

# Diskrete Mathematik

Martin Avanzini   Arne Dür   Christoph Kollreider   Georg Moser

Fakultät für Mathematik, Informatik und Physik © UIBK  
Sommersemester 2010



## Zusammenfassung der letzten LV

### Definition

### Minimierungsalgorithmus

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

konstruiere  $B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ :

- 1 unerreichbare Zustände eliminieren
- 2 partitioniere die Zustände in äquivalente Blöcke
- 3  $Q_B$  sind die verschiedenen Blöcke
- 4  $\forall S$  ein Block,  $\forall a$  ein Eingabesymbol:  
 $\exists$  Block  $T$ , sodass für alle  $q \in S$  gilt  $\delta(q, a) \in T$   
 setze  $\delta_B(S, a) = T$
- 5  $q_B$  ist der Block, der  $q_0$  enthält
- 6  $F_B$  ist die Menge der Blöcke, die Zustände aus  $F$  enthalten

### Satz

der Minimierungsalgorithmus ist optimal und eindeutig

# Übersicht

Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion, Automaten mit  $\epsilon$ -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,

Einführung in die Komplexitätstheorie, Laufzeitkomplexität, die Klassen P und NP, logarithmisch platzbeschränkte Reduktionen, Speicherplatzkomplexität

## Einführung in die Berechenbarkeitstheorie

### Frage

Ist jedes Problem **algorithmisch** lösbar?

### Antwort

**Nein!**

Ein einfaches Programm

“hello,world” Programm

```
main()
{
    printf(‘‘hello, world\n’’);
}
```

## Beispiel

## Program F

```

main()
{
    int n, summe, x, y, z;
    scanf("%d", &n);
    summe = 3;
    while (1) {
        for (x=1; x <= summe-2; x++)
            for (y=1; y<=summe-x-1; y++) {
                z = summe - x - y;
                if (exp(x,n) + exp(y,n) == exp(z,n))
                    printf("hello, world\n");
            }
        summe++;
    }
}

```

Program F ist kein "hello, world" Programm

## Beispiel

## Programm G

```

main()
{
    int n, x, y, z, test;
    summe=4;
    while (1) {
        test = 0;
        for (x=2; x <= summe; x++) {
            y = summe - x;
            if (primes(x) && primes(y))
                test = 1;
        }
        if !test printf("hello, world\n");
        summe = summe + 2;
    }
}

```

Program G ist **wahrscheinlich** kein "hello, world" Programm

## Satz

es kann kein Testprogramm für “hello-world” Programme geben

## Beweisskizze

- wir betrachten ein hypothetisches Programm  $H$ , das wir “hello, world” Tester nennen



- man kann zeigen, dass ein “hello, world”-Tester **nicht** existieren kann  
Beweismethode: **Diagonalisierung**
- es ist **unentscheidbar**, ob ein beliebiges Programm ein “hello, world” Programm ist

## Unvollständige Liste unentscheidbarer Probleme

## Definition

## Halteproblem

als **Halteproblem** bezeichnen wir das Problem, ob ein beliebiges Programm auf seiner Eingabe hält

## Definition

## PCP

**Postsches Korrespondenzproblem**: Gegeben zwei Listen von Strings der gleichen Länge  $w_1, w_2, \dots, w_n$  und  $x_1, x_2, \dots, x_n$ . Gesucht sind Indizes  $i_1, i_2, \dots, i_m$ , sodass

$$w_{i_1} w_{i_2} \dots w_{i_m} = x_{i_1} x_{i_2} \dots x_{i_m}$$

## Satz

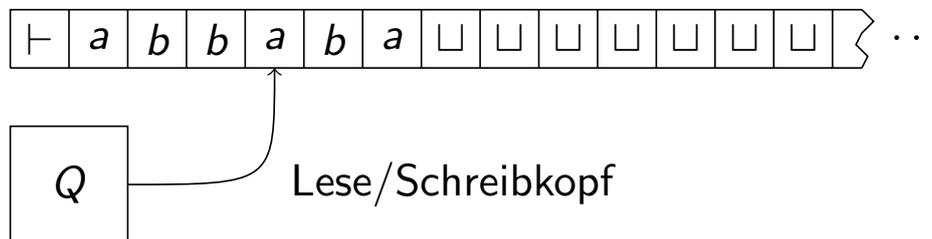
die folgenden Probleme sind **unentscheidbar**:

- das Halteproblem
- das Postsche Korrespondenzproblem
- ist eine beliebige Sprache regulär?

