

**Modulhandbuch**  
für den  
**Bachelor- und Master-Studiengang Informatik**

**Institut für Informatik  
der Mathematisch-Naturwissenschaftliche Fakultät  
der Heinrich-Heine-Universität**

**Herausgegeben vom  
Ausschuss für die Bachelor- und Master-Prüfung  
im Fach Informatik**



## **Vorwort**

Das Modulhandbuch soll eine Orientierung über den grundständigen Bachelor-Studiengang Informatik und den konsekutiven Master-Studiengang Informatik liefern. Insbesondere soll es die Wahl der Lehrveranstaltungen erleichtern und die Organisation des Studiums unterstützen.

Das Modulhandbuch führt die gängigen Lehrveranstaltungen auf. Es ist jedoch keine vollständige, abschließende oder endgültige Auflistung. Vielmehr wird das Modulhandbuch kontinuierlich aktualisiert und gibt so die Entwicklung in Forschung und Lehre am Institut für Informatik wieder.

Beachten Sie jedoch, dass in allen Fragen zum Studium sowie zum Prüfungswesen die jeweiligen Bachelor- oder Master-Prüfungsordnungen des Fachs Informatik maßgeblich sind.

Düsseldorf, im Frühjahr 2018

Der Ausschuss für die Bachelor- und Master-Prüfung im Fach Informatik



# Inhaltsverzeichnis

0.	Studienverlaufspläne .....	- 10 -
1.	Pflichtmodule der Informatik (1.–4. Fachsemester) .....	- 12 -
1.1	Programmierung.....	- 12 -
1.2	Algorithmen und Datenstrukturen.....	- 13 -
1.3	Einführung Rechnernetze, Datenbanken und Betriebssysteme .....	- 14 -
1.4	Rechnerarchitektur.....	- 15 -
1.5	Professionelle Softwareentwicklung (Programmierpraktikum I) .....	- 17 -
1.6	Softwareentwicklung im Team (Programmierpraktikum II).....	- 18 -
1.7	Theoretische Informatik .....	- 19 -
2.	Pflichtmodule der Mathematik (1.–4. Fachsemester).....	- 22 -
2.1	Lineare Algebra I.....	- 22 -
2.2	Analysis I.....	- 23 -
2.3	Analysis II.....	- 23 -
2.4	Angewandte Mathematik.....	- 24 -
3.	Nebenfachmodule .....	- 28 -
4.	Praxis- und Berufsorientierung.....	- 28 -
4.1	Praxis- und Berufsorientierung.....	- 28 -
5.	Lehreinheiten für Wahlpflichtbereiche (B.Sc.) .....	- 30 -
5.1	Algorithmen in der Bioinformatik.....	- 30 -
5.2	Algorithmen zur Entscheidungsfindung unplugged .....	- 31 -
5.3	Algorithmische Eigenschaften von Wahlsystemen I .....	- 33 -
5.4	Algorithmische Geometrie .....	- 34 -
5.5	Algorithmische Komplexitätstheorie .....	- 35 -
5.6	Angewandte Algorithmik.....	- 36 -
5.7	Approximative Algorithmen .....	- 38 -
5.8	Betriebssysteme .....	- 39 -
5.9	Betriebssysteme und Systemprogrammierung.....	- 40 -
5.10	Compilerbau .....	- 41 -
5.11	Datenbanksysteme .....	- 42 -
5.12	Einführung in die logische Programmierung.....	- 43 -
5.13	Graphenalgorithmen I .....	- 44 -
5.14	Grundlagen Verteilter Systeme .....	- 46 -
5.15	Knapsack Problems.....	- 47 -
5.16	Kryptokomplexität I.....	- 48 -

