The Boolean Hierarchy over **NP** (Part 2) and *Kadin's Theorem*

Frank Heitmann
heitmann@informatik.uni-hamburg.de

Gastvortrag im Rahmen der Vorlesung
Komplexitätstheorie
von
Prof. Dr. Matthias Jantzen
30.06.2008

Zur Wiederholung...

Das Färbungsproblem für Graphen:

- $\{(G,k) \mid \chi(G) \leq k\} \in \mathbf{NP}$
- $\{(G,k) \mid \chi(G) \geq k\} \in \mathbf{coNP}$
- $\{(G,k) \mid \chi(G) = k\} \in \mathbf{NP} \wedge \mathbf{coNP}$

Zur Wiederholung...

Anmerkung I

Durch solche *exakten* Varianten von **NP**-vollständigen Optimierungsproblemen motiviert, führten Papadimitriou und Yannakakis 1982 die Klasse **DP** ein.

Anmerkung II

- Exakte Probleme $\Longrightarrow$ `Exact-`i`-Colorability`, `Exact-`i`-DNP`
- Kritische Probleme $\Longrightarrow$ `Minimal-3-Uncolorability`
- Unique Solution Problems $\Longrightarrow$ `Unique-SAT`

Die Klasse **DP**

Definition (DP)

$\mathbf{DP} = \mathbf{NP} \wedge \mathbf{coNP} = \{L \mid L = L_1 \cap L_2, L_1 \in \mathbf{NP}, L_2 \in \mathbf{coNP}\}$

DP ist

- der Abschluss von $\mathbf{NP} \cup \mathbf{coNP}$ gegenüber Schnittbildung;
- vermutlich nicht gegenüber Vereinigung abgeschlossen;
- vermutlich nicht gegenüber Komplement abgeschlossen.

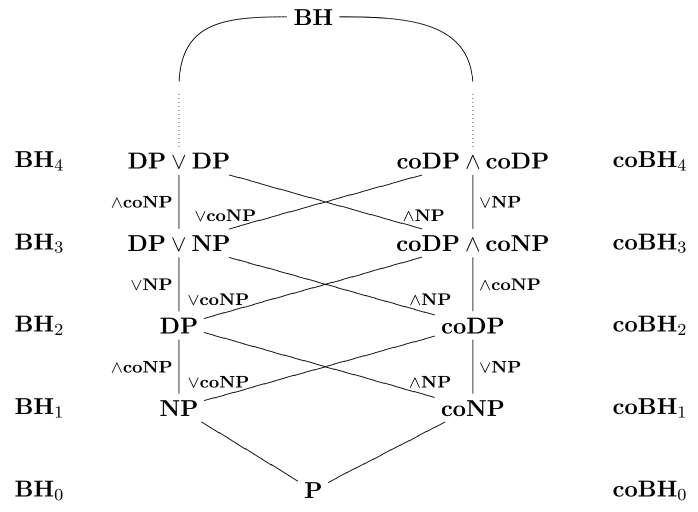
Es lässt sich daher fortfahren...

Die Boolesche Hierarchie

Definition (Die Boolesche Hierarchie)

Die *Boolesche Hierarchie* wird induktiv definiert durch:

$$\begin{aligned}
\mathbf{BH}_0 &= \mathbf{P}, \\
\mathbf{BH}_1 &= \mathbf{NP}, \\
\mathbf{BH}_k &= \begin{cases} \mathbf{BH}_{k-1} \wedge \mathbf{coNP} & \text{falls } k \text{ gerade} \\ \mathbf{BH}_{k-1} \vee \mathbf{NP} & \text{falls } k \text{ ungerade} \end{cases} \quad \text{für } k \geq 2, \\
\mathbf{BH} &= \bigcup_{k \geq 0} \mathbf{BH}_k.
\end{aligned}$$



Die Boolesche Hierarchie - Alternative Darstellungen

Definition (Boolesche Hierarchie 2)

Die *Hierarchie verschachtelter Differenzen* wird induktiv definiert durch:

$$\begin{aligned}
\mathbf{BH}'_0 &= \mathbf{P}, \\
\mathbf{BH}'_k &= \left\{ L \;\middle|\; \begin{array}{l} L = A_1 - A_2 - \ldots - A_k \text{ für Mengen } A_i \in \mathbf{NP} \\ 1 \leq i \leq k, \text{ mit } A_k \subseteq A_{k-1} \subseteq \ldots \subseteq A_1 \end{array} \right\}, \\
\mathbf{BH}' &= \bigcup_{k \geq 0} \mathbf{BH}'_k.
\end{aligned}$$

Die Boolesche Hierarchie - Alternative Darstellungen

Definition (Boolesche Hierarchie 3)

$$\begin{aligned}
\mathbf{BH}''_0 &= \mathbf{P} \\
\mathbf{BH}''_1 &= \mathbf{NP} \\
\mathbf{BH}''_2 &= \mathbf{DP} \\
\mathbf{BH}''_k &= \begin{cases} \mathbf{BH}''_{k-1} \vee \mathbf{NP} & \text{falls } k \text{ ungerade} \\ \mathbf{BH}''_{k-2} \vee \mathbf{DP} & \text{falls } k \text{ gerade} \end{cases} \quad \text{für } k \geq 3 \\
\mathbf{BH}'' &= \bigcup_{k \geq 0} \mathbf{BH}''_k
\end{aligned}$$

Die Boolesche Hierarchie - Alternative Darstellungen

Definition (Boolescher Abschluss)

Sei $\mathcal{C}$ eine beliebige Klasse von Mengen. Der *Boolesche Abschluss*, $\mathbf{BC}(\mathcal{C})$, von $\mathcal{C}$ ist die kleinste Klasse $\mathcal{B}$ von Mengen, die $\mathcal{C}$ enthält und gegenüber den folgenden Booleschen Operationen abgeschlossen ist:

1. $\mathcal{B}$ ist gegenüber Vereinigung abgeschlossen, d.h. mit $A, B \in \mathcal{B}$ ist auch $A \cup B \in \mathcal{B}$;
2. $\mathcal{B}$ ist gegenüber Durchschnitt abgeschlossen, d.h. mit $A, B \in \mathcal{B}$ ist auch $A \cap B \in \mathcal{B}$;
3. $\mathcal{B}$ ist gegenüber Komplementbildung abgeschlossen, d.h. mit $A \in \mathcal{B}$ ist auch $\overline{A} \in \mathcal{B}$.

