets ein y_n wie angegeben gefunden werden.

— Mit $B = \bigcup B_n$ und $C = \bigcup C_n$ gilt: $B \cap C = \emptyset$.

Wäre $L_{[B]} \in \mathbf{P}^B$, so würde $L_{[B]}$ z.B. von der det. Orakel-TM M_n mit Orakel B akzeptiert. O.E. (Warum?) wähle n so groß, dass 2^n größer als die (polynomielle) Rechenzeit von M_n auf 0^n ist.

$0^n \in L_{[B]} \iff \exists w \in B : |w| = n$
 \iff Es wurde auf n -ter Stufe gewählt: $y_n \in \{0, 1\}^n \setminus C_n$
 $\iff M_n$ mit Orakel B_{n-1} akzeptiert nicht 0^n
 $\iff M_n$ mit Orakel B akzeptiert nicht 0^n
 \leadsto **Widerspruch**

Anmerkungen

- (1) Die Frage $\mathbf{P} = \mathbf{NP}$ kann nicht mit Hilfe von Orakel-Turingmaschinen gelöst werden.
- (2) Es gibt keinen Beweis für $\mathbf{P} = \mathbf{NP}$ oder $\mathbf{P} \neq \mathbf{NP}$, der auf Orakel-Turingmaschinen übertragbar ist.
- (3) $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ ist nicht zu Unrecht eine \$1 000 000 - Frage.

Sind \mathcal{K} und \mathcal{K}' zwei Komplexitätsklassen, bei denen $\mathcal{K}^A = \mathcal{K}^B$ für beliebige \mathcal{K}' -vollständige Probleme A, B gilt, so definieren wir

$$\mathcal{K}^{\mathcal{K}'} := \mathcal{K}^A$$

Falls \leq_{\log} - bzw. \leq_p -Reduktionen in der Klasse \mathcal{K} möglich sind:

Definition von $\mathcal{K}^{\mathcal{K}'}$ vom konkreten vollständigen Problem A unabhängig!

Damit z.B. $\text{CHROMATIC NUMBER} \in \mathbf{P}^{\mathbf{NP}}$,

zudem:

$$\mathbf{P}^{\mathbf{P}} = \mathbf{P} \quad , \quad \mathbf{NP}^{\mathbf{P}} = \mathbf{NP}$$

Unbekannt:

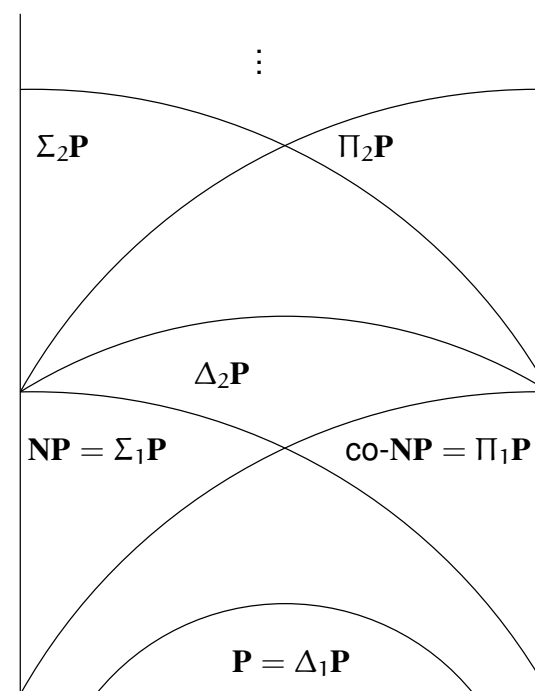
$$\mathbf{P}^{\mathbf{NP}} \stackrel{?}{=} \mathbf{NP}$$

Die **Polynomialzeithierarchie** ist die folgende induktiv definierte Folge von Komplexitätsklassen:

- $\Delta_0\mathbf{P} := \Sigma_0\mathbf{P} := \Pi_0\mathbf{P} := \mathbf{P}$
- $\Delta_{i+1}\mathbf{P} := \mathbf{P}^{\Sigma_i\mathbf{P}}$
- $\Sigma_{i+1}\mathbf{P} := \mathbf{NP}^{\Sigma_i\mathbf{P}}$
- $\Pi_{i+1}\mathbf{P} := \text{co} - (\mathbf{NP}^{\Sigma_i\mathbf{P}})$

für beliebige $i > 0$. Zudem sei

$$\mathbf{PH} := \bigcup_{i \geq 0} \Sigma_i\mathbf{P}$$



Aufbau der Polynomialzeit-Hierarchie

Satz 3 Für $L \subseteq \Sigma^*$ und $i > 0$ gilt

$$L \in \Sigma_i \mathbf{P}$$

genau dann, wenn eine polynomial balancierte Relation R existiert mit

$$L = \{x \mid \exists y \ (x, y) \in R\}$$

und

$$\{x; y \mid (x, y) \in R\} \in \Pi_{i-1} \mathbf{P}$$

Aufbau der Polynomialzeit-Hierarchie; Beweisteil I

Für $i = 1$ ist die Aussage bereits bewiesen (siehe letzte Vorlesung), denn $\Sigma_1\mathbf{P} = \mathbf{NP}$ und $\Pi_0\mathbf{P} = \mathbf{P}$.

