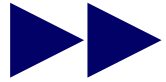


# Theoretische Informatik

## Kap 3: Komplexitätstheorie

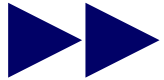


### *Gliederung der Vorlesung*

- 0. Grundbegriffe
- 1. Formale Sprachen/Automatentheorie
  - 1.1. Grammatiken
  - 1.2. Reguläre Sprachen
  - 1.3. Kontextfreie Sprachen
- 2. Berechnungstheorie
  - 2.1. Berechenbarkeitsmodelle
  - 2.2. Die Churchsche These
  - 2.3. Unentscheidbarkeit
- 3. Komplexitätstheorie
  - 3.1. Nicht-deterministische Turing Maschinen
  - 3.2. Komplexitätsmaße**
  - 3.3. Das P=NP? Problem

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### zentraler Begriff

es sei  $\Sigma$  das zugrunde liegende Alphabet

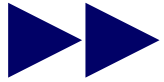
es sei  $M$  eine sprachakzeptierende det. Turing-Maschine mit dem Eingabealphabet  $\Sigma$

$M$  stoppt bei jeder Eingabe, falls für alle  $w \in \Sigma^*$  gilt:

- $M$  akzeptiert die Eingabe  $w$  (/\* d.h.  $M$  überführt die Anfangskonfiguration  $z_0w$  in eine akzeptierende Konfiguration  $uz_e v$  \*/) oder
- $M$  kommt bei der Eingabe  $w$  in einen Fehlerzustand (/\* d.h.  $M$  überführt die Anfangskonfiguration  $z_0w$  in eine Konfiguration  $uz_i v$ , in der keine Aktion mehr möglich ist \*/) )

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### **Ressourcen: Rechenzeit und Speicherplatz**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $M$  eine sprachakzeptierende det. Turing-Maschine mit dem Eingabealphabet  $\Sigma$ , die bei jeder Eingabe stoppt

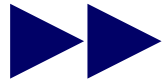
es sei  $w \in \Sigma^*$

$T_M(w)$  bezeichnet die Anzahl der Schritte, die  $M$  bei Eingabe von  $w$  rechnet, bevor  $M$  stoppt.

$S_M(w)$  bezeichnet die Anzahl der Zellen des Arbeitsbands, die  $M$  bei Eingabe von  $w$  besucht, bevor  $M$  stoppt.

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### Zeit- und Platzschranken

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $M$  eine sprachakzeptierende det. Turing-Maschine mit dem Eingabealphabet  $\Sigma$ , die bei jeder Eingabe stoppt

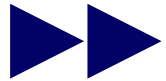
es seien  $t: \mathbb{N} \rightarrow \mathbb{N}$  und  $s: \mathbb{N} \rightarrow \mathbb{N}$

$M$  heißt  $t(n)$ -zeitbeschränkt, falls für alle  $w \in \Sigma^*$  gilt:  $T_M(w) \leq t(|w|)$ .

$M$  heißt  $s(n)$ -bandbeschränkt, falls für alle  $w \in \Sigma^*$  gilt:  $S_M(w) \leq s(|w|)$ .

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### Komplexitätsmaße

**Beispiel** eine det. sprachakzeptierende TM M für die Sprache L mit  
 $L = \{ w\underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w} \text{ ist die gespiegelte Version von } w \}$

	a	b	B
$z_0$	$z_1, B, R$	$z_3, B, R$	
$z_1$	$z_1, a, R$	$z_1, b, R$	$z_2, B, L$
$z_2$	$z_5, B, L$		
$z_3$	$z_3, a, R$	$z_3, b, R$	$z_4, B, L$
$z_4$		$z_5, B, L$	
$z_5$	$z_6, a, L$	$z_6, b, L$	$z_e, B, N$
$z_6$	$z_6, a, L$	$z_6, b, L$	$z_0, B, R$

sinnvolle Zeitschranke:

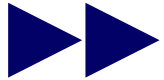
$$t(n) = (n+1)^2$$

sinnvolle Platzschranke:

$$s(n) = n+1$$

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

**eine wichtige det. Zeitkomplexitätsklasse**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

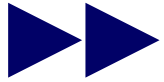
$L \in P$ , falls es eine det. sprachakzeptierende TM  $M$  gibt, die auf jeder Eingabe stoppt und die folgende Eigenschaften hat:

- $L(M) = L$
- es gibt Zahlen  $r, c > 0$ , so daß für alle  $w \in \Sigma^*$  gilt:  $T_M(w) \leq c \cdot |w|^r$

...  $P$  bezeichnet die Klasse aller Sprachen  $L \subseteq \Sigma^*$ , die mit det. TM akzeptiert werden können, die polynomielle Rechenzeit benötigen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

**eine wichtige det. Platzkomplexitätsklasse**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

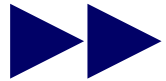
$L \in \text{PSPACE}$ , falls es eine det. sprachakzeptierende TM  $M$  gibt, die auf jeder Eingabe stoppt und die folgende Eigenschaften hat:

