Merkblatt

Prof. Dr. W. Thomas, Berechenbarkeit und Komplexität, WS 2004/2005

Autor: Ulrich Loup

Es handelt sich hierbei um eine Zusammenfassung der wichtigsten Notationen, Definitionen und Sätze der Vorlesung. Anspruch auf Vollständigkeit und Richtigkeit wird nicht erhoben. Wünsche oder Bemerkungen an Ulrich.Loup@rwth-aachen.de.

1 Syntax und Semantik

1.1 Turing Maschine (TM)

- *Definition:* TM M ist 6-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_s)$, wobei Q endliche Zustandsmenge, Σ Eingabe-, Γ Arbeitsalphabet, δ die Übergangsfunktion, q_0 der Anfangszustand und q_s der Stoppzustand ist.
- $w_1 q_i w_2$ für $w_1, w_2 \in \Gamma$ heißt *Konfigurationswort* und $w_1 q_i w_2 \vdash w_1' q_j w_2'$ für $w_1, w_2 \in \Gamma$ *Folgekonfigurationswort.* Benutze $\vdash^*$ zum Überspringen einiger leicht einzusehender Konfigurationen.
- $q_i\ a\ b \star q_j \quad \forall a, b \in \Gamma$ mit $\star \in \{L, R, N\}$ heißt *Turingzeile*, alle Turingzeilen einer TM heißt *Turingtafel.*
- Bei formaler Konstruktion einer TM sollen immer auch einige Beispielkonfigurationsfolgen angegeben werden.

1.2 GOTO$_{\mathrm{m}}$-Programme

- *Registermaschine:* m Register bezeichnet mit X1,…,Xm, die vom Typ Integer ($\in \mathbb{N}$) sind, die Zeilen werden nummeriert.
- *Zuweisung:* j Xi:=Xi*1 //* $\in \{+, -\}$ mit $0 - r = 0$ für $x \in \mathbb{N}$ in Zeile j $\in [1, n]$
- *bedingter Sprung:* j if Xi=0 goto l else goto k //Zeilen j,l,k $\in [1, n]$
- Alleiniges goto l für l$\in$ $[1, n]$ ist erlaubt, desweiteren steht in Zeile n (letzte) stop und es gilt GOTO$_{m=\infty}$ =: GOTO.

1.3 WHILE$_{\mathrm{m}}$-Programme

- Maschine, Variablenbezeichnung, Zuweisungen sind wie bei GOTO$_m$-Programmen nur ohne Zeilennummerierung.
- Außerdem sind folgende Zuweisungen erlaubt: Xi:=0, Xi:=Xj, Xi:=Xj*Xk mit * $\in \{+, -, div, \cdot, mod\}$, und noch die Bedingungen(<Bed>): Xi>0, Xi=0
- Blöcke (<Block>):
  - <Wertzuseisung>
  - bedingte Ausdrücke: if <Bedingung> then <Block1> else <Block2> end
  - für if <Bed> then <Block1> else X1:=X1 end auch: if <Bed> then <Block1> end
  - Schleifen: while <Bedingung> do <Block> end und loop <Var> begin <Block> end
  - Es gilt bei mehreren Blöcken: <Block1>;<Block2>
- Notation: LOOP$_m$-Programme sind WHILE$_m$-Programme ohne while-Konstrukte.
- Notation: Bei WHILE^0/LOOP^0-Programmen sind nur Wertzuweisungen der Form Xi:=Xi*1 für * $\in \{+, -\}$ erlaubt.

1.4 Semantische Funktion

Sei P GOTO$_m$- oder WHILE$_m$-Programm.

- Die Ausgabe bzw. das Ergebnis $[P]$ von P ist der Wert des Registers `X1` nach Programmausführung.
- Semantische Funktion: $f_P^{(n)} := f_P^{(n)}(X_1, \ldots, X_n) = [P](X_1, \ldots, X_n, 0, \ldots, 0)$ für $1 < n < m$ ist die Ausgabe bzw. das Ergebnis von P für die Belegung `X1:=`X_1,...,`Xn:=`X_n.

2 Entscheidbarkeit/Unentscheidbarkeit

- *kanonische Reihenfolge:* zuerst nach Länge des Worts dann lexikographisch; $\delta(i) = i$-tes Wort in kanonischer Reihenfolge und $\gamma(\underline{k}_r \ldots \underline{k}_0) = \sum_{i=0}^{r} k_i m^i$ mit $\underline{k}_i \in \Sigma_m$ und $k_i \in [1, m]$ über dem Alphabet Σ_m mit $|\Sigma_m| = m$.
- Sprache $L \subseteq \Sigma^*$ ist *entscheidbar* $\iff \exists$ Entscheidungsalgorithmus A mit $A : w \to \begin{cases} \text{„ja“} & w \in L \\ \text{„nein“} & w \notin L \end{cases}$, für $w \in \Sigma^*$.
- $L \subseteq \Sigma^*$ ist *semi-entscheidbar* $\iff \exists$ Semi-Entscheidungsalgorithmus A mit $A : w \to \begin{cases} \text{„ja“} & w \in L \\ \perp & w \notin L \end{cases}$, $w \in \Sigma^*$.
- $L \subseteq \Sigma^*$ ist *aufzählbar* $\iff \exists$ Aufzählalgorithmus mit $w \in L \Leftrightarrow w$ wird irgendwann von A ausgegeben (evtl. Wdhlg.). *Merke:* Ein Aufzählalgorithmus gibt immer *alle* $w \in L$ aus durch testen *aller* $w \in \Sigma^*$ (unendlich viele). Terminierung kann nur unter Verwendung eines Diagonlaschemas mittels Schrittzahlbegrenzung erreicht werden.
- $f : \Sigma^* - \to \Sigma^*$ berechnebar $\iff f$ Turing-berechnebar $\iff f$ GOTO/WHILE-berechenbar.
- $f : \Sigma^* - \to \Sigma^*$ berechnebar $\iff G_f = \{(w, f(w)) \mid w \in \Sigma^*\}$ semi-entscheidbar (Kodierung etwa: $w\#f(w)$).
- $L \subseteq \Sigma^*$ entscheidbar $\iff L$ semi-entscheidbar und $\Sigma^* \setminus L$ semi-entscheidbar.
- $L \subseteq \Sigma^*$ semi-entscheidbar $\iff L = \mathrm{Def}(f)$ für $f : \Sigma^* - \to \Sigma^*$ berechnebar.
- Seien $P = (\mathrm{Inst}_P, \mathrm{Pos}_P)$, $Q = (\mathrm{Inst}_Q, \mathrm{Pos}_Q)$. $P \leq Q$ („Q ist mindestens so schwer wie P“) $\iff$ ex. berechenbare Funktion $f : \mathrm{Inst}_P \to \mathrm{Inst}_Q$ mit $\forall x \in \mathrm{Inst}_P :\ x \in \mathrm{Pos}_P \Leftrightarrow f(x) \in \mathrm{Pos}_Q$. Prüfe zur Wohldefiniertheit von f: 1. $x \in \mathrm{Pos}_P \Rightarrow f(x) \in \mathrm{Pos}_Q$ 2. $x \notin \mathrm{Pos}_P \Rightarrow f(x) \notin \mathrm{Pos}_Q$.
- Unentscheidbare Probleme: (M)PCP: (Modifiziertes) Postsches Korrespondenzproblem, D: Dominoproblem, ÄP: Äquivalenzproblem, HP: Halteproblem, WP: Wortproblem

WP:	Gegeben:	TM M über $\Sigma = \{0, 1\}$, $w \in \Sigma^*$.
	Frage:	$M : w \to \mathit{stop}$?
HP:	Gegeben:	TM M über $\Sigma = \{0, 1\}$, $w \in \Sigma^*$.
	Frage:	$M : \epsilon \to \mathit{stop}$?
ÄP:	Gegeben:	TM M_1, M_2 über $\Sigma = \{0, 1\}$, $w \in \Sigma^*$.
	Frage:	Berechnen M_1 und M_2 dieselbe Funktion $f : \Sigma^* \to \Sigma^*$?
D:	Gegeben:	Dominospiel $\mathcal{D} = (d_1, \ldots, d_n)$ mit Dominotypen d_i.
	Frage:	ex. Parkettierung von $\mathbb{Z} \times \mathbb{Z}$, d.h., $\mathcal{D}$ gut?
PCP:	Gegeben:	$x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, $x_i, y_i \in \Sigma^*$.
	Frage:	ex. Indexfolge $i_1, \ldots, i_k$ mit $x_{i_1}, \ldots x_{i_n} = y_{i_1}, \ldots y_{i_n}$?
	Gegeben:	$x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, $x_i, y_i \in \Sigma^*$.
	Frage:	ex. Indexfolge $i_1, \ldots, i_k$ mit $i_1 = 1$ und $x_{i_1}, \ldots x_{i_n} = y_{i_1}, \ldots y_{i_n}$?

3 Komplexität/Vollständigkeit

- Sei $g : \mathbb{N} \to \mathbb{N}$. $\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{N} \mid \exists c > 0\ \exists n_0\ \forall n \geq n_0 :\ f(n) \leq c \cdot g(n)\}$.
- TM M heißt $\mathcal{O}(g)$-Zeitbeschränkt für ein Polynom $g(n)$, falls $f \in \mathcal{O}(g)$ mit $f(n) = \max\{m \mid ex.w \in \Sigma^*$ mit $|w| = n$ und M läuft mit eingabe w mSchritte$\}$
- Bei einer *nichtdeterministischen* TM (NTM) existieren zu einem Zustand $(q, a) \in Q \times \Gamma$ mehrere Turingzeilen (NTM Probiert alle aus), d.h., es existieren auch mehrere Folgekonfigurationen.
- Bei einer *Offline*-Turingmaschine ist eine Turingzeile $q_i\ a_1\ b_1\ \star_1\ b_2\ \star_2\ q_j$ mit $\star_1, \star_2 \in \{L, R, N\}$, a_1 auf dem Eingabeband und b_i auf dem Arbeitsband, wobei $\star_1$ die Bewegung auf dem Eingabe- $\star_2$ die auf dem Ausgabeband beschreibt.
- $f : \Sigma^* \to \Sigma^*$ ist polynomzeit-berechenbar $\Leftrightarrow$ ex. polynomial zeitbeschränkte TM, die f berechnet.
- Sprache $L \in$ NP
  1. $\Longleftrightarrow$ ex. polynomzeitbeschräkte NTM M mit $M : w \to 1\ \Leftrightarrow\ w \in L$
  2. $\Longleftrightarrow$ ex. polynomzeit-entscheidbare Relation $R \subseteq \Sigma^* \times \Gamma^*$ und Polynom $q(n)$, so dass $w \in L \Leftrightarrow \exists v(|v| \leq (|w|) \wedge (w, v) \in R)$, d.h. „Teste $w \in L$ durch Probieren aller Lösungskandidaten v, deren *Länge* polynomial in $|w|$ beschränkt ist.“
- $L_1 \leq_p L_2 :\Leftrightarrow$ ex. polynomzeitberechenbare Funktion $f : \Sigma^* \to \Sigma^*$ mit $w \in L_1 \Leftrightarrow f(w) \in L_2$.
- $L_1 \leq_p L_2$ und $L_2 \in$ P $\Rightarrow L_1 \in$ P.
- L_0 ist NP-vollständig $\Longleftrightarrow$ 1. $L_0 \in$ NP 2. $\forall L \in NP :\ L \leq_p L_0$.
- NP-vollständige Probleme:

SAT(3):	Gegeben:	3-KNF Formel $\varphi \in$ AL (je 3 Literale pro Klausel).
	Frage:	Ist φ erfüllbar?
SAT:	Gegeben:	Formel $\varphi \in$ AL.
	Frage:	Ist φ erfüllbar?
COLOR(3):	Gegeben:	ungerichteter Graph $G = (V, E)$.
	Frage:	Ex. eine 3-Färbung von G?
CLIQUE:	Gegeben:	ungerichteter Graph $G = (V, E)$, Zahl $k \in \mathbb{N}$.
	Frage:	Existiert eine k-Clique in G?

- DLOG $\subseteq$ NLOG $\subseteq$ P $\subseteq$ NP $=$ DPSPACE $\subseteq$ und DLOG $\subset$ DPSPACE.