



## Diskrete Mathematik

Ralph Bottesch    David Obwaller  
 Burak Ekici        Vincent van Oostrom  
 Johannes Koch    Oleksandra Panasiuk  
**Georg Moser**

cbr.uibk.ac.at

## Inhalte der Lehrveranstaltung (cont'd)

### Reguläre Sprachen

deterministische Automaten, nichtdeterministische Automaten, endliche Automaten mit Epsilon-Übergängen, reguläre Ausdrücke, Abgeschlossenheit, Schleifenlemma

### Berechenbarkeitstheorie

deterministische TM, nichtdeterministische TM, universelle TMs, Äquivalenzen

### Komplexitätstheorie

Grundlagen, die Klassen P und NP, polynomielle Reduktionen, logspace Reduktionen, die Klassen NLOGSPACE und PSPACE

## Zusammenfassung der letzten LVA

### Satz (Kontraposition des Schleifenlemmas)

Sei  $L$  eine formale Sprache über  $\Sigma$ , sodass:

- für alle  $n \in \mathbb{N}$  existiert ein Wort  $w \in L$  mit  $\ell(w) \geq n$ , sodass
- für alle  $x, y, z \in \Sigma^*$  mit  $w = xyz$ ,  $y \neq \epsilon$  und  $\ell(xy) \leq n$  existiert  $k \in \mathbb{N}$  mit  $x(y)^k z \notin L$

Dann ist  $L$  nicht regulär. ■

### Beispiel

Sei  $\Sigma = \{1\}$ ; dann ist

$$D = \{w \in \Sigma^* \mid \ell(w) \text{ ist eine Primzahl}\}$$

nicht regulär

## Rekursiv, Rekursiv Aufzählbar, Co-Rekursiv Aufzählbar

### Definition

Eine Sprache  $L$  (oder allgemein eine Menge) heißt

- **rekursiv aufzählbar**, wenn eine TM  $M$  existiert, sodass  $L = L(M)$
- **co-rekursiv aufzählbar** wenn sie das Komplement einer rekursiv aufzählbaren Sprache ist
- **rekursiv**, wenn es eine **totale** TM  $M$  gibt, sodass  $L = L(M)$

### Church-Turing-These

Jedes algorithmisch lösbare Problem ist auch mit Hilfe einer Turingmaschine lösbar (das gilt auch für Quantenrechner)

**Satz**

Sei  $\Sigma$  ein Alphabet und  $L \subseteq \Sigma^*$  rekursiv; dann ist  $\sim L$  rekursiv

**Beweis.**

Da  $L$  rekursiv ist, gibt es eine totale TM  $M$  mit  $L = L(M)$ . Wir definieren eine TM  $M'$ , wobei der akzeptierende und der verwerfende Zustand von  $M$  vertauscht werden. Weil  $M$  total ist, ist auch  $M'$  total. Somit akzeptiert  $M'$  ein Wort genau dann, wenn  $M$  es verwirft und es folgt  $\sim L = L(M')$ , d.h.  $\sim L$  ist rekursiv. ■

**Satz**

Jede rekursive Menge ist rekursiv aufzählbar. Andererseits ist nicht jede rekursiv aufzählbare Menge rekursiv.

**Beweis.**

Der erste Teil des Satzes ist eine Konsequenz der Definitionen; den zweiten Teil holen wir nach ■

**Definition**

Eine **nichtdeterministische (einbändige) Turingmaschine**  $N$  (kurz NTM) ist ein 9-Tupel

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

sodass

- 1  $Q$  eine endliche Menge von **Zuständen**,
- 2  $\Sigma$  eine endliche Menge von **Eingabesymbolen**,
- 3  $\Gamma \supseteq \Sigma$  eine endliche Menge von **Bandsymbolen**,
- 4  $\vdash \in \Gamma \setminus \Sigma$ , der **linke Endmarker**,
- 5  $\sqcup \in \Gamma \setminus \Sigma$ , ( $\sqcup \neq \vdash$ ), das **Leerzeichen**,
- 6  $\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$  die **Übergangsfunktion**,
- 7  $s \in Q$ , der **Startzustand**,
- 8  $t \in Q$ , der **akzeptierende Zustand** und
- 9  $r \in Q$ , der **verwerfende Zustand** mit  $t \neq r$ .

**Satz**

Wenn  $L$  und  $\sim L$  rekursiv aufzählbar sind, dann ist  $L$  rekursiv.

