

Theoretische Informatik II

Sommersemester 2006

Christoph Kreitz / Holger Arnold

Theoretische Informatik

<http://www.cs.uni-potsdam.de/ti/lehre/06-Theorie-II>



1. Das Team
2. Lernziele und Lehrinhalte
3. Organisatorisches

DAS TEAM



Christoph Kreitz

Raum 1.18, Telephon 3060

kreitz@cs.uni-potsdam.de



Holger Arnold

Raum 1.21, Telephon 3066

holger@cs.uni-potsdam.de

Tutoren

Marcel Goehring (marcelgoehring@arcor.de)

Ellen König (ekoenig@rz.uni-potsdam.de)

Kristine Jetzke (jetzke@uni-potsdam.de)

Hendrik Radke (hermes9@gmx.net)

Jan Schwarz (miracelwhipp@gmx.de)

Jens Steinborn (steinbor@uni-potsdam.de)

Holger Trölenberg (troelenb@uni-potsdam.de)

RÜCKBLICK: THEMEN DER TI-I

- **Mathematische Methodik in der Informatik**

- Allgemeine Methodik des Problemlösens
- Beweisführung: deduktiv, indirekt, induktiv

- **Automatentheorie und Formale Sprachen**

- **Reguläre Sprachen**

\mathcal{L}_3

- Modelle: Endliche Automaten, Reguläre Ausdrücke, Grammatiken
- Abschlußeigenschaften, Testverfahren (Lexikalische Analyse), Grenzen der Ausdruckskraft

- **Kontextfreie Sprachen**

\mathcal{L}_2

- Modelle: Pushdown Automaten, Kontextfreie Grammatiken
- Abschlußeigenschaften, Testverfahren (Syntaxanalyse), Grenzen

- **Allgemeine Berechenbarkeit**

\mathcal{L}_0

- Turingmaschinen vs. Typ-0 Grammatiken

● Theorie der Berechenbarkeit

- Berechenbarkeitsmodelle:
 - *Wie leistungsstark sind verschiedene Architekturen / Sprachen?*
- Abstrakte Berechenbarkeit, Aufzählbarkeit und Entscheidbarkeit
 - *Welche Eigenschaften müssen für alle Computer gelten?*
- Unlösbare Probleme
 - *Gibt es prinzipielle Grenzen dessen, was Computern können?*

● Komplexitätstheorie

- Komplexitätsmaße und -klassen für Algorithmen und Probleme
 - *Wieviel Rechenzeit und Speicherplatz sind nötig?*
- Nicht handhabbare Probleme (\mathcal{NP} -Vollständigkeit)
 - *Welche Probleme haben keine effiziente Lösung?*
- Effizientere Alternativen zu konventionellen Verfahren
 - *Wie kann man nicht handhabbare Probleme angehen?*

DER LEHRSTOFF

● Reihenfolge und Notation folgt Leittext

- J. Hopcroft, R. Motwani, J. Ullman: *Einführung in die Automaten-
theorie, Formale Sprachen und Komplexitätstheorie*, Pearson 2002
- Einige ergänzende Themen sind nicht im Buch enthalten
- Vorlesungsfolien sind im Voraus auf dem Webserver erhältlich

● Lesenswerte Zusatzliteratur

- G. Vossen, K.-U. Witt: *Grundkurs Theoretische Informatik*. Vieweg 2004
- M. Sipser: *Introduction to the Theory of Computation*. PWS 1997
- A. Asteroth, C. Baier: *Theoretische Informatik*, Pearson 2002
- I. Wegener: *Theoretische Informatik*, Teubner Verlag 1993
- U. Schöning: *Theoretische Informatik - kurzgefaßt*, Spektrum-Verlag 1994
- K. Erk, L. Priese: *Theoretische Informatik*, Springer Verlag 2000
- H. Lewis, C. Papadimitriou: *Elements of the Theory of Computation*, Prentice-Hall 1998

WAS WIRD ANDERS ALS IM ERSTEN SEMESTER?

● **Der Inhalt wird anspruchsvoller**

- Interessante Resultate benötigen mehr Tiefe und höhere Dichte
- Die wirkliche Stärke von Theorie ist die Fähigkeit zur Abstraktion
- Es ist wie im Leistungssport: das Training wird immer intensiver

● **Sie brauchen zusätzliche Quellen**

- Die Vorlesung alleine ist nicht genug
- Details finden Sie in Büchern, im Web ... und durch eigene Versuche
- Die Möglichkeiten sind viel größer als noch vor 10 Jahren

● **Wir erwarten mehr Selbständigkeit**

- Sie lernen nur, wenn Sie selbst aktiv werden
- Sie müssen eigene Methoden ausprobieren
- Sie sollten lernen im Team zusammen zu arbeiten
- Wir sind nur die Trainer: die Schritte müssen Sie selbst gehen

BEDEUTUNG DER LEHR- UND LERNFORMEN

- **Selbststudium** wird immer bedeutender
 - Eigenständige Bearbeitung verschiedener Quellen (Literatur, Web,...)
 - Trainieren durch Lösung von Beispielaufgaben im Team (!)
 - Verständnis ist wichtiger als Faktenwissen
- **Vorlesung** ist nur eine Anleitung Di 11:00–12:30
 - Vorstellung und Illustration zentraler Konzepte und Zusammenhänge
 - Bewußt “unvollständig”: die Idee zählt mehr als das Detail
 - Nutzen Sie das optionale Tutorium zweiwöchentlich Mi 15:15–16:40
- **Übungen** helfen, selbst aktiv zu werden
 - Kurzquiz als Selbsttest und Klärung von Fragen
 - Eigenständiges Lösen von Aufgaben unter Anleitung
 - Bearbeitung von Hausaufgaben gibt Feedback & Korrektur
- **Sprechstunden** werden viel zu wenig genutzt
 - Optimierung des individuellen Lernstils ... auch für bessere Studenten
 - Klärung von Schwierigkeiten ... aber nicht Lösung der Hausaufgaben

LEISTUNGSERFASSUNG HAT NEUE OPTION

- **Eine Klausur entscheidet über die Note**
 - Hauptklausur Anfang Oktober – alle (Papier-)Unterlagen erlaubt
 - Nachprüfung für durchgefallene Studenten – maximal 4.0 erreichbar
 - Zulassung zur Klausur: 50% der Punkte in den Hausaufgaben
 - Gruppenarbeit (bis 4 Studenten) ist dringend zu empfehlen
- **Anerkennung eigenständiger Leistungen möglich**
 - Freiwilliges selbstgewähltes Projekt zu Themen der TI-2
 - Literaturstudie, Programmierung von Methoden, ...
 - Projekt wird mit bis zu 10% auf die Klausurpunkte angerechnet
 - Kriterien: Originalität, Schwierigkeitsgrad, Tiefe, Textqualität, ..
 - Risiko und Chance zugleich – es gibt keine Garantien
 - **Maximal ein Projekt pro Student** (einzeln oder in Kleingruppen)
 - Projektantrag (ca 4-6 Zeilen) muß von uns “genehmigt” werden
 - **Deliverable**: 4-6 seitiges pdf Dokument bei Literaturstudie
 - Programm, Dokumentation+Demo bei Programmierprojekt
 - **Deadline**: 3 Wochen nach Genehmigung (nur elektronische Abgabe)