

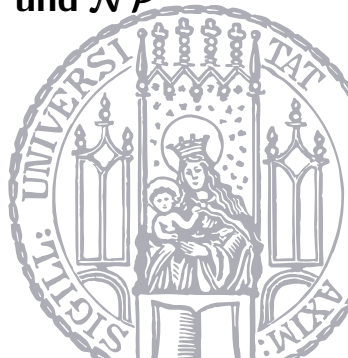
## 11b

# Laufzeitkomplexität und die Klassen $\mathcal{P}$ und $\mathcal{NP}$

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für  
Theoretische Informatik und Theorembeweisen

Stand: 10. Mai 2024  
Basiert auf Folien von PD Dr. David Sabel



Die Komplexitätstheorie ist ein Teilgebiet der Theoretischen Informatik, das sich mit der Komplexität von **entscheidbaren Problemen** befasst.

Maße zum Messen des Ressourcenbedarfs von Algorithmen:

- ▶ Rechenzeit
- ▶ Platzbedarf
- ▶ usw.

Die Komplexität eines Problems ist die Komplexität des **besten** Algorithmus bezüglich des Maßes.

Das Ziel ist die Einordnung von Problemen in **Komplexitätsklassen**.

## 1. Teilbereich: Nachweis von **oberen Schranken**:

- ▶ Finde möglichst guten Algorithmus für konkretes Problem.
- ▶ Analysiere die Laufzeit- und Platzkomplexität.
- ▶ Gesucht sind möglichst genaue Schranken (bezüglich der  $O$ -Notation).

## 2. Teilbereich: Nachweis von **unteren Schranken**:

- ▶ Zeige, dass es keinen besseren Algorithmus gibt.
- ▶ Schwieriger, da man über **alle** Algorithmen argumentieren muss.
- ▶ Gesucht werden möglichst genaue Schranken (bezüglich der  $\Omega$ -Notation).

3. Teilbereich: Auswirkung der **Maschinenmodelle auf den Ressourcenbedarf**  
insbesondere **Determinismus vs. Nichtdeterminismus**

- ▶ Eine wichtige ungelöste Frage ist das  **$\mathcal{P}$ -vs.- $\mathcal{NP}$ -Problem**.
- ▶ Komplexitätsklassen sind im Allgemeinen ungenauer (größer) als in den vorher genannten Teilbereichen.
- ▶ Wir beschäftigen uns im letzten Teil des Kurses hiermit.

Turingmaschinen:

- ▶ Wir betrachten nur **entscheidbare** Sprachen.
- ▶ Deswegen nehmen wir an:  
Die Turingmaschinen, welche diese Sprachen entscheiden,  
**halten auf jeder Eingabe an**.
- ▶ DTMs: Wir nehmen an, dass es „Verwirf“-Zustände gibt,  
für die die Turingmaschine keine Nachfolgekonfiguration besitzen.
- ▶ NTMs haben solche Zustände von Haus aus.

Mehrband- vs. Einband-Turingmaschinen:

- ▶ Wir nehmen **Mehrband-TMs**, da sie besser zu „normalen“ Rechnern passen: Kein „Vergeuden“ von Rechenzeit, nur um die Eingaben zu suchen.
- ▶ Unterschied:  $n$  Schritte auf Mehrband-TM können in  $O(n^2)$  Schritten auf Einband-TM ausgeführt werden.
- ▶ Der Unterschied macht sich in den Komplexitätsklassen  $\mathcal{P}$  und  $\mathcal{NP}$  nicht bemerkbar.

# Die Klasse $TIME(f(n))$

## Definition

Sei  $M$  eine stets anhaltende Mehrband-DTM mit Startzustand  $z_0$ .

Für Eingabe  $w$  definieren wir

$$time_M(w) := i$$

wenn für die Startkonfiguration für  $z_0$  und  $w$  nach  $i$  Schritten ein Endzustand oder Verwirf-Zustand erreichbar ist.



# Die Klasse $TIME(f(n))$

## Definition

Sei  $M$  eine stets anhaltende Mehrband-DTM mit Startzustand  $z_0$ .

Für Eingabe  $w$  definieren wir

$$time_M(w) := i$$

wenn für die Startkonfiguration für  $z_0$  und  $w$  nach  $i$  Schritten ein Endzustand oder Verwirf-Zustand erreichbar ist.