5.17	Machine Learning .....	- 49 -
5.18	Matching.....	- 50 -
5.19	Networks, Crowds, and Markets .....	- 52 -
5.20	Nichtkooperative Spieltheorie.....	- 53 -
5.21	Präferenzaggregation durch Wählen: Algorithmik und Komplexität .....	- 54 -
5.22	Rechnernetze.....	- 56 -
5.23	Statistische Datenanalyse.....	- 58 -
5.24	Überblick Künstliche Intelligenz .....	- 59 -
5.25	Verfahren zur kollektiven Entscheidungsfindung.....	- 60 -
5.26	Von NAND zu Tetris .....	- 61 -
6.	Bachelor-Arbeit.....	- 63 -
6.1	Bachelor-Arbeit.....	- 63 -
7.	Lehreinheiten für Wahlpflichtbereiche (M.Sc.).....	- 65 -
7.1	Algebraische und Strukturelle Graphentheorie .....	- 65 -
7.2	Algorithmen für Ad-hoc- und Sensornetzwerke.....	- 66 -
7.3	Algorithmen für planare Graphen .....	- 68 -
7.4	Algorithmen für schwere Probleme .....	- 69 -
7.5	Algorithmen in der Cheminformatik .....	- 70 -
7.6	Algorithmische Eigenschaften von Wahlsystemen II .....	- 71 -
7.7	Algorithmische Spieltheorie .....	- 72 -
7.8	Android-Programmierung .....	- 74 -
7.9	Angewandte Bioinformatik.....	- 75 -
7.10	Approximationsalgorithmen.....	- 77 -
7.11	Architektur Verteilter Systeme.....	- 78 -
7.12	Betriebssystem-Entwicklung .....	- 79 -
7.13	Betriebssysteme Vertiefung .....	- 80 -
7.14	Big Data Techniques, Technologies and Trends .....	- 81 -
7.15	Cake-cutting Algorithms .....	- 82 -
7.16	Causality .....	- 84 -
7.17	Computational Systems Biology .....	- 85 -
7.18	Computer Vision .....	- 86 -
7.19	Data Warehouses .....	- 88 -
7.20	Dynamische Programmiersprachen .....	- 89 -
7.21	Effiziente Algorithmen.....	- 90 -
7.22	Einführung in die Geoinformatik .....	- 91 -

7.23	Einführung in die statistische Analyse mittels Computersimulationen .....	- 92 -
7.24	Entwurfs- und Analysetechniken für Algorithmen .....	- 93 -
7.25	Funktionale Programmierung.....	- 95 -
7.26	Fuzzy-Clusteranalyse .....	- 96 -
7.27	Fuzzy Systeme.....	- 97 -
7.28	Gerechte Aufteilungsverfahren .....	- 98 -
7.29	Graphenalgorithmen II .....	- 99 -
7.30	Grundlagen der Linearen Optimierung .....	- 101 -
7.31	Information Retrieval und Natural Language Processing.....	- 102 -
7.32	iOS-Programmierung .....	- 103 -
7.33	Judgment Aggregation.....	- 104 -
7.34	Knowledge Discovery in Databases .....	- 106 -
7.35	Knowledge Discovery in Databases - Ausgewählte Themen .....	- 107 -
7.36	Kooperative Spieltheorie .....	- 108 -
7.37	Kryptokomplexität II .....	- 110 -
7.38	Machine Learning – Deep Learning .....	- 111 -
7.39	Master-Seminar über Algorithmen für perfekte Graphen .....	- 112 -
7.40	Master-Seminar über Graphparameter für gerichtete Graphen .....	- 114 -
7.41	Master-Seminar über Kollektive Entscheidungsfindung .....	- 115 -
7.42	Master-Seminar über Modellierung biologischer Zellen.....	- 116 -
7.43	Master-Seminar zu aktuellen Themen der theoretischen und computergestützten Biologie .....	- 118 -
7.44	Master-Seminar zu Informatik unplugged .....	- 118 -
7.45	Master-Seminar zu Kombinatorische Optimierung in der Bioinformatik .....	- 120 -
7.46	Master-Seminar zu Machine Learning.....	- 121 -
7.47	Master-Seminar zu Reproduzierbarkeit bioinformatischer Forschungsergebnisse .....	- 122 -
7.48	Master-Seminar zu Verteilte und Parallele Systeme.....	- 123 -
7.49	Mathematische Hintergründe der Wahlmanipulation.....	- 124 -
7.50	Mobilkommunikation .....	- 126 -
7.51	Model Checking .....	- 127 -
7.52	Modellierung metabolischer Netzwerke.....	- 128 -
7.53	Multimedia-Datenbanksysteme .....	- 129 -
7.54	Netzwerksicherheit.....	- 131 -
7.55	Opportunistische und P2P-basierte Netzwerke .....	- 132 -
7.56	Paralleles Rechnen mit Grafikkarten .....	- 133 -
7.57	Peer-to-Peer Systeme.....	- 134 -

7.58	Programmsynthese und Transformation .....	- 136 -
7.59	Sicherheitskritische Systeme .....	- 137 -
7.60	Transaktionsverwaltung .....	- 138 -
7.61	Verteilte Systeme .....	- 139 -
7.62	Verteilte und Parallele Programmierung.....	- 141 -
7.63	Vertiefung Logische Programmierung.....	- 142 -
7.64	Vertiefung Rechnernetze.....	- 143 -
7.65	Zeichnen von Graphen .....	- 144 -
8.	Projektarbeit.....	- 146 -
8.1	Projektarbeit.....	- 146 -
9.	Master-Arbeit .....	- 148 -
9.1	Master-Arbeit .....	- 148 -





