A short look at Query Hierarchies over **NP** & The Graph Isomorphism Problem

Frank Heitmann
heitmann@informatik.uni-hamburg.de

Gastvortrag im Rahmen der Vorlesung
Komplexitätstheorie
von
Prof. Dr. Matthias Jantzen
14.07.2008

Die Query Hierarchie

Definition (Query Hierarchie)

Die *Anfragehierarchie* über **NP** ist definiert durch:

$$\mathbf{P}^{\mathbf{NP}[k]} = \left\{ L \;\middle|\; \begin{array}{l} L = L(M^B) \text{ für ein } B \in \mathbf{NP} \text{ und eine} \\ \text{DPOTM } M\text{, die nicht mehr als } k \\ \text{Orakelfragen stellt} \end{array} \right\}$$

$$\mathbf{P}^{\mathbf{NP}[\mathcal{O}(1)]} = \bigcup_{k \in \mathbb{N}} \mathbf{P}^{\mathbf{NP}[k]}$$

Die parallele Query Hierarchie

Definition (Parallele Query Hierarchie)

Die *parallele Anfragehierarchie* über **NP** ist definiert durch:

$$\mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} = \left\{ L \;\middle|\; \begin{array}{l} L = L(M^B) \text{ für ein } B \in \mathbf{NP} \text{ und eine} \\ \text{DPOTM } M\text{, die nicht mehr als } k \\ \text{Orakelfragen } \textit{in parallel} \text{ stellt} \end{array} \right\}$$

$$\mathbf{P}^{\mathbf{NP}}_{\text{btt}} = \bigcup_{k \in \mathbb{N}} \mathbf{P}^{\mathbf{NP}}_{k\text{-tt}}$$

Die parallele Query Hierarchie

Dabei bedeutet *'in parallel'*, dass die Orakelfragen gleichzeitig gestellt werden. Anders als bisher wird also ein Tupel $(q_1, \ldots, q_n)$ von Orakelfragen an das Orakel als Frage gestellt. Das Orakel B antwortet dann mit einem Tupel $(x_1, \ldots, x_n)$, wobei $x_i = 1$ ist genau dann, wenn $q_i \in B$ ist. Die Orakel-Turingmaschine wird dafür so abgeändert, dass nach einer Orakelfrage das Orakelband geleert und das obige Tupel mit Antworten auf dem Band notiert ist. Die Zustände z_{ja} und z_{nein} sind dann nicht mehr nötig.

Verzahnung mit der **BH**

Satz

1. $\mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} = \mathbf{P}^{A[1]}$ *für eine* $\mathbf{BH}_k$*-vollständige Menge* A *(bzgl.* $\leq^p_m$*-Reduktion).*

Verzahnung mit der **BH**

Satz

1. $\mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} = \mathbf{P}^{A[1]}$ *für eine* $\mathbf{BH}_k$*-vollständige Menge* A *(bzgl.* $\leq^p_m$*-Reduktion).*
2. $\mathbf{P}^{\mathbf{NP}[k]} = \mathbf{P}^{\mathbf{NP}}_{(2^k-1)\text{-tt}}$.

Verzahnung mit der BH

Satz

1. $\mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} = \mathbf{P}^{A[1]}$ *für eine* $\mathbf{BH}_k$*-vollständige Menge* A *(bzgl.* $\leq^p_m$*-Reduktion).*
2. $\mathbf{P}^{\mathbf{NP}[k]} = \mathbf{P}^{\mathbf{NP}}_{(2^k-1)\text{-tt}}$.
3. $\mathbf{BH}_k \cup \mathbf{coBH}_k \subseteq \mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} \subseteq \mathbf{BH}_{k+1} \cap \mathbf{coBH}_{k+1}$.

