

Berechenbarkeit und formale Sprachen

Felix Leidl

26. März 2024

Inhaltsverzeichnis

Turingmaschine	2
1-Band TM	2
Deltatable	2
Konfiguration	2
Begriffe	2
Programmiertechniken	3
Endlicher Speicher	3
Unterprogramme	3
Spurtechnik	3
Gödelnummer	3
Universelle TM	3
Halteproblem	3
Allgemeines Halteproblem	3
Initiales Halteproblem	3
Reduktion	4
$L_1 \leq L_2$	4
Nichtdeterministische Turingmaschine	4
Begriffe	4
P-NP	4
Sprachprobleme	4
Clique	4
Independent-Set	4
Coloring	5
Traveling-Salesman	5
Vertex-Cover	5
Binary-Programming	5
Verifizierer	5

Turingmaschine

1-Band TM

Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$:

- Q : endliche Zustandsmenge
- Σ : endliches Eingabealphabet
- Γ : endliches Bandalphabet $\Sigma \subsetneq \Gamma$
- B : Blank, $B \in \Gamma, B \notin \Sigma$
- $q_0 : q_0 \in Q$ Startzustand
- F : akzeptierende Endzustände, $F \subseteq Q$
- das Programm $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, L, N\}$ eine partielle Funktion, wobei es für Endzustände keine Übergänge geben soll
- Zu Beginn steht der Lese-/Schreibkopf auf dem ersten Zeichen der Eingabe
- Eingabe: $w = w_1 w_2 \dots w_n \in \Sigma^*$
- ϵ : leeres Wort
- $L \subseteq \Sigma^*$ ist Sprache über dem Alphabet Σ

Deltatabelle

$Q = \{q_0, q_1\}, \Sigma = \{0, 1\}, \Gamma\{0, 1, B\}, F = \{q_1\}$

δ	0	1	B
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	-
q_1	-	-	-

Konfiguration

TM M ist in Konfiguration $K = \alpha q \beta$ ($\Gamma^* \times Q \times \Gamma^*$), wobei der Schreib-/Lesekopf auf dem ersten Zeichen von β steht.

Eine direkte Nachfolgekonfiguration von $\alpha q \beta$ ist: $\alpha q \beta \vdash \alpha' q' \beta'$

i -te Nachfolgekonfiguration $\alpha q \beta \vdash K_1 \vdash \dots \vdash K_{i-1} \vdash \alpha' q' \beta' = \alpha q \beta \vdash^i \alpha q \beta$

Nachfolgekonfiguration: $\alpha q \beta \vdash^* \alpha' q' \beta'$

Begriffe

- akzeptieren: Falls es $\alpha, \beta \in \Gamma^*$ und $q \in F$ gibt mit $q_0 x \vdash^* \alpha q \beta$
- $L(M)$: Menge aller von M akzeptierter Eingaben $x \in \Sigma^*$
- entscheidet: M hält mit Eingabe $x \in \Sigma^*$ nach endlich vielen Schritten
- rekursiv aufzählbar:

- $L \subseteq \Sigma^*$ ist rekursiv aufzählbar, wenn es eine TM M gibt mit $L(M) = L$
- es gibt eine surjektive Funktion $g : 0, 1^* \rightarrow L$
- entscheidbar/rekursiv:
 - wenn es eine deterministische 1-Band-TM M gibt, die L entscheidet
 - L und \bar{L} sind rekursiv aufzählbar

Programmiertechniken

Endlicher Speicher

Man merkt sich die Zeichen im Zustand

$\Gamma = \Sigma \cup \{B\}, Q = (\{q_0\} \times \Sigma) \cup \{q_0, q_1\}$, Startzustand $q_0, F = \{q_1\}$

Unterprogramme

Wenn man eine TM „programmiert“, kann man sagen: Man benutzt ein Unterprogramm um eine bestimmte Aufgabe zu lösen

Spurtechnik

	U	N	I	
	E	R	L	
	N	B	G	

Das erste Zeichen wäre $\begin{pmatrix} U \\ E \\ N \end{pmatrix}$

Gödelnummer

$\langle M \rangle$ ist die Gödelnummer (Bauplan von M). Sie ist die Repräsentation der TM M als natürliche Zahl

Universelle TM

Eine TM \tilde{M} heißt universell, wenn sie sich mit der Eingabe $\langle M \rangle x, x \in \{0, 1\}^*$ so verhält, wie M gestartet mit x

Halteproblem

Allgemeines Halteproblem

$H = \{\langle M \rangle \mid M \text{ ist deterministische 1-Band-TM, die, gestartet mit Eingabe } w, \text{ hält}\}$

Initiales Halteproblem

$H_\epsilon = \{\langle M \rangle \mid M \text{ ist deterministische 1-Band-TM, die, gestartet mit Eingabe } \epsilon, \text{ hält}\}$

Reduktion

- Eine Funktion ist berechenbar, wenn es eine TM M_f gibt, für die mit $x \in \{0, 1\}^*$ gilt:
 - Ist $f(x)$ definiert, so hält M_f mit der Eingabe x und $f(x)$ steht auf dem Band
 - Ist $f(x)$ undefiniert, so hält M_f gestartet mit x nicht
- Eine Funktion ist total, wenn alle $x \in \{0, 1\}^*$ definiert und berechenbar sind

Eine Reduktion ist eine total berechenbare Funktion $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$, für die gilt:

$$x \in L_1 \Leftrightarrow f(x) \in L_2$$

Wir schreiben „ $L_1 \leq L_2$ “ und sagen „ L_1 wird auf L_2 reduziert“,

$$L_1 \leq L_2$$

- L_2 entscheidbar $\Rightarrow L_1$ entscheidbar
- L_2 rekursiv aufzählbar $\Rightarrow L_1$ rekursiv aufzählbar
- L_1 unentscheidbar $\Rightarrow L_2$ unentscheidbar
- L_1 nicht rekursiv aufzählbar $\Rightarrow L_2$ nicht rekursiv aufzählbar

Nichtdeterministische Turingmaschine

Nichtdeterministische TM $M = \{Q, \Sigma, \Gamma, \delta, q_0, F\}$, wobei nur δ anderes ist, als bei einer deterministischen TM. $\delta : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{R, N, L\})$

Begriffe

- akzeptieren: wenn es eine Rechnung $q_0 x \vdash^* \alpha q \beta$ gibt

P-NP

- $\text{DTIME}(t(n)) := \{L \mid \text{Es gibt eine deterministische } \mathcal{O}(t(n))\text{-zeitbeschränkte TM, die } L \text{ entscheidet}\}$
- $\text{NTIME}(t(n)) := \{L \mid \text{Es gibt eine nichtdeterministische } \mathcal{O}(t(n))\text{-zeitbeschränkte TM, die } L \text{ akzeptiert}\}$

Sprachprobleme

Clique

$\text{CLIQUE} := \{\langle G, k \rangle \mid k \in \mathbb{N}, G \text{ ist ein ungerichteter Graph} \\ \text{der einen vollständigen Teilgraphen der Größe } k \text{ enthält}\}$

Independent-Set

$\text{IS} := \{\langle G, k \rangle \mid k \in \mathbb{N}, G = (V, E) \text{ ist ungerichteter Graph}, \\ \exists U \subseteq V, |U| = k : \forall u, v \in U : \{u, v\} \notin E\}$

Coloring

$$\begin{aligned}\text{COL} &:= \{\langle G, k \rangle \mid G \text{ ist ein ungerichteter Graph und } G \text{ ist } k\text{-färbbar}\} \\ \text{3COL} &:= \{\langle G \rangle \mid G \text{ ist ein ungerichteter Graph und } G \text{ ist 3-färbbar}\}\end{aligned}$$

Traveling-Salesman

$$\text{TSP} := \{\langle G, c, k \rangle \mid \text{der Graph } G \text{ mit Kantengewicht } c : E \rightarrow \mathbb{R} \\ \text{enthält eine Rundreise mit Gewicht } \leq k\}$$

Vertex-Cover

$$\text{VC} := \{\langle G, k \rangle \mid k \in \mathbb{N}, G \text{ ist ein ungerichteter Graph} \\ \text{und hat eine Knotenüberdeckung der Größe } k\}$$

Binary-Programming

$$\text{BP} := \{\langle A, \vec{b} \rangle \mid A \text{ ist eine } m \times n\text{-Matrix mit ganzzahligen Einträgen, } \vec{b} \text{ ist ein Vektor mit } m \\ \text{ganzzahligen Einträgen und es gibt einen 0-1-Vektor } \vec{x} \in \{0, 1\}^n \text{ mit } A \cdot \vec{x} \leq \vec{b}\}$$

Verifizierer

Eine deterministische Turingmaschine V_L heißt $t(n)$ -beschränkter Verifizierer für L , wenn gilt:

1. Die Eingaben von V_L sind von der Form $x\#w$, $w, x \in \{0, 1\}^*$
2. Die Laufzeit ist in $\mathcal{O}(t(|x|))$
3. Für alle $x \in \{0, 1\}^*$:

$$x \in L \Leftrightarrow \exists w : |w| \leq t(|x|) \text{ und } V_L \text{ akzeptiert } x\#w$$

Dieses heißt Zertifikat von w