Kapitel 5: Die Polynomialzeit-Hierarchie und Alternierungen

5.1 Die Klasse Σ_2^P

Zur Erinnerung:

Das folgende Problem ist NP-vollständig:

$$\text{INDSET} = \{ \langle G, k \rangle : \text{Graph } G \text{ besitzt eine unabhängige Menge der Größe } k \}$$

Frage 5.1:

Was ist mit den folgenden Problemen?

(a) $\text{EXACT-INDSET} := \{ \langle G, k \rangle : k \text{ ist die Größe der } \underline{\text{größten}} \text{ unabhängigen Menge von } G \}$

Klar: $\langle G, k \rangle \in \text{EXACT-INDSET} \iff$

$\exists S_1 \subseteq V(G) \;\; \forall S_2 \subseteq V(G):$
$|S_1| = k \wedge S_1$ ist eine unabhängige Menge $\wedge$
$(S_2$ ist eine unabhängige Menge $\rightarrow |S_2| \leq k)$

(b) MIN-EQ-DNF :=

$\{ \langle \varphi, k \rangle$: φ ist eine aussagenlogische Formel, $k \in \mathbb{N}$ s.d. für die <u>kürzeste</u> zu φ äquivalente Formel ψ in DNF gilt: $|\psi| = k$ $\}$

disjunktive Normalform

(Warum ist dieses Problem interessant?
— Weil das Erfüllbarkeitsproblem für DNF-Formeln in Polynomialzeit lösbar ist (siehe Vorlesung "Diskrete Modellierung") und man das SAT-Problem lösen kann, indem man eine gegebene CNF-Formel φ zunächst in eine äquivalente, möglichst kurze DNF-Formel ψ transformiert und dann diese Formel in Zeit poly($|\psi|$) auf Erfüllbarkeit testet.
<u>Problem</u>: Es ist bekannt, dass $|\psi|$ exponentiell größer sein kann als $|\varphi|$ (vgl. Vorlesung/Übung "Diskrete Modellierung")
)

<u>klar</u>: $\langle \varphi, k \rangle \in$ MIN-EQ-DNF $\iff$

$\exists \psi_1 \; \forall \psi_2$:
ψ_1 in DNF $\wedge$ $\psi_1 \equiv \varphi$ $\wedge$ $|\psi_1| \leq k$ $\wedge$
$((\psi_2$ in DNF $\wedge$ $\psi_2 \equiv \varphi) \rightarrow |\psi_2| \geq k$ $)$

Die beiden Probleme EXACT-INDSET und MIN-EQ-DNF gehören zur folgenden Klasse Σ_2^P:

<u>Definition 5.2</u> (Σ_2^p – die 2te Stufe der Polynomialzeit-Hierarchie)

Die Klasse Σ_2^p besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die es eine det. Polynomialzeit-TM M und ein Polynom $q : \mathbb{N} \to \mathbb{N}$ gibt, so dass f.a. $x \in \{0,1\}^*$ gilt:

$$x \in L \iff \exists u_1 \in \{0,1\}^{q(|x|)} \; \forall u_2 \in \{0,1\}^{q(|x|)} \; M(\langle x, u_1, u_2 \rangle) = 1.$$

<u>Beachte:</u> $\Sigma_2^p \supseteq NP$ und $\Sigma_2^p \supseteq coNP$.

5.2 Die Polynomialzeit-Hierarchie

Σ_2^p ist durch 2 wechselnde Quantoren ($\exists \forall$) definiert. Dies lässt sich natürlich für beliebige Zahlen k von wechselnden Quantoren verallgemeinern:

<u>Definition 5.3</u> (Σ_k^p , Π_k^p und PH)

<u>(a)</u> Sei $k \in \mathbb{N}$. Die Klasse Σ_k^p besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die es eine det. Polynomialzeit-TM M und ein Polynom $q : \mathbb{N} \to \mathbb{N}$ gibt, s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$x \in L \iff \exists u_1 \in \{0,1\}^{q(|x|)} \; \forall u_2 \in \{0,1\}^{q(|x|)} \cdots Q_k u_k \in \{0,1\}^{q(|x|)}$$
$$M(\langle x, u_1, u_2, \dots, u_k \rangle) = 1$$

wobei $Q = \{ \exists$ falls k ungerade

<u>Insbes.:</u> $\Sigma_1^P = NP$, $\Sigma_0^P = P$

(b) Sei $k \in \mathbb{N}$ $\quad \Pi_k^P := co\Sigma_k^P \stackrel{\text{Def}}{=} \{L : \overline{L} \in \Sigma_k^P\}$.

<u>Insbes.:</u> $\Pi_1^P = coNP$, $\Pi_0^P = P$.

Man sieht leicht, dass f.a. $L \subseteq \{0,1\}^*$ gilt: $L \in \Pi_k^P \iff$ es gibt eine det. Polynomialzeit-TM M und ein Polynom $q: \mathbb{N} \to \mathbb{N}$, s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$x \in L \iff \forall u_1 \in \{0,1\}^{q(|x|)} \; \exists u_2 \in \{0,1\}^{q(|x|)} \; \dots \; Q_k u_k \in \{0,1\}^{q(|x|)}$$

$$M(\langle x, u_1, u_2, \dots, u_k \rangle) = 1$$

wobei $Q_k = \begin{cases} \forall & \text{falls } k \text{ ungerade} \\ \exists & \text{falls } k \text{ gerade und } k \neq 0. \end{cases}$

(c) Die <u>Polynomialzeit-Hierarchie</u> ist die Klasse

$$PH := \bigcup_{k \in \mathbb{N}} \Sigma_k^P$$

Anhand der Definition von Σ_k^P, Π_k^P und PH sieht man leicht, dass folgendes gilt (f.a. $k \in \mathbb{N}$):

$$\Sigma_k^P \subseteq \Pi_{k+1}^P \subseteq \Sigma_{k+2}^P \subseteq PH$$

und daher auch $PH = \bigcup_{k \in \mathbb{N}} \Pi_k^P$.