- $L(M) = L$
- es gibt Zahlen  $r, c > 0$ , so daß für alle  $w \in \Sigma^*$  gilt:  $S_M(w) \leq c \cdot |w|^r$

... PSPACE bezeichnet die Klasse aller Sprachen  $L \subseteq \Sigma^*$ , die mit det. TM akzeptiert werden können, die polynomiellen Speicherplatz benötigen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

**eine andere wichtige det. Platzkomplexitätsklasse**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

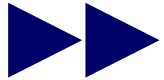
$L \in \text{DLBA}$ , falls es eine det. sprachakzeptierende TM  $M$  gibt, die auf jeder Eingabe stoppt und die folgende Eigenschaften hat:

- $L(M) = L$
- es gibt eine Zahl  $c > 0$ , so daß für alle  $w \in \Sigma^*$  gilt:  $S_M(w) \leq c \cdot |w|$

... DLBA bezeichnet die Klasse aller Sprachen  $L \subseteq \Sigma^*$ , die mit det. TM akzeptiert werden können, die linearen Speicherplatz benötigen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### zentraler Begriff

es sei  $\Sigma$  das zugrunde liegende Alphabet

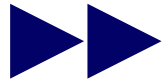
es sei  $N$  eine sprachakzeptierende nicht-det. Turing-Maschine mit dem Eingabealphabet  $\Sigma$

$N$  stoppt bei jeder Eingabe, falls für alle  $w \in \Sigma^*$  gilt:

- $N$  akzeptiert die Eingabe  $w$  (/\* d.h.  $N$  kann die Anfangskonfiguration  $z_0w$  in eine akzeptierende Konfiguration  $uz_e v$  überführen \*/) oder
- $N$  kommt bei der Eingabe  $w$  in einen Fehlerzustand (/\* d.h.  $N$  überführt die Anfangskonfiguration  $z_0w$  stets in eine Konfiguration  $uz_1 v$ , in der keine Aktion mehr möglich ist \*/) )

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### **Ressourcen: Rechenzeit und Speicherplatz**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $N$  eine sprachakzeptierende det. Turing-Maschine mit dem Eingabealphabet  $\Sigma$ , die bei jeder Eingabe stoppt

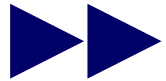
es sei  $w \in \Sigma^*$

Wenn  $w \in L(N)$  gilt, so bezeichnet  $T_N(w)$  die minimale Anzahl der Schritte, die  $N$  bei Eingabe von  $w$  rechnet, bevor  $N$  die Eingabe  $w$  akzeptiert. Sonst ist  $T_N(w) = 1$ .

Wenn  $w \in L(N)$  gilt, so bezeichnet  $S_N(w)$  die minimale Anzahl der Zellen des Arbeitsbands, die  $M$  bei Eingabe von  $w$  besucht, bevor  $N$  die Eingabe  $w$  akzeptiert. Sonst ist  $S_N(w) = 1$ .

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### Zeit- und Platzschranken

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $N$  eine sprachakzeptierende nicht-det. Turing-Maschine mit dem Eingabealphabet  $\Sigma$ , die bei jeder Eingabe stoppt

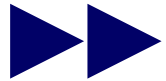
es seien  $t: \mathbb{N} \rightarrow \mathbb{N}$  und  $s: \mathbb{N} \rightarrow \mathbb{N}$

$N$  heißt  $t(n)$ -zeitbeschränkt, falls für alle  $w \in \Sigma^*$  gilt:  $T_N(w) \leq t(|w|)$ .

$N$  heißt  $s(n)$ -bandbeschränkt, falls für alle  $w \in \Sigma^*$  gilt:  $S_N(w) \leq s(|w|)$ .

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### Komplexitätsmaße

**Beispiel** eine nicht-det. sprachakzeptierende TM N für die Sprache L mit  $L = \{ w \mid w \in \{ a,b \}^*, w \text{ enthält das Teilwort baba} \}$

	a	b	B
$z_0$	$z_0, a, R$	$z_0, b, R$ $z_1, b, R$	
$z_1$	$z_2, a, R$		
$z_2$		$z_3, b, R$	
$z_3$	$z_e, a, N$		

sinnvolle Zeitschranke:

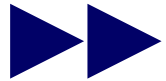
$$t(n) = n$$

sinnvolle Platzschranke:

$$s(n) = n$$

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

**eine wichtige nicht-det. Zeitkomplexitätsklasse**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

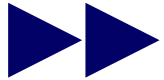
$L \in \text{NP}$ , falls es eine nicht-det. sprachakzeptierende TM  $N$  gibt, die auf jeder Eingabe stoppt und die folgende Eigenschaften hat:

- $L(N) = L$
- es gibt Zahlen  $r, c > 0$ , so daß für alle  $w \in \Sigma^*$  gilt:  $T_N(w) \leq c \cdot |w|^r$

... NP bezeichnet die Klasse aller Sprachen  $L \subseteq \Sigma^*$ , die mit nicht-det. TM akzeptiert werden können, die polynomielle Rechenzeit benötigen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