## Definition

Für eine Funktion  $f : \mathbb{N} \rightarrow \mathbb{N}$  sei die Klasse  $TIME(f(n))$  genau die Menge der Sprachen  $L$ , für die es eine stets anhaltende Mehrband-DTM  $M$  gibt mit  $L(M) = L$  und  $time_M(w) \leq f(|w|)$  für alle  $w \in \Sigma^*$ .

# Die Klasse $TIME(f(n))$

## Definition

Sei  $M$  eine stets anhaltende Mehrband-DTM mit Startzustand  $z_0$ .

Für Eingabe  $w$  definieren wir

$$time_M(w) := i$$

wenn für die Startkonfiguration für  $z_0$  und  $w$  nach  $i$  Schritten ein Endzustand oder Verwirf-Zustand erreichbar ist.

## Definition

Für eine Funktion  $f : \mathbb{N} \rightarrow \mathbb{N}$  sei die Klasse  $TIME(f(n))$  genau die Menge der Sprachen  $L$ , für die es eine stets anhaltende Mehrband-DTM  $M$  gibt mit  $L(M) = L$  und  $time_M(w) \leq f(|w|)$  für alle  $w \in \Sigma^*$ .

Eine Sprache ist daher in  $TIME(f(n))$ , wenn sie von einer DTM für jede Eingabe der Länge  $n$  in  $\leq f(n)$  Schritten entschieden wird.

## Definition

Ein **Polynom** ist eine Funktion  $p : \mathbb{N} \rightarrow \mathbb{N}$  von der Form

$$p(n) = \sum_{i=0}^k a_i \cdot n^i = a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0$$

mit  $a_i \in \mathbb{N}$  und  $k \in \mathbb{N}$ .

## Definition

Die Klasse  $\mathcal{P}$  ist definiert als

$$\mathcal{P} := \bigcup_{p \text{ Polynom}} \text{TIME}(p(n))$$

## Definition

Die Klasse  $\mathcal{P}$  ist definiert als

$$\mathcal{P} := \bigcup_{p \text{ Polynom}} \text{TIME}(p(n))$$

Sprechweise: Problem ist **effizient lösbar** = Problem in  $\mathcal{P}$ .

# Zugehörigkeit zu $\mathcal{P}$

---

Formale Sprache  $L$  ist in Klasse  $\mathcal{P}$  enthalten, wenn:

Es gibt eine stets anhaltende DTM  $M$  und ein Polynom  $p$  mit

- ▶  $L(M) = L$
- ▶  $time_M(w) \leq p(|w|)$  für jedes  $w \in \Sigma^*$ .

# Nichtdeterminismus

---

Auch für NTMs nehmen wir an, dass sie auf allen Berechnungspfaden anhalten.

Beachte: Ein Wort wird **akzeptiert**, wenn es auf **einen** Berechnungspfad akzeptiert wird.

Auch für NTMs nehmen wir an, dass sie auf allen Berechnungspfaden anhalten.

Beachte: Ein Wort wird **akzeptiert**, wenn es auf **einen** Berechnungspfad akzeptiert wird.

## Definition

Sei  $M$  eine stets anhaltende Mehrband-NTM, die für jede Eingabe anhält.  
Dann definieren wir die Laufzeit von  $M$  als

$$ntime_M(w) := \max \{i \mid \text{für die Startkonfiguration für } z_0 \text{ und } w \text{ hält } M \text{ in } i \text{ Schritten}\}$$



Auch für NTMs nehmen wir an, dass sie auf allen Berechnungspfaden anhalten.

Beachte: Ein Wort wird **akzeptiert**, wenn es auf **einen** Berechnungspfad akzeptiert wird.

## Definition

Sei  $M$  eine stets anhaltende Mehrband-NTM, die für jede Eingabe anhält.  
Dann definieren wir die Laufzeit von  $M$  als

$$ntime_M(w) := \max \{i \mid \text{für die Startkonfiguration für } z_0 \text{ und } w \text{ hält } M \text{ in } i \text{ Schritten}\}$$

Beachte: Schöning verwendet eine andere Definition. Es macht für die Definition der Klasse  $\mathcal{NP}$  aber keinen Unterschied.

# Die Klassen $NTIME(f(n))$ und $\mathcal{NP}$

## Definition

Für eine Funktion  $f : \mathbb{N} \rightarrow \mathbb{N}$  sei die Klasse  $NTIME(f(n))$  genau die Menge der Sprachen  $L$ , für die es eine stets anhaltende Mehrband-NTM  $M$  gibt mit  $L(M) = L$  und  $ntime_M(w) \leq f(|w|)$  für alle  $w \in \Sigma^*$ .