Wir betrachten dies als Induktionsanfang und diskutieren den Induktionsschritt $(i - 1) \rightarrow i$:

\Leftarrow : Sei R gegeben; z.z.: für das zugehörige L muss gelten: $L \in \Sigma_i\mathbf{P}$.

Gesucht ist also nichtdet. TM M mit Orakel A aus $\Sigma_{i-1}\mathbf{P}$, die L akzeptiert.

Nach Vorauss.: $\{x; y \mid (x, y) \in R\} \in \Pi_{i-1}\mathbf{P}$ mit Balance $|y| \leq |x|^k$.

$\leadsto A := \{x; y \mid (x, y) \notin R, |y| \leq |x|^k\} \in \Sigma_{i-1}\mathbf{P}$

(*längenbeschränktes Komplement* von R).

Wähle dieses A als Orakel. Bei Eingabe x rät M ein y mit $|y| \leq |x|^k$ und überprüft (Orakelanfrage), ob $x; y \in A$. Falls $x; y \notin A$, akzeptiert M .

$\leadsto L \in \mathbf{NP}^{\Sigma_{i-1}\mathbf{P}} = \Sigma_i\mathbf{P}$.

Hilfsüberlegungen

Die Berechnung einer nichtdet. Orakel-TM M bei Eingabe x lässt sich mit zwei Hilfswörtern $s(x)$ und $o(x)$ (jeweils über $\{0, 1\}$) charakterisieren:

— $s(x) = s_1 \dots s_{|s(x)|}$ steht für die Folge der nichtdeterministischen Auswahlen, O.E. seien diese Auswahlen stets “binär”.

— $o(x) = o_1 \dots o_{|o(x)|}$ steht für die Folge der Antworten des Orakels.

Per def. sind die Antworten auf Orakelanfragen binär.

Ist M polynomzeitbeschr., so gilt $|s(x)| \leq |x|^k$ und $|o(x)| \leq |x|^k$ für geeignetes k .

Das Prädikat: $P_M^1(x; s; o)$ ist in \mathbf{P} berechenbar, wobei P_M^1 bedeuten soll:

“ M hält auf x bei Auswahlwort s und Orakelantworten o .”

Aufbau der Polynomialzeit-Hierarchie; Beweisteil II

Für $i = 1$ ist die Aussage bereits bewiesen, denn $\Sigma_1\mathbf{P} = \mathbf{NP}$ und $\Pi_0\mathbf{P} = \mathbf{P}$.

Wir betrachten dies als Induktionsanfang und diskutieren den Induktionsschritt $(i - 1) \rightarrow i$:

\Rightarrow : Sei $L \in \Sigma_i\mathbf{P} = \mathbf{NP}^{\Sigma_{i-1}\mathbf{P}}$.

$\rightsquigarrow \exists$ nichtdet. pol.zeitbeschränkte Orakel-TM M mit Orakel $A \in \Sigma_{i-1}\mathbf{P}$, die L akzeptiert.

Nach Induktionsannahme: \exists polynomial balancierte Relation S (mit Balance k), sodass

(1) $\{x; y \mid (x, y) \in S\} \in \Pi_{i-2}\mathbf{P}$ und (2) $A = \{x \mid \exists y \ (x, y) \in S\}$.

Betrachte wieder "längenbeschr. Komplement": $S' := \{(x, y) \mid (x, y) \notin S, |y| \leq |x|^k\} \in \Sigma_{i-2}\mathbf{P}$.

Es gilt: $A = \{x \mid \forall y \ (x, y) \notin S'\}$.

$x \in L$ gdw. $(\exists s \in \{0, 1\}^*, |s| \leq |x|^k) (\exists o \in \{0, 1\}^*, |o| \leq |x|^k)$ mit (1) $P_M^1(x; s; o)$ und (2) Die Antworten von Orakel A bei der Berechnung auf x (mit s) stimmen mit o überein.