# Turingmaschinen

## Definition - informell

deterministische, einbändige Turingmaschine:



## Church-Turing These

jedes algorithmisch lösbare Problem ist mit einer Turingmaschine lösbar

## Definition - formal

## Turingmaschine

eine deterministische, einbändige Turingmaschine  $M$  ist ein 9-Tupel

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

sodass

- 1  $Q$  eine endliche Menge von Zuständen,
- 2  $\Sigma$  eine endliche Menge von Eingabesymbolen,
- 3  $\Gamma$  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  $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  die Übergangsfunktion,
- 7  $s \in Q$ , der Startzustand,
- 8  $t \in Q$ , der akzeptierende Zustand und
- 9  $r \in Q$ , der verwerfende Zustand mit  $t \neq r$ .

## Beispiel: Binärer Nachfolger

betrachte

$$M = (\{s, t, r, p\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \delta, s, t, r)$$

mit  $\delta$ :

$p \in Q$	$a \in \Gamma$	$\delta(p, a)$
$s$	$0$	$(s, 0, R)$
$s$	$1$	$(s, 1, R)$
$s$	$\sqcup$	$(p, \sqcup, L)$
$s$	$\vdash$	$(s, \vdash, R)$
$p$	$0$	$(t, 1, L)$
$p$	$1$	$(p, 0, L)$
$p$	$\vdash$	$(t, \vdash, R)$

### Zusatzbedingungen

- $\forall p \in Q, \exists q \in Q$  sodass:

$$\delta(p, \vdash) = (q, \vdash, R)$$

- $\forall b \in \Gamma \exists c, c' \in \Gamma$  und  $d, d' \in \{L, R\}$ :

$$\delta(t, b) = (t, c, d)$$

$$\delta(r, b) = (r, c', d')$$

### Beispiel

### Binärer Nachfolger (2)

$p \in Q$	$a \in \Gamma$	$\delta(p, a)$
$t$	$*$	$*$
$r$	$*$	$*$

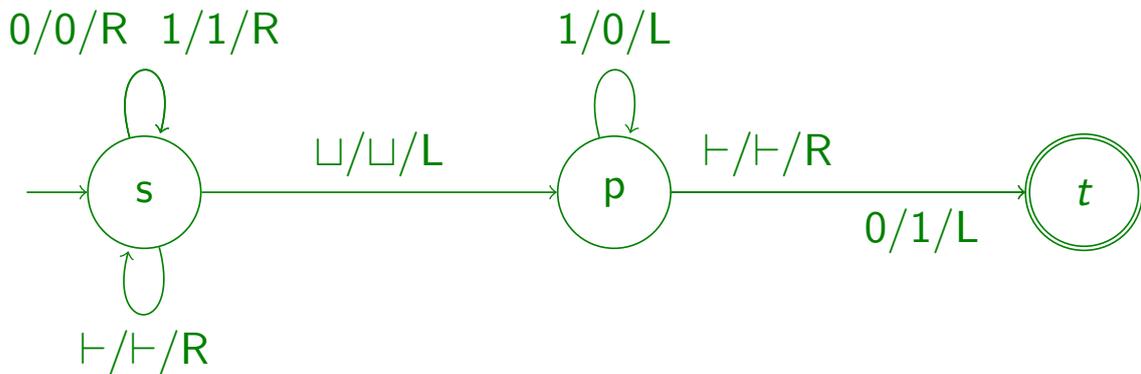
sodass **Zusatzbedingungen** erfüllt

## Beispiel: Binärer Nachfolger (3)

betrachte

$$M = (\{s, t, r, p\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \delta, s, t, r)$$

mit  $\delta$ :



## Demo: Turing Machine Simulator

### Installation

[http://sourceforge.net/project/showfiles.php?group\\_id=210379&package\\_id=252627](http://sourceforge.net/project/showfiles.php?group_id=210379&package_id=252627)

### Beispiel

```
java -jar TuataraTMSim.jar
```

### Definition

Konfiguration

eine **Konfiguration** einer TM  $M$  ist ein Tripel  $(p, x, n)$ , sodass

- $p \in Q$  Zustand,
- $x = y \sqcup^\infty$  Bandinhalt
- $n \in \mathbb{N}$  Position des Lese/Schreibkopfes