# Die Klassen $NTIME(f(n))$ und $\mathcal{NP}$

## Definition

Für eine Funktion  $f : \mathbb{N} \rightarrow \mathbb{N}$  sei die Klasse  $NTIME(f(n))$  genau die Menge der Sprachen  $L$ , für die es eine stets anhaltende Mehrband-NTM  $M$  gibt mit  $L(M) = L$  und  $ntime_M(w) \leq f(|w|)$  für alle  $w \in \Sigma^*$ .

## Definition

Die Klasse  $\mathcal{NP}$  ist definiert als

$$\mathcal{NP} := \bigcup_{p \text{ Polynom}} NTIME(p(n))$$

$$\mathcal{P} \subseteq \mathcal{NP}$$

---

### Lemma

Es gilt  $TIME(f(n)) \subseteq NTIME(f(n))$  und damit auch  $\mathcal{P} \subseteq \mathcal{NP}$ .

**Lemma**

Es gilt  $TIME(f(n)) \subseteq NTIME(f(n))$  und damit auch  $\mathcal{P} \subseteq \mathcal{NP}$ .

**Beweis** Jede DTM kann leicht als NTM aufgefasst werden, wobei es nur einen möglichen Berechnungspfad gibt.

**Lemma**

Es gilt  $TIME(f(n)) \subseteq NTIME(f(n))$  und damit auch  $\mathcal{P} \subseteq \mathcal{NP}$ .

**Beweis** Jede DTM kann leicht als NTM aufgefasst werden, wobei es nur einen möglichen Berechnungspfad gibt.

Dieser bestimmt  $ntime_M(w)$  und es gilt für diese Turingmaschine  $time_M(w) = ntime_M(w)$ .

**Lemma**

Es gilt  $TIME(f(n)) \subseteq NTIME(f(n))$  und damit auch  $\mathcal{P} \subseteq \mathcal{NP}$ .

**Beweis** Jede DTM kann leicht als NTM aufgefasst werden, wobei es nur einen möglichen Berechnungspfad gibt.

Dieser bestimmt  $ntime_M(w)$  und es gilt für diese Turingmaschine  $time_M(w) = ntime_M(w)$ .

Damit folgt, dass aus  $L \in TIME(f(n))$  auch  $L \in NTIME(f(n))$  folgt.

**Lemma**

Es gilt  $TIME(f(n)) \subseteq NTIME(f(n))$  und damit auch  $\mathcal{P} \subseteq \mathcal{NP}$ .

**Beweis** Jede DTM kann leicht als NTM aufgefasst werden, wobei es nur einen möglichen Berechnungspfad gibt.

Dieser bestimmt  $ntime_M(w)$  und es gilt für diese Turingmaschine  $time_M(w) = ntime_M(w)$ .

Damit folgt, dass aus  $L \in TIME(f(n))$  auch  $L \in NTIME(f(n))$  folgt.

Schließlich zeigt dies auch  $\mathcal{P} \subseteq \mathcal{NP}$ . □



Die Frage „Gilt  $\mathcal{P} = \mathcal{NP}$  oder  $\mathcal{P} \neq \mathcal{NP}$ ?“ ist bis heute **ungelöst**.

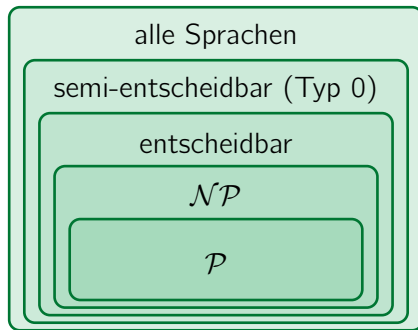
- ▶ Das  $\mathcal{P}$ -vs.- $\mathcal{NP}$ -Problem ist eines der sieben sogenannten Millennium-Probleme, die vom Clay Mathematics Institute im Jahr 2000 als Liste ungelöster Probleme der Mathematik herausgegeben wurde.
- ▶ Für dessen Lösung wurde ein Preisgeld von einer Million US-Dollar ausgelobt.

Ein **wesentlicher Grund**, der für  $\mathcal{P} \neq \mathcal{NP}$  spricht:

Man müsste für  $\mathcal{P} = \mathcal{NP}$  einen deterministischen Polynomialzeitalgorithmus finden, für ein Problem, für das bisher nur deterministische Exponentialzeitalgorithmen bekannt sind.

Viele haben gesucht, keiner hat einen solchen Algorithmus gefunden.

# Lage der Komplexitätsklasse



Dabei unklar, ob  $\mathcal{NP} = \mathcal{P}$