Verzahnung mit der **BH**

Satz

1. $\mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} = \mathbf{P}^{A[1]}$ *für eine* $\mathbf{BH}_k$*-vollständige Menge* A *(bzgl.* $\leq^p_m$*-Reduktion).*
2. $\mathbf{P}^{\mathbf{NP}[k]} = \mathbf{P}^{\mathbf{NP}}_{(2^k-1)\text{-tt}}$.
3. $\mathbf{BH}_k \cup \mathbf{coBH}_k \subseteq \mathbf{P}^{\mathbf{NP}}_{k\text{-tt}} \subseteq \mathbf{BH}_{k+1} \cap \mathbf{coBH}_{k+1}$.
4. $\mathbf{BH}_{2^k-1} \cup \mathbf{coBH}_{2^k-1} \subseteq \mathbf{P}^{\mathbf{NP}[k]} \subseteq \mathbf{BH}_{2^k} \cap \mathbf{coBH}_{2^k}$.

Verzahnung mit der **BH**

Satz

$$\mathbf{BH}(\mathbf{NP}) = \mathbf{P}^{\mathbf{NP}[\mathcal{O}(1)]} = \mathbf{P}^{\mathbf{NP}}_{\mathrm{btt}} = \mathbf{BC}(\mathbf{NP})$$

Verzahnung mit der BH

Satz

$\mathbf{BH}(\mathbf{NP}) = \mathbf{P}^{\mathbf{NP}[\mathcal{O}(1)]} = \mathbf{P}^{\mathbf{NP}}_{\mathrm{btt}} = \mathbf{BC}(\mathbf{NP})$

Satz

$\mathbf{BH}(\mathbf{NP}) \subseteq \mathbf{P}^{\mathbf{NP}[\log]} = \mathbf{P}^{\mathbf{NP}}_{\mathrm{tt}} = \mathbf{\Theta}_2$

Verfeinerung des SvK

Satz

Ist $\mathbf{BH}_k(\Sigma_i) = \mathbf{coBH}_k(\Sigma_i)$ *für* $i, k \geq 1$*, so folgt* $\mathbf{PH} \subseteq \Sigma_{i+2}$.

Verfeinerung des SvK

Satz

Ist $\mathbf{BH}_k(\Sigma_i) = \mathbf{coBH}_k(\Sigma_i)$ *für* $i, k \geq 1$, *so folgt* $\mathbf{PH} \subseteq \Sigma_{i+2}$.

Satz

1. *Gilt* $\mathbf{P}^{\Sigma_i}_{k\text{-tt}} = \mathbf{P}^{\Sigma_i}_{(k+1)\text{-tt}}$ *für* $i, k \geq 1$, *so folgt* $\mathbf{PH} \subseteq \Sigma_{i+2}$.
2. *Gilt* $\mathbf{P}^{\Sigma_i[k]} = \mathbf{P}^{\Sigma_i[k+1]}$ *für* $i, k \geq 1$, *so folgt* $\mathbf{PH} \subseteq \Sigma_{i+2}$.

Intro...

Wir werden nun zwei weitere Hierarchien einführen...

- die *Low Hierarchy* und
- die *High Hierarchy*

in **NP**.

Intro...

Wir werden nun zwei weitere Hierarchien einführen...

- die *Low Hierarchy* und
- die *High Hierarchy*

in **NP**.

Diese werden mittles der polynomiellen Hierarchie definiert und es uns erlauben **NP** (!) genauer zu untersuchen!

Die Low Hierarchy

Definition

Die *Low Hierarchy* in **NP** wird definiert durch

$$\begin{aligned} \textbf{Low}_k &= \{L \mid L \in \textbf{NP} \text{ und } \Sigma_k^L \subseteq \Sigma_k\} \text{ für jedes } k \geq 0 \\ \textbf{LH} &= \cup_{k \geq 0} \textbf{Low}_k \end{aligned}$$

Die High Hierarchy

Definition

Die *High Hierarchy* in **NP** wird definiert durch

$$\begin{aligned}\mathbf{High}_k &= \{H \mid H \in \mathbf{NP} \text{ und } \Sigma_{k+1} \subseteq \Sigma_k^H\} \text{ für jedes } k \geq 0\\ \mathbf{HH} &= \cup_{k\geq 0}\mathbf{High}_k\end{aligned}$$