Die Boolesche Hierarchie - Alternative Darstellungen

Satz (Die BH - Äquivalenzen)

1. $\mathbf{BH}_k = \mathbf{BH}'_k = \mathbf{BH}''_k$ *für alle* $k \geq 0$
2. $\mathbf{BH} = \mathbf{BH}' = \mathbf{BH}''$
3. $\mathbf{BH} = \mathbf{BC}(\mathbf{NP})$

Anmerkung

(i) $L \in \mathbf{BH}_k$

(ii) $L = (\dots(((L_1 - L_2) + L_3) - L_4)\dots)(-1)^{k-1}L_k$ für Mengen $L_1, \dots, L_k \in \mathbf{NP}$

(iii) $L = (\dots(((L_1 - L_2) + L_3) - L_4)\dots)(-1)^{k-1}L_k$ für Mengen $L_1, \dots, L_k \in \mathbf{NP}$ mit $L_k \subseteq L_{k-1} \subseteq \dots \subseteq L_1$

(iv) $L \in \mathbf{BH}'_k$

(v) $L \in \mathbf{BH}''_k$

Zusammenbruch

Satz (Zusammenbruch der Booleschen Hierarchie)

1. *Gilt* $\mathbf{BH}_k = \mathbf{BH}_{k+1}$ *für ein* $k \geq 0$*, so folgt* $\mathbf{BH}_k = \mathbf{coBH}_k = \mathbf{BH}_{k+1} = \mathbf{coBH}_{k+1} = \dots = \mathbf{BH}$.
2. *Gilt* $\mathbf{BH}_k = \mathbf{coBH}_k$ *für ein* $k \geq 1$*, so folgt* $\mathbf{BH}_k = \mathbf{coBH}_k = \mathbf{BH}_{k+1} = \mathbf{coBH}_{k+1} = \dots = \mathbf{BH}$.

Verallgemeinerung von SAT

Als Verallgemeinerung von SAT erhält man vollständige Probleme für jede Stufe der Booleschen Hierarchie:

$$\begin{aligned} L_{\mathbf{BH}_2} &= \{(F_1, F_2) \mid F_1 \in \mathtt{SAT} \text{ und } F_2 \in \overline{\mathtt{SAT}}\} \\ L_{\mathbf{BH}_3} &= \{(F_1, F_2, F_3) \mid (F_1 \in \mathtt{SAT} \text{ und } F_2 \in \overline{\mathtt{SAT}}) \text{ oder } F_3 \in \mathtt{SAT}\} \\ &\dots \end{aligned}$$

$$\begin{aligned} \mathbf{BH}_1 &= \mathbf{NP} \\ \mathbf{BH}_2 &= \mathbf{NP} \wedge \mathbf{coNP} \\ \mathbf{BH}_3 &= (\mathbf{NP} \wedge \mathbf{coNP}) \vee \mathbf{NP} \end{aligned}$$

Verallgemeinerung von $\overline{\mathrm{SAT}}$

Ebenso definiert man vollständige Probleme für $\mathbf{coBH}_k$:

$$\begin{aligned} L_{\mathbf{coBH}_2} &= \{(F_1, F_2) \mid F_1 \in \overline{\mathtt{SAT}} \text{ oder } F_2 \in \mathtt{SAT}\} \\ L_{\mathbf{coBH}_3} &= \{(F_1, F_2, F_3) \mid (F_1 \in \overline{\mathtt{SAT}} \text{ oder } F_2 \in \mathtt{SAT}) \text{ und } F_3 \in \overline{\mathtt{SAT}}\} \\ &\ldots \end{aligned}$$

$$\begin{aligned} L_{\mathbf{BH}_2} &= \{(F_1, F_2) \mid F_1 \in \mathtt{SAT} \text{ und } F_2 \in \overline{\mathtt{SAT}}\} \\ L_{\mathbf{BH}_3} &= \{(F_1, F_2, F_3) \mid (F_1 \in \mathtt{SAT} \text{ und } F_2 \in \overline{\mathtt{SAT}}) \text{ oder } F_3 \in \mathtt{SAT}\} \\ &\ldots \end{aligned}$$

Mehr vollständige Probleme

Satz

Für jedes $k \geq 1$ gilt

1. $L_{\mathbf{BH}_k}$ *ist* $\mathbf{BH}_k$*-vollständig und*
2. $L_{\mathbf{coBH}_k}$ *ist* $\mathbf{coBH}_k$*-vollständig.*

Beweis.

Der Beweis von 1. ist induktiv und verläuft ansonsten völlig analog zu dem vorherigen Beweis. Anschließend folgt 2. dann direkt aus 1. durch Komplementbildung. □

Anmerkung

Mit den Problemen $L_{\mathbf{BH}_k}$ ist $\mathbf{BH} \subseteq \mathbf{P}^{\mathbf{NP}}$ klar.

Das Lemma von Wagner

Satz (Wagner)

Sei A eine **NP***-vollständige Menge, B eine beliebige Menge und $k \geq 1$ fest. Existiert eine in Polynomialzeit berechenbare Funktion f derart, dass für alle $v_1, v_2, \ldots, v_k \in \Sigma^*$ mit $v_{j+1} \in A \Rightarrow v_j \in A$*

$$||\{i \mid v_i \in A\}|| \text{ ist ungerade} \Leftrightarrow f(v_1, \ldots, v_k) \in B$$

erfüllt ist, so ist B $\mathbf{BH}_k$*-schwierig.*

Anmerkung

$v_{j+1} \in A \Rightarrow v_j \in A$ ist keine weitere (störende) Vorbedingung, sondern eine *Einschränkung* der Vorbedingung!

