Kap 6 Komplexitätstheorie

Def 6.1 (NDTM) Das Tupel NDTM = $(Q, \Sigma, \Gamma, \Delta, q_0, B, \{q_E\})$ heißt Nichtdeterministische Turingmaschine, wobei

$Q \triangleq$ Zustandsmenge
$\Sigma =$ Eingabealphabet
$\Gamma =$ Bandalphabet
$\Delta : (Q \times \Gamma) \to \mathcal{P}(Q \times \Gamma \times \{L, R, N\})$
$q_0 \triangleq$ Startzustand
$B =$ Leerzeichen
$q_E \triangleq$ Finalzustand

Def. 6.2 (Akzeptanz auf NDTM) Eine Sprache $L \subseteq \Sigma^*$ heißt akzeptiert von einer NDTM, wenn es für jede beliebige Eingabe $v \in \Sigma^*$ eine Übergangsfolge gibt, dass NDTM mit True anhält, falls $v \in L$.

Def. 6.3 (Zeitkomplexität einer TM) Die Zeitkomplexität $T_i(n)$ einer Turingmaschine TM_i ist definiert als die Anzahl der Takte n, die TM_i zur Berechnung der Funktion benötigt:

$$T_i(n) = \max\{m \mid \exists w \in \Sigma^* \text{ mit } |w| = n,\ TM_i \text{ rechnet } m \text{ Schritte bei Eingabe } w\}$$

Für NDTM wird der kürzeste Weg betrachtet.

Die Zeitkomplexität $T(n)$ eines Problems Π ist definiert durch $T(n) = \min_i T_i(n)$ für alle TM$_i$, die das Problem Π löst.

<u>Def 6.4 (Zeitbeschränktheit)</u> Eine Turingmaschine TM$_i$ heißt $f(n)$-zeitbeschränkt, wenn für die Zeitkomplexität $T_i(n)$ mit $n > n_0$ gilt $T_i(n) \leq f(n)$.

<u>Bem</u>: <u>These von Edmonds</u>
Probleme mit $T(n) \leq c \cdot n^k$ mit c, k konstant gelten als praktisch lösbar.
Alle anderen gelten als praktisch unlösbar (?)

<u>Def 6.5 (Klassen zeitbeschränkter TM)</u>

$DTIME(f(n)) = \{L \mid \exists f(n)\text{-zeitbeschränkte DTM, die die Sprache } L \text{ entscheidet}\}$

$NTIME(f(n)) = \{L \mid \exists f(n)\text{-zeitbeschränkter NDTM, die die Sprache } L \text{ akzeptiert}.$

<u>Lemma 6.6</u> Es gilt $DTIME(f(n)) \subseteq NTIME(f(n))$

Def 6.7 (Klasse P und Klasse NP)

Sei p ein Polynom n-te Grades. Dann definieren wir die Klasse P

$$P = \bigcup_p DTIME(p(n))$$

und die Klasse NP

$$NP = \bigcup_p NTIME(p(n))$$

Lemma 6.8 Es gilt $P \subset NP$

Lemma 6.9 Die kontextfreien Sprachen liegen in P.

Beweis: Der CYK-Algorithmus hat Laufzeit $O(n^3)$.

Es gibt eine Vielzahl von Problemen, für die schnellste deterministische TM exponentielle Rechenzeit benötigt, jedoch kann man leicht eine entsprechende NDTM konstruieren, die in polynomialer Zeit die Sprache akzeptiert.

Es stellt sich die Frage, ob $P = NP$.

Satz 6.10 Für jede Sprache $L \in NP$ gibt es ein Polynom p und eine deterministische TM T, so dass T in der Zeit $2^{p(n)}$ die Sprache entscheidet.

Bew: Reischuk 1990 Einführung in die Komplexitätstheorie.

Def. 6.11 (polynomiell reduzierbar $\leq_{pol}$)

Eine Sprache L heißt polynomiell reduzierbar auf L_0 ($L \leq_{pol} L_0$) wenn eine mittels einer DTM in polynomieller Zeit berechenbare Funktion f gibt so dass für alle Wörter w gilt $w \in L \Leftrightarrow f(w) \in L_0$.

Def. 6.12 (NP-vollständig, Klasse NPC, NP-hard (NP-schwer))

Eine Sprache L_0 heißt NP-vollständig, wenn gilt:

1. $L_0 \in NP$
2. $\forall L \subseteq \Sigma^* : L \in NP \Rightarrow L \leq_{pol} L_0$

Eine Sprache L_0 heißt NP-hard, wenn es ein $L \in NP$ mit $L \leq_{pol} L_0$ gibt.

Die Klasse der NP-vollständigen Sprachen wird mit NPC bezeichnet.