**Beweis.**

- $\exists$  TM  $M_1, M_2$  mit  $L = L(M_1)$  und  $\sim(L) = L(M_2)$
- definiere TM  $M'$ , sodass das Band zwei Hälften hat (oder eine 2-Band TM):

$b$	$\hat{b}$	$a$	$b$	$a$	$a$	$a$	$a$	$b$	$a$	$a$	$a$	} ...
$c$	$c$	$c$	$d$	$d$	$d$	$c$	$\hat{c}$	$d$	$c$	$d$	$c$	

- $M_1$  wird auf der oberen und  $M_2$  auf der unteren Hälfte simuliert
- wenn  $M_1$   $x$  akzeptiert,  $M'$  akzeptiert  $x$
- wenn  $M_2$   $x$  akzeptiert,  $M'$  verwirft  $x$  ■

**Beispiel**

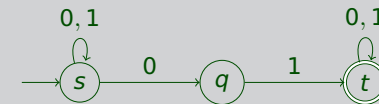
Für die NTM  $N = (\{s, q, r, t\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \vdash, \sqcup, \delta, s, t, r)$  mit  $\delta$  gegeben durch

	$\vdash$	0	1	$\sqcup$
$s$	$\{(s, \vdash, R)\}$	$\{(s, 0, R), (q, 0, R)\}$	$\{(s, 1, R)\}$	$\{(r, \sqcup, R)\}$
$q$	.	$\{(r, 0, R)\}$	$\{(t, 1, R)\}$	$\{(r, \sqcup, R)\}$

es gilt  $0011 \in L(N)$

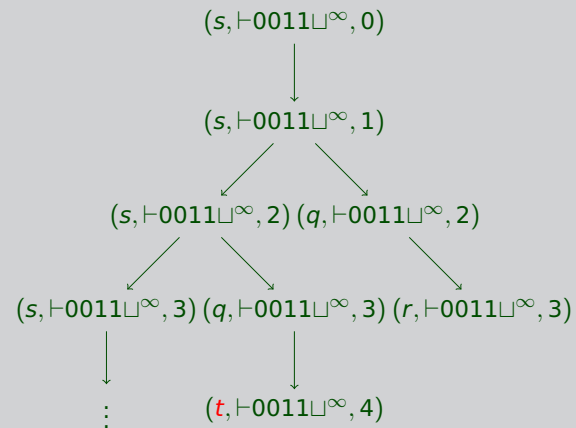
**Beispiel (Fortsetzung)**

$N$  entspricht dem folgendem NEA (plus Fangzustand  $r$ ):



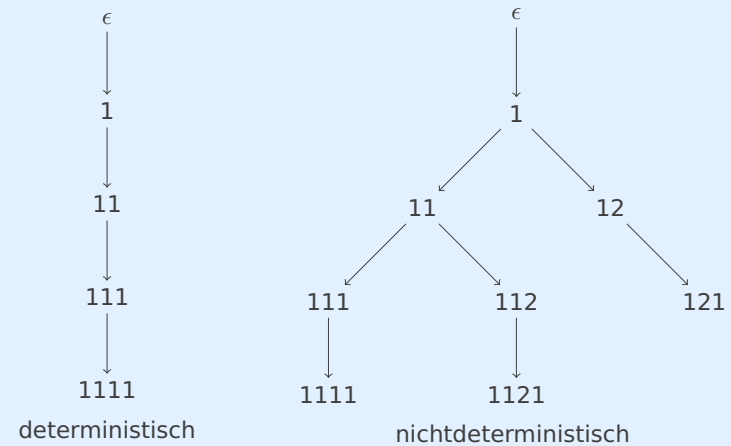
## Beispiel (Fortsetzung)

Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:



8

## Adressierung des Berechnungsbaumes



9

## Satz

Sei  $N$  eine NTM. Dann existiert eine DTM  $M$ , sodass  $L(M) = L(N)$ . Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.

## Beweis des ersten Teils.

Sei  $N$  eine NTM; wir konstruieren eine dreibändige DTM  $M$  mit  $L(N) = L(M)$ ; sei  $x$  das Eingabewort für  $N$ :

- Das erste Band von  $M$  wird immer nur diese Eingabe  $x$  enthalten
- Auf dem zweiten Band simulieren wir die Rechenschritte von  $N$  bezüglich eines Weges im Berechnungsbaum
- Schließlich dient das dritte Band dazu, den aktuellen Weg zu adressieren

Mit Hilfe der Adressierung von Wegen können wir nun das Verhalten von  $N$  in  $M$  simulieren