# 0. Studienverlaufspläne

## [Studienablaufplan](#)

**Bachelor-Studiengang (Beginn im Wintersemester):** siehe Webseite: [Musterstudienplan Wintersemester](#)

Semester	Lehrveranstaltungen mit zugehörigen Leistungspunkten				
1	Programmierung (10)	Algorithmen und Datenstrukturen (10)	Lineare Algebra 1 (10)		30 LP
2	Rechnerarchitektur (9) V+Ü (5) BV+PÜ (4)	Professionelle Softwareentwicklung (Programmierpraktikum I) (8)	Einf. Rechnernetze, Datenbanken und Betriebssysteme (5)	Analysis 1 (10)	32 LP
3	Softwareentwicklung im Team (Programmierpraktikum II) (8)	Analysis 2 (10)	Nebenfach (10)		28 LP
4	Theoretische Informatik (10)	Angewandte Mathematik (10): Numerik oder Stochastik <b>oder</b> ↓ Nebenfach (10)	Wahlbereich (10)		30 LP
5	Wahlbereich (20)	Angewandte Mathematik (10): Stochastik oder Numerik <b>oder</b> ↓ Nebenfach (10)			30 LP
6	Bachelorarbeit (15)	Praxis- und Berufsorientierung (5LP)	Nebenfach (10)		30 LP
Gesamtleistungspunkte: 180 LP					

**Bachelor-Studiengang (Beginn im Sommersemester):** siehe Webseite: [Musterstudienplan Sommersemester](#)

Semester	Lehrveranstaltungen mit zugehörigen Leistungspunkten				
1	Rechnerarchitektur (9) V+Ü (5)	Einf. Rechnernetze, Datenbanken und Betriebssysteme (5)	Lineare Algebra 1 (10)	Analysis 1 (10)	30 LP
2	Programmierung (10)	Algorithmen und Datenstrukturen (10)	Analysis 2 (10)		30 LP
3	Professionelle Softwareentwicklung (Programmierpraktikum I) (8)	Angewandte Mathematik (10): Numerik oder Stochastik <b>oder</b> ↓ Nebenfach (10)	Rechnerarchitektur (9) BV+PÜ (4)	Theoretische Informatik (10)	32 LP
4	Softwareentwicklung im Team (Programmierpraktikum II) (8)	Angewandte Mathematik (10): Stochastik oder Numerik <b>oder</b> ↓ Nebenfach (10)	Wahlbereich (10)		28 LP
5	Wahlbereich (20)		Nebenfach (10)		30 LP
6	Bachelorarbeit (15)	Praxis- und Berufsorientierung (5LP)	Nebenfach (10)		30 LP
Gesamtleistungspunkte: 180 LP					

## Master-Studiengang

### [Studienablaufplan](#)

Semester	Lehrveranstaltungen mit zugehörigen Leistungspunkten		
7	Praktische oder Technische Informatik (15)	Schwerpunkt (30)	Projektarbeit über zwei Semester (20)
8	und		
9	Theoretische Informatik (15)	Individuelle Ergänzung (10)	
10	Masterarbeit im Schwerpunkt (30)		
	Gesamtleistungspunkte: 120 LP		

# Kapitel 1

## 1. Pflichtmodule der Informatik (1.–4. Fachsemester) Compulsory Modules in Computer Science

### 1.1 Programmierung Programming

#### Studiengang

Bachelor-Studiengang Informatik

#### Leistungspunkte

10 LP

#### Lehrveranstaltungen

- Vorlesung „Programmierung“, 4 SWS
- Übung, 2 SWS
- Praktische Übung, 2 SWS

#### Inhalte

Dieses Modul vermittelt grundlegende Programmierkenntnisse in einer objektorientierten Programmiersprache. Darüber hinaus werden einführend Aspekte von Algorithmen und Datenstrukturen behandelt. Es wird keine Programmiererfahrung vorausgesetzt.