$y \in \Gamma^*$

### Definition

Startkonfiguration

**Startkonfiguration** bei Eingabe  $x \in \Sigma^*$ :

$$(s, \vdash x \sqcup^\infty, 0)$$

### Definition

$\xrightarrow[M]{1}$

Relation  $\xrightarrow[M]{1}$  ist wie folgt definiert:

$$(p, z, n) \xrightarrow[M]{1} \begin{cases} (q, z', n - 1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n + 1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

$z'$  den String, den wir aus  $z$  erhalten, wenn  $z_n$  durch  $b$  ersetzt

### Definition

$\xrightarrow[M]{*}$

reflexive, transitive Hülle  $\xrightarrow[M]{*}$ :

- $\alpha \xrightarrow[M]{*} \alpha$
- $\alpha \xrightarrow[M]{n+1} \beta$ , wenn  $\alpha \xrightarrow[M]{n} \gamma \xrightarrow[M]{1} \beta$  für Konfiguration  $\gamma$
- $\alpha \xrightarrow[M]{*} \beta$ , wenn  $\exists n \alpha \xrightarrow[M]{n} \beta$

$p \in Q$	$a \in \Gamma$	$\delta(p, a)$	
$s$	$0$	$(s, 0, R)$	$(s, \vdash 0010 \sqcup^\infty, 0) \xrightarrow[M]{*}$
$s$	$1$	$(s, 1, R)$	
$s$	$\sqcup$	$(p, \sqcup, L)$	$(s, \vdash 0010 \sqcup^\infty, 5) \xrightarrow[M]{1}$
$s$	$\vdash$	$(s, \vdash, R)$	$(p, \vdash 0010 \sqcup^\infty, 4) \xrightarrow[M]{1}$
$p$	$0$	$(t, 1, L)$	
$p$	$1$	$(p, 0, L)$	$(t, \vdash 0011 \sqcup^\infty, 3)$
$p$	$\vdash$	$(t, \vdash, R)$	

## Definition

Turingmaschine  $M$ 

- **akzeptiert**  $x \in \Sigma^*$ , wenn  $\exists y, n$ :

$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (t, y, n)$$

- **verwirft**  $x \in \Sigma^*$ , wenn  $\exists y, n$ :

$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (r, y, n)$$

- **hält** bei Eingabe  $x$ , wenn entweder  $x$  akzeptiert oder verworfen, andernfalls **hält**  $M$  **nicht**
- ist **total**, wenn  $M$  auf **allen** Eingaben hält

## Definition

Sprache einer TM

$$L(M) := \{x \in \Sigma^* \mid M \text{ akzeptiert } x\}$$

## Beispiel

betrachte  $M = (\{s, t, r, q_0, q_1, q'_0, q'_1\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \delta, s, t, r)$  mit  $\delta$ :

$p \in Q$	$a \in \Gamma$	$\delta(p, a)$	$p \in Q$	$a \in \Gamma$	$\delta(p, a)$
$s$	$0$	$(q_0, \vdash, R)$	$q'_0$	$0$	$(q, \sqcup, L)$
$s$	$1$	$(q_1, \vdash, R)$	$q'_0$	$1$	$(r, 1, L)$
$s$	$\vdash$	$(s, \vdash, R)$	$q'_0$	$\vdash$	$(r, \vdash, R)$
$s$	$\sqcup$	$(t, \sqcup, L)$	$q'_1$	$0$	$(r, 1, L)$
$q_0$	$0$	$(q_0, 0, R)$	$q'_1$	$1$	$(q, \sqcup, L)$
$q_0$	$1$	$(q_0, 1, R)$	$q'_1$	$\vdash$	$(r, \vdash, R)$
$q_0$	$\sqcup$	$(q'_0, \sqcup, L)$	$q$	$0$	$(q, 0, L)$
$q_1$	$0$	$(q_1, 0, R)$	$q$	$1$	$(q, 1, L)$
$q_1$	$1$	$(q_1, 1, R)$	$q$	$\vdash$	$(s, \vdash, R)$
$q_1$	$\sqcup$	$(q'_1, \sqcup, L)$			

es gilt;  $L(M) = \{ww^R \mid w \in \{0, 1\}^*\}$