10

## Beweis (Fortsetzung).

- 1 Anfangs enthält Band 1 von  $M$  das Eingabewort  $x$ , die 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 jeder Stelle in der Berechnung, in der die Übergangsfunktion  $\delta$  mehrere Möglichkeiten zulässt, sieht  $M$  auf Band 3 nach, welche Möglichkeit gewählt werden soll
- 4 Wenn die Simulation von  $N$  akzeptiert, dann akzeptiere
- 5 Ersetze das Wort auf Band 3 durch seinen unmittelbaren Nachfolger in der graduiert-lexikographischen Ordnung auf Wörtern  $\leq_{\text{gradlex}}$
- 6 Gehe zu Schritt (2)

11

## Entscheidbarkeit, Unentscheidbarkeit

### Definition

Sei  $\Sigma$  ein Alphabet. Eine Eigenschaft  $P$  von Wörtern über  $\Sigma$  heißt

- **entscheidbar** genau dann, wenn die Menge  $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$  rekursiv ist
- **semi-entscheidbar** genau dann, wenn die Menge  $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$  rekursiv aufzählbar ist

### Beispiel

Sei  $P(x) := x$  ist ein Palindrom gerader Länge; dann ist  $P$  entscheidbar

### Beispiel

Jedes entscheidbare Problem ist semi-entscheidbar

12

### Bemerkung

Ein Problem  $P$  ist

- semi-entscheidbar, wenn es eine TM  $M$  gibt, deren Sprache alle Wörter sind, welche die Eigenschaft  $P$  haben
- entscheidbar, wenn es eine **totale** TM  $M$  gibt, sodass  $M$  genau jene Wörter akzeptiert, welche die Eigenschaft  $P$  haben

13

## Grenzen der Berechenbarkeit

### Was können Turingmaschinen?

- Codierung von Turingmaschinen
- **universelle Turingmaschine**; eine TM als Universalinterpreter

### Warum hat die Informatik dann Grenzen?

- Definition einer TM mit einem anderen Verhalten (in Bezug auf Termination)  $\overline{LD}$  als **alle anderen**
- Unentscheidbarkeit des Halteproblems

14

### Codierung von TMs

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

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

### Beispiel

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

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

entspricht  $Q = \{0, \dots, n-1\}$ ,  $\Gamma = \{0, \dots, m-1\}$ ,  $\Sigma = \{0, \dots, k-1\}$ , ( $k \leq m$ ),  $s$  Startzustand,  $t$  akzeptierend,  $r$  verwerfend,  $u$  linker Endmarker,  $v$  Blanksymbol; das Zeichen **1** dient als Trennzeichen der Codierung

15

### Beispiel (Fortsetzung)

betrachte  $M$  und kodiere  $\delta(p, a) = (q, b, d)$ , wobei  $c = 0$  wenn  $d = L$  und  $c = 1$  wenn  $d = R$

$$0^p 1 0^a 1 0^q 1 0^b 1 0^c 1$$

### Beispiel

Wir kodieren  $M' = (\{s, p, t, r\}, \{0, 1\}, \{0, 1, \vdash, \sqcup\}, \vdash, \sqcup, \delta, s, t, r)$  mit

	$\vdash$	0	1	$\sqcup$
s	(s, $\vdash$ , R)	(s, 0, R)	(s, 1, R)	(p, $\sqcup$ , L)
p	(t, $\vdash$ , R)	(t, 1, L)	(p, 0, L)	.

Zunächst erhalten wir

$$\underbrace{0000}_{n=4} \underbrace{10000}_{m=4} \underbrace{100}_{k=2} 1 \underbrace{\epsilon}_s \underbrace{100}_t \underbrace{1000}_{r} \underbrace{100}_{\vdash} \underbrace{1000}_{\sqcup} 1 \dots$$

und etwa  $\delta(p, \vdash) = (t, \vdash, R)$  wird zu  $010^210^210^2101$

16

## Simulation durch eine Universelle Turingmaschine

### Notation

Vereinfachend läßt man oft den Hinweis auf die Kodierung weg:

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

### Simulation

- 1 UTM  $U$  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 den simulierten Bandinhalt des Bandes von  $M$
- 3 wenn  $M$  akzeptiert, so akzeptiert  $U$ ; wenn  $M$  verwirft, so verwirft  $U$

18

### Definition

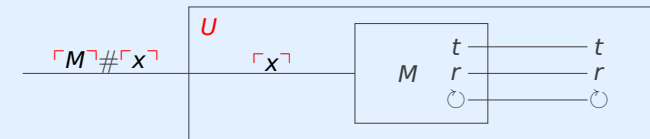
eine TM  $U$  heißt **universell (UTM)**, wenn bei Eingabe

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

die TM  $U$ , die TM  $M$  auf  $x$  **simuliert**, das heißt

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

### UTM schematisch



17

### Lemma

Angenommen  $U$  eine UTM und  $M$  eine beliebige TM. Dann existiert eine Spezialisierung von  $U$ , genannt  $U_M$ , die  $M$  auf allen Eingaben simuliert.