Für jedes $k \in \mathbb{N}$ definieren wir

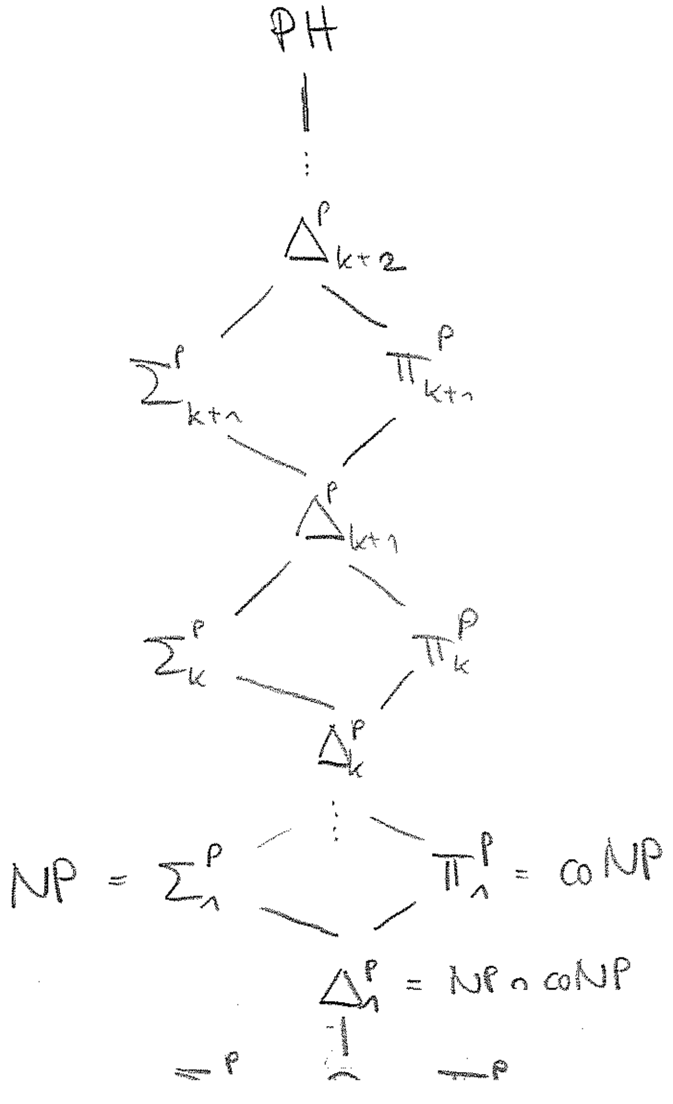
$$\Delta_k^p := \Sigma_k^p \cap \Pi_k^p .$$

<u>Klar:</u>

$$\Delta_k^p \subseteq \Sigma_k^p \subseteq \Delta_{k+1}^p$$
$$\Delta_k^p \subseteq \Pi_k^p \subseteq \Delta_{k+1}^p$$

denn: $\Sigma_k^p \subseteq \Pi_{k+1}^p \cap \Sigma_{k+1}^p = \Delta_{k+1}^p$
und $\Pi_k^p \subseteq \Sigma_{k+1}^p \cap \Pi_{k+1}^p = \Delta_{k+1}^p$

Somit weisen die Stufen der Polynomialzeit-Hierarchie die folgende Inklusionsstruktur auf:



Man vermutet, dass die Stufen alle verschieden sind, d.h. dass $\Sigma_{k-1}^P \neq \Sigma_k^P$ und $\Sigma_k^P \neq \Pi_k^P$

f.a. $k \geq 1$ gilt.

(<u>Beachte</u>: für $k=1$ besagt diese Vermutung gerade, dass $P \neq NP$ und $NP \neq coNP$ ist.)

Diese Vermutung wird oft in der Aussage "Die Polynomialzeit-Hierarchie ist strikt" bzw. "Die Polynomialzeit-Hierarchie kollabiert nicht" zusammengefasst.

<u>Satz 5.4</u>

<u>(a)</u> Falls $P = NP$ ist, so gilt $PH = P$
(d.h. die Polynomialzeit-Hierarchie kollabiert zu ihrer 0-ten Stufe $P = \Sigma_0^P$)

<u>(b)</u> Für jedes $k \geq 1$ gilt:
Falls $\Sigma_k^P = \Pi_k^P$, so ist $PH = \Sigma_k^P$
(d.h. die Polynomialzeit-Hierarchie kollabiert zu ihrer k-ten Stufe Σ_k^P)

<u>Beweis:</u>

<u>(a)</u> Wir nehmen an, dass $P = NP$ ist und zeigen per Induktion nach k, dass f.a. $k \in \mathbb{N}$ gilt:
$\Sigma_k^P = P$.

<u>$k=0$:</u> klar (da $\Sigma_0^p \stackrel{\text{Def}}{=} P$)

<u>$k=1$:</u> $\Sigma_1^p = NP = P$ gemäß Annahme

<u>$k \to k+1$:</u> <u>Ind.annahme:</u> $\Sigma_k^p = P$

(für $k \geq 1$) <u>zu zeigen:</u> $\Sigma_{k+1}^p = P$

<u>Beweis:</u> Gemäß Ind.annahme gilt $\Sigma_k^p = P$

Somit gilt auch $\Pi_k^p \stackrel{\text{Def}}{=} co\Sigma_k^p = coP = P$, also (*): $\Pi_k^p = P$.

klar: $P \subseteq \Sigma_{k+1}^p$, zum Beweis von $\Sigma_{k+1}^p \subseteq P$

sei nun $L \in \Sigma_{k+1}^p$. Zu zeigen: $L \in P$.

Wegen $L \in \Sigma_{k+1}^p$ gibt es eine det. Polynomialzeit-TM M und ein Polynom $q: \mathbb{N} \to \mathbb{N}$ s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$x \in L \iff \exists u_1 \in \{0,1\}^{q(|x|)}\ \forall u_2 \in \{0,1\}^{q(|x|)} \ldots Q_{k+1} u_{k+1} \in \{0,1\}^{q(|x|)}$$
$$M(\langle x, u_1, u_2, \ldots, u_{k+1} \rangle) = 1$$

mit $Q_{k+1} = \begin{cases} \exists & \text{falls } k+1 \text{ ungerade} \\ \forall & \text{falls } k+1 \text{ gerade.} \end{cases}$

Sei $L' := \{ \langle x, u_1 \rangle : \forall u_2 \in \{0,1\}^{q(|x|)} \ldots Q_{k+1} u_{k+1} \in \{0,1\}^{q(|x|)}$
$M(\langle x, u_1, u_2, \ldots, u_{k+1} \rangle) = 1 \}$

klar: $L' \in \Pi_k^p$. Wegen (*) gilt $\Pi_k^p = P$. Daher gibt es eine Polynomialzeit-TM M' s.d. f.a. $\langle x, u_1 \rangle$ gilt:

$$\langle x, u_1 \rangle \in L' \iff M'(\langle x, u_1 \rangle) = 1$$

Somit gilt f.a. $x \in \{0,1\}^*$

$$x \in L \iff \exists u_1 \in \{0,1\}^{q(|x|)} \quad \langle x, u_1 \rangle \in L'$$
$$\iff \exists u_1 \in \{0,1\}^{q(|x|)} \quad M'(\langle x, u_1 \rangle) = 1$$

Also ist $L \in NP$. Gemäß Voraussetzung ist $NP = P$, also $L \in P$. Somit ist $\Sigma_{k+1}^P \subseteq P$. □

(b): analog (Details: Übung!)

