Vorlesung Komplexitätstheorie
7. November 2006
Relativierung der PvsNP-Frage

Theorem 6 Es gibt ein Orakel $B \subseteq \{0,1\}^*$ mit P $\neq$ NP
Beweis:
Man definiere zu einer Sprache $B \subseteq \{0,1\}^*$ eine Menge von Strings
$L_B := \{0^n; \exists x \in B,\ |x| = n\}$.

Offenbar ist $L_B \in NP^B$, dieser nichtdeterministische Algorithmus erkennt L_B:
input x
falls $x \notin 0^k$ verwerfe
rate w mit $|w| = |x|$
akzeptiere, falls $w \in B$, sonst verwerfe.

Es bleibt eine Sprache B zu konstruieren, so dass L_B nicht in P^B liegt.
Die Behauptung wird mit der Technik der Diagonalisierung bewiesen, dazu brauchen wir eine Aufzählung $T_1, T_2, \ldots, T_i, \ldots$ von Orakel-DTM so dass für alle Orakel $X \subseteq \{0,1\}^*$ gilt:

- $P^X = \{L(T_1^X), L(T_2^X), \ldots, L(T_i^X), \ldots\}$
- für alle $y \in \{0,1\}^*$ und $k \in \mathbb{N}$ hält T_k^X bei input y nach höchstens $|y|^k + k$ Schritten.

Wir definieren B induktiv $B = \bigcup_{i \in \mathbb{N}} B_i$ zusammen mit einer Schranke $n_i \in \mathbb{N}$ so dass für alle $x \in B_i$ mit $|x| < n_i$ gilt:

- Induktionsanfang: $B_0 := \emptyset, n_0 := 0$
- Induktionsschritt: $n_{i+1} := min\{m; m > n_i^i + i \text{ und } 2^m > m^{i+1} + i + 1\}$
  (damit erfllt n_{i+1} folgende Gleichungen:
  $n_{i+1} > n_i^i + i$ und $2^{i+1} > n_{i+1}^{i+1} + (i+1)$.
  n_i ist so gewählt, dass die (i-1)-te Maschine nicht alle Wörter schreiben kann und die i-te Maschine keine Zeit hat alle Wörter durchzutesten.)

  Zur Definition von B_{i+1} betrachte Berechnung von $T_{i+1}^{B_i}$ bei input $0^{n_{i+1}}$ und setze B_{i+1} genau andersherum:
  Falls diese akzeptiert, setze $B_{i+1} := B_i$.
  Falls die Maschine nicht akzeptiert wählen wir ein y_{i+1} der Länge $|y_{i+1}| = n_{i+1}$ für das die Berechnung nicht das Orakel befragt.
  So ein y existiert immer, da $T_{i+1}^{B_i}$ nur $n_{i+1}^{i+1} + (i+1) < 2^{n_{i+1}}$ Anfragen an das Orakel stellt.
  Setze $B_{i+1} := B_i \cup \{y_{i+1}\}$.

Beobachtung: Für alle $i \in \mathbb{N}$ ist die Berechnung von T_i^B bei Eingabe 0^{n_i} gleich der von $T_i^{B_{i-1}}$ bei 0^{n_i}.
Begründung: Das einzige Wort y_i in $B_i \setminus B_{i-1}$ ist so gewählt, dass $T_i^{B_i}$ bei Eingabe 0^{n_i} das Orakel nicht befragt. Es gibt somit keinen Unterschied in der Berechnung der Maschinen $T_i^{B_{i-1}}$ und $T_i^{B_i}$.
Alle anderen Wörter $w \in B \setminus B_i$ sind zu lang, um von T_i^B bei input 0^{n_i} befragt

zu werden: $|w| \geq n_{i+1} > n_i^i + i >$ Laufzeit von T_i^B bei input 0^{n_i}.

