Diskrete Mathematik

Martin Avanzini Arne Dür Christoph Kollreider Georg Moser

Fakultät für Mathematik, Informatik und Physik @ UIBK
Sommersemester 2010

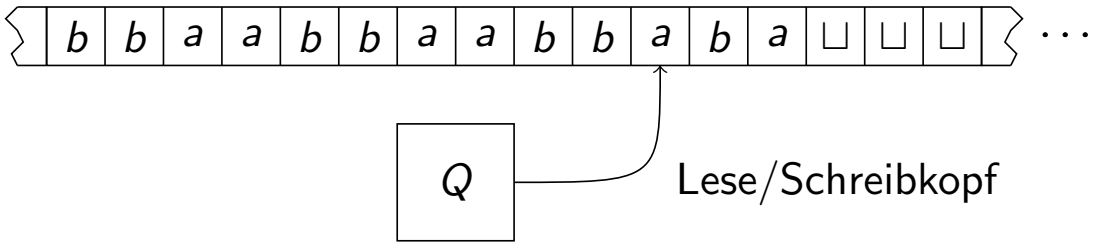
Zusammenfassung der letzten LV

Satz

$\forall$ deterministische TM mit k Bändern $\exists$ einbändige, deterministische TM M', sodass $\mathrm{L}(M) = \mathrm{L}(M')$

Definition

zweiseitig unbeschränktes Band



Satz

$\forall$ M eine einbändige, deterministische TM, dessen Band in beide Richtungen unbeschränkt ist $\exists$ einbändige, deterministische TM M', sodass $\mathrm{L}(M) = \mathrm{L}(M')$

Übersicht

Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion, Automaten mit ϵ-Übergängen, Umwandlung endlicher Automaten in reguläre Ausdrücke, Algebraische Gesetze für reguläre Ausdrücke, Pumpinglemma, Minimierung

Einführung in die Berechenbarkeitstheorie, Turing Maschinen, Entscheidungsprobleme, Äquivalente Formulierungen, Universelle Maschinen und Diagonalisierung,

Komplexität der Umwandlung von Repräsentationen, Entscheidungsprobleme, Einführung in die Komplexitätstheorie, Laufzeitkomplexität, die Klassen P und NP, logarithmisch platzbeschränkte Reduktionen, Speicherplatzkomplexität

Nichtdeterministische Turingmaschine

Definition NTM

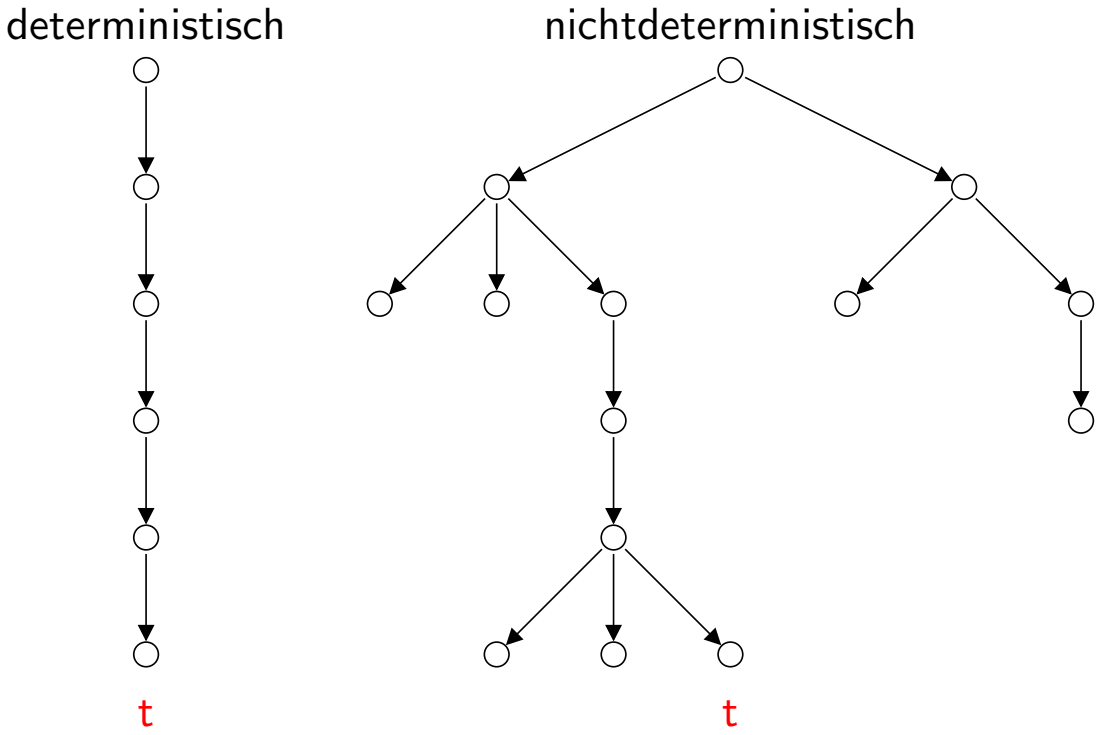
eine nichtdeterministische, k-bändige Turingmaschine N ist ein 9-Tupel