- Grundlegende Begriffe der Informatik
- Primitive Datentypen und Variablen
- Kontrollstrukturen
- Eigene Datentypen (Klassen) und Arrays
- Programmstrukturen im Speicher (Heap, Stack)
- Konzepte der Objektorientierung (Polymorphie, Schnittstellen)
- Rekursion
- Fehlerbehandlung
- Dynamische Datenstrukturen (Listen, Binärbäume, Hashing)
- Suchen und Sortieren (ausgewählte Algorithmen, u.a. binäre Suche, BubbleSort, QuickSort)
- Datenströme (Standard-Eingabe und -Ausgabe, einfache 2D-Grafik, Dateien)

#### Lernergebnisse/Kompetenzen

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Begriffe der Informatik und der Programmierung zu nennen und zu erläutern
- einfache Algorithmen (iterativ und rekursiv) zu verstehen, deren Ablauf zu beschreiben, sowie selbst zu erstellen
- eigene Datentypen zu konzipieren und anzuwenden
- einfache objektorientierte Programme mit Polymorphie, Vererbung und Schnittstellen zu entwickeln
- die behandelten dynamischen Datenstrukturen anzuwenden

#### Literatur

- R. Schiedermeier, „Programmieren mit Java“, Pearson Studium, 2010.
- C. Ullenboom, „Java ist auch eine Insel“, 11. Aufl., 2014
- R. Sedgewick & K. Wayne, „Introduction to Programming in Java“, Addison-Wesley, 2007.

#### Verwendbarkeit des Moduls

- Pflichtbereich
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

- Modul D1 im Studienbereich Informatik (D) im Integrativen Bachelor-Studiengang Sprachtechnologie und Informationswissenschaft

### **Teilnahmevoraussetzungen**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- schriftliche Prüfung (Klausur, i.d.R. 90 Minuten)

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Harmeling, Prof. Dr. Michael Schöttner

## **1.2 Algorithmen und Datenstrukturen Algorithms and Data Structures**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

10 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmen und Datenstrukturen“, 4 SWS
- Übung, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 4 SWS

### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen.

- Algorithmen und ihre formalen Grundlagen
- Rechenmodelle, Effizienzmaße
- Sortierverfahren (Quicksort, Heapsort, Mergesort, ...)
- Aufwandsabschätzung im Mittel
- Suchstrategien (Binärsuche, Interpolationsuche, Textsuche, ...)
- Dictionaries (offene Hashverfahren, dynamische Hashverfahren)
- Suchbäume (AVL-Bäume, B-Bäume, Splay-Trees, ...)
- Vorrangwarteschlangen (Heaps, Binominal Queues, Fibonacci-Heaps, ...)
- Amortisierte Laufzeitanalysen
- Einführung in Graphenalgorithmen (Tiefensuche, Breitensuche, Zusammenhangsprobleme, ...)

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein Basisverständnis der wichtigsten Grundlagen über Algorithmen. Sie haben die Fähigkeit zur Problemspezifikation und algorithmischen Problembearbeitung erworben.

### **Empfohlene Literatur**

- Thomas Ottmann, Peter Widmayer: Algorithmen und Datenstrukturen, Spektrum Akademischer Verlag, 5. Auflage, 2012
- Richard Johnsonbaugh, Marcus Schäfer: Algorithms, Pearson Education, 2004
- Jon Kleinberg, Eva Tardos: Algorithm Design, Addison Wesley, 2006

### **Verwendbarkeit des Moduls**

- Pflichtbereich
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 45 Minuten) am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski, Prof. Dr. Gunnar W. Klau, Prof. Dr. Martin Lercher, Prof. Dr. Egon Wanke

## **1.3 Einführung Rechnernetze, Datenbanken und Betriebssysteme Introduction to computer networks, databases and operating systems**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Vorlesung „Einführung Rechnernetze, Datenbanken und Betriebssysteme“, 2 SWS
- Übung, 1 SWS
- Selbststudium, 100h

### **Inhalte**

Die Vorlesung gibt einen Überblick über verschiedene Themen der praktischen Informatik. In der Übung werden diese Inhalte eingeübt.

- Aufbau und grundlegende Funktionsweise des Internets
- Socketprogrammierung
- Speicherverwaltung (virtueller Speicher, Heap, Stack)
- Grundlegende Funktionsweise von Dateisystemen
- Nebenläufigkeit (Threads, Scheduling, Synchronisierung)
- Entity-Relationship-Modellierung
- Relationenmodell
- Grundlagen von SQL

### **Lernergebnisse/Kompetenzen**

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- die wichtigsten Komponenten des Internets benennen,
- sowie deren Beziehungen untereinander und Ihre grundsätzliche Funktion beschreiben,
- einfache Programme mit Netzwerknutzung schreiben,

- grundlegende Funktionen eines Betriebssystems benennen und erläutern,
- einfache nebenläufige Programme schreiben,
- die Aufgaben eines Datenbanksystems benennen,
- Datenbanken in Form von ER Modellen darstellen und
- einfache Datenbankabfragen in SQL schreiben.

