

# Diagonalisierung und polynomielle Hierarchie

Corvin Paul    Matthias Schimek

13.05.2015

# Gliederung

- 1 Einleitung
  - Erster Unterabschnittstitel
  - Zweiter Unterabschnittstitel

# Gliederung

- 1 Einleitung
  - Erster Unterabschnittstitel
  - Zweiter Unterabschnittstitel

Überschriften müssen informativ sein.  
Korrekte Groß-/Kleinschreibung beachten.  
Untertitel sind optional.

- Viel `itemize` benutzen.
- Sehr kurze Sätze oder Satzglieder verwenden.

# Überschriften müssen informativ sein.

Man kann Overlays erzeugen. . .

- mit dem `pause`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
  - Erster Punkt.
  - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.

# Überschriften müssen informativ sein.

Man kann Overlays erzeugen. . .

- mit dem `pause`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
  - Erster Punkt.
  - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.

# Überschriften müssen informativ sein.

Man kann Overlays erzeugen. . .

- mit dem `pause`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
  - Erster Punkt.
  - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.

# Überschriften müssen informativ sein.

Man kann Overlays erzeugen. . .

- mit dem `pause`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
  - Erster Punkt.
  - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.



# Überschriften müssen informativ sein.

Man kann Overlays erzeugen. . .

- mit dem `pause`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
  - Erster Punkt.
  - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.

# Überschriften müssen informativ sein.

Man kann Overlays erzeugen. . .

- mit dem `pause`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
  - Erster Punkt.
  - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
  - Erster Punkt.
  - Zweiter Punkt.

# Vorraussetzungen

Wiederholung :

- Für  $i \in \mathbb{N}$  beschreibt  $i$  die TM  $M_i$
- Jede TM wird von unendlich vielen  $i \in \mathbb{N}$  beschrieben
- Es existiert eine universelle TM  $U$ , die jede TM mit logarithmischem Overhead simulieren kann

# Vorraussetzungen

Wiederholung :

- Für  $i \in \mathbb{N}$  beschreibt  $i$  die TM  $M_i$
- Jede TM wird von unendlich vielen  $i \in \mathbb{N}$  beschrieben
- Es existiert eine universelle TM  $U$ , die jede TM mit logarithmischem Overhead simulieren kann

# Vorraussetzungen

Wiederholung :

- Für  $i \in \mathbb{N}$  beschreibt  $i$  die TM  $M_i$
- Jede TM wird von unendlich vielen  $i \in \mathbb{N}$  beschrieben
- Es existiert eine universelle TM  $U$ , die jede TM mit logarithmischem Overhead simulieren kann

# Vorraussetzungen

## Beispiel (Universelle TM)

TM  $M_i$  läuft bei Eingabe  $x$  in  $O(f(n)) \Rightarrow$  TM  $U$  läuft bei Eingabe  $i, x$  in  $O(f(n)\log(fn))$

# Vorraussetzungen

## Definition (time-constructible functions)

Wir nennen eine Funktion  $f$  time-constructible, falls gilt :  
 $f(n)$  ist in  $O(f(n))$  berechenbar.

## Definition

$\mathbf{DTIME}(f(n)) = \{ L \mid \exists \text{ deterministische Turingmaschine, die } L \text{ in } O(f(n)) \text{ entscheidet} \}$

# Vorraussetzungen

## Definition (time-constructible functions)

Wir nennen eine Funktion  $f$  time-constructible, falls gilt :  
 $f(n)$  ist in  $O(f(n))$  berechenbar.

## Definition

**DTIME** $(f(n)) = \{ L \mid \exists \text{ deterministische Turingmaschine, die } L \text{ in } O(f(n)) \text{ entscheidet} \}$



# Deterministische Time Hierarchy

## Satz (Time Hierarchy Theorem, 65)

*Wenn  $f, g$  time-constructible Funktionen sind die  $f(n)\log(f(n))=o(g(n))$  erfüllen, dann gilt*  
 **$\text{DTIME}(f(n)) \subsetneq \text{DTIME}(g(n))$**

Frage : Warum brauchen wir den Faktor  $\log(f(n))$  ?

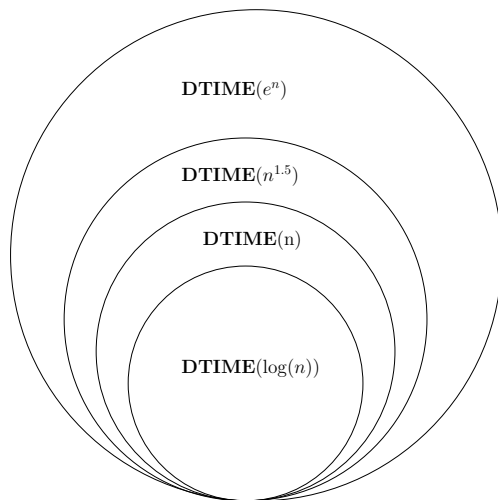
# Deterministische Time Hierarchy

## Satz (Time Hierarchy Theorem, 65)

*Wenn  $f, g$  time-constructible Funktionen sind die  $f(n)\log(f(n))=o(g(n))$  erfüllen, dann gilt*

$$\mathbf{DTIME}(f(n)) \subsetneq \mathbf{DTIME}(g(n))$$

Frage : Warum brauchen wir den Faktor  $\log(f(n))$  ?



# Gliederung

1

## Einleitung

- Erster Unterabschnittstitel
- Zweiter Unterabschnittstitel

# Überschriften müssen informativ sein.

# Überschriften müssen informativ sein.

# Zusammenfassung

- Die **erste Hauptbotschaft** des Vortrags in ein bis zwei Zeilen.
- Die **zweite Hauptbotschaft** des Vortrags in ein bis zwei Zeilen.
- Eventuell noch eine **dritte Botschaft**, aber nicht noch mehr.
- Ausblick
  - Etwas, was wir noch nicht lösen konnten.
  - Nochwas, das wir noch nicht lösen konnten.