$$N = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

1. Q eine endliche Menge von Zuständen,
2. Σ eine endliche Menge von Eingabesymbolen,
3. Γ eine endliche Menge von Bandsymbolen, sodass $\Sigma \subseteq \Gamma$,
4. $\vdash \in \Gamma \setminus \Sigma$, der linke Endmarker,
5. $\sqcup \in \Gamma \setminus \Sigma$, das Blanksymbol,
6. $\delta\colon Q \times \Gamma^k \to \mathcal{P}(Q \times \Gamma^k \times \{\mathrm{L}, \mathrm{R}\}^k)$ die Übergangsfunktion,
7. $s \in Q$, der Startzustand,
8. $t \in Q$, der akzeptierende Zustand und
9. $r \in Q$, der verwerfende Zustand mit $t \neq r$.

Nichtdeterministischer Berechnungsbaum



Beachte

damit NTM N akzeptiert, genügt ein Pfad, sodass N auf einen akzeptierenden Zustand trifft

Nichtdeterminismus vs. Determinisums

Satz

- $\forall$ N nichtdeterministische TM, $\exists$ deterministische TM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$
- jede deterministische TM ist auch eine nichtdeterministische TM

Adressierung des Pfades

- sei b der "Grad des Nichtdeterminismus" beziehungsweise b ist der Verzweigungsgrad des Berechnungsbaums
- String über dem Alphabet $\Sigma_b = \{1, 2, \ldots, b\}$ nennen wir Adresse
- Die Adresse ist ungültig, oder bezeichnet eine eindeutige Position im Berechnungsbaum

Beweis

- sei N 1-Band NTM, wir konstruieren eine 3-Band deterministische TM M sodass $\mathrm{L}(M) = \mathrm{L}(N)$
- sei x das Eingabewort; Simulation ist uniform für jedes x
- erste Band von M wird immer nur Eingabe x enthalten
- zweites Band simuliert Rechenvorgänge von N
- dritte Band adressiert den aktuellen Pfad

Simulation für einen Pfad

1. Anfangs enthält Band 1 von M das Eingabewort x
   Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2
3. Verwende Band 2, um die Rechenschritte von N auf x zu simulieren
   Bei nichtdeterministischen Entscheidungen: siehe Band 3
4. Ersetze das Wort auf Band 3 durch den nächstgrößeren in der graduiert-lexikographische Ordnung ■

- $\forall$ deterministische TM mit k Bändern $\exists$ einbändige, deterministische TM M', sodass $\mathrm{L}(M) = \mathrm{L}(M')$
- $\forall$ M eine einbändige, deterministische TM, dessen Band in beide Richtungen unbeschränkt ist $\exists$ einbändige, deterministische TM M', sodass $\mathrm{L}(M) = \mathrm{L}(M')$
- $\forall$ N nichtdeterministische TM, $\exists$ deterministische TM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$

Satz

die folgenden Erweiterungen von Turingmaschien verändern die Ausdrucksfähigkeit nicht:

- Nichtdeterminismus
- mehrere Bänder
- mehrere Lese/Schreibköpfe
- zweifach unendliches Band

Klasse der rekursiven, rekursiv aufzählbaren Mengen wird nicht verändert

Universelle Turingmaschinen

Kodierung

TMs können kodiert werden indem alle notwendigen Informationen als Wörter über $\{0,1\}$ dargestellt werden:

1. Anzahl der Zustände
2. Übergangsfunktion
3. Eingabe- und Bandalphabet
4. ...

Beispiel einer Kodierung

sei $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ eine TM; Kodierung über Alphabet $\{0,1\}$

$$0^n\ 1\ 0^m\ 1\ 0^k\ 1\ 0^s\ 1\ 0^t\ 1\ 0^r\ 1\ 0^u\ 1\ 0^v\ 1 \quad \cdots$$

entspricht $Q = \{0, \ldots, n-1\}$, $\Gamma = \{0, \ldots, m-1\}$, $\Sigma = \{0, \ldots, k-1\}$, $(k \leqslant m)$, s Startzustand, t akzeptierend, r verwerfend, u linken Endmarker, v das Blanksymbol

Kodierung (Fortsetzung)

Betrachte $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ und kodiere die Übergangsfunktion $\delta(p,a) = (q,b,d)$

$$0^p\ 1\ 0^a\ 1\ 0^q\ 1\ 0^b\ 1\ \underbrace{1}_{\text{Richtung } d}$$

Beobachtung

Kodierung muss injektiv sein, aber nicht notwendigerweise bijektiv

Definition

UTM

eine TM U heißt universell, wenn U bei Eingabe

- des Codes $\ulcorner M \urcorner$ einer TM M
- und des Code $\ulcorner x \urcorner$ einer Eingabe x für M

U die TM M auf x simuliert, das heißt

$$\mathrm{L}(U) = \{\ulcorner M \urcorner \sharp \ulcorner x \urcorner \mid x \in \mathrm{L}(M)\}$$

Konstruktion einer UTM

Simulation

1. U kontrolliert Korrektheit der Codes; wenn inkorrekt, verwirft U
2. U simuliert M mit 3 Bändern auf der Eingabe x
   - Band 1 enthält die Beschreibung von M
   - Band 2 enthält das (dekodierte) Eingabewort x
   - Band 3 enthält (simulierten) Bandinhalt des Bandes von M
3. wenn M jemals auf der Eingabe x hält, hält U ebenfalls und akzeptiert beziehungsweise verwirft entsprechend

Konvention

wir schreiben

$$\mathrm{L}(U) = \{M\sharp x \mid x \in \mathrm{L}(M)\}$$

Unentscheidbarkeit des Halteproblems

Definition

definiere Halteproblem und Zugehörigkeitsproblem von TMs

$$\mathrm{HP} := \{M\sharp x \mid M \text{ hält bei Eingabe } x\}$$
$$\mathrm{MP} := \{M\sharp x \mid x \in \mathrm{L}(M)$$

Definition

M_x

- M_x ist TM (mit Eingabealphabet) $\{0,1\}$
  deren Code (mit Kodierungsalphabet $\{0,1\}$) gleich x
- wenn x kein Code, definiere M_x beliebig

Aufzählung aller Turingmaschinen

$$M_\epsilon, M_0, M_1, M_{00}, M_{01}, M_{10}, M_{000}, \ldots$$

Matrix aller TMs auf allen Eingaben

	ϵ	0	1	00	01	10	11	000	001	010	...
M_ϵ	✓	↻	↻	✓	✓	↻	✓	↻	✓	✓	
M_0	↻	↻	✓	✓	↻	✓	✓	↻	↻	✓	
M_1	↻	✓	↻	✓	↻	✓	✓	↻	↻	✓	
M_{00}	✓	↻	↻	✓	✓	✓	✓	↻	↻	✓	
M_{01}	✓	✓	✓	✓	↻	↻	↻	✓	✓	↻	...
M_{10}	✓	✓	↻	✓	✓	↻	✓	✓	↻	✓	
M_{11}	✓	✓	↻	↻	✓	↻	✓	↻	✓	↻	
M_{000}	✓	✓	✓	✓	↻	✓	✓	↻	✓	↻	
M_{001}	↻	✓	✓	✓	✓	↻	✓	✓	✓	✓	
⋮						⋮					⋱

Diagonalisierung

die dem Komplement der Diagonale entsprechende Sprache
wird von keiner der TMs akzeptiert

$$M_\epsilon, M_0, M_1, M_{00}, M_{01}, M_{10}, M_{000}, \ldots$$

Behauptung

die dem Komplement der Diagonale entsprechende Sprache
wird von keiner TM akzeptiert

