

(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-TM'en

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

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

$$\Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \text{ nicht } \exists u_2 \in \{0,1\}^{q(|x|)} : M(\langle x, u_1, u_2 \rangle) = 0$$

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

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

Klar: $L' \in NP$ 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.

Ziel: 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

$$\langle x, u_1 \rangle \in L' \Leftrightarrow f(\langle x, u_1 \rangle) \in \text{SAT}$$

.."
 $\Psi_{\langle x, u_1 \rangle}$

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

Klar: $\Phi_{\langle x, u_1 \rangle} \in \Sigma_1\text{-SAT} \Leftrightarrow \langle x, u_1 \rangle \in L' \quad (**)$

Um zu zeigen, dass $L \in NP^{\Sigma_2^P}$ ist, konstruieren wir eine ndet. Orakel-TM \tilde{M} , die bei Eingabe von $x \in \{0,1\}^*$ folgendermaßen vorgeht:

- 1) wähle ein $u_1 \in \{0,1\}^{q(|x|)}$
- 2) nutze die Polynomzeit-Reduktion f , um die QBF $\Phi_{\langle x, u_1 \rangle}$ zu konstruieren
- 3) befrage das Orakel, ob $\Phi_{\langle x, u_1 \rangle} \in \Sigma_2^P$ -SAT ist
- 4) akzeptiere genau dann, wenn das Orakel "nein" antwortet.

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

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

$$\begin{aligned}
 x \in L & \stackrel{(*)}{\iff} \exists u_1 \in \{0,1\}^{q(|x|)} : \langle x, u_1 \rangle \notin L' \\
 & \stackrel{(**)}{\iff} \exists u_1 \in \{0,1\}^{q(|x|)} : \Phi_{\langle x, u_1 \rangle} \notin \Sigma_2^P\text{-SAT} \\
 & \iff \tilde{M} \text{ akzeptiert } x
 \end{aligned}$$

Somit ist $L \in NP^{\Sigma_2^P}$.

" \supseteq ": Sei nun $L \in NP^{\Sigma_2^P}$ durch eine ndet. Polynomialzeit-Orakel-TM N mit Orakel Σ_2^P -SAT.

Beachte: Im Laufe ihrer Berechnung kann N polynomial 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 w in $\{0,1\}^t$ getestet.
 ndet. Entscheidungen geraten sowie alle Orakel-Anfragen,
 die N im Laufe der Berechnung stellt und alle
 Antworten, die das Orakel gibt.

Genauer: Sei $q: \mathbb{N} \rightarrow \mathbb{N}$ die (polynomielle) Zeitschranke von N
 D.h.: N läuft bei Eingabe x maximal $t := q(|x|)$ Schritte.

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

Für $1 \leq i \leq t$ sei $\Phi_i := \exists \bar{u}_i \varphi_i(\bar{u}_i)$ die i -te Orakel-Anfrage,
 die N bei Eingabe x unter Verwendung von \bar{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. $a_i = 1 \Rightarrow \varphi_i$ ist erfüllbar)

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

$x \in L$

$$\Leftrightarrow \exists \bar{c} \in \{0,1\}^t, \exists (\Phi_i = \exists \bar{u}_i \varphi_i(\bar{u}_i))_{1 \leq i \leq t} \exists a_1 \dots a_t \in \{0,1\}^t$$

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

Für alle $i = 1, \dots, t$ gilt:

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

und der Oracle-Antworten w_1, \dots, w_i

$\Phi_i = \exists u_i \forall v_i (i(u_i))$ die i -te Oracle-Anfrage, die

N bei Eingabe x stellt, und N beendet seine Berechnung im Zustand q_{accept}

Man sieht leicht, dass Φ von einer det. polynomialzeit-TM durchgeführt werden kann.

Insgesamt erhalten wir daher gemäß Def 5.2

(und Wahl eines geeigneten Padding, damit das unter "3" quantifizierte Wort genauso lang ist wie das unter "4" 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 $Q \setminus \{q_{\text{halt}}, q_{\text{accept}}\}$ 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 für Akzeptanz führt
- für eine Konfiguration, deren Zustand mit "∀" markiert ist, müssen alle Nachfolgekonfigurationen für 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 \Rightarrow 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 \Leftrightarrow die Startkonfiguration von M bei Eingabe x hat in $G_{M,x}$ die Markierung ACCEPT.