Zu zeigen: $L_B \notin P^B$.
Widerspruchsbeweis, sei also $L_B \in P^B$.
Dann ist $L_B = L(T_j^B)$ für ein $j \in \mathbb{N}$.
Betrachte die Berechnung der Maschinen T_j^B und $T_j^{B_j}$ bei Eingabe 0^{n_j}:
$T_j^{B_j}(0^{n_j}) = T_j^B(0^{n_j})$.
Falls die Maschinen akzeptieren, dann gibt es in B kein Wort der Länge n_j, also $0^{n_j} \notin L_B$.
Falls die Maschinen verwerfen, dann existiert ein $y_j \in B$ mit $|y_j| = n_j$, also $0^{n_j} \in L_B$.
Also akzeptiert T_j^B die Sprache L_B nicht, Widerspruch.
Somit ist $L_B \notin P^B$, q.e.d.

Zwischen den Mengen P, NP und coNP besteht die Relation P $\subseteq$ NP$\cap$coNP. Daraus ergeben sich die Fragen, ob NP = coNP und ob P = NP$\cap$coNP ?
Ähnlich wie im Theorem 6 lassen sich Orakel C, D, E konstruieren, so dass folgende Verhältnisse zwischen P, NP und coNP bzgl. der Orakel bestehen:
$NP^C = coNP^C$, aber $P^C \neq coNP^C$
$NP^D \neq coNP^D$, aber $P^D = NP^D \cap coNP^D$
$NP^E \neq coNP^E$, aber $P^E \neq NP^E \cap coNP^E$
Damit können diese Fragen von solchen Beweisideen, die sich auf Orakel-TM fortsetzen lassen, nicht beantwortet werden.

Kapitel 4 Platzkomplexität

Der Begriff Speicher bezieht sich in der Vorlesung auf den Arbeitsspeicher. Im passenden Maschinenmodell werden die Ein- und Ausgabebänder ignoriert, deswegen dürfen diese nur zum Lesen bzw. Schreiben benutzt werden.

Wir definieren einen platzbeschränkten Akzeptor als Akzeptor-TM

- einem Eingabeband, welches nicht überschrieben werden kann (read-only)
- mindestens zwei Arbeitsbändern

Ein platzbeschränkter Übersetzer hat zusätzlich ein Ausgabeband. Dieses kann nur beschrieben werden, d.h. der Schreib/Lese-Kopf darf auf diesem Band nur stehenbleiben oder nach rechts gehen.

Definition des Speicherverbrauchs
Für eine DTM T sei $SPACE_T(x)$:= Anzahl der Bandzellen auf den Arbeitsbändern, die T bei input x beschreibt.
Für eine NTM T sei

$$NSPACE_T(x) := \begin{cases} \text{minimale Anzahl der Bandzellen einer akzeptierenden} \\ \text{Berechnung, falls } x \in L(T). \\ \text{sonst: minimale Anzahl der Bandzellen einer} \\ \text{Berechnung überhaupt.} \end{cases}$$

Definition der Komplexitätklassen

$$
\begin{aligned}
L &:= \{A \in \Sigma^*; \exists \,\text{DTM}\, T \,\text{mit}\, L(T) = A \,\text{und}\, SPACE_T(x) \leq O(\log(|x|))\} \\
PSPACE &:= \{A \in \Sigma^*; \exists \,\text{DTM}\, T \,\text{mit}\, L(T) = A \,\text{und}\, SPACE_T(x) \leq |x|^{O(1)}\} \\
NL &:= \{A \in \Sigma^*; \exists \,\text{NTM}\, T \,\text{mit}\, L(T) = A \,\text{und}\, NSPACE_T(x) \leq O(\log(|x|))\} \\
L &:= \{A \in \Sigma^*; \exists \,\text{NTM}\, T \,\text{mit}\, L(T) = A \,\text{und}\, NSPACE_T(x) \leq |x|^{O(1)}\} \\
FL &:= \{f : \Sigma^* \to \Sigma^*; \text{f wird von DTM T berechnet mit} SPACE_T(x) \leq O(\log(|x|))\}
\end{aligned}
$$