Beweis

sei $\Sigma \supseteq \{✓, ↻\}$ ein Alphabet
$s_0, s_1, s_2, \ldots$ eine Folge unendlicher Wörter über $\{✓, ↻\}$

$$\begin{aligned} s_0 &= s_{00}s_{01}s_{02}s_{03}s_{04}\ldots \\ s_1 &= s_{10}s_{11}s_{12}s_{13}s_{14}\ldots \\ s_1 &= s_{20}s_{21}s_{22}s_{23}s_{24}\ldots \\ &\vdots \end{aligned}$$

dann ist die Folge

$$d_n = \begin{cases} ↻ & \text{wenn } s_{nn} = ✓ \\ ✓ & \text{wenn } s_{nn} = ↻ \end{cases}$$

eine neue Folge ■

Satz

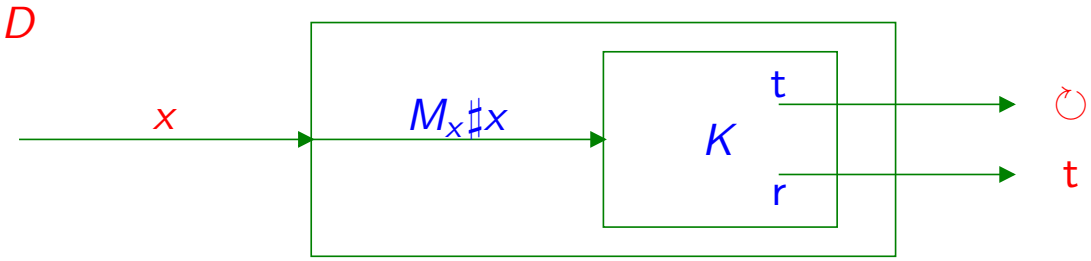
HP ist nicht rekursiv, aber rekursiv aufzählbar

Beweis

wir zeigen Nicht-Rekursivität
angenommen $\exists$ totale TM K, sodass HP $=$ L(K)

Definition

Diagonaliserungsmaschine



D hält auf $x \Leftrightarrow K$ verwirft $M_x\sharp x$ Definition von D

$\Leftrightarrow M_x$ hält nicht auf x Definition von K

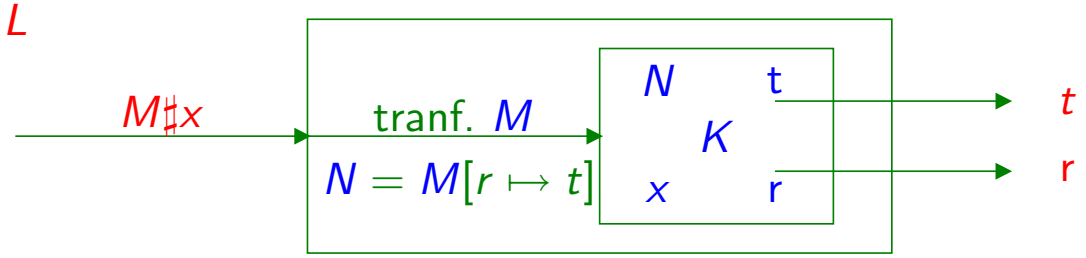
Verhalten von D verschieden von jeder TM M_x in der Aufzählung; Widerspruch

Satz

Menge MP ist rekursiv aufzählbar, aber nicht rekursiv

Beweisskizze

- um zu zeigen, dass MP r.e. ist definiere UTM U
  die bei Eingabe $M\sharp x$, TM M auf x simuliert, U akzeptiert, wenn M akzeptiert
- um zu zeigen, dass MP nicht rekursiv ist, verwende Reduktion vom Halteproblem
  sei K eine totale TM, die MP entscheidet,
  definiere L (totale) TM, die HP entscheidet:



Satz

1. jede rekursive Menge ist rekursiv aufzählbar
2. aber nicht jede rekursiv aufzählbare Menge rekursiv

Beweis

nach Definition ist jede rekursive Sprache auch rekursiv aufzälhbar; MP (HP) ist rekursiv aufzählbar, aber nicht rekursiv ■

Satz

es kann kein Testprogramm für "hello-world" Programme geben

Beweisskizze

Reduktion vom Halteproblem ■

Satz

die folgenden Probleme sind unentscheidbar:

1. das Postsche Korrespondenzproblem
2. ist eine beliebige Sprache regulär?