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

Definition 5.11 (Alternierende Zeit $\text{ATIME}(T(n))$)

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

(a) 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 $\leq T(|x|)$

(d.h. M läuft bei jeder Wahl von ndt Entscheidungen höchstens $T(|x|)$ Schritte).

(b) Die Klasse $\text{ATIME}(T(n))$ 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 $S: \mathbb{N} \rightarrow \mathbb{N}$

(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 $L \in \{0,1\}^*$, die von einer $O(S(n))$ -platzbeschränkten ATM entschieden werden können.

Satz 5.13

Sei $T: \mathbb{N} \rightarrow \mathbb{N}$ mit $T(n) \geq n$ zeitkonstruierbar und

sei $S: \mathbb{N} \rightarrow \mathbb{N}$ mit $S(n) \geq \log n$ platzkonstruierbar.

Dann gilt:

(a) $ATIME(T(n)) \subseteq SPACE(T(n))$

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

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

(d) $DTIME(T(n)) \subseteq ASPACE(\log T(n))$

Beweis (Skizze):

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

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

Details:

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

Um zu zeigen, dass $L \in \text{ATIME}(S(n)^2)$ ist, nutzen wir die folgende "parallele Implementierung" des im Beweis des Satzes von Savitch verwendeten rekursiven Prozedurs $\text{REACH?}(u, v, i)$, die entscheidet, ob es im Konfigurationsgraph $G_{M, i}$ 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 $G_{M, i}$ eine Kante von u nach v gibt, STOPP mit Antwort "ja"

Sonst: STOPP mit Antwort "nein"

Falls $i > 0$:

Wähle einen beliebigen Knoten w von $G_{M, i}$
(Bem.: Unsere ATM verwendet dazu $O(S(n))$ Schritte und nutzt Zustände die mit "E" markiert sind)

Geh dann in einen mit "H" markierten Zustand, von dem aus mit der ndet Entscheidung "0" bzw "1"

Anfrage von $\text{PARALLEL-REACH?}(u, w, i-1)$ bzw. $\text{PARALLEL-REACH?}(w, v, i-1)$ gestartet werden.

Mit der gleichen Analyse wie im Beweis des Satzes von Savitch erhält man, dass

$\text{PARALLEL-REACH?}(C_{\text{start}}, C_{\text{accept}}, \log(|V(G_{M, i})|))$

durch eine $O(|V(G_{n+1})| \cdot S(n))$ -zeitbeschränkte ATM implementiert werden kann.

Wegen $\log(|V(G_{n+1})|) = O(S(n))$ erhalten wir, dass $L \in \text{ATIME}(S(n)^2)$.

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

Details: Übung

(d) Hier ohne Beweis.

(Details werden im nächsten Kapitel nachgeliefert).

D

Definition 5.14 (Alternierende Komplexitätsklassen)

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

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

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

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

Theorem 5.16 (PSPACE vs alternierende Zeit)

(a) $AL = P$

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

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

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

Beweis:

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

• $P \stackrel{\text{Satz 5.13(d)}}{\subseteq} ASPACE(O(\log n)) = AL \quad \checkmark$

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

• $PSPACE \stackrel{\text{Def}}{=} \bigcup_{c>0} SPACE(n^c) \stackrel{\text{Satz 5.14(b)}}{\subseteq} \bigcup_{c>0} ATIME((n^c)^2) = \bigcup_{d>0} ATIME(n^d) = AP \quad \checkmark$

(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 $\Gamma_{n,x}$ nur $(k-1)$ -mal zwischen mit "∃" markierten und mit "∀" 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 $T: \mathbb{N} \rightarrow \mathbb{N}$.

(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 "∃" markiert ist und bei der für alle Eingaben $x \in \{0,1\}^*$ gilt:

Entlang jedes in C_{start} beginnender Pfads in $A_{n,x}$ wird höchstens $(k-1)$ -mal zwischen mit "E" und mit "V" markierten Zuständen gewechselt.

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

Satz 5.18 ($\Sigma_k^P = \Sigma_k TIME(n^{O(k)})$)

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

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

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

Beweis: Übung!

