Komplexitätstheorie

Weitere Komplexitätsklassen und Beziehungen zwischen Komplexitätsklassen

Vortrag im Seminar „Theoretische Informatik und Mathematik“

Von Jens Mahnke

Einleitung

- Bisher: Betrachtung von P≠NP bzw. P=NP

- Jetzt: vernünftige und stärkere Annahmen
  - Innerer Aufbau von NP und coNP
  - Klassen oberhalb NP ∪ coNP (Orakelklassen)
  - Die polynomielle Hierarchie (PH)
  - Einordnung von BPP und NP in die PH

Komplexitätsklassen innerhalb von NP und coNP

- Wenn P=NP gilt
  - P = coP => P = coNP

  => Nur eine Klasse von Problemen exisitert
- Viel interessanter: Wenn P≠NP gilt
  - Drei Äquivalenzklassen in P:
    - Nichts akzeptieren, Alles akzeptieren, der Rest
  - Klasse der NP-vollständigen Probleme in NP
- Was ist mit dem Rest:
  - NPI = NP – (P ∪ NPC) ={}?

Die Klasse NPI

- Kandidaten nach Garey und Johnson 1979:
  - Lineares Programmieren (LP)
  - Primzahltests (PRIMES)
  - Graphenisomorphie (GI)
- Heute bewiesen: LP und PRIMES in P
- Aber vermutlich: **GI$\notin$P und GI$\notin$NPC**

  $\Rightarrow$Vermutung: **GI$\in$NPI**
- *Theorem 10.2.1: Wenn P$\neq$NP ist, dann ist NPI nicht leer und enthält Probleme, die in Bezug auf $\leq_P$ nicht vergleichbar sind.*

Darstellung von NP und coNP

- Kapitel 5.3 Einführung alternativen Darstellung von L ∈ NP
  - Polynomieller Existenzquantor
  - Polynomzeiteigenschaft, mit L‘∈P:

$$L = \left\{ x \mid \exists z \in \{0,1\}^{p(|x|)} : (x,z) \in L' \right\}$$

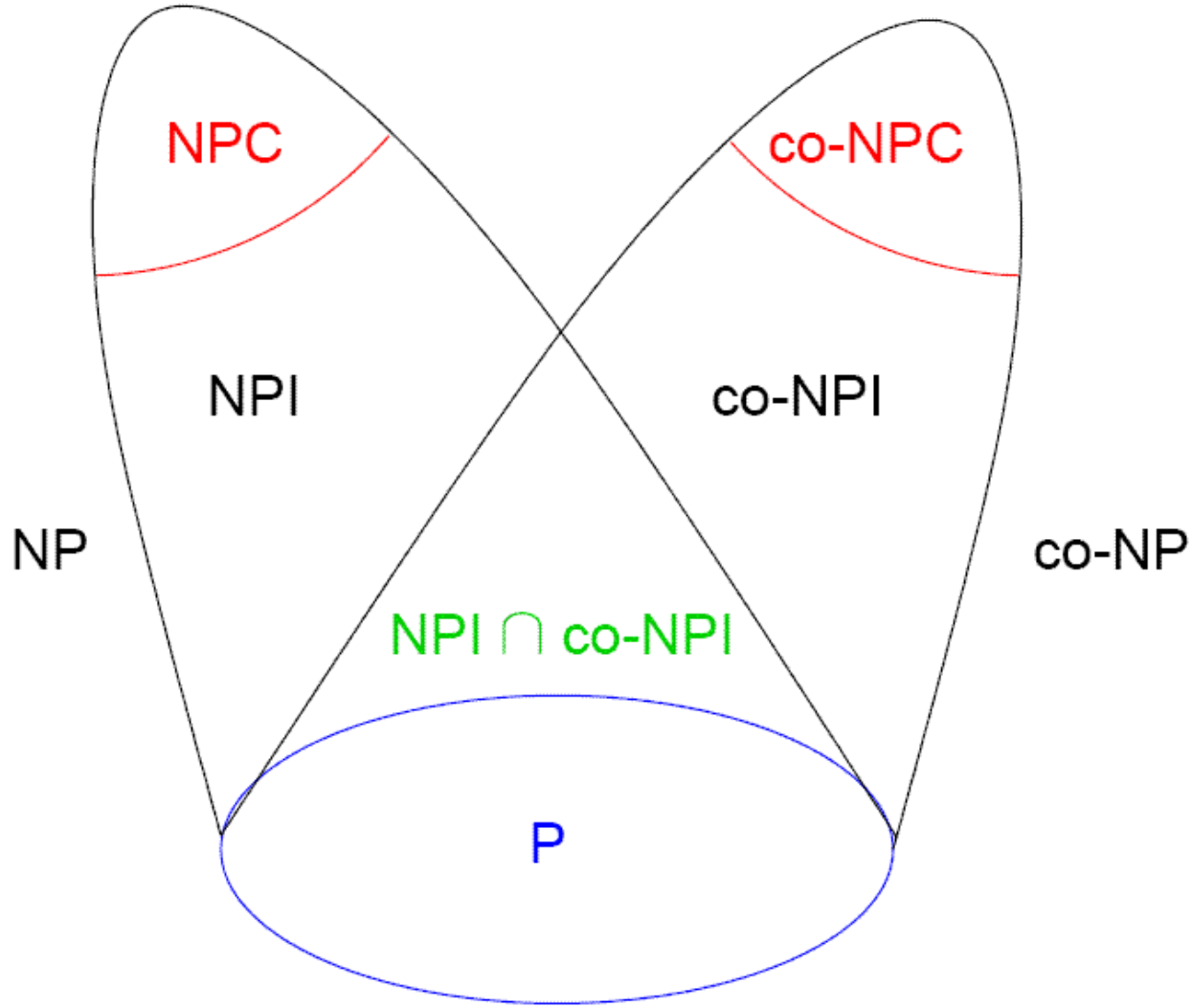
- Darstellung von coNP durch Austausch Existenz- durch All-quantor