Wagners Lemma an anderer Stelle

Satz (Wagner)

Sei A eine **NP***-vollständige Menge und B eine beliebige Menge. Existiert eine in Polynomialzeit berechenbare Funktion f derart, dass für alle $k \geq 1$ und alle $x_1, x_2, \ldots, x_{2k} \in \Sigma^*$ mit $\chi_A(x_1) \geq \chi_A(x_2) \geq \ldots \geq \chi_A(x_{2k})$*

$$||\{i \mid x_i \in A\}|| \text{ ist ungerade} \Leftrightarrow f((x_1, \ldots, x_{2k})) \in B$$

erfüllt ist, so ist B $\mathbf{\Theta}_2$*-schwierig.*

Vollständigkeit in der BH

Mit dem Lemma von Wagner wollen wir nun ein komplizierteres Problem als Vollständig für die Boolsche Hierarchie nachweisen. Dazu

- Definieren wir das *domatic number problem*.
- Zeigen, dass dieses **NP**-vollständig ist.
- Definieren die exakten Varianten.
- Zeigen, dass dieses **DP**- bzw. $\mathbf{BH}_k$-vollständig ist.

Defintion des DNP

Definition (Domatic Number Problem)

Sei G ein ungerichteter Graph. Ein *dominating set* von G ist eine Teilmenge $D \subseteq V(G)$ derart, dass zu jedem Knoten $u \in V(G) \setminus D$ ein Knoten $v \in D$ existiert mit $\{v, u\} \in E$. Die maximale Anzahl an disjunkten dominating sets wird mit $\delta(G)$ bezeichnet.
Die Entscheidungsvariante des *domatic number problems* wird definiert durch:

$$\mathtt{DNP} = \{(G, k) \mid G \text{ ist ein Graph mit } \delta(G) \geq k\}$$

Anmerkung

Es gilt stets $\delta(G) \leq min - deg(G) + 1$.

$\mathtt{DNP} \in$ NPC - Ziel

Wir wollen nun $\mathtt{DNP}$ als **NP**-vollständig nachweisen...

$$\mathtt{3\text{-}Colorability} \leq_m^p \mathtt{DNP}$$

Dazu geben wir ein $f \in \mathbf{FP}$ an mit $f(G) = (H, 3)$, wobei

$$G \in \mathtt{3\text{-}Colorability} \implies \delta(H) = 3$$
$$G \notin \mathtt{3\text{-}Colorability} \implies \delta(H) = 2$$

Anmerkung

- $\chi(G) \geq 3$
- G hat keine isolierten Knoten (impliziert $\delta(G) \geq 2$)

$\mathtt{DNP} \in$ NPC - Konstruktion

H entsteht aus G indem

- auf jede Kante von G ein neuer Knoten gesetzt wird und
- alle Knoten in G in H verbunden werden (Clique)

Anmerkung

- Jede Kante in G induziert ein Dreieck in H.
- Jedes Paar in G nicht verbundener Knoten ist in H verbunden.

DNP $\in$ **NPC** - Konstruktion (formal)

Sei $V(G) = \{v_1, \dots, v_n\}$. Nun sei

$$\begin{aligned} V(H) &= V(G) \cup \{u_{i,j} \mid \{v_i, v_j\} \in E(G)\} \\ E(H) &= \{\{v_i, u_{i,j}\}, \{u_{i,j}, v_j\} \mid \{v_i, v_j\} \in E(G) \cup \\ &\quad \{\{v_i, v_j\} \mid 1 \leq i, j \leq n, i \neq j\} \end{aligned}$$

Anmerkung

Wegen $min - deg(H) = 2$ und da H keine isolierten Knoten hat, gilt

$$2 \leq \delta(H) \leq 3$$

DNP $\in$ **NPC**

$G \in$ 3-Colorability $\Longrightarrow \delta(H) = 3$

- Seien $C_k = \{v_i \mid v_i \in V(G)$ ist mit k gefärbt$\}$, $k = 1, 2, 3$
- Seien nun $C_k' = C_k \cup \{u_{i,j} \mid \{v_i, v_j\} \in E(G) \wedge v_i, v_j \notin C_k\}$
- Wegen $C_k' \cap V(G) \neq \emptyset$ für jedes k und da $V(G)$ in H eine Clique induziert, dominiert jedes C_k' $V(G)$ in H.
- Jedes Dreieck $\{v_i, u_{i,j}, v_j\}$ in H enthält ein Element aus jedem C_k', also dominiert jedes C_k' auch $\{u_{i,j} \mid \{v_i, v_j\} \in E(G)\}$.
- Also: $\delta(H) = 3$.

DNP $\in$ **NPC**

$G \in$ 3-Colorability $\Longleftarrow \delta(H) = 3$

- Sei C_1', C_2', C_3' eine Partition von H in drei dominating sets
- Färbe C_k' mit Farbe k
- Jedes Dreieck $\{v_i, u_{i,j}, v_j\}$ ist jetzt dreigefärbt
- Diese Färbung von $V(H)$ induziert eine legale Färbung von G
- Wegen $2 \leq \delta(H) \leq 3$ folgt damit dann auch $G \notin$ 3-Colorability $\Longrightarrow \delta(H) = 2$

Die exakte Variante

Definition (Exact Domatic Number Problem)

Sei $i \in \mathbb{N}$ fest. Das *exact domatic number problem* wird definiert durch:

$$\texttt{Exact-}i\texttt{-DNP} = \{(G, k) \mid G \text{ ist ein Graph mit } \delta(G) = k\}$$

Definition (Exact Domatic Number Problem (Generalized Version))

Sei $M_k \subseteq \mathbb{N}$ eine Menge von k Zahlen. Das *allgemeine exact domatic number problem* wird definiert durch:

$$\texttt{Exact-}M_k\texttt{-DNP} = \{G \mid G \text{ ist ein Graph mit } \delta(G) \in M_k\}$$

Das Ziel...