Vollständige Probleme für die einzelnen Stufen von PH

Definition 5.5

Sei K eine der Klassen PH, Σ_k^P, Π_k^P (für $k \geq 1$).
Eine Sprache $L \subseteq \{0,1\}^*$ heißt K-vollständig, falls gilt:

(1) $L \in K$ und

(2) L ist K-hart, d.h. f.a. $L' \in K$ gilt: $L' \leq_p L$.

Beobachtung 5.6: (Vermutung: Es gibt kein PH-vollständiges Problem)

Falls es eine PH-vollständige Sprache L gibt, dann gibt es ein $k \in \mathbb{N}$ s.d. $PH = \Sigma_k^P$ (d.h. die Polynomialzeit-Hierarchie kollabiert zu ihrer k-ten Stufe Σ_k^P).

<u>Beweis:</u>

Sei $L \subseteq \{0,1\}^*$ PH-vollständig.

Wegen $L \in PH = \bigcup_{k \in \mathbb{N}} \Sigma_k^p$ gibt es ein $k \in \mathbb{N}$

s.d. $L \in \Sigma_k^p$ via TM M und Polynom q gemäß Def. 5.3.

Sei $L' \in PH$. Zu zeigen: $L' \in \Sigma_k^p$.

Da L PH-hart ist, ist $L' \leq_p L$ durch eine Polynomialzeit-Reduktion $f: \{0,1\}^* \to \{0,1\}^*$.

Somit gilt f.a. $x \in \{0,1\}^*$:

$$x \in L' \iff f(x) \in L$$
$$\iff \exists u_1 \in \{0,1\}^{q(|f(x)|)} \; \forall u_2 \in \{0,1\}^{q(|f(x)|)} \dots Q_k u_k \in \{0,1\}^{q(|f(x)|)}$$
$$M(\langle f(x), u_1, u_2, \dots, u_k \rangle) = 1$$

Wir können oBdA annehmen, dass $|f(x)| = |x|^c$ für ein geeignetes $c \in \mathbb{N}$ ist (da f in Polynomialzeit berechnet werden kann – und ggf unter Verwendung eines geeigneten Paddings). Da $f(x)$ in Polynomialzeit berechnet werden kann ist daher $L' \in \Sigma_k^p$.

□

Beobachtung 5.7 (PH und PSPACE)

(a) $PH \subseteq PSPACE$

(b) Falls $PH = PSPACE$, so gibt es ein $k \in \mathbb{N}$ s.d. $PH = \Sigma_k^P$. (— Also gilt vermutlich: $PH \neq PSPACE$)

Beweis:

(a): Analog zum Nachweis, dass $NP \subseteq PSPACE$ und $TQBF \in PSPACE$ zeigt man für jedes $k \in \mathbb{N}$, dass $\Sigma_k^P \subseteq PSPACE$.

(b) Wir wissen, dass TQBF PSPACE-vollständig ist. Falls $PH = PSPACE$, so ist TQBF PH-vollständig. Somit folgt die Behauptung aus Beobachtung 5.6. □

Beobachtung 5.8 (Vollständige Probleme für Σ_k^P und Π_k^P)

(a) Für jedes $k \geq 1$ ist das folgende Problem Σ_k^P-vollständig

Σ_k-SAT $:= \{ \Phi : \Phi$ ist eine wahre QBF der Form $\exists \overline{u_1} \forall \overline{u_2} \cdots Q_k \overline{u_k}\, \varphi$, wobei $\overline{u_1}, \ldots, \overline{u_k}$ Listen von Variablen sind und φ eine aussagenlogische Formel über den Variablen $\overline{u_1}, \ldots, \overline{u_k}$ ist, und $Q_k = \begin{cases} \exists & \text{falls } k \text{ ungerade} \\ \forall & \text{falls } k \text{ gerade} \end{cases}$ $\}$

(b) Für jedes $k \geq 1$ ist das folgende Problem Π_k^P-vollständig:

Π_k-SAT $:= \{ \Phi : \Phi$ ist eine wahre QBF der Form $\forall \bar{u}_1 \exists \bar{u}_2 \cdots Q_k \bar{u}_k \; \varphi$, wobei $\bar{u}_1, \dots, \bar{u}_k$ Listen von Variablen sind und φ eine aussagenlogische Formel über den Variablen $\bar{u}_1, \dots, \bar{u}_k$ ist und $Q_k = \begin{cases} \forall & \text{falls } k \text{ ungerade} \\ \exists & \text{falls } k \text{ gerade} \end{cases}$ $\}$

Beweis:

(b) folgt leicht aus (a), da $\Pi_k^P = \text{co}\,\Sigma_k^P$.

zu (a): Σ_k-SAT $\in \Sigma_k^P$ folgt unmittelbar aus der Definition von Σ_k-SAT und Σ_k^P.

Die Σ_k^P-Härte von Σ_k-SAT lässt sich leicht aus dem Beweis des Satzes von Cook und Levin folgern.

Details: Übung!

Bemerkung 5.3

(a) Das Problem EXACT-INDSET (vgl. Frage 5.1 (a)) ist vermutlich nicht Σ_2^P-vollständig, da es bereits in $\Sigma_2^P \cap \Pi_2^P$ liegt.

(b) Das Problem MIN-EQ-DNF (vgl. Frage 5.1(b)) ist Σ_2^P-vollständig (hier ohne Beweis)

5.3 Charakterisierung der PH durch Orakel-TMen

Zur Erinnerung (vgl. Kapitel 3.3)

- M eine Orakel-TM, $O \subseteq \{0,1\}^*$, $x \in \{0,1\}^*$ $\Rightarrow$ schreibe $M^O(x)$, um die Ausgabe von M bei Eingabe x mit Orakel O zu bezeichnen
- NP^O: die Klasse aller Sprachen $L \subseteq \{0,1\}^*$, die durch eine ndet. Orakel-TM mit Orakel O in polynomiell vielen Schritten entschieden werden können.

Satz 5.10

Für jedes $k \geq 2$ gilt: $\Sigma_k^P = NP^{\Sigma_{k-1}\text{-SAT}}$.

