

## Theoretische Informatik 3

### WS 2004/05

Volker Strehl  
Informatik 8

October 18, 2004



## Empfohlene Literatur

- ▶ M. AIGNER, Diskrete Mathematik, Vieweg, 2005 (5. Aufl.).
- ▶ T. CORMEN, C. LEISERSON, R. RIVEST, Introduction to Algorithms, MIT Press, 2001 (2. A.).
- ▶ R. GRAHAM, D. KNUTH, O. PATASHNIK, Concrete Mathematics, Addison-Wesley, 1994 (2. A.).
- ▶ V. HEUN, Grundlegende Algorithmen, Vieweg, 2003 (2. A.).
- ▶ D. KNUTH, The Art of Computer Programming (1–3), Addison-Wesley, 1962–1997.
- ▶ U. SCHÖNING, Algorithmik, Spektrum-Verlag, 2001.
- ▶ R. SEDGEWICK, P. FLAJOLET, An Introduction to the Analysis of Algorithms, Addison-Wesley, 1996.
- ▶ H. WILF, Algorithms and Complexity, Prentice-Hall 1986.



## Organisatorisches

- ▶ Vorlesungstermine
  - Montags und Donnerstag, 16:00–17:30 Uhr im H9
- ▶ Übungen in 9 Gruppen, Termine und Eintragung → Webseite
- ▶ Webseite:
  - [www8.informatik.uni-erlangen.de/IMMD8/Lectures/THINF/](http://www8.informatik.uni-erlangen.de/IMMD8/Lectures/THINF/) enthält Materialien zur Vorlesung, Übungsaufgaben, aktuelle Informationen etc. (NB: nicht alles ist von aussen zugänglich)
- ▶ Kontakt:
  - ▶ Adresse: Haberstrasse 2, 3. Stock
  - ▶ Telefon: 85-28712
  - ▶ email: [strehl@cs.fau.de](mailto:strehl@cs.fau.de)
  - ▶ Sprechstunde: Freitag vormittags (9–12), möglichst vorher anmelden



## Übersicht

- ▶ Einleitung – Begriffe und Beispiele
- ▶ Mathematische Hilfsmittel
- ▶ Sortierkomplexität und Entropie
- ▶ Komplexität arithmetischer Operationen
- ▶ Problemkomplexität: P versus NP



## Part I

## Einleitung

## Szenario

## Beispiele

Zwei formale Sprachen

Multiplikation von Polynomen

Sortierkomplexität

Erfüllbarkeit: ein NP-vollständiges Problem

## Probleme

## ▶ Was ist ein Problem?

- ▶  $D$  = domain = Definitionsbereich = "Instanzen"
- ▶  $R$  = range = Wertebereich
- ▶  $f : D \rightarrow R$  berechenbare Funktion
- ▶ falls  $R = \{\text{true}, \text{false}\}$  : Entscheidungsproblem
- ▶ Grösse von Instanzen  $\text{size} : D \rightarrow \mathbb{N}$
- ▶ oft:  $D_n = \{d \in D ; \text{size}(d) = n\}$  ist endlich für  $n \geq 0$ , dann  $d_n = \#D_n$

## Kosten

## ▶ Algorithmen und Kosten

- ▶  $\mathcal{A}$  : Algorithmus zur Berechnung von  $f : D \rightarrow R$   
(auf TM oder RAM oder dergl.), d.h.

$$\forall d \in D : \mathcal{A}(d) = f(d)$$

- ▶ Kostenfunktion (modellabhängig)

$$\text{cost}_{\mathcal{A}} : D \rightarrow \mathbb{N} : d \mapsto \text{cost}_{\mathcal{A}}(d)$$

soll Laufzeitverhalten (oder Speicherverbrauch) von  $\mathcal{A}$  auf einem Maschinenmodell (TM, RAM) anhand der bestimmenden Einflussgrößen (z.B. Anzahl der elementaren Rechenschritte einer TM, Anzahl der arithmetischen oder Vergleichsoperationen einer RAM) wiedergeben



## Begriffe

- ▶ *Komplexitätsanalyse* beschäftigt sich mit dem Entwurf der Analyse von möglichst effizienten Algorithmen für konkrete Probleme
- ▶ *Komplexitätstheorie* untersucht die Komplexität von Problemen und die Komplexitäts-Beziehungen zwischen Problemen. *Komplexitätsklassen* (wie P, NP, EXPTIME, PSPACE) fassen Probleme "gleicher Schwierigkeit" zu Klassen zusammen und untersuchen Beziehungen zwischen diesen Klassen.

## Beispiel: reguläre formale Sprache

- ▶ Für  $F = (a + bb)^* \subseteq \{a, b\}^*$  gilt

$$\begin{aligned} \ell_0(F) &= \ell_1(F) = 1 \\ \ell_{n+2}(F) &= \ell_{n+1}(F) + \ell_n(F) \quad (n \geq 0) \end{aligned}$$

und daher

$$\ell_n(F) = F_{n+1} \text{ Fibonacci-Zahl}$$

Asymptotisch exponentielles Wachstum:

$$F_n = \frac{\phi^n - \hat{\phi}^n}{\sqrt{5}} \sim \frac{\phi^n}{\sqrt{5}}$$

mit  $\phi = \frac{1+\sqrt{5}}{2} = 1.61803\dots$  (goldener Schnitt),  
 $\hat{\phi} = \frac{1-\sqrt{5}}{2} = -0.61803\dots$

## Zwei formale Sprachen

- ▶ Reguläre Sprache  $F = (a + bb)^* \subseteq \{a, b\}^*$ .  
 Wieviele Wörter der Länge  $n$  enthält  $F$ ?
- ▶ Kontextfreie Sprache der korrekten Klammerungen  
 $D \subseteq \{a, b\}^*$ , erzeugt durch

$$D \rightarrow aDbD \mid \lambda.$$

Wieviele Wörter der Länge  $n$  enthält  $D$ ?

- ▶ Allgemein:  
 $\Sigma$  endliches Alphabet,  $L \subseteq \Sigma^*$  formale Sprache  
 Wie verhält sich  $\ell_n(L) := \#(L \cap \Sigma^n)$  für  $n \rightarrow \infty$ ?  
 Welche Rolle spielt dabei der Chomsky-Typ von  $L$ ?

## Beispiel: kontextfreie formale Sprache

- ▶ Für die durch

$$D \rightarrow aDbD \mid \lambda$$

erzeugte kontextfreie Sprache der korrekten Klammerungen (äquivalent: Binärbäume) gilt  $\ell_{2n+1}(D) = 0$  für  $n \geq 0$  und für  $d_n := \ell_{2n}(D)$

$$d_{n+1} = d_0 \cdot d_n + d_1 \cdot d_{n-1} + d_2 \cdot d_{n-2} + \dots + d_n \cdot d_0$$

Wir werden später sehen:

$$d_n = \ell_{2n}(D) = \frac{1}{n+1} \binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi n^3}}$$





## Part II

### Mathematische Hilfsmittel

#### Asymptotische Notation

#### Abschätzung von Summen

Harmonische Zahlen

Stirlings Formel

Polynomsummen

#### Rekursionsgleichungen



## Rekursionen

- ▶ Die Komplexität rekursiver Algorithmen (“divide-and-conquer”) wird durch Rekursionsgleichungen beschrieben!
- ▶ Explizite und exakte Lösungen von Rekursionsgleichungen sind nur in wenigen Fällen möglich — aber oft kann man trotzdem Information über das asymptotische Verhalten von Kostenfunktionen gewinnen.