  => Untermauerung der These NP≠coNP

Die Klassen NP und coNP

- *Theorem 10.2.2: Wenn L $\in$NPC und L $\in$coNP folgt NP=coNP.*
  - Beweis:
    - Reduktionen aus beiden Klassen möglich
    - n.det. TM konstruieren...
- D.h: NP$\neq$coNP => für L $\in$ NP $\cap$ coNP
  - L $\notin$ NPC bzw. L $\notin$ coNPC
- Beispiel: GI $\in$ NP !
  - Ist GI in coNP oder nicht?

Komplexitätsklassen



Orakelklassen

- Die Welt oberhalb von NP $\cup$ coNP
- *Definition 10.3.1: Die Komplexitätsklasse P(L) für das Entscheidungsproblem L enthält alle Entscheidungsprobleme L' mit L' $\leq_T$ L enthält, d.h. alle Probleme L', die mit Hilfe eines Orakels für L mit einem Polynomzeit-Algorithmus entschieden werden können.*
- *Die Komplexitätsklasse P(C) für eine Klasse C von Entscheidungsproblemen ist die Vereinigung aller P(L) mit L $\in$ C.*
  - $P(L):=\{L' \in ENT | L' \leq_T L\}$

Orakelklassen

- *Definition 10.3.2: Die Komplexitätsklasse NP(L) definiert für das Entscheidungsproblem L, welches alle Entscheidungsprobleme L' enthält, für die ein nichtdeterministischer polynomzeit Algorithmus mit Orakel L existiert.*
- *Die Komplexitätsklasse NP(C) für eine Klasse C von Entscheidungsproblemen ist die Vereinigung aller NP(L) mit L $\in$ C.*
  - NP(L):={L'$\in$ENT| existiert n.det. poly. Alg. für L' der Orakel L benutzt}

Beispiel: Minimal Circuit (MC)

- *Theorem 10.3.3: MC $\in$ coNP(NP),*
  - *und vermutlich gilt MC $\notin$ NP oder NP(NP)*
  - *Beweis: -> Tafel*

Polynomielle Hierarchie

- Definiert weitere sinnvolle Klassen
- Neue Notation für die Klassen

Definition 10.4.1:

Sei $\Sigma_1 := NP$, $\Pi_1 := coNP$ und $\Delta_1 := P$. Für $k \geq 1$, sei

$$\Sigma_{k+1} := NP(\Sigma_k),$$
$$\Pi_{k+1} := co\Sigma_{k+1} = coNP(\Sigma_k)$$
$$\Delta_{k+1} := P(\Sigma_k)$$

Die polynomielle Hierarchie (PH) ist die Vereinigung aller Σk für $k \geq 1$.

Für $k = 0$ sei $\Sigma_0 = \Pi_0 = \Delta_0 = P$.

Beispiel: $\Sigma_3=NP(NP(NP))$, $\Pi_3=coNP(NP(NP))$, $\Delta_3=P(NP(NP))$

Lemmas zur Polynomiellen Hierarchie

Lemma 10.4.2.a:

$$\Delta_k = co\Delta_k = P(\Delta_k) \subseteq \Sigma_k \cap \Pi_k \subseteq \Sigma_k \cup \Pi_k \subseteq \Delta_{k+1} = P(\Pi_k)$$