Wir wollen nun zeigen:

Satz

Sei $k \geq 1$ fest gewählt. Sei $M_k = \{4k+1, 4k+3, \ldots, 6k-1\}$. Dann ist `Exact-`$M_k$`-DNP` ***BH***$_{2k}$*-vollständig.*

Satz

`Exact-5-DNP` *ist* ***DP****-vollständig.*

Exact-M_k-DNP $\in$ **BHC**$_{2k}$ - Vorarbeiten

Wir werden Wagners Lemma nutzen:

Satz (Wagner)

Sei A eine ***NP****-vollständige Menge, B eine beliebige Menge und $k \geq 1$ fest. Existiert eine in Polynomialzeit berechenbare Funktion f derart, dass für alle $v_1, v_2, \ldots, v_k \in \Sigma^*$ mit $v_{j+1} \in A \Rightarrow v_j \in A$*

$$||\{i \mid v_i \in A\}|| \text{ ist ungerade} \Leftrightarrow f(v_1, \ldots, v_k) \in B$$

erfüllt ist, so ist B ***BH***$_k$*-schwierig.*

Exact-M_k-DNP $\in$ **BHC**$_{2k}$ - Vorarbeiten

- `Exact-`M_k`-DNP` $\in$ **BH**$_{2k}$ ist klar (Definition 3 der BH)
- Dabei ist k fest, $M_k = \{4k+1, 4k+3, \ldots, 6k-1\}$
- `3-Colorability` ist A, `Exact-`M_k`-DNP` ist B
- Seien also $G_1, G_2, \ldots, G_{2k}$ Graphen mit

$$G_{i+1} \in \texttt{3-Colorability} \Rightarrow G_i \in \texttt{3-Colorability}$$

- Annahmen (wegen `3-SAT` $\leq_m^p$ `3-Colorability` ok):
  - $3 \leq \chi(G_i) \leq 4$
  - Die G_i haben keine isolierten Knoten

Exact-M_k-DNP $\in$ **BHC**$_{2k}$ - Vorarbeiten

- Sei $g \in$ **FP** mit $g(G) = (H, 3)$ aus dem Beweis der **NP**-vollständigkeit von `DNP`, wobei

$$\begin{aligned} G \in \texttt{3-Colorability} &\implies \delta(H) = 3 \\ G \notin \texttt{3-Colorability} &\implies \delta(H) = 2 \end{aligned}$$

- Sei nun $H_j = g(G_j)$, $1 \leq j \leq 2k$, so gilt
  - $\delta(H_j) \in \{2, 3\}$
  - $\delta(H_{j+1}) = 3 \Rightarrow \delta(H_j) = 3$

$\texttt{Exact-}M_k\texttt{-DNP} \in \mathbf{BHC}_{2k}$ - Ziel

Gesucht ist nun ein $f \in \mathbf{FP}$ derart, dass

$$||\{i \mid G_i \in \texttt{3-Colorability}\}|| \text{ ist ungerade} \iff f(G_1, \ldots, G_{2k}) = H \in \texttt{Exact-}M_k\texttt{-DNP}$$

erfüllt ist.
Dazu wird ein Graph H aus $H_1, \ldots, H_{2k}$ konstruiert mit

$$\delta(H) = \sum_{j=1}^{2k} \delta(H_j)$$

$\texttt{Exact-}M_k\texttt{-DNP} \in \mathbf{BHC}_{2k}$ - Der Fall $k = 1$

Sei zunächst $k = 1$, $H_1 = g(G_1)$, $H_2 = g(G_2)$. Wir konstruieren folgendes 'Gadget', um H_1 und H_2 so zu H zu verbinden, dass $\delta(H) = \delta(H_1) + \delta(H_2)$ gilt:

- Sei $V(T_1) = \{v_q^{T_1}, u_{q,r}^{T_1}, v_r^{T_1}\}$ ein fest gewähltes Dreieck in H_1 und
- sei $V(T_2) = \{v_s^{T_2}, u_{s,t}^{T_2}, v_t^{T_2}\}$ ein fest gewähltes Dreieck in H_2
- Füge sechs neue Knoten $a_1^{T_1,T_2}, a_2^{T_1,T_2}, \ldots, a_6^{T_1,T_2}$ und neue Kanten ein so, *wie Frank das jetzt anmalt...*
- Verbinde *jedes* Paar von Dreiecken aus H_1 und H_2 durch solche Gadgets. Der resultierende Graph ist der Graph H. Beachte: $\delta(H) \leq 6$.

$\texttt{Exact-}M_k\texttt{-DNP} \in \mathbf{BHC}_{2k}$ - Der Fall $k = 1$

Es gibt nun drei Fälle:

- $\delta(H_1) = 3$ und $\delta(H_2) = 3$
- $\delta(H_1) = 3$ und $\delta(H_2) = 2$
- $\delta(H_1) = 2$ und $\delta(H_2) = 2$
- ($\delta(H_1) = 2$ und $\delta(H_2) = 3$ kann nicht auftreten.)

Seien $D_1, \ldots, D_{\delta(H_1)}$ die disjunkten dominierenden Mengen von H_1 und $D_{\delta(H_1)+1}, \ldots, D_{\delta(H_1)+\delta(H_2)}$ jene von H_2. Zu zeigen ist nun stets $\delta(H) = \delta(H_1) + \delta(H_2)$, wobei $\delta(H) \leq 6$ gilt.

$\delta(H_1) = 3$ und $\delta(H_2) = 3$

- Sei D_j, $1 \leq j \leq 3$ fest. Jedes Dreieck in H_1 hat genau einen Knoten in D_j.
- Sei T_1 mit $V(T_1) = \{v_q^{T_1}, u_{q,r}^{T_1}, v_r^{T_1}\}$ fest gewählt (!).
- Sei $V(T_1) \cap D_j = \{v_q^{T_1}\}$ (andere Fälle analog)
- Genau *einer* der a_i ist nun *nicht* mit $v_q^{T_1}$ verbunden. Dieser Knoten dominiert das zugehörige Dreieck T_2 in H_2.
- Sei $D_j' = D_j \cup \{a_3^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\}$

$\delta(H_1) = 3$ und $\delta(H_2) = 3$

$D'_j = D_j \cup \{a_3^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\}$

- D'_j enthält D_j und dominiert also H_1.
- D'_j dominiert wegen der $a_3^{T_1,T_2}$ auch H_2.
- Da $v_q^{T_1,T_2} \in D_j$ mit jedem $a_i^{T_1,T_2}$, $i \neq 3$ benachbart ist, dominiert D'_j auch die Gadget-Knoten.

$\Rightarrow$ D'_j dominiert H.

$\delta(H_1) = 3$ und $\delta(H_2) = 3$

Die Argumentation verläuft analog für die D_j mit $4 \leq j \leq 6$. Nach der Konstruktion sind die D'_j disjunkt, so dass insgesamt $\delta(H) = 6 = \delta(H_1) + \delta(H_2)$ folgt.

