

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:

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

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

Die parallele Query Hierarchie

Definition (Parallele Query Hierarchie)

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

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

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

Die parallele Query Hierarchie

Dabei bedeutet 'in parallel', dass die Orakelfragen gleichzeitig gestellt werden. Anders als bisher wird also ein Tupel (q_1, \dots, q_n) von Orakelfragen an das Orakel als Frage gestellt. Das Orakel B antwortet dann mit einem Tupel (x_1, \dots, 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 $\mathbf{P}_{k\text{-tt}}^{\text{NP}} = \mathbf{P}^{A[1]}$ für eine \mathbf{BH}_k -vollständige Menge A
(bzgl. \leq_m^P -Reduktion).
- ② $\mathbf{P}^{\text{NP}[k]} = \mathbf{P}_{(2^k-1)\text{-tt}}^{\text{NP}}$.
- ③ $\mathbf{BH}_k \cup \mathbf{coBH}_k \subseteq \mathbf{P}_{k\text{-tt}}^{\text{NP}} \subseteq \mathbf{BH}_{k+1} \cap \mathbf{coBH}_{k+1}$.
- ④ $\mathbf{BH}_{2^k-1} \cup \mathbf{coBH}_{2^k-1} \subseteq \mathbf{P}^{\text{NP}[k]} \subseteq \mathbf{BH}_{2^k} \cap \mathbf{coBH}_{2^k}$.

Verzahnung mit der BH

Satz

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

Satz

$$\mathbf{BH}(\mathbf{NP}) \subseteq \mathbf{P}^{\text{NP}[\log]} = \mathbf{P}_{\text{tt}}^{\text{NP}} = \Theta_2$$

Verfeinerung des SvK

Satz

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

Satz

- ① Gilt $\mathbf{P}_{k\text{-tt}}^{\Sigma_i} = \mathbf{P}_{(k+1)\text{-tt}}^{\Sigma_i}$ 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 \mathbf{NP} .

Diese werden mittels der polynomiellen Hierarchie definiert und es uns erlauben \mathbf{NP} (!) genauer zu untersuchen!

Die Low Hierarchy

Definition

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

$$\mathbf{Low}_k = \{L \mid L \in \mathbf{NP} \text{ und } \Sigma_k^L \subseteq \Sigma_k\} \text{ f\u00fcr jedes } k \geq 0$$

$$\mathbf{LH} = \bigcup_{k \geq 0} \mathbf{Low}_k$$

Die High Hierarchy

Definition

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

$$\mathbf{High}_k = \{H \mid H \in \mathbf{NP} \text{ und } \Sigma_{k+1} \subseteq \Sigma_k^H\} \text{ f\u00fcr jedes } k \geq 0$$

$$\mathbf{HH} = \bigcup_{k \geq 0} \mathbf{High}_k$$

Charakterisierung der unteren Stufen

Satz

- ① $\mathbf{Low}_0 = \mathbf{P}$
- ② $\mathbf{Low}_1 = \mathbf{NP} \cap \mathbf{coNP}$
- ③ $\mathbf{High}_0 = \{H \mid H \text{ ist } \leq_T^p\text{-vollst\u00e4ndig f\u00fcr } \mathbf{NP}\}$
- ④ $\mathbf{High}_1 = \{H \mid H \text{ ist } \leq_T^{np}\text{-vollst\u00e4ndig f\u00fcr } \mathbf{NP}\}$

Einige Ergebnisse I

Satz

- ① $\mathbf{Low}_0 \subseteq \mathbf{Low}_1 \subseteq \dots \subseteq \mathbf{Low}_k \subseteq \dots \subseteq \mathbf{LH} \subseteq \mathbf{NP}$
- ② $\mathbf{High}_0 \subseteq \mathbf{High}_1 \subseteq \dots \subseteq \mathbf{High}_k \subseteq \dots \subseteq \mathbf{HH} \subseteq \mathbf{NP}$
- ③ $\mathbf{Low}_k \cap \mathbf{High}_k \neq \emptyset \iff \Sigma_k = \Sigma_{k+1} = \dots = \mathbf{PH}$
- 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 $L \in \text{NP} \setminus (\text{Low}_k \cup \text{High}_k) \iff \Sigma_k \neq \Sigma_{k+1}$
- ② $L \in \text{NP} \setminus (\text{LH} \cup \text{HH}) \iff \text{die PH ist strictly infinite}$

Graphenisomorphie II

Warum?

- Wäre $GI \in \text{NPC}$, dann wäre es insbesondere in High_0 .
- Dann wäre GI auch in High_2 .
- Dann wäre aber $\text{Low}_2 \cap \text{High}_2 \neq \emptyset$, was wiederum den Zusammenbruch der polynomiellen Hierarchie auf Σ_2 nach sich zöge! (*considered unlikely...*)

Graphenisomorphie I

- Man kann zeigen, dass das Graphenisomorphieproblem GI in Low_2 ist. (Nutzt eine weitere Hierarchie.)
- Dies gilt als starker (!) Anhaltspunkt, dass GI *nicht* NP -vollständig sein kann.

Graphenisomorphie - Meilensteine

Satz

 GI ist in NP

Satz

 GNI ist in AM

Satz

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

Nächstes Mal...

Nächstes Mal:

- ?
- ?
- ?

Ende...

Danke für die Aufmerksamkeit
!

Ende...

Spannende 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