### **Empfohlene Literatur**

- J. F. Kurose, K. W. Ross: Computer Networking – A Top-Down Approach Featuring the Internet; 6th Edition; Pearson, 2012
- A. Heuer, G. Saake, K.-U. Sattler: Datenbanken – Konzepte und Sprachen, 5. Aufl., mitp-Verlag, Bonn, 2013
- A. Tanenbaum: Modern Operating Systems, 4. Aufl., Prentice Hall, 2014.
- W. Stallings, Operating Systems: Internals and Design Principles, Prentice Hall, 8. Aufl., 2014.

### **Verwendbarkeit des Moduls**

- Pflichtbereich
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive und erfolgreiche Teilnahme an den Übungen
- schriftliche Prüfung (i.d.R. 60 Minuten) am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad, Prof. Dr. Martin Mauve, Prof. Dr. Michael Schöttner

## **1.4 Rechnerarchitektur Computer Architecture**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

9 LP ab PO 2016, alte PO's 10 LP

### **Lehrveranstaltungen**

- Vorlesung „Rechnerarchitektur“, 2 SWS
- Übung, 1 SWS
- Vorlesung „Hardwarenahe Programmierung“, 15h (im Block)
- Praktische Übung „Hardwarenahe Programmierung“, 30h (im Block)
- Selbststudium, 180h

### **Inhalte**

Die Vorlesung „Rechnerarchitektur“ sowie die dazugehörige Übung behandelt den Aufbau eines Rechners. Dabei wird insbesondere auf folgende Themengebiete eingegangen:

- Datendarstellung

- einfache Fehlererkennende und -korrigierende Codes
- Konzepte zur effizienten Datenverarbeitung (Pipelines, Caches)
- digitale Logik
- digitale Schaltungen
- Mikroprogrammierung

Die Vorlesung „Hardwarenahe Programmierung“ und die praktische Übung vermitteln Kenntnisse in der Assemblerprogrammierung sowie einer systemnahen imperativen Programmiersprache. Es wird vorausgesetzt, dass die Teilnehmer übliche Programmelemente, wie Variablen, Verzweigungen, Schleifen und Funktionen sicher verwenden können.

- Grundlagen der Assembler Programmierung
- Programmierung in einer systemnahen imperativen Programmiersprache
- dynamische Speicherverwaltung inkl. Identifizierung von Speicherlecks
- ein zur Programmiersprache passendes Build-Tool sowie eine geeignete Testumgebung

### **Lernergebnisse/Kompetenzen**

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- wiedergeben wie ein moderner Computer aufgebaut ist,
- die verschiedenen Schichten einer Rechnerarchitektur beschreiben und dabei auf ihre Verbindung untereinander eingehen,
- erklären, wie eine CPU/ALU aus elementaren digitalen Schaltungen konstruiert wird,
- zentrale Funktionen eines Betriebssystems identifizieren und ihre Arbeitsweise an einfachen Beispielen darstellen,
- einfache digitale Schaltungen entwerfen und optimieren,
- einfache Assemblerprogramme in x86 Assembler entwickeln,
- Programme in der Programmiersprache C unter Berücksichtigung dynamischer Speicherverwaltung entwickeln und
- Werkzeuge für typische Aufgaben bei der Programmierung (Speicherverwaltung, Build-Prozesse, Tests) verwenden.

### **Empfohlene Literatur**

Die primären Lehrbücher zu den verschiedenen Bestandteilen dieses Moduls sind:

- Andrew S. Tanenbaum and Todd Austin: Structured Computer Organization; 6th Edition; Pearson; 2013. Prentice Hall; 5th Edition; 2006
- David Griffiths and Dawn Griffiths (dt. Lars Schulten): C von Kopf bis Fuß; O'Reilly Verlag; 2012
- Paul A. Carter: PC Assembly Language; Online; 2003

### **Verwendbarkeit des Moduls**

- Pflichtbereich
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive und erfolgreiche Teilnahme an den Übungen zur Vorlesung „Rechnerarchitektur“
- erfolgreiche Bearbeitung sämtlicher Praktikumsaufgaben
- schriftliche Prüfung (Klausur, i.d.R. 60 Minuten)

### **Häufigkeit des Angebots, modulare Schiene**

- Die Vorlesung „Rechnerarchitektur“ mit zugehöriger Übung wird jedes Sommersemester angeboten.
- Die Vorlesung „Assemblerprogrammierung“ und die praktische Übung werden in der vorlesungsfreien Zeit im Anschluss an das Sommersemester angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**



## **1.5 Professionelle Softwareentwicklung (Programmierpraktikum I) Professional Software Development (Programming Project I)**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

8 LP ab PO 2016, alte PO's 10 LP

### **Lehrveranstaltungen**

- Vorlesung „Professionelle Softwareentwicklung“, 2 SWS
- Übung, 2 SWS
- Selbststudