Berechenbarkeit und formale Sprachen

Felix Leitl

29. April 2024

Inhaltsverzeichnis

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: $w = w_1 w_2 \ldots w_n \in \Sigma^*$
- 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 $\alpha q \beta \vdash K_1 \vdash \cdots \vdash K_{i-1} \vdash \alpha' q' \beta' = \alpha q \beta \vdash^i \alpha q \beta$
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 $g : 0, 1^* \to L$
- entscheidbar/rekursiv:
  - wenn es eine deterministische 1-Band-TM M gibt, die L entscheidet
  - L und $\overline{L}$ 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 $\tilde{M}$ hießt 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

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

Initiales Halteproblem

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

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 $f : \{0,1\}^* \to \{0,1^*\}$, 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. $\delta : Q \times \Gamma \to P(Q \times, \Gamma \times \{R, N, L\})$

Begriffe

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

P-NP

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

Sprachprobleme

Satisfiability Problem

$$\text{SAT} := \{\langle\Phi\rangle | \Phi \text{ ist eine erfüllbare } KNF\}$$

Clique

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

Independent-Set

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

Coloring

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

Traveling-Salesman

$$\begin{aligned}\text{TSP} := \{\langle G,c,k\rangle | \text{ der Graph } G \text{ mit Kantengewicht } c : E \to \mathbb{R}\\ \text{enthält eine Rundreise mit Gewicht } \leq k\}\end{aligned}$$

Vertex-Cover

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

Binary-Programming

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

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

$L \in NTIME(t(n)) \Leftrightarrow$ es gibt einen $t(n)$-beschränkten Verifizierer V_L für L
$NP = \{L|$ es gibt einen polynomiellen Verifizierer für $L\}$

NP-Vollständig

Polynomiell reduzierbar

Seien $L_1 \subseteq \Sigma_1^*, L_2 \subseteq \Sigma_2^*$.
L_1 ist genau dann polynomiell reduzierbar auf $L_2 (L_1 \leq_p L_2)$, wenn gilt:

1. $L_1 \leq L_2$ mittels Reduktionsfunktion f
2. Die Laufzeit zur Berechnung von $f(x)$ ist $\mathcal{O}(|x|^k)$ für $k \in \mathbb{N}$

NP-schwer

$$\forall L' \in NP : L' \leq_p L$$

NP-vollständig

1. $L \in NP$
2. L ist NP-schwer

Formale Sprachen

Eine Grammatik G vom Typ Chomsky-0 ist beschrieben durch ein 4-Tupel(V, Σ, P, S) mit:

- V: Endliche Menge an Variablen
- Σ: Endliche Menge der Terminalsymbole
- $S : S \in V$: Startsymbol
- $P \subseteq ((V \cup \Sigma)^+ \backslash \Sigma^*) \times (V \cup \Sigma)^*$: Endliche Menge von Produktionen/Ableitungsregeln: $u \to v$

Die erzeugte Sprache:

$$L(G) := \{w | S \xrightarrow{*} w, w \in \Sigma^*\}$$

Chomsky-Grammatiken

Sprachtyp	Erzeuger	Erkenner	Definition
rekursiv aufzählbar	Chomsky-0	Turingmachinen	
kontextsensitiv	Chomsky-1	Linear beschränkte Turningmaschinen	$\lvert u\rvert \leq \lvert v\rvert$
kontextfrei	Chomsky-2	Kellerautomaten	$u \in V \& \lvert u\rvert = 1$
regulär	Chomsky-3	Automaten	$u \in V \& v \in \{\epsilon\} \cup \Sigma$ / $u \in V \& v \in \Sigma \circ V$

Endliche Automaten

$A = (Q, \Sigma, \delta, q_0, F)$:

- Q: Endliche Menge der Zustände
- Σ: Endliches Alphabet, $Q \cap \Sigma = \emptyset$
- δ: $Q \times \Sigma \to Q$ Übergangsfunktion

- q_0: Startzustand
- F: $F \subseteq Q$, akzeptierende Endzustände

δ	0	1
q_0	q_0	q_0
q_1	q_0	q_1

$\Rightarrow \delta(q_0, 001) = \delta(q_0, 01) = \delta(q_0, 1) = q_1$

Reguläre Pump-Eigenschaft

$\exists n_L \in \mathbb{N}\ \forall z \in L, |z| \geq n_L\ \exists u, v, w \in \Sigma^* : uvw = z$ und

- $|uv| \leq n_L$
- $v \neq \epsilon$
- $\forall i \geq 0 : uv^iw \in L$

$\Rightarrow$ Die Sprache L ist regulär

Kontextfreie Pum-Eigenschaft

$\exists n_L \in \mathbb{N}\ \forall z \in L, |z| \geq n_L\ \exists u, v, w, x, y \in \Sigma^* : uvwxy = z$ und

- $|vwx| \leq n_L$
- $vx \neq \epsilon$
- $\forall i \geq 0 : uv^iwx^iy \in L$

$\Rightarrow$ Die Sprache L ist kontextfrei