$\delta(H_1) = 3$ und $\delta(H_2) = 2$

- Wie eben können zunächst die D'_i aus den D_i gewonnen werden. Dies zeigt $5 \leq \delta(H) \leq 6$ und es muss nur noch $\delta(H) = 6$ ausgeschlossen werden.
- Widerspruchsbewies: Angenommen $\delta(H) = 6$
- Sei wieder T_1 mit $V(T_1) = \{v_q^{T_1}, u_{q,r}^{T_1}, v_r^{T_1}\}$ fest gewählt. H kann jetzt nur auf eine Weise in 6 dominating sets unterteilt werden:

$\delta(H_1) = 3$ und $\delta(H_2) = 2$

- $\{v_q^{T_1}\} \cup \{a_3^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\} \subseteq E_1$
- $\{u_{q,r}^{T_1}\} \cup \{a_2^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\} \subseteq E_2$
- $\{v_r^{T_1}\} \cup \{a_1^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\} \subseteq E_3$
- $\{v_s^{T_2}, a_6^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\} \subseteq E_4$
- $\{u_{s,t}^{T_2}, a_5^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\} \subseteq E_5$
- $\{v_t^{T_2}, a_4^{T_1,T_2} \mid T_2 \text{ ist ein Dreieck in } H_2\} \subseteq E_6$

$\delta(H_1) = 3$ und $\delta(H_2) = 2$

- Nun aber sind alle Knoten von H_2 auf die Mengen E_4, E_5 und E_6 aufgeteilt.

$\Rightarrow$ Partition von H_2 in drei dominating sets

$\Rightarrow$ Widerspruch zu $\delta(H_2) = 2 \neq 3$.

$\Rightarrow$ Also $\delta(H) = 5 = \delta(H_1) + \delta(H_2)$

$\delta(H_1) = 2$ und $\delta(H_2) = 2$

Ähnlich wie die vorherigen beiden, wobei hier nur ein Widerspruch zu $\delta(H) = 5$ geführt werden muss (der Rest ist analog zu obigem). Auch hier wird dann angenommen es gäbe 5 disjunkte dominating sets und dies zum Widerspruch geführt. Der Trick ist zu zeigen, dass weder T_1 noch T_2 zwei Knoten im gleichen dominating set haben können. (Es wird dabei ein Widerspruch zu $\delta(H_2) = 2$ erzeugt.)

$\texttt{Exact-}M_k\texttt{-DNP} \in \mathbf{BHC}_{2k}$ - Der Fall $k = 1$

Sei nun $f(G_1, G_2) = H$. Es gilt $f \in \mathbf{FP}$ und

$$\begin{aligned} & G_1 \in \texttt{3-Colorability} \wedge G_2 \notin \texttt{3-Colorability} \\ \Leftrightarrow \quad & \delta(H_1) = 3 \wedge \delta(H_2) = 2 \\ \Leftrightarrow \quad & \delta(H) = \delta(H_1) + \delta(H_2) = 5 \\ \Leftrightarrow \quad & f(G_1, G_2) = H \in \texttt{Exact-5-DNP} \end{aligned}$$

Satz

$\texttt{Exact-5-DNP}$ *ist* **DP**-*vollständig.*

$\texttt{Exact-}M_k\texttt{-DNP} \in \mathbf{BHC}_{2k}$ - Der allgemeine Fall

- Gegeben sind die $2k$ Graphen $H_1, \dots, H_{2k}$.
- Sei $T_1, \dots, T_{2k}$ eine feste Sequenz von Dreiecken (T_i aus H_i).
- Füge $a_1, \dots, a_{6k}$ hinzu und assoziere $a_{1+3(i-1)}, a_{2+3(i-1)}$ und a_{3i} mit dem Dreieck T_i.
- Für jedes i, $1 \leq i \leq 2k$ verbinde T_i mit *jedem* T_j, $1 \leq j \leq 2k$, $i \neq j$ mittels der mit T_i assozierten Knoten und zwar genau so, wie T_1 mit T_2 mittels der Knoten a_1, a_2, a_3 verbunden wurde.
- Es ist $d(a_i) \leq 6k - 1$ und darum $\delta(H) \leq 6k$.
- Ähnlich wie oben zeigt man $\delta(H) = \sum_{j=1}^{2k} \delta(H_j)$.

Exact-M_k-DNP $\in$ **BHC**$_{2k}$ - Der allgemeine Fall

$$
\begin{aligned}
& \|\{i \mid G_i \in \texttt{3-Colorability}\}\| \text{ ist ungerade} \\
\Leftrightarrow\ & \exists i : \chi(G_1) = \ldots = \chi(G_{2i-1}) = 3 \text{ und} \\
& \chi(G_{2i}) = \ldots = \chi(G_{2k}) = 4 \\
\Leftrightarrow\ & \exists i : \delta(H_1) = \ldots = \delta(H_{2i-1}) = 3 \text{ und} \\
& \delta(H_{2i}) = \ldots = \delta(H_{2k}) = 2 \\
\Leftrightarrow\ & \exists i : \delta(H) = \sum_{j=1}^{2j} \delta(H_j) = 3(2i-1) + 2(2k-2i+1) \\
\Leftrightarrow\ & \exists i : \delta(H) = 4k + 2i - 1 \\
\Leftrightarrow\ & \delta(H) \in \{4k+1, 4k+3, \ldots, 6k-1\} \\
\Leftrightarrow\ & f(G_1, \ldots, G_{2k}) = H \in \texttt{Exact-}M_k\texttt{-DNP}
\end{aligned}
$$

Exact-M_k-DNP $\in$ **BHC**$_{2k}$ - Der allgemeine Fall

- Mit dem Lemma von Wagner folgt dann, dass Exact-M_k-DNP Vollständig für **BH**$_{2k}$ ist.
- Wir wissen nun insbesondere, dass Exact-5-DNP **DP**-vollständig ist.
- Ferner ist Exact-2-DNP $\in$ **coNP** bekannt.
- Die genaue Komplexität von Exact-3-DNP und Exact-4-DNP ist *unbekannt*!
- Beide sind **coNP**-schwierig und in **DP** enthalten ....

Exakte Färbbarkeit

Für das exakte Färbungsproblem ist die Sachlage klarer (aber noch nicht lange):

- Exact-3-Colorability $\in$ **NP**
- Exact-4-Colorability ist **DP**-vollständig
- Der Beweis ist sehr schwer (kompliziertes Gadget)

Infinite Towers collapsing... or not?!

And now to something different...

