Theoretische Informatik

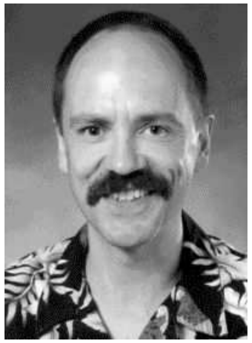
Sommersemester 2004

Christoph Kreitz

Theoretische Informatik, Raum 1.18, Telephon 3060

kreitz@cs.uni-potsdam.de

http://www.cs.uni-potsdam.de/ti/kreitz



1. Themen und Lernziele

2. Organisatorisches

Was ist Theoretische Informatik?

Mathematische Analyse von Grundsatzfragen

- **Problemstellungen**
  - Präzisierung: Wie beschreibt man Probleme?
  - Berechenbarkeit: Ist ein Problem überhaupt lösbar?
  - Effizienz: Ist ein Problem schwer oder leicht lösbar?
- **Algorithmen: abstrakte Lösungsmethoden**
  - Was ist Algorithmus und welche Beschreibungsformen gibt es?
  - Welche grundsätzlichen Merkmale und Eigenschaften haben Algorithmen?
- **Programme: konkrete Lösungsvorschriften**
  - Welche grundsätzlichen Arten von Sprachen gibt es?
  - Syntax: Wie kann man Sprachen beschreiben, erkennen und erzeugen?
  - Semantik: wie beschreibt man die Bedeutung von Programmen
- **Maschinen: Ausführung von “Berechnungen”**
  - Welche grundsätzlichen Typen und Merkmale von Maschinen gibt es?
  - Was können bestimmte Maschinentypen leisten?

Typische Theoretische Fragestellungen

- **Sind Korrekheitsgarantien möglich?**
  - Kein Systemabsturz, kein "Aufhängen", keine falschen Resultate
  - Wie erzeugt man korrekte Programme?
- **Wie effizient kann Software sein?**
  - Wieviel RAM, Plattenplatz, Rechenzeit wird benötigt?
  - Skalierbarkeit: wie groß darf das Problem werden?
- **Wie einfach kann Interaktion gemacht werden?**
  - Wieviel Freiheiten sind in der Formulierung von Programmen möglich?
  - Wie aufwendig wird Syntaxanalyse und Compilierung?
- **Gibt es Grenzen?**
  - Können alle Aufgaben irgendwann von Computern übernommen werden?
  - Werden ineffiziente Programme durch Hardwaresteigerungen handhabbar?
  - Können Computer irgendwann jede natürliche Sprache verstehen?

Themen der Theoretischen Informatik

- **Mathematische Methodik in der Informatik**
- **Automatentheorie und Formale Sprachen**
  - Endliche Automaten und Reguläre Sprachen
  - Kontextfreie Sprachen und Pushdown Automaten
  - Die Chomsky Hierarchie
- **Theorie der Berechenbarkeit**
  - Berechenbarkeitsmodelle
  - Aufzählbarkeit und Entscheidbarkeit
  - Unlösbare Probleme (Unentscheidbarkeit)
- **Komplexitätstheorie**
  - Komplexitätsmaße und -klassen für Algorithmen und Probleme
  - Nicht handhabbare Probleme ($\mathcal{NP}$-Vollständigkeit)

Lernziele in diesem Semester

- **Was kann man überhaupt mit Computern lösen?**
  - Sind bestimmte Berechnungsmodelle/-sprachen besser als andere?
  - Gibt es Grenzen dessen, was prinzipiell möglich ist?
- **Was kann man effizient mit Computern lösen?**
  - Wieviel Rechenzeit und Speicherplatz benötigt ein Programm?
  - Sind bestimmte Lösungen/Programme besser als andere?
  - Wie gut kann eine Lösung bestenfalls werden?
  - Skalierbarkeit: wie groß darf das Problem maximal werden?
  - Gibt es relevante Probleme, die nicht effizient lösbar sind?
- **Wie über den Stand der Technik hinausgehen?**
  - Unlösbare Probleme können heuristisch angegangen werden
  - Approximierende und probabilistische Lösungen können effizienter sein

Theoretische Analysen sind billiger als fehlgeschlagene Experimente

Organisatorisches

- **Lehrveranstaltungen**
  - **Vorlesung**: Vorstellung und Illustration zentraler Konzepte
    "unvollständig": die Idee (Verstehen) zählt mehr als das Detail
    · 13:30–16:45 – 17.&24.April, 8.&15.Mai, 5.Juni + 13:30–15:00 12.Juni
  - **Übung**: Vertiefung/Anwendung durch Aufgaben, Klären von Fragen
    · 17:00–18:30 – 24.April, 8.&15.Mai, 5.Juni
  - **Abschlußprüfung**: 12.Juni 15:15–16:45
- **Vorlesungsfolien auf dem Webserver erhältlich**
  - Lehrbücher decken Inhalt i.w. ab (Achtung! Andere Notation)
- **Literaturhinweise**
  - A. Asteroth, C. Baier: Theoretische Informatik, Pearson 2002
  - J. Hopcroft, R. Motwani, J. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson 2002

  Auch lesenswert
  - I. Wegener: Theoretische Informatik, Teubner Verlag 1993
  - U. Schöning: Theoretische Informatik - kurzgefaßt, Spektrum-Verlag 1994
  - K. Erk, L. Priese: Theoretische Informatik, Springer Verlag 2000

Leistungserfassung

- **Eine Klausur entscheidet über die Note**
  - Hauptklausur am 12.Juni
  - Probeklausur wird am 5. Juni zum Selbsttraining ausgegeben
- **Vorbereitung auf die Klausur**
  - Feedback durch Besprechung von Hausaufgaben
  - 5-Minuten Kurzquiz in Übungsstunde
  - Präsentation eigener Lösungen zu Übungsaufgaben
  - Klärung von Fragen in Übung und Sprechstunden