Ansätze für den Beweis $P \stackrel{?}{=} NP$.

Satz 6.13 Sei L_0 eine NP-vollständige Sprache. Dann gilt $L_0 \in P \Leftrightarrow P = NP$

Bew:

"$\Leftarrow$" wegen $L_0 \in NPC$ folgt $L_0 \in NP$. mit $P = NP$ folgt $L_0 \in P$

"$\Rightarrow$" Sei $L \in NP$ beliebig. Dann ist zu zeigen, dass $L \in P$.
wegen $L_0 \in NPC$ ist $L \leq_{pol} L_0$ und es gilt

$\forall w (w \in \Sigma^* : w \in L \Leftrightarrow f(w) \in L_0$, für eine deterministisch in Polynomialzeit berechenbare Funktion f.

Mit der Voraussetzung $L_0 \in P$ kann die Frage $f(w) \overset{?}{\in} L_0$ in Polynomialzeit entschieden werden. Wegen $v \in L \Leftrightarrow f(w) \in L_0$ folgt $L \in P$.

Gelingt es also für ein einziges NP-vollständiges Problem zu zeigen, dass es in der Klasse P liegt, so ist $P = NP$.

Es gibt über 1000 bekannte NP-vollständige Probleme. Eines davon ist das sogenannte SAT-Problem.

<u>Def. 6.14</u> (<u>SAT-Problem</u>) Das Satisfiability-Problem (SAT-Problem) ist wie folgt definiert.
Sei F eine Formel in konjunktiver Normalform mit Variablen $\{x_1 \ldots x_m\}$ $((x_1 \vee x_2 \vee \neg x_3) \wedge \ldots \wedge (x_i \vee x_j \ldots x_n))$ dann ist die zu entscheidende Frage, ob es eine Belegung für $(x_1 \ldots x_m)$ gibt, so dass F wahr wird.

<u>Satz 6.15</u> (Satz von Cook) Das SAT-Problem ist NP-vollständig.

Nun stellt sich die Frage ob es „schwierigere Probleme" als NP-vollständige Probleme gibt. Übertragen auf die Sprachtheorie heißt das: Gibt es eine entscheidbare Sprache, die nicht in der Klasse DTIME(P) liegt?

Die Antwort lautet ja

man kann sogar zeigen, dass es für entscheidbare Sprache überhaupt keine Zeitkomplexitätsschranken gibt.

Satz 6.16 Sei $f: \mathbb{N} \to \mathbb{N}$ eine turing-berechenbare Fkt. Dann gibt es eine Sprache $L_D \subseteq \{0,1\}^*$ mit $L_D \notin DTIME(f)$.

Beweis: Sei $T_{w_1}, T_{w_2}, \ldots$ eine effektive Nummerierung aller Turingmaschinen ($w_1, w_2, \ldots \in \{0,1\}^*$).
(→ Gödelisierung)

$$L_D(f) = \{w \mid w \text{ als Eingabe von } T_w \text{ wird nicht in } f(|w|) \text{ Schritten akzeptiert}\}$$

1. $L_D(f)$ ist entscheidbar:

Sei w eine Eingabe.
Die TM kann $f(|w|)$ bestimmen, und dann wie eine universelle TM $f(|w|)$ Schritte auf T_w ausführen.
Hat T_w akzeptiert, so akzeptiert TM nicht, sonst akzeptiert TM.

2. $L_D(f) \notin DTIME(f)$:

Wir nehmen an, $L_D(f)$ wäre für die Eingabe w in $f(|w|)$ Schritten entscheidbar. Dann gibt es eine f-zeitbeschränkte TM, die $L_D(f)$ akzeptiert.

Sei w die Nummer von TM in der effektiven Nummerierung der Turingmaschinen: $T_w = TM$.

TU hält für die Eingabe von w und höchstens $f(|w|)$ Schritten an.

Nun gibt es zwei Möglichkeiten:

1. w wird akzeptiert $\Rightarrow w \notin L_D(t)$.
2. w wird nicht akzeptiert (False) $w \in L_D(t)$

Beide Fälle sind ein Widerspruch zu der Annahme, dass TM = TU die Sprache $L_D(t)$ akzeptiert.

Einordnung der N bzw. NP in die Chomsky-Hierarchie

allg. Sprachen

$\cup$

entscheidbare Sprachen

$\cup$ $\quad$ $\supset$

NP

kontextsensitive Sprachen $\quad$ $\cup|$

P

$\cup$ $\quad$ $\cup$ $\{a^n b^n c^n \mid n \geq 1\}$

NDDA

$\cup$

DPDA

$\cup$

reguläre Sprachen