Der Satz von Kadin

Satz (Kadin, 1988)

Wenn ein $k \geq 1$ mit $\mathbf{BH}_k = \mathbf{coBH}_k$ existiert, dann bricht die polynomielle Hierarchie zusammen: $\mathbf{PH} \subseteq \Sigma_3$.

Anmerkung

Der Satz von Kadin gilt als starker Anhaltspunkt, dass die Boolesche Hierarchie **BH** nicht zusammenbricht.

Die Polynomielle Hierarchie (Wdh.)

Definition (Polynomielle Hierarchie)

Die *polynomielle Hierarchie* wird induktiv definiert:

- $\Delta_0 = \Sigma_0 = \Pi_0 = \mathbf{P}$
- Für $i \geq 0$ sei

$$\Delta_{i+1} = \mathbf{P}^{\Sigma_i}, \Sigma_{i+1} = \mathbf{NP}^{\Sigma_i} \text{ und } \Pi_{i+1} = \mathbf{co}\Sigma_{i+1}$$

- $\mathbf{PH} = \bigcup_{k \geq 0} \Sigma_k$

Teilmengenbeziehungen (Wdh.)

Satz (Teilmengenbeziehungen)

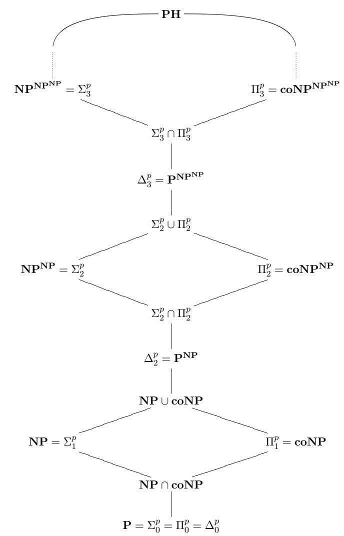
1. *Für jedes $i \geq 0$ gilt:*

$$\Sigma_i \cup \Pi_i \subseteq \Delta_{i+1} \subseteq \Sigma_{i+1} \cap \Pi_{i+1}$$

2. *Für jedes $i \geq 0$ gilt:*

$$\Sigma_i \subseteq \Pi_{i+1} \text{ und } \Pi_i \subseteq \Sigma_{i+1}$$

3. $\mathbf{PH} \subseteq \mathbf{PSPACE}$



Zusammenbruch (Wdh.)

Satz (Zusammenbruch der polynomiellen Hierarchie)

1. *Gilt* $\Sigma_i = \Sigma_{i+1}$ *für ein* $i \geq 0$*, so folgt* $\Sigma_i = \Pi_i = \Delta_{i+1} = \Sigma_{i+1} = \Pi_{i+1} = \ldots = \mathbf{PH}$.
2. *Gilt* $\Sigma_i = \Pi_i$ *für ein* $i \geq 1$*, so folgt* $\Sigma_i = \Pi_i = \Delta_{i+1} = \Sigma_{i+1} = \Pi_{i+1} = \ldots = \mathbf{PH}$.

Präfixsuche - Die Idee

Ein wichtiges Verfahren, das mit Orakelmaschinen realisiert werden kann, ist die Präfixsuche.

- Speichern eines Wortes *und* dessen Präfixe im Orakel.
- Sukzessives Auslesen/Aufbauen des Wortes.
- Beispiel: Präfixsuche für das Wort 11001?
- Was passiert, wenn auch das Wort 1001 im Orakel ist?

$\Rightarrow$ Darum wird mit *markierten* Präfixen gearbeitet!

$\Rightarrow$ Länge des interessierenden Wortes muss bekannt sein

$\Rightarrow$ Es wird stets (nur) das lexikographisch kleinste Wort ermittelt

Präfixsuche

Präfixsuche

```
if 2^n ∉ S then
    return ε
end if
F ← ε
for i = 1 to n do
    if F02^(n−i) ∈ S then
        F ← F0
    else
        F ← F1
    end if
end for
return F
```


Reduktionen

Definition (Reduktionen)

- Die *Many-One-Reduktion* $\leq_m^p$ wird definiert durch $A \leq_m^p B$ genau dann, wenn $\exists f \in \mathbf{FP} : x \in A \Leftrightarrow f(x) \in B$.
- Die *Polynomialzeit-Turing-Reduktion* $\leq_T^p$ wird definiert durch $A \leq_T^p B$ genau dann, wenn eine DPOTM M existiert mit $A = L(M^B)$.
- Die *Orakel-Reduktion* relativ zu einem Orakel S $\leq_m^{p;S}$ wird definiert durch $A \leq_m^{p;S} B$ genau dann, wenn $\exists f^S \in \mathbf{FP}^S : x \in A \Leftrightarrow f^S(x) \in B$ genau dann, wenn eine DOTM M existiert mit $x \in A \Leftrightarrow M^S(x) \in B$.

Reduktionen

Anmerkung

Wenn $A \leq B$ gilt, so gilt auch $\overline{A} \leq \overline{B}$ mit $\leq \in \{\leq_m^p, \leq_T^p, \leq_m^{p,S}\}$.

Kurze Wiederholung

Was ist wichtig? (Für den folgenden SvK ...)

- Orakel-Turingmaschine, Präfixsuche (mittels OTM)
- $A \leq_m^{p,S} B$ genau dann, wenn $\exists f^S \in \mathbf{FP}^S : x \in A \Leftrightarrow f^S(x) \in B$ genau dann, wenn eine DOTM M existiert mit $x \in A \Leftrightarrow M^S(x) \in B$
- $\Delta_{i+1} = \mathbf{P}^{\Sigma_i}$, $\Sigma_{i+1} = \mathbf{NP}^{\Sigma_i}$ und $\Pi_{i+1} = \mathbf{co}\Sigma_{i+1}$
- $\mathbf{BH}_k = \begin{cases} \mathbf{BH}_{k-1} \wedge \mathbf{coNP} & \text{falls } k \text{ gerade} \\ \mathbf{BH}_{k-1} \vee \mathbf{NP} & \text{falls } k \text{ ungerade} \end{cases}$ für $k \geq 2$
- Die vollständigen Probleme $L_{\mathbf{BH}_k}$ und $L_{\mathbf{coBH}_k}$. In Kurzform:
  - $L_{\mathbf{BH}_3} = (\mathtt{SAT}\ \&\ \overline{\mathtt{SAT}}) \mid \mathtt{SAT}$
  - $L_{\mathbf{coBH}_3} = (\overline{\mathtt{SAT}} \mid \mathtt{SAT})\ \&\ \overline{\mathtt{SAT}}$

Das Lemma von Yap

Satz (Kadin, 1988)

Wenn ein $k \geq 1$ mit $\mathbf{BH}_k = \mathbf{coBH}_k$ existiert, dann bricht die polynomielle Hierarchie zusammen: $\mathbf{PH} \subseteq \Sigma_3$.