Charakterisierung der unteren Stufen

Satz

1. $\mathbf{Low}_0 = \mathbf{P}$
2. $\mathbf{Low}_1 = \mathbf{NP} \cap \mathbf{coNP}$
3. $\mathbf{High}_0 = \{H \mid H \text{ ist } \leq_T^p\text{-vollständig für } \mathbf{NP}\}$
4. $\mathbf{High}_1 = \{H \mid H \text{ ist } \leq_{sT}^{np}\text{-vollständig für } \mathbf{NP}\}$

Einige Ergebnisse I

Satz

1. $\mathbf{Low}_0 \subseteq \mathbf{Low}_1 \subseteq \ldots \subseteq \mathbf{Low}_k \subseteq \ldots \subseteq \mathbf{LH} \subseteq \mathbf{NP}$
2. $\mathbf{High}_0 \subseteq \mathbf{High}_1 \subseteq \ldots \subseteq \mathbf{High}_k \subseteq \ldots \subseteq \mathbf{HH} \subseteq \mathbf{NP}$
3. $\mathbf{Low}_k \cap \mathbf{High}_k \neq \emptyset \iff \Sigma_k = \Sigma_{k+1} = \ldots = \mathbf{PH}$
4. *Entweder* $\mathbf{Low}_k \cap \mathbf{High}_k = \emptyset$ *oder* $\mathbf{NP} = \mathbf{Low}_k = \mathbf{High}_k$

Einige Ergebnisse II

Satz (Schöning)

1. $L \in \mathbf{NP} \setminus (\mathbf{Low}_k \cup \mathbf{High}_k) \iff \Sigma_k \neq \Sigma_{k+1}$
2. $L \in \mathbf{NP} \setminus (\mathbf{LH} \cup \mathbf{HH}) \iff$ *die PH ist strictly infinite*

Graphenisomorphie I

- Man kann zeigen, dass das Graphenisomorphieproblem GI in $\mathbf{Low}_2$ ist. (Nutzt eine weitere Hierarchie.)

Graphenisomorphie I

- Man kann zeigen, dass das Graphenisomorphieproblem `GI` in $\mathbf{Low}_2$ ist. (Nutzt eine weitere Hierarchie.)
- Dies gilt als starker (!) Anhaltspunkt, dass `GI` *nicht* **NP**-vollständig sein kann.

Graphenisomorphie II

Warum?

Graphenisomorphie II

Warum?

- Wäre GI $\in$ **NPC**, dann wäre es insbesondere in $\mathbf{High}_0$.

Graphenisomorphie II

Warum?

- Wäre GI $\in$ **NPC**, dann wäre es insbesondere in $\mathbf{High}_0$.
- Dann wäre GI auch in $\mathbf{High}_2$.

Graphenisomorphie II

Warum?

- Wäre GI $\in$ **NPC**, dann wäre es insbesondere in $\mathbf{High}_0$.
- Dann wäre GI auch in $\mathbf{High}_2$.
- Dann wäre aber $\mathbf{Low}_2 \cap \mathbf{High}_2 \neq \emptyset$, was wiederum den Zusammenbruch der polynomiellen Hierarchie auf Σ_2 nach sich zöge! (*considered unlikely...*)

Graphenisomorphie - Meilensteine

Satz

GI ist in **NP**

Graphenisomorphie - Meilensteine

Satz

GI ist in **NP**

Satz

GNI ist in **AM**

Graphenisomorphie - Meilensteine

Satz

GI ist in **NP**

Satz

GNI ist in **AM**

Satz

$\mathbf{NP} \cap \mathbf{coAM} \subseteq \mathbf{Low}_2$

Nächstes Mal...

Nächstes Mal:

- ?
- ?
- ?

Ende...

Spannede Bücher, in die ihr mal gucken könnt:

- Computational Complexity, Papadimitriou, Addison-Wesley 1994
- Complexity Theory and Cryptology, Jörg Rothe, Springer, 2005 (dt. 2008)
- Complexity Theory Companion, Hemaspaandra, Springer 2002

Ende...

Danke für die Aufmerksamkeit
!