### Beweis.

- Wir betrachten eine Variante  $U'$  von  $U$ , sodass das zweite Band von  $U'$  die Beschreibung der zu simulierenden TMs enthält und das erste Band die (dekodierte) Eingabe
- Nun spezialisieren wir  $U'$  zur gesuchten TM  $U_M$  indem wir den Code von  $M$  fix auf das zweite Band schreiben (also hardcoden)
- Nach Definition führt  $U_M$  alle Schritte von  $M$  auf der Eingabe  $x$  aus

### Bemerkung

Metaprogrammierung oder Programmiermakros stammen von UTM's

19

### Definition

definiere **Halteproblem** und **Zugehörigkeitsproblem** von TMs

$$HP := \{M\#x \mid M \text{ hält bei Eingabe } x\}$$

$$MP := \{M\#x \mid x \in L(M)\}$$

### Definition

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

### Aufzählung aller Turingmaschinen

$$M_\epsilon, M_0, M_1, M_{00}, M_{01}, M_{10}, M_{11}, M_{000}, \dots$$

(geordnet bezüglich der graduiert lexikographischen Ordnung)

20

### Behauptung

die dem Komplement der Diagonale entsprechende Sprache  $\overline{LD}$  wird von keiner TM in der Aufzählung akzeptiert

### Beweis.

sei  $\Sigma \supseteq \{!, \circ\}$  ein Alphabet

$s_0, s_1, s_2, \dots$  eine Folge unendlicher Wörter über  $\{!, \circ\}$

$$s_0 = s_{00}s_{01}s_{02}s_{03}s_{04} \dots$$

$$s_1 = s_{10}s_{11}s_{12}s_{13}s_{14} \dots$$

$$s_2 = s_{20}s_{21}s_{22}s_{23}s_{24} \dots$$

$\vdots$

dann ist die Folge

$$d_n = \begin{cases} \circ & \text{wenn } s_{nn} = ! \\ ! & \text{wenn } s_{nn} = \circ \end{cases}$$

eine neue Folge

22

### Zweidimensionale Matrix

Indiziert mit Wörtern  $w \in \{0, 1\}^*$  und andererseits mit den Turingmaschinen

	$\epsilon$	0	1	00	01	10	11	000	001	010	...
$M_\epsilon$	!	○	○	!	!	○	!	○	!	!	
$M_0$	○	○	!	!	○	!	!	○	○	!	
$M_1$	○	!	○	!	○	!	!	○	○	!	
$M_{00}$	!	○	○	!	!	!	!	○	○	!	
$M_{01}$	!	!	!	!	○	○	○	!	!	○	...
$M_{10}$	!	!	○	!	!	○	!	!	○	!	
$M_{11}$	!	!	○	○	!	○	!	○	!	○	
$M_{000}$	!	!	!	!	○	!	!	○	!	○	
$M_{001}$	○	!	!	!	!	○	!	!	!	!	
$\vdots$						$\vdots$				$\ddots$	

21

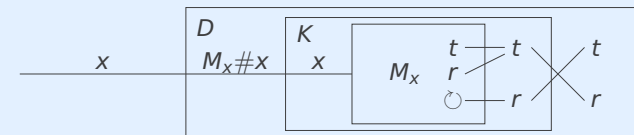
### Satz

HP ist nicht rekursiv, aber rekursiv aufzählbar

### Beweis.

wir zeigen zunächst Nicht-Rekursivität

- 1 angenommen  $\exists$  totale TM  $K$ , sodass  $HP = L(K)$
- 2 Definition von TM  $D$

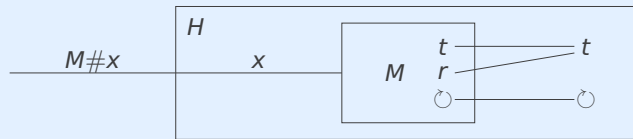


- 3  $D$  akzeptiert genau die dem Komplement der Diagonale entsprechende Sprache
- 4 Verhalten von  $D$  verschieden von jeder TM  $M_x$  in der Aufzählung

23

### Beweis (Fortsetzung).

Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende (nicht notwendigerweise totale) TM  $H$



### Folgerung

Die Menge  $\sim$ HP ist nicht rekursiv aufzählbar

### Beweis.

Angenommen  $\sim$ HP wäre rekursiv aufzählbar; dann wären HP und  $\sim$ HP rekursiv aufzählbar und somit auch HP rekursiv. Widerspruch