Lemma (Yap, 1983)

Existiert eine dünne Menge S mit $\mathbf{coNP} \subseteq \mathbf{NP}^S$, so bricht die polynomielle Hierarchie zusammen: $\mathbf{PH} \subseteq \Sigma_3$.

Satz (Kadin)

Wenn ein $k \geq 1$ mit $\mathbf{BH}_k = \mathbf{coBH}_k$ existiert, dann existiert eine dünne Menge S mit $\mathbf{coNP} \subseteq \mathbf{NP}^S$.

Das Lemma von Yap

Lemma (Yap, 1983)

Existiert eine dünne Menge S mit $\mathbf{coNP} \subseteq \mathbf{NP}^S$, so bricht die polynomielle Hierarchie zusammen: $\mathbf{PH} \subseteq \Sigma_3$.

Definition

Sei $S \subseteq \Sigma^*$ eine Menge von Wörtern. Es sei

$$S^{\leq n} := \{w \in S \mid |w| \leq n\}.$$

S ist eine *dünne Menge*, wenn ein Polynom p existiert mit

$$||S^{\leq n}|| \leq p(n)$$

für alle $n \in \mathbb{N}$.

Beweisskizze

Beweis.

Was ist zu beweisen?

- Aufgrund des Lemmas von Yap reicht es die Existenz eines dünnen Orakels S mit $\mathbf{coNP} \subseteq \mathbf{NP}^S$ zu beweisen.
- Es reicht zu zeigen, dass $\overline{\mathtt{SAT}} \leq_m^{p,S} \mathtt{SAT}$ möglich ist.

□

Nachfolgend wird der Beweis für den Fall, dass $\mathbf{DP} = \mathbf{coDP}$ gilt geführt (also für $k = 2$). Die Idee ist dann verallgemeinerbar.

Idee 1: Die Struktur der **BH**

Die Struktur der vollständigen Probleme $L_{\mathbf{DP}}$ und $L_{\mathbf{coDP}}$ ausnutzen:

- $L_{\mathbf{DP}} = L_{\mathbf{BH}_2} = \mathtt{SAT}\ \&\ \overline{\mathtt{SAT}}$
- $L_{\mathbf{coDP}} = L_{\mathbf{coBH}_2} = \overline{\mathtt{SAT}} \mid \mathtt{SAT}$

$L_{\mathbf{coDP}}$ ist **coDP**-vollständig, d.h. jedes Problem aus **coDP** lässt sich auf $L_{\mathbf{coDP}}$ reduzieren. Wegen der Prämisse **DP** = **coDP** gilt insbesondere

$$L_{\mathbf{DP}} \leq_m^p L_{\mathbf{coDP}}$$

Eine *Konjunktion* wird auf eine *Disjunktion* abgebildet. Dies kann verallgemeinert werden.

Idee 2: Die Easy-Hard-Technik

Unterteilung der Formeln aus $\overline{\mathtt{SAT}}$ in *einfache* und *schwierige* Formeln.

- Für die einfachen kann man schnell einen **NP**-Algorithmus angeben.
- Für die schwierigen braucht man etwas Hilfe und bedient sich des Orakels.

Der Beweis für $k = 2$

Beweis

Gelte **DP** = **coDP**. Es gibt eine Reduktion h derart, dass

$$\mathtt{SAT}\ \&\ \overline{\mathtt{SAT}} \leq_m^p \overline{\mathtt{SAT}} \mid \mathtt{SAT}$$

Sei nun eine Formel F der Länge n gegeben. Ein **NP**-Algorithmus kann wie folgt einen *Teil* von $\overline{\mathtt{SAT}}$ entscheiden:

- Rate F' der Länge n.
- Berechne $h(F', F) = (G', G)$
- Prüfe ob $G \in \mathtt{SAT}$.

$\Rightarrow$ Formeln wie F werden als *einfach* bezeichnet.

Der Beweis für $k = 2$

Beweis (Fortsetzung)

Sei M_e die TM, die obigen Algorithmus ausführt. Es gilt

- $L(M_e) \subseteq \overline{\mathtt{SAT}}$ und
- $L(M_e) \leq_m^p \mathtt{SAT}$

Sei die Reduktion für letzteres h_e.

Anmerkung

Sind alle Formeln in $\overline{\mathtt{SAT}}^{=n}$ einfach, so ist h_e eine Reduktion von $\overline{\mathtt{SAT}}^{=n}$ auf $\mathtt{SAT}$, *aber das muss nicht so sein...*

Der Beweis für $k = 2$

Sind nicht alle Formeln einfach, dann muss es *schwierige* geben.

- Eine Formel F heisst *schwierig* genau dann, wenn $F \in \overline{\mathtt{SAT}}$ gilt und F nicht einfach ist, d.h. genau dann, wenn $F \in \overline{\mathtt{SAT}}$ gilt und für alle F' gleicher Länge $\pi_2(h(F', F)) \in \overline{\mathtt{SAT}}$ gilt.

Eine Formel F heisst *einfach* genau dann, wenn eine Formel F' gleicher Länge existiert derart, dass $\pi_2(h(F', F)) \in \mathtt{SAT}$ gilt.