(2) bedeutet: $\neg \exists i \leq |o| \exists$ Orakelanfrage q_i (bei x und s): $o_i \neq "q_i \in A"$. Das heißt formaler:

$\neg \exists i \leq |o| \exists q_i ((o_i \wedge (\exists y_1 : (q_i, y_1) \in S)) \vee (\neg o_i \wedge \neg(\exists y_2 : (q_i, y_2) \in S'))))$.

$P_M^2(x; s; o) := (\exists i \leq |o| \exists q_i (P_M^1(x; s; o) \wedge (\exists y_1 : (o_i \wedge (q_i, y_1) \in S) \vee \neg \exists y_2 : o_i \wedge (q_i, y_2) \in S'))))$

Definiere: $R := \{(x; s; o) \mid P_M^2(x; s; o), x \text{ Eingabe}, s, o \in \{0, 1\}^*, |s| \leq |x|^k, |o| \leq |x|^k\}$.

$R \in \Sigma_{i-1}\mathbf{P}$, da dies für die einzelnen Formelbestandteile gilt.

\rightsquigarrow längenbeschr. Komplement $R' \in \Pi_{i-1}\mathbf{P}$, $L = \{x \mid \exists s, o : (x; s; o) \in R'\}$.

Folgerung 4 Für $L \subseteq \Sigma^*$ und $i > 0$ gilt:

$$L \in \Pi_i \mathbf{P}$$

genau dann, wenn eine polynomial balancierte Relation R (mit Balance k) existiert mit

$$L = \{x \mid (\forall y, |y| \leq |x|^k) (x, y) \in R\}$$

und

$$\{x; y \mid (x, y) \in R\} \in \Sigma_{i-1} \mathbf{P}.$$

Folgerung 5 Für $L \subseteq \Sigma^*$ und $i > 0$ gilt

$$L \in \Sigma_i \mathbf{P}$$

genau dann, wenn eine polynomial balancierte und in polynomialer Zeit entscheidbare $(i + 1)$ -stellige Relation R existiert mit

$$L = \{x \mid (\exists y_1)(\forall y_2)(\exists y_3) \dots (Qy_i) \ (x, y_1, y_2, \dots, y_i) \in R\}$$

Dabei ist $Q = \exists$ bei ungeradem i und $Q = \forall$ bei geradem i .

Diese Struktur entspricht voll-quantifizierten Formeln.

vgl. $\Sigma\text{-QBF}$ s.u. im nachfolgenden Abschnitt über **PSPACE**

Man kann für jede "Ebene" der Polynomialzeit-Hierarchie vollständige QBF -Probleme bauen, indem die Quantorenalternierungen auf der entsprechenden Ebene abbrechen.

Damit gilt auch:

Folgerung 6 $\mathbf{PH} \subseteq \mathbf{PSPACE}$.

Frage $\mathbf{NP} = \text{co-NP}$? taucht auf jeder Stufe der Hierarchie auf:

Satz 7 Gilt für ein $i \geq 1$ $\Sigma_i\mathbf{P} = \Pi_i\mathbf{P}$, so gilt auch für alle $j > i$

$$\Sigma_j\mathbf{P} = \Pi_j\mathbf{P} = \Delta_j\mathbf{P} = \Sigma_i\mathbf{P},$$

d.h. die Polynomialzeit-Hierarchie *kollabiert* dann auf i -ter Stufe.

Satz 8 Gilt $\mathbf{P} = \mathbf{NP}$ oder $\mathbf{NP} = \text{co-NP}$, so kollabiert die Polynomialzeit-Hierarchie bereits auf Ebene 1.

Satz 9 Gibt es ein **PH**-vollständiges Problem, so kollabiert die Hierarchie auf endlicher Stufe.

(Ohne Beweis)

Mehr Orakel

In unserem bisherigen Orakelmodell durften beliebig viele Orakelanfragen hintereinander gestellt werden.

Vgl.: Turing-Reduktionen vs. many-one Reduktionen vs. andere Reduktionsmodelle.

Darf die TM nur einmal, aber dann evtl. mehrere parallele Anfragen stellen, gelangt man zu Komplexitätsklassen wie $P_{||}^{NP}$, auch P_{tt}^{NP} aus logischer Sichtweise genannt.

Man kann auch z.B. nur logarithmisch viele Anfragen hintereinander zulassen: $P^{NP[\log]}$.

Satz 10 $P_{||}^{NP} = P_{tt}^{NP} = P^{NP[\log]} = L^{NP}$. (ohne Beweis)

Beispiel für ein L^{NP} -vollständiges Problem: VERTEX COVER MEMBER:

Eingabe: unger. Graph $G = (V, E)$, Knoten v ;

Frage: Gibt es eine kleinstmögliche Knotenüberdeckung von G , die v enthält ?