5.5 Zeit vs. Alternierungen: Zeit-Platz-Tradeoffs

Definition 5.10 (Zeit-Platz-Klasse $TISP(T(n), S(n))$)

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

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

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

Beobachtung 5.20

$$\begin{aligned} \text{TISP}(T(n), S(n)) &\subseteq \text{DTIME}(T(n)) \quad \text{und} \\ \text{TISP}(T(n), S(n)) &\subseteq \text{SPACE}(S(n)) \end{aligned}$$

Lemma 5.21

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

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 es gibt Konfigurationen $C_0, C_1, C_2, \dots, C_{n^6}$, so dass gilt:

\circledast $C_0 = C_{\text{start}}$, C_{n^6} ist akzeptierend und für jedes $i \in \{1, \dots, n^6\}$ gilt: M kann in $\leq n^6$ Schritten von Konfiguration C_{i-1} zu Konfiguration C_i gelangen.

⊕ kann von einer ATM in Zeit $O(n^2)$ wie folgt berechnet werden: 142

Eine ATM \tilde{M} kann durch Nutzen von mit "E" markierten Zuständen die Konfigurationen C_0, C_1, \dots, C_n 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 nutzt \tilde{M} Zustände, die mit "V" 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 $L \in \mathbb{Z}_2\text{TIME}(n^8)$ ist. □

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

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

(b) $\text{SAT} \not\subseteq \text{TISP}(n^{1/1}, n^{0,1})$

Beweis:

(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 $\Sigma_2 TIME(n^8) \subseteq NTIME(n^{3,6})$

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:

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

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

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

↳ Widerspruch zum (indet.) 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 det. TM M gibt, die bei Eingabe von $\langle x, u, v \rangle$ höchstens $O(|x|^{18})$ Schritte läuft, so dass f.a. $x \in \{0,1\}^*$ gilt:

$$\begin{aligned}
 x \in L & \Leftrightarrow \exists u \in \{0,1\}^{c \cdot |x|^8} \forall v \in \{0,1\}^{d \cdot |x|^8} : M(\langle x, u, v \rangle) = 1 \\
 & \Leftrightarrow \exists u \in \{0,1\}^{c \cdot |x|^8} \text{ nicht } \exists v \in \{0,1\}^{d \cdot |x|^8} : M(\langle x, u, v \rangle) = 0 \\
 & \Rightarrow \exists u \in \{0,1\}^{c \cdot |x|^8} : \langle x, u \rangle \notin L, \quad (*)
 \end{aligned}$$

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

Klar: $L \in \text{NTIME}(n)$ (da M det. ist, höchstens $O(|x|^8)$ Schritte, läuft und $n := |\langle x, u \rangle| = \Theta(|x|^8)$ ist.)

Gemäß Voraussetzung gilt $\text{NTIME}(n) \subseteq \text{DTIME}(n^{1/2})$.

Daher ist $L \in \text{DTIME}(n^{1/2})$ und wird durch eine $O(n^{1/2})$ -zeitbeschränkte det TM M' entschieden.

Mit $(*)$ folgt:

$$\begin{aligned}
 x \in L & \Leftrightarrow \exists u \in \{0,1\}^{c \cdot |x|^8} : \langle x, u \rangle \notin L' \\
 & \Leftrightarrow \exists u \in \{0,1\}^{c \cdot |x|^8} : M'(\langle x, u \rangle) = 0
 \end{aligned}$$

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.

$$\text{Beachte: } |\langle x, u \rangle|^{1/2} = (O(|x|^8))^{1/2} = O((|x|^8)^{1/2}) = O(|x|^{9/6})$$

Somit ist $L \in \text{NTIME}(n^{3/6})$

□ Beh 1

Beweis von Behauptung 2:

Wir nutzen ein einfaches Padding-Argument: Sei $L \in \text{NTIME}(n^{10})$
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 $L' \in \text{NTIME}(n)$

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

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 $O((|x|^{10})^{1/2}) = O(|x|^{1/2})$ Schritte und
verbraucht Platz $O((|x|^{10})^{0/2}) = O(|x|^2)$.

Man kann M'' leicht zu einer $O(n^{1/2})$ -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 $L \in \text{TISP}(n^{1/2}, n^2)$

Wir haben also gezeigt, dass $\text{NTIME}(n^{10}) \subseteq \text{TISP}(n^{1/2}, n^2)$ ist

□ Beh 2

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