Der Beweis für $k = 2$

Schwierige Formeln haben nun eine praktische Eigenschaft:

- Sei F_s eine schwierige Formel der Länge n und F eine beliebige Formel gleicher Länge.
- $h(F, F_s) = (G, G')$ mit $G' \in \overline{\mathtt{SAT}}$.
- $(F, F_s) \in \mathtt{SAT}\ \&\ \overline{\mathtt{SAT}} \Leftrightarrow (G, G') \in \overline{\mathtt{SAT}} \mid \mathtt{SAT}$
- $(F, F_s) \in \mathtt{SAT}\ \&\ \overline{\mathtt{SAT}} \Leftrightarrow G \in \overline{\mathtt{SAT}}$
- $F \in \mathtt{SAT} \Leftrightarrow G \in \overline{\mathtt{SAT}}$
- $F \in \overline{\mathtt{SAT}} \Leftrightarrow G \in \mathtt{SAT}$

Eine Formel F heisst *schwierig* genau dann, wenn $F \in \overline{\mathtt{SAT}}$ gilt und für alle F' gleicher Länge $\pi_2(h(F', F)) \in \overline{\mathtt{SAT}}$ gilt.

Der Beweis für $k = 2$

Gegeben eine schwierige Formel F_s der Länge n, kann $\overline{\mathtt{SAT}}^{=n}$ auf $\mathtt{SAT}$ in Polynomialzeit reduziert werden:

- Berechne $h(F, F_s) = (G, G')$. (F ist die Eingabe!)
- Gib G aus.

Sei $h_{s(F_s)}$ diese Reduktion. Es muss jetzt noch F_s ermittelt werden ...

Anmerkung

- Ob F selbst schwierig oder einfach ist, ist gar nicht bekannt.
- Eine schwierige Formel, dient als *Schlüsselformel* für ganz $\overline{\mathtt{SAT}}^{=n}$.

Der Beweis für $k = 2$

Um die schwierigen Formeln zu ermitteln bedienen wir uns eines *Orakels* und der *Präfixsuche*.

- $S_n = \left\{ \alpha 2^{n-|\alpha|} \;\middle|\; \begin{array}{l} \exists F_{s_n} \in \{0,1\}^* : |F_{s_n}| = n \text{ und} \\ F_{s_n} \text{ ist schwierig und} \\ \alpha \text{ ist Präfix der lexikographisch} \\ \text{kleinsten schwierigen Formel} \\ \text{der Länge } n \end{array} \right\}$.
- $S = \bigcup_{n \geq 1} S_n$.

- S_n ist entweder leer oder enthält genau $n+1$ Elemente.
- S ist eine dünne Menge: $|S^{\leq n}| = \Sigma_{i=1}^{i=n} |S_i| \leq (n+1)^2$.
- Mit Präfixsuche kann eine schwierige Formel der Länge n ermittelt werden.

Der Beweis für $k = 2$

Jetzt kann $\overline{\texttt{SAT}} \leq_m^{p,S} \texttt{SAT}$ gezeigt werden:

- S sei obige dünne Menge.
- Bei vorgelegter Formel F der Länge n wird zunächst geprüft, ob $2^n \in S$ gilt.
- Falls ja, wird F_s ermittelt und $h_{s(F_s)}(F)$ ausgegeben.
- Falls nein, wird $h_e(F)$ ausgegeben.

Anmerkung

Die einzelnen Schritte sind alle in Polynomialzeit möglich, da insbesondere h eine Reduktion in Polynomialzeit ist.

Der Beweis für $k = 2$ (Zusammenfassung)

1. Aufgrund der Prämisse die Struktur der $L_{\mathbf{BH}_k}$ und $L_{\mathbf{coBH}_k}$ nutzen.
   - Konjuktion auf Disjunktion abbilden.
2. Die Formeln aus $\overline{\texttt{SAT}}$ in einfache und schwierige Formeln einteilen.
   - Die einfachen sofort auf $\texttt{SAT}$ reduzierbar.
   - Die schwierigen mittels des Orakels behandeln.
3. Schwierige Formeln untersuchen. Jede schwierige Formel ist eine *Schlüsselformel* für ganz $\overline{\texttt{SAT}}^{=n}$.
4. Mit einem dünnen Orakel und der Präfixsuche schwierige Formeln ermitteln.
5. Alles in richtiger Reihenfolge zusammenbauen.

Und der allgemeine Fall?

Satz

Wenn für ein $k > 1$ eine dünne Menge S existiert mit $L_{\mathbf{BH}_k} \leq_m^{p,S} L_{\mathbf{coBH}_k}$, dann gibt es eine dünne Menge S' mit $L_{\mathbf{BH}_{k-1}} \leq_m^{p,S'} L_{\mathbf{coBH}_{k-1}}$.

Beweis.

Ist in der Originalliteratur zu finden... □

Und das Lemma von Yap?

Lemma (Yap, 1983)

Existiert eine dünne Menge S mit $\mathbf{coNP} \subseteq \mathbf{NP}^S$, so bricht die polynomielle Hierarchie zusammen: $\mathbf{PH} \subseteq \Sigma_3$.

Beweis.

Ist auch in der Originalliteratur zu finden... $\square$

Fragen?

Fragen
?!?

Verfeinerungen

Das Ergebnis von Kadin wurde unter der gleichen Prämisse verfeinert:

- Wagner zeigt, dass $\mathbf{PH} \subseteq \Delta_3$ und sogar $\mathbf{PH} \subseteq \mathbf{BH}(\Sigma_2)$ gilt.
- Chang und Kadin zeigten, dass der Zusammenbruch der Booleschen Hierarchie auf die k-te Stufe $\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{NP}^{\mathbf{NP}}[k]}$ impliziert.
- Weitere Ergebnisse von Beigel, Chang und Ogihara (1993) und E. und L. Hemaspaandra und Hempel (1998).

Sparse sets

In wie weit ging es um die Boolesche Hierarchie? In wie weit nicht eher um dünne Mengen?

Satz

Wenn es eine dünne Menge $S \in \mathbf{NP}$ gibt, die $\leq_T^p$ -vollständig ist (für $\mathbf{NP}$), dann gilt $\mathbf{PH} = \mathbf{P}^{\mathbf{NP}[\mathcal{O}(log)]}$.

Satz (Karp-Lipton)

Wenn es eine dünne Menge $S \in \mathbf{NP}$ gibt, die $\leq_T^p$ -schwierig ist, dann gilt $\mathbf{PH} = \mathbf{NP}^{\mathbf{NP}}$.

Nächstes Mal...

Nächstes Mal:

- Die Query Hierarchy (diesmal wirklich!)
- Die parallele Query Hierarchie (ebenso!)
- Der Satz, der diese Hierarchien (und die BH) miteinander verzahnt (mit Beweis; *mind change technique*)

Ende...

Danke für die Aufmerksamkeit
!