Lemma 10.4.2.b:

$$\Sigma_{k+1} = NP(\Pi_k) = NP(\Delta_{k+1})$$

Lemma 10.4.2.c:

$$\Pi_{k+1} = coNP(\Pi_k) = coNP(\Delta_{k+1})$$

Lemma 10.4.2.d:

$$\Sigma_k \subseteq \Pi_k \Rightarrow \Sigma_k = \Pi_k$$

Komplexitäts-Landkarte der polynomiellen Hierarchie

- Pfeile stehen für Teilmengen
- Komplexitätstheoretische Hypothesen:

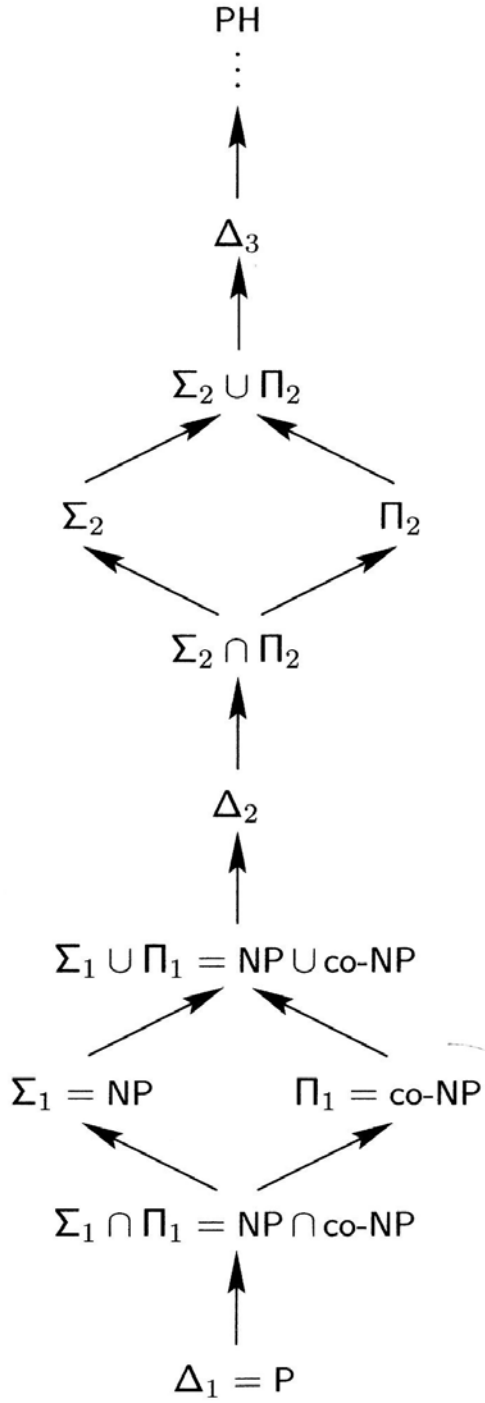
$$\forall k:$$

$$\Sigma_k \neq \Sigma_{k+1}$$

$$\Pi_k \neq \Pi_{k+1}$$

$$\Sigma_k \neq \Pi_k$$

$$\Delta_k \neq \Sigma_k \cap \Pi_k \neq \Sigma_k \neq \Sigma_k \cup \Pi_k \neq \Delta_{k+1}$$



Logikorientierte Sicht auf Σ_k

Theorem 10.4.3: Ein Entscheidungsproblem L gehört zur Klasse Σ_k wenn ein Polynom p und ein Entscheidungsproblem L' $\in$ P existiert, so dass für $A=\{0,1\}^{p(|x|)}$ gilt:

$$L = \{x \mid \exists y_1 \in A \forall y_2 \in A \exists y_3 \in A ... Q y_k \in A : (x, y_1, y_2, y_3, ..., y_k) \in L'\}$$

Wobei Q für einen Quantor (All- oder Existenz-) steht, so dass eine alternierende Folge von Quantoren entsteht. Bei geradem k entspricht das Q einem Allquantor und bei ungeradem k einem Existenzquantor.

Logikorientierte Sicht auf Π_k

Korollar 10.4.4: Ein Entscheidungsproblem L ist in Π_k, wenn und nur wenn ein Polynom p und ein Entscheidungsproblem L' aus P existiert, so dass $A = \{0,1\}^{p(|x|)}$ gesetzt werden kann und es gilt:

$$L = \{x \mid \forall y_1 \in A \exists y_2 \in A, ..., Q y_k \in A : (x, y_1, ..., y_k) \in L'\}$$

Bei ungeradem k entspricht das Q einem Allquantor und bei geradem k einem Existenzquantor.