#### eine wichtige nicht-det. Platzkomplexitätsklasse

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

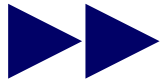
$L \in \text{NPSPACE}$ , falls es eine nicht-det. sprachakzeptierende TM  $N$  gibt, die auf jeder Eingabe stoppt und die folgende Eigenschaften hat:

- $L(N) = L$
- es gibt Zahlen  $r, c > 0$ , so daß für alle  $w \in \Sigma^*$  gilt:  $S_N(w) \leq c \cdot |w|^r$

... NPSPACE bezeichnet die Klasse aller Sprachen  $L \subseteq \Sigma^*$ , die mit nicht-det. TM akzeptiert werden können, die polynomiellen Speicherplatz benötigen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

**eine andere wichtige nicht-det. Platzkomplexitätsklasse**

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

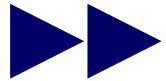
$L \in \text{LBA}$ , falls es eine nicht-det. sprachakzeptierende TM  $N$  gibt, die auf jeder Eingabe stoppt und die folgende Eigenschaften hat:

- $L(N) = L$
- es gibt eine Zahl  $c > 0$ , so daß für alle  $w \in \Sigma^*$  gilt:  $S_N(w) \leq c \cdot |w|$

... LBA bezeichnet die Klasse aller Sprachen  $L \subseteq \Sigma^*$ , die mit nicht-det. TM akzeptiert werden können, die linearen Speicherplatz benötigen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

ein interessantes Resultat und zwei wichtige offene Fragen

#### **Satz**

$PSPACE = NPSPACE$ .

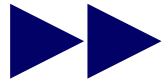
#### **offene Fragen**

- (1) Gilt  $DLBA = LBA$  ?
- (2) Gilt  $P = NP$  ?

Vermutung: beide Fragen sind mit „Nein“ zu beantworten

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

Warum ist die Frage, ob  $DLBA = LBA$  gilt, wichtig?

es sei  $\Sigma$  das zugrunde liegende Alphabet

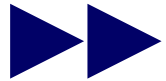
es sei  $L \subseteq \Sigma^*$

#### **Satz**

Wenn  $L$  eine kontextsensitive Sprache ist, so gilt  $L \in LBA$ .

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### Komplexitätsmaße

ein Beispiel zur Illustration

es sei  $L = L(G)$

$\Sigma = \{ a, b, c \}$

$V = \{ S, S', H \}$

S

$S \rightarrow abc$

$S \rightarrow aS'bc$

$S' \rightarrow abH$

$S' \rightarrow aS'bH$

$Hb \rightarrow bH$

$Hc \rightarrow cc$

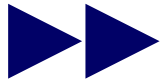
es sei  $w = aabbcc$

*Nachweis für  $w \in L(G)$*

...	B	a	a	b	b	c	c	B	...
...	B	a	a	b	b	c	c	B	...
...	B	a	a	b	b	H	c	B	...
...	B	a	a	b	b	H	c	B	...
...	B	a	a	b	H	b	c	B	...
...	B	a	a	b	H	b	c	B	...
...	B	a	S'	b	c	B	B	B	...
...	B	S				B	B	B	...
...	B	S	B	B	B	B	B	B	...

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### Komplexitätsmaße

#### Beweisidee

es sei  $\Sigma$  das zugrunde liegende Alphabet, es sei  $L \subseteq \Sigma^*$  eine kontextsensitive Sprache

seien  $(l_1, r_1), \dots, (l_n, r_n)$  die Regeln einer kontextsensitiven Grammatik für  $L$

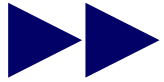
bei Eingabe von  $w = a_1 \dots a_m$  geht die gesuchte nicht-det. TM  $N$  wie folgt vor:

1.  $N$  wählt eine Regel  $(l_k, r_k)$
2.  $N$  sucht ein Vorkommen von  $r_k$  im aktuellen Bandinhalt
  - falls die Suche erfolgreich war, so ersetzt  $N$  das Vorkommen von  $r_k$  durch  $l_k$  und verschiebt den Bandinhalt so, daß „Lücken“ beseitigt werden
3. falls nur noch das Startsymbol auf dem Band steht, so geht  $N$  in einen akzeptierenden Zustand; andernfalls geht  $N$  zu 1

→ wegen  $|l_k| \leq |r_k|$  für alle  $k$ , besucht  $N$  höchstens  $|w|$  viele Zellen

# Theoretische Informatik

## Kap 3: Komplexitätstheorie



### *Komplexitätsmaße*

es sei  $\Sigma$  das zugrunde liegende Alphabet

es sei  $L \subseteq \Sigma^*$

#### **Satz**

Wenn  $L \in \text{LBA}$  gilt, so ist  $L$  eine kontextsensitive Sprache.

... die Frage, ob  $\text{DLBA} = \text{LBA}$  gilt, entspricht der Frage, ob det. Turing-Maschinen, die nur linear viel Speicherplatz benötigen, für das Akzeptieren kontextsensitiver Sprachen „ausreichen“