Beweis: Wir zeigen hier die Aussage für $k=2$ (die allgemeine Aussage für $k \geq 2$ lässt sich analog beweisen

Zu zeigen: $\Sigma_2^P = NP^{\Sigma_1\text{-SAT}}$

"$\subseteq$": Sei $L \in \Sigma_2^P$. D.h. sei M eine det. Polynomialzeit-TM und sei $q: \mathbb{N} \to \mathbb{N}$ ein Polynom s.d. f.a. $x \in \{0,1\}^*$ gilt:

$x \in L \iff \exists u_1 \in \{0,1\}^{q(|x|)}\ \forall u_2 \in \{0,1\}^{q(|x|)} : M(\langle x, u_1, u_2\rangle) = 1$

$\iff \exists u_1 \in \{0,1\}^{q(|x|)}$ <u>nicht</u> $\exists u_2 \in \{0,1\}^{q(|x|)} : M(\langle x, u_1, u_2\rangle) = 0$

$\iff \exists u_1 \in \{0,1\}^{q(|x|)} : \langle x, u_1\rangle \notin L'$, (*)

wobei $L' := \{ \langle x, u_1\rangle : x \in \{0,1\}^*,\ u_1 \in \{0,1\}^{q(|x|)}$ und $\exists u_2 \in \{0,1\}^{q(|x|)} : M(\langle x, u_1, u_2\rangle) = 0 \}$

<u>klar:</u> $L' \in NP$ und

$L \in NP^{L'}$ (durch eine ndet. Orakel-TM, die bei Eingabe von x zunächst ein $u_1 \in \{0,1\}^{q(|x|)}$ rät, dann das Orakel befragt, ob $\langle x, u_1\rangle \in L'$ ist und genau dann akzeptiert, wenn das Orakel "nein" antwortet.

<u>Ziel:</u> Wir wollen zeigen, dass $L \in NP^{\Sigma_1\text{-SAT}}$ ist.

Wir wissen, dass SAT NP-vollständig ist. Wegen $L' \in NP$ gibt es eine Polynomialzeit-Reduktion f von L' auf SAT. D.h. f.a. x, u_1 gilt

$$\langle x, u_1\rangle \in L' \iff \underbrace{f(\langle x, u_1\rangle)}_{=: \varphi_{\langle x, u_1\rangle}} \in \text{SAT}$$

Sei $\Phi_{\langle x, u_1\rangle}$ die QBF, die aus $\varphi_{\langle x, u_1\rangle}$ entsteht, indem vorne alle in $\varphi_{\langle x, u_1\rangle}$ vorkommenden Variablen existentiell quantifiziert werden.

<u>klar:</u> $\Phi_{\langle x, u_1\rangle} \in \Sigma_1\text{-SAT} \iff \langle x, u_1\rangle \in L'$ (**)

Um zu zeigen, dass $L \in NP^{\Sigma_i\text{-SAT}}$ ist, konstruieren wir eine ndet. Orakel-TM $\tilde{M}$, die bei Eingabe von $x \in \{0,1\}^*$ folgendermaßen vorgeht:

1) rate ein $u_1 \in \{0,1\}^{q(|x|)}$

2) nutze die Polyn.zeit-Reduktion f, um die QBF $\Phi_{\langle x, u_1 \rangle}$ zu konstruieren

3) befrage das Orakel, ob $\Phi_{\langle x, u_1\rangle} \in \Sigma_i\text{-SAT}$ ist

4) akzeptiere genau dann, wenn das Orakel "nein" antwort[et]

Klar: • All dies geht in Polynomialzeit (da q ein Polynom und f eine Polynomialzeit-Reduktion ist)

• Außerdem gilt f.a. $x \in \{0,1\}^*$:

$$
\begin{aligned}
x \in L &\overset{(*)}{\iff} \exists u_1 \in \{0,1\}^{q(|x|)} : \langle x, u_1 \rangle \notin L' \\
&\overset{(**)}{\iff} \exists u_1 \in \{0,1\}^{q(|x|)} : \Phi_{\langle x, u_1 \rangle} \notin \Sigma_i\text{-SAT} \\
&\iff \tilde{M} \text{ akzeptiert } x
\end{aligned}
$$

Somit ist $L \in NP^{\Sigma_i\text{-SAT}}$.

"$\supseteq$": Sei nun $L \in NP^{\Sigma_i\text{-SAT}}$ durch eine ndet. Polynomialzeit-Orakel-TM N mit Orakel Σ_i-SAT.

Beachte: Im Laufe ihrer Berechnung kann N polynomiell viele Anfragen ans Orakel stellen — und jede Anfrage kann von den vorherigen Orakel-Antworten und dem bisherigen Verlauf der Berechnung abhängen.

Um zu zeigen, dass $L \in \Sigma_2^P$ ist, müssen wir dies unter Verwendung von einem $\exists$-Quantor gefolgt von einem $\forall$-Quantor ausdrücken.

Idee: Durch den $\exists$-Quantor werden alle von N getroffenen ndet. Entscheidungen geraten sowie alle Orakel-Anfragen, die N im Laufe der Berechnung stellt und alle Antworten, die das Orakel gibt.

Genauer: Sei $q: \mathbb{N} \to \mathbb{N}$ die (polynomielle) Zeitschranke von N

D.h.: N läuft bei Eingabe x maximal $t := q(|x|)$ Schritte.

Sei $\overline{c} = c_1 \cdots c_t \in \{0,1\}^t$ eine Folge von ndet. Entscheidungen die N im Laufe ihrer Berechnung trifft.

Für $1 \leq i \leq t$ sei $\Phi_i := \exists \overline{u}_i \, \varphi_i(\overline{u}_i)$ die i-te Orakel-Anfrage, die N bei Eingabe x unter Verwendung von $\overline{c}$ stellt (d.h. N fragt, ob die aussagenlogische Formel φ_i erfüllbar ist).