Theorem 10.4.5: Wenn $\Sigma_k = \Pi_k$, dann ist $PH = \Sigma_k$.

$$Beweis: z.Z. \Sigma_k = \Pi_k \Rightarrow \Sigma_{k+1} = \Pi_{k+1} = \Sigma_k$$

$$Beispiel: k = 4, \Sigma_4 = \Pi_4$$

$$\exists\forall\exists\forall P = \forall\exists\forall\exists P \; (Theorem 10.4.3 / Korollar 10.4.4)$$

$$\Sigma_5 = \exists(\forall\exists\forall\exists P) = \exists\exists\forall\exists\forall P = \exists\forall\exists\forall P = \Sigma_4$$

$$ana\log \Pi_5 = \Pi_4$$

$$\Rightarrow \Sigma_5 = \Sigma_4 = \Pi_5 = \Pi_4$$

$$\Sigma_{k+1} \neq \Sigma_k \; ist \; strenger \; als \; \Sigma_k \neq \Sigma_{k-1}$$

$$NP \neq P \; Hypothese \; ist \; die \; schwächste \; aller \; Annahmen$$

$$Für \; NP = P \Rightarrow nur \; \Sigma_1 = \Pi_1 \; und \; PH = P$$

Korollar 10.4.6: Wenn $\Sigma_k = \Sigma_{k+1}$, dann ist PH = Σ_k.

Beweis :

Es gilt nach Lemma 10.4.2 : $\Sigma_k \subseteq \Pi_{k+1}$ mit

Voraussetzung $\Sigma_k = \Sigma_{k+1}$

$\Rightarrow \Sigma_{k+1} \subseteq \Pi_{k+1}$ und mit Lemma 10.4.2 $\Sigma_{k+1} = \Pi_{k+1}$

$\Rightarrow$ nach Theorem 10.4.5 $PH = \Sigma_{k+1}$

$\Rightarrow$ nach Voraussetzung $PH = \Sigma_k$

BPP, NP und die Polynomielle Hierarchie

- Welcher Zusammenhang besteht zwischen den Klassen NP und BPP?
  - NP und BPP basieren auf randomisierten Algorithmen.
  - NP mit einseitigem, BPP mit beidseitigem Fehler.
  - NP mit sehr hoher, BPP mit beschränkter Fehlerwahrscheinlichkeit.
  - Vermutet wird eine Teilmengenbeziehung zwischen NP und BPP

Theorem 10.5.1: $BPP \subseteq \Sigma_2 \cap \Pi_2$

- Beweis: z.Z. BPP Teilmenge von Σ_2, da gilt

Definition 10.5.2:

Für ein Entscheidungsproblem L enthält die Komplexitätsklasse RP(L) alle Entscheidungsprobleme L', für die es einen RP-Algorithmus mit einem Orakel für L gibt. Die Komplexitätsklasse RP(C) für die Klasse der Entscheidungsprobleme C ist die Vereinigung über alle RP(L) mit L $\in$ C.

Analog werden die Klassen RP(C); ZPP(L); ZPP(C); BPP(L); BPP(C); PP(L); PP(C) definiert.

Theorem 10.5.3: $BPP \subseteq RP(NP) \cap coRP(NP)$

- Beweis: bei Theorem 10.5.1 bereits gezeigt, dass mindestens die hälfte der Vektoren „gute“ Vektoren sind. D.h. bei zufälliger Auswahl

=> Pr(Fehler) ≤ 1/2

Theorem 10.5.4: *BPP (BPP) = BPP*

- L ∈BPP(BPP), dann existiert Orakel L‘∈BPP
  - mit L ∈BPP(L‘)
- Beweis: Angabe eines BPP Algorithmus für L

Korollar 10.5.6: $NP \subseteq BPP \Rightarrow PH \subseteq BPP$

- Beweis: Voraussetzung Lemma 10.5.7, zu zeigen, das Σ_k Teilmenge von BPP für alle k ist

$$\Sigma_{k+1} = NP(\Sigma_k) \subseteq NP(BPP) \subseteq BPP(NP) \subseteq BPP(BPP) = BPP$$

Lemma 10.5.7: $NP(BPP) \subseteq BPP(NP)$

- Für Beweis von Theorem 10.5.6
- Beweis: Über Konstruktion eines BPP Algorithmus mit NP Orakel und Fehlerwahrscheinlichkeit 1/4

Theorem 10.5.9: $NP \subseteq BPP \Rightarrow NP = RP$

Zusammenfassung

- Erweiterung der Vorstellung der Komplexitätstheorie-Klassen
- Einführung der Klasse NPI zwischen P und NPC
- Neue Klassen oberhalb von NP und coNP, die Orakelklassen, definiert und betrachtet
- Zusammenhang von NP mit BPP und RP vertieft
- Erhärtung der NP≠P These

Literatur

Wegener, Ingo: “Complexity Theory – Exploring the Limits of Efficient Algorithms”, Berlin/Heidelberg 2005.

Wegener, Ingo: „Effiziente Algorithmen und Komplexitätstheorie mit dem Schwerpunkt Komplexitätstheorie“, Vorlesung im WS 2004/05 an der Universität Dortmund.