Sei $a_i \in \{0,1\}$ die Antwort des Orakels (d.h. $a_i = 1 \iff \varphi_i$ ist erfüllbar)

Es gilt f.a. $x \in \{0,1\}^*$ und $t := q(|x|)$:

$x \in L$

$$\iff \exists \, \overline{c} \in \{0,1\}^t, \ \exists \left(\Phi_i = \exists \overline{u}_i \, \varphi_i(\overline{u}_i)\right)_{1 \leq i \leq t} \ \exists \, a_1 \cdots a_t \in \{0,1\}^t$$

$$\exists \left(\overline{z}_i \in \{0,1\}^{|\overline{u}_i|}\right)_{1 \leq i \leq t} \quad \forall \left(\overline{z}'_i \in \{0,1\}^{|\overline{u}'_i|}\right)_{1 \leq i \leq t} \ :$$

(*) Für alle $i = 1, \ldots, t$ gilt:

- falls $a_i = 1$, so ist $\varphi_i(\overline{z}_i) = 1$
  (d.h. φ_i ist erfüllbar und $\overline{z}_i$ ist eine erfüllende Belegung)
- falls $a_i = 0$, so ist $\varphi_i(\overline{z}'_i) = 0$
  (wegen des $\forall$-Quantors vor $\overline{z}'_i$ ist dann φ_i nicht erfüllbar), und
- es gilt:
  Unter Verwendung der ndet. Entscheidungen $\overline{c}$

und der Orakel-Antworten ... ist

$\Phi_i = \exists \bar{u}_i \, \varphi_i(\bar{u}_i)$ die i-te Orakel-Anfrage, die

N bei Eingabe x stellt, und N beendet seine

Berechnung im Zustand q_{accept}

Man sieht leicht, dass (*) von einer <u>det.</u> polynomialzeit-TM durchgeführt werden kann.

Insgesamt erhalten wir daher gemäß Def 5.2 (und Wahl eines geeigneten Paddings, damit das unter "$\exists$" quantifizierte Wort genauso lang ist wie das unter "$\forall$" quantifizierte Wort), dass $L \in \Sigma_2^P$ ist.

Somit: $NP^{\Sigma_1\text{-SAT}} \subseteq \Sigma_2^P$ $\square$

5.4 Charakterisierung der PH durch Alternierende TMen

Alternierende TMen

Eine alternierende TM (kurz: ATM) ist eine NDTM, bei der jeder Zustand aus $Q \setminus \{q_{halt}, q_{accept}\}$ mit genau einem der beiden Symbole $\exists, \forall$ markiert ist.

Die Akzeptanz-Bedingung einer ATM ist eine Verallgemeinerung der Akzeptanz-Bedingung einer NDTM

- für eine Konfiguration, deren Zustand mit "∃" markiert ist, muss es mind. eine Nachfolgekonfiguration geben, die zur Akzeptanz führt
- für eine Konfiguration, deren Zustand mit "∀" markiert ist, müssen <u>alle</u> Nachfolgekonfigurationen zur Akzeptanz führen.

Präzise lässt sich das wie folgt definieren:

Sei M eine ATM und $x \in \{0,1\}^*$ eine Eingabe für M.
Sei $G_{M,x}$ der Konfigurationsgraph von M bei Eingabe x

Wir nutzen folgendes Verfahren, um einige Knoten von $G_{M,x}$ mit der Markierung ACCEPT zu versehen:

- Jede Konfiguration mit Zustand q_{accept} erhält die Markierung ACCEPT
- Wiederhole so lange, bis sich nichts mehr ändert:
  - Für jede Konfiguration C, deren Zustand mit $\exists$ markiert ist:
    Markiere C mit ACCEPT, falls es eine Kante von C zu einer bereits mit ACCEPT markierten Konfiguration C' gibt
  - Für jede Konfiguration C, deren Zustand mit $\forall$ markiert ist

Markiere C mit ACCEPT, falls alle Konfigurationen C', zu denen C eine Kante hat, bereits mit ACCEPT markiert sind.

Wir sagen:

- M akzeptiert x $:\Longleftrightarrow$ die Startkonfiguration von M bei Eingabe x hat in $G_{M,x}$ die Markierung ACCEPT.

- M entscheidet $L \subseteq \{0,1\}^*$ $:\Longleftrightarrow$ $L = \{x \in \{0,1\}^* : M \text{ akzeptiert } x\}$ und jeder in C_{start} beginnende Pfad von $G_{M,x}$ hat endliche Länge

<u>Definition 5.11</u> (Alternierende Zeit $ATIME(T(n))$)

Sei $T: \mathbb{N} \to \mathbb{N}$.

<u>(a)</u> Eine ATM M läuft in Zeit $T(n)$, falls für jede Eingabe $x \in \{0,1\}^*$ gilt: Alle in C_{start} beginnenden Pfade in $G_{M,x}$ haben die Länge $\leq T(|x|)$

(dh M läuft bei jeder Wahl von ndet Entscheidungen höchstens $T(|x|)$ Schritte).

<u>(b)</u> Die Klasse $ATIME(T(n))$ besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, die von einer ATM M in Zeit $O(T(n))$ entschieden werden können.

Definition 5.12 (Alternierender Platz ASPACE($S(n)$))

Sei $S: \mathbb{N} \to \mathbb{N}$

(a) Eine ATM M ist $S(n)$-platzbeschränkt, falls sie bei jeder Eingabe $x \in \{0,1\}^*$ und jeder Wahl von ndet. Entscheidungen auf jedem ihrer Arbeitsbänder höchstens $S(|x|)$ Zellen benutzt.

(b) Die Klasse ASPACE($S(n)$) besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, die von einer $O(S(n))$-platzbeschränkten ATM entschieden werden können.

Satz 5.13

Sei $T: \mathbb{N} \to \mathbb{N}$ mit $T(n) \geq n$ zeitkonstruierbar und sei $S: \mathbb{N} \to \mathbb{N}$ mit $S(n) \geq \log n$ platzkonstruierbar. Dann gilt

(a) $\mathrm{ATIME}(T(n)) \subseteq \mathrm{SPACE}(T(n))$

(b) $\mathrm{NSPACE}(S(n)) \subseteq \mathrm{ATIME}(S(n)^2)$

(c) $\mathrm{ASPACE}(S(n)) \subseteq \mathrm{DTIME}(2^{O(S(n))})$

(d) $\mathrm{DTIME}(T(n)) \subseteq \mathrm{ASPACE}(\log T(n))$

Beweis (Skizze):

(a) Analog zum Beweis, dass TQBF $\in$ PSPACE (Theorem 4.17) oder dass PH $\subseteq$ PSPACE (Beobachtung 5.7). Details: Übung.

(b) Analog zum Beweis des Satzes von Savitch (Theorem 4.12).

Details:

Sei $L \in NSPACE(S(n))$ und sei M eine $c \cdot S(n)$-platz-beschränkte NDTM, die L entscheidet (für ein $c \in \mathbb{N}$).

Um zu zeigen, dass $L \in ATIME(S(n)^2)$ ist, nutzen wir die folgende "parallele Implementierung" der im Beweis des Satzes von Savitch verwendeten rekursiven Prozedur REACH?(u,v,i), die entscheidet, ob es im Konfigurationsgraph $G_{M,x}$ einen Weg der Länge $\leq 2^i$ von Knoten u zu Knoten v gibt:

PARALLEL-REACH?(u,v,i):

Falls $i=0$:

Falls $u=v$ oder es in $G_{M,x}$ eine Kante von u nach v gibt, s
STOPP mit Antwort "ja"

Sonst:
STOPP mit Antwort "nein"

Falls $i>0$:

Rate einen beliebigen Knoten w von $G_{M,x}$
(Bem: Unsere ATM verwendet dazu $O(S(n))$ Schritte und nutzt Zuständen, die mit "$\exists$" markiert sind)

Geh dann in einen mit "$\forall$" markierten Zustand, von dem aus mit der ndet. Entscheidung "0" bzw "1"

Aufrufe von
PARALLEL-REACH?$(u,w,i-1)$ bzw. PARALLEL-REACH?$(w,v,i-1)$
gestartet werden.

Mit der gleichen Analyse wie im Beweis des Satzes von Savitch erhält man, dass
PARALLEL-REACH?$(C_{start}, C_{accept}, \log(|V(G_{M,x})|))$

durch eine $O(|V(G_{M,x})| \cdot S(n))$ - zeitbeschränkte ATM implementiert werden kann.

Wegen $\log(|V(G_{M,x})|) = O(S(n))$ erhalten wir, dass $L \in \mathrm{ATIME}(S(n)^2)$.

(c) Analog zum Beweis, dass $\mathrm{NSPACE}(S(n)) \subseteq \mathrm{DTIME}(2^{O(S(n))})$

Details: Übung

(d) Hier ohne Beweis.

(Details werden im nächsten Kapitel nachgeliefert).

□

Definition 5.14 (Alternierende Komplexitätsklassen)

$$\mathrm{AL} := \mathrm{ASPACE}(\log n)$$

$$\mathrm{AP} := \bigcup_{c>0} \mathrm{ATIME}(n^c)$$

$$\mathrm{APSPACE} := \bigcup_{c>0} \mathrm{ASPACE}(n^c)$$

$$\mathrm{AEXP} := \bigcup_{c>0} \mathrm{ATIME}(2^{(n^c)})$$

Theorem 5.16 (Platz vs alternierende Zeit)

(a) $\mathrm{AL} = \mathrm{P}$

(b) $\mathrm{AP} = \mathrm{PSPACE}$

(c) $\mathrm{APSPACE} = \mathrm{EXP}$

(d) $\mathrm{AEXP} = \mathrm{EXPSPACE} := \bigcup_{c>0} \mathrm{SPACE}(2^{(n^c)})$

Beweis:

(a) • $AL \overset{\text{Def}}{=} ASPACE(\log n) \overset{\text{Satz 5.13(c)}}{\subseteq} DTIME(2^{O(\log n)}) \subseteq P$

• $P \underset{\text{Satz 5.13(d)}}{\subseteq} ASPACE(O(\log n)) = AL$ ✓

(b) • $AP \overset{\text{Def}}{=} \bigcup_{c>0} ATIME(n^c) \overset{\text{Satz 5.13(a)}}{\subseteq} \bigcup_{c>0} SPACE(n^c) \overset{\text{Def}}{=} PSPACE$

• $PSPACE \underset{\text{Def}}{=} \bigcup_{c>0} SPACE(n^c) \underset{\text{Satz 5.13(b)}}{\subseteq} \bigcup_{c>0} ATIME((n^c)^2) = \bigcup_{d>0} ATIME(n^d)$

$= AP$. ✓

(c), (d): analog.

□

Als Verfeinerung der Klasse $ATIME(T(n))$ betrachten wir die Klassen $\Sigma_k TIME(T(n))$ und $\Pi_k TIME(T(n))$, bei denen Berechnungspfade in $G_{M,x}$ nur $(k-1)$-mal zwischen mit "$\exists$" markierten und mit "$\forall$" markierten Zuständen wechseln dürfen:

Definition 5.17 ($\Sigma_k TIME(T(n))$ und $\Pi_k TIME(T(n))$)

Sei $k \in \mathbb{N}_{>0}$ und sei $T: \mathbb{N} \to \mathbb{N}$.

(a) $\Sigma_k TIME(T(n))$ besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, die von einer $T(n)$-zeitbeschränkten ATM M entschieden werden, deren Startzustand mit "$\exists$" markiert ist und bei der für alle Eingaben $x \in \{0,1\}^*$ gilt:

Entlang jedes in C_{start} beginnenden Pfads in $G_{M,x}$ wird höchstens $(k-1)$-mal zwischen mit "$\exists$" und mit "$\forall$" markierten Zuständen gewechselt.

(b) $\Pi_k TIME(T(n))$ ist analog definiert, wobei der Startzustand mit "$\forall$" markiert sein muss.

<u>Satz 5.18</u> ($\Sigma_k^P = \Sigma_k TIME(n^{O(1)})$)

Für jedes $k \in \mathbb{N}_{>0}$ gilt:

$$\Sigma_k^P = \bigcup_{c>0} \Sigma_k TIME(n^c) \quad \text{und}$$

$$\Pi_k^P = \bigcup_{c>0} \Pi_k TIME(n^c).$$

<u>Beweis:</u> Übung!

5.5 Zeit vs. Alternierungen: Zeit-Platz-Tradeoffs

<u>Definition 5.19</u> (Zeit-Platz-Klasse $TISP(T(n), S(n))$)

Seien $T, S: \mathbb{N} \to \mathbb{N}$. Die Klasse

$TISP(T(n), S(n))$

besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, die von einer (det.) TM entschieden werden, die in Zeit $T(n)$ läuft <u>und</u> $S(n)$-platzbeschränkt ist.

Beobachtung 5.20

$$TISP(T(n), S(n)) \subseteq DTIME(T(n))$$ und
$$TISP(T(n), S(n)) \subseteq SPACE(S(n))$$

Lemma 5.21

$$TISP(n^{12}, n^2) \subseteq \Sigma_2 TIME(n^8)$$

Beweis:

Sei M eine (det.) TM, die eine Sprache L in Zeit n^{12} und auf Platz n^2 entscheidet.

Für jede Eingabe $x \in \{0,1\}^*$ betrachte den Konfigurationsgraphen $G_{M,x}$:

Jeder Knoten von $G_{M,x}$ kann durch einen Bitstring der Länge $O(n^2)$ repräsentiert werden (da M n^2-platzbeschränkt ist). Und es gilt:

$x \in L \iff$ in $G_{M,x}$ gibt es einen in C_{start} beginnenden Pfad der Länge $\leq n^{12}$ zu einer akzeptierenden Konfiguration

$\iff$ (*) $\begin{cases} \text{es gibt Konfigurationen} \\ C_0, C_1, C_2, \ldots, C_{n^6}, \text{ so dass gilt:} \\ C_0 = C_{start},\ C_{n^6} \text{ ist akzeptierend und} \\ \text{für } \underline{\text{jedes}}\ i \in \{1, \ldots, n^6\} \text{ gilt: } M \text{ kann in} \\ \leq n^6 \text{ Schritten von Konfiguration } C_{i-1} \\ \text{zu Konfiguration } C_i \text{ gelangen.} \end{cases}$

(*) kann von einer ATM in Zeit $O(n^8)$ wie folgt berechnet werden:
Eine ATM $\tilde{M}$ kann durch Nutzen von mit "$\exists$" markierten Zuständen die Konfigurationen $C_0, C_1, \dots, C_{n^6}$ raten und macht dabei $O(n^6 \cdot n^2) = O(n^8)$ Schritte (da jede Konfiguration durch $O(n^2)$ Bits repräsentiert wird).
Dann nützt $\tilde{M}$ Zustände, die mit "$\forall$" markiert sind um ein $i \in \{1, \dots, n^6\}$ zu raten und überprüft dann, ob die Original-TM M in $\leq n^6$ Schritten von C_{i-1} zu C_i gelangt (dazu simuliert $\tilde{M}$ einfach n^6 Schritte von M).
Insgesamt zeigt dies, dass $L \in \Sigma_2\mathrm{TIME}(n^8)$ ist. $\square$

<u>Theorem 5.22</u> (Zeit-Platz-Tradeoff für SAT;
Fortnow 1997 und
Fortnow, Lipton, van Melkebeek, Viglas 2000)

(a) $\mathrm{NTIME}(n) \not\subseteq \mathrm{TISP}(n^{1,2}, n^{0,2})$

(b) $\mathrm{SAT} \notin \mathrm{TISP}(n^{1,1}, n^{0,1})$

<u>Beweis:</u>

(a) Der Beweis beruht auf einer Kombination von Lemma 5.21, dem (ndet.) Zeithierarchiesatz und den beiden folgenden Behauptungen:

Behauptung 1:

Falls $NTIME(n) \subseteq DTIME(n^{1,2})$, so $\Sigma_2 TIME(n^8) \subseteq NTIME(n^{9,6})$

Behauptung 2:

Falls $NTIME(n) \subseteq TISP(n^{1,2}, n^{0,2})$, so $NTIME(n^{10}) \subseteq TISP(n^{12}, n^2)$

Unter Verwendung dieser beiden Behauptungen können wir aus der Annahme, dass $NTIME(n) \subseteq TISP(n^{1,2}, n^{0,2})$ folgendermaßen einen Widerspruch herleiten:

$$NTIME(n) \subseteq TISP(n^{1,2}, n^{0,2})$$

$$\underset{\text{Beh2}}{\Longrightarrow} NTIME(n^{10}) \subseteq TISP(n^{12}, n^2) \underset{\text{Lemma 5.21}}{\subseteq} \Sigma_2 TIME(n^8) \underset{\text{Beh1, da } TISP(n^{1,2}, n^{0,2}) \subseteq DTIME(n^{1,2})}{\subseteq} NTIME(n^{9,6})$$

Somit: $NTIME(n^{10}) \subseteq NTIME(n^{9,6})$

↯ Widerspruch zum (ndet.) Zeithierarchiesatz.

Um den Beweis von Theorem 5.22 (a) abzuschließen müssen nur noch Beh 1 und Beh 2 bewiesen werden.

Beweis von Behauptung 1:

Sei $L \in \Sigma_2 TIME(n^8)$

Analog zum Beweis von Satz 5.18 sieht man leicht, dass es dann Konstanten $c, d \in \mathbb{N}$ und eine <u>det.</u> TM M gibt, die bei Eingabe von $\langle x, u, v \rangle$ höchstens $O(|x|^8)$ Schritte läuft,

so dass f.a. $x \in \{0,1\}^*$ gilt:

$$x \in L \iff \exists u \in \{0,1\}^{c\cdot|x|^8}\ \forall v \in \{0,1\}^{d\cdot|x|^8} : M(\langle x,u,v\rangle) = 1$$

$$\iff \exists u \in \{0,1\}^{c\cdot|x|^8}\ \underline{\text{nicht}}\ \exists v \in \{0,1\}^{d\cdot|x|^8} : M(\langle x,u,v\rangle) = 0$$

$$\iff \exists u \in \{0,1\}^{c\cdot|x|^8} : \langle x,u\rangle \notin L', \quad (*)$$

wobei $L' := \{ \langle x,u\rangle : x \in \{0,1\}^*,\ u \in \{0,1\}^{c\cdot|x|^8}$ und $\exists v \in \{0,1\}^{d\cdot|x|^8}$ s.d. $M(\langle x,u,v\rangle) = 0 \}$

<u>klar</u>: $L' \in \mathrm{NTIME}(n)$ (da M det. ist, höchstens $O(|x|^8)$ Schritte läuft und $n := |\langle x,u\rangle| = \Theta(|x|^8)$ ist.

Gemäß Voraussetzung gilt $\mathrm{NTIME}(n) \subseteq \mathrm{DTIME}(n^{1,2})$.
Daher ist $L' \in \mathrm{DTIME}(n^{1,2})$ und wird durch eine $O(n^{1,2})$-zeitbeschränkte <u>det</u> TM M' entschieden.

Mit (*) folgt:

$$x \in L \iff \exists u \in \{0,1\}^{c\cdot|x|^8} : \langle x,u\rangle \notin L'$$

$$\iff \exists u \in \{0,1\}^{c\cdot|x|^8} : M'(\langle x,u\rangle) = 0$$

Somit kann L von einer ndet TM entschieden werden, die zunächst ein $u \in \{0,1\}^{c\cdot|x|^8}$ rät und dann M' mit Eingabe $\langle x,u\rangle$ simuliert.
M' läuft bei Eingabe $\langle x,u\rangle$ höchstens $O(|\langle x,u\rangle|^{1,2})$ Schritte.
Beachte: $|\langle x,u\rangle|^{1,2} = \left(O(|x|^8)\right)^{1,2} = O\left((|x|^8)^{1,2}\right) = O(|x|^{9,6})$

Somit ist $L \in \mathrm{NTIME}(n^{9,6})$

$\square$ Beh 1

Beweis von Behauptung 2:

Wir nützen ein einfaches Padding-Argument: Sei $L \in NTIME(n^{10})$ und sei M eine n^{10}-zeitbeschränkte NDTM, die eine Sprache L entscheidet.

Sei $L' := \{ \langle x, 1^{(|x|^{10})} \rangle : x \in L \}$.

Aus M erhält man leicht eine $O(n)$-zeitbeschränkte NDTM M', die L' entscheidet. Somit ist $L' \in NTIME(n)$

$$L' \in NTIME(n) \underset{\text{Voraussetzung}}{\subseteq} TISP(n^{1,2}, n^{0,2}).$$

Sei M'' eine $n^{1,2}$-zeit- und $n^{0,2}$-platzbeschränkte det TM, die L' entscheidet. D.h. bei Eingabe von $\langle x, 1^{(|x|^{10})} \rangle$ macht M'' nur $O((|x|^{10})^{1,2}) = O(|x|^{12})$ Schritte und verbraucht Platz $O((|x|^{10})^{0,2}) = O(|x|^2)$.

Man kann M'' leicht zu einer $O(n^{12})$-zeit- und $O(n^2)$-platzbeschränkten det. TM umbauen, die bei Eingabe von $x \in \{0,1\}^*$ zunächst $\langle x, 1^{(|x|^{10})} \rangle$ konstruiert und dann M'' bei Eingabe $\langle x, 1^{(|x|^{10})} \rangle$ simuliert und dadurch entscheidet, ob $x \in L$ ist.

Insgesamt ist also $L \in TISP(n^{12}, n^2)$

Wir haben also gezeigt, dass $NTIME(n^{10}) \subseteq TISP(n^{12}, n^2)$ ist

$\square$ Beh 2

Insgesamt beendet dies den Beweis von Theorem 5.22 (a).

(b) <u>zu zeigen</u>: SAT $\notin$ TISP$(n^{1,1}, n^{0,1})$

<u>Beweisskizze</u>:

Angenommen, SAT $\in$ TISP$(n^{1,1}, n^{0,1})$.
Wir wollen zeigen, dass dann NTIME$(n) \subseteq$ TISP$(n^{1,2}, n^{0,2})$ ist
— was im Widerspruch zu (a) steht.

Sei dazu $L \in$ NTIME(n). <u>zu zeigen</u>: $L \in$ TISP$(n^{1,2}, n^{0,2})$.

<u>Idee</u>:
- Sei M eine det. $n^{1,1}$-Zeit- und $n^{0,1}$-Platzbeschränkte TM, die SAT entscheidet.
- Sei f eine geeignete Reduktion von L auf SAT.
- Sei M' eine det. TM, die bei Eingabe von $x \in \{0,1\}^*$ die Berechnung von M bei Eingabe $\varphi_x := f(x)$ simuliert
- <u>Klar</u>: M' entscheidet L
  <u>Ziel</u>: Führe die Details in der Konstruktion von M' so aus, dass M' $n^{1,2}$-Zeit- und $n^{0,2}$-platzbeschränkt ist

<u>Einige Details</u>:

- <u>Zur "geeigneten Reduktion f"</u>:
  1) Zu $L \in$ NTIME(n) gibt es eine <u>stereotype</u> 2-Band-NDTM N die L in Zeit $O(n \cdot \log n)$ entscheidet
  (Hennie und Stearns, 1966)
  2) Unter Betrachtung von N erhält man für jedes $n \in \mathbb{N}$ eine CNF-Formel ψ_n s.d. gilt:
     - ψ_n hat Länge $\leq O(n \cdot (\log n)^2)$
     - ψ_n benutzt Variablen $u_1, \dots, u_n$ und weitere Variablen $y_1, y_2, \dots$
     - Ist für $x \in \{0,1\}^n$ φ_x die Formel, die aus ψ_n entsteht, indem die Variablen $u_1, \dots, u_n$ mit den Werten $x_1, \dots, x_n$ belegt werden, so gilt: $x \in L \iff \varphi_x$ ist erfüllbar.

  Und es gibt eine Zahl $r \in \mathbb{N}$ und einen Algorithmus A der bei Eingabe von n und j das j-te Bit der Repräsentation von ψ_n in Zeit und auf Platz $(\log n)^r$ berechnet

$\Rightarrow$ Es gibt einen Algorithmus A', der bei Eingabe von $x \in \{0,1\}^n$ und j das j-te Bit der Repräsentation von φ_x in Zeit und auf Platz $(\log n)^r$ berechn

<u>Beachte</u>: A' benötigt dazu "random access" auf die Bits von – d.h. bei Angabe von i kann A' in einem Schritt auf x_i zugreifen.

Dieses Berechnungsmodell wird durch so genannte <u>Random-Access-Turingmaschinen</u> modelliert.

Die Aussage von Teil (<u>a</u>) gilt auch, wenn die Klasse $\mathrm{TISP}(n^{1,2}, n^{0,2})$ bezüglich dieser Random-Access-Turingmaschinen definiert wird

- <u>Zur Konstruktion von M'</u>:

Ähnlich wie beim Nachweis der Transitivität von $\le_\ell$ (logspace-Reduktionen, vgl. Kapitel 4):

M' schreibt φ_x auf ein "virtuelles Eingabeband", merkt sich die aktuelle Kopfposition und nutzt in jedem Schritt, mit dem sie M bei Eingabe φ_x simuliert, den Algorithmus A' um das aktuell von φ_x zu lesende Bit zu ermitteln.

M' ist somit eine det. Random-Access-Turingmaschine, die L entscheidet und bei Eingabe von $x \in \{0,1\}^n$ höchstens

- $|\varphi_x|^{1,1} \cdot (\log n)^r$ Berechnungsschritte durchführt

$$\le \left(c \cdot n \cdot (\log n)^2\right)^{1,1} \cdot (\log n)^r \le \tilde{c} \cdot n^{1,1} \cdot (\log n)^{2,2+r} \le \tilde{c} \cdot n^{1,}$$

und

- $|\varphi_x|^{0,1} + (\log n)^r$ Platz benutzt

$$\le \left(c \cdot n \cdot (\log n)^2\right)^{0,1} + (\log n)^r \le \tilde{c} \cdot n^{0,1} \cdot (\log n)^{0,2} + (\log n)^r \le \tilde{c} \cdot n^{0,}$$

(für geeignete Konstanten $c, \tilde{c} \in \mathbb{N}$).

Somit gehört L zur Variante der Klasse TISP$(n^{1.2}, n^{0.2})$, die mittels Random-Access-TMs definiert ist. Der Widerspruch folgt mit der entsprechenden Variante von (a). $\square$

Details zum Beweis von Theorem 5.22 finden sich in dem Artikel

"Time-Space Lower Bounds for Satisfiability"
von L. Fortnow, R. Lipton, D. van Melkebeek, A. Viglas.
Journal of the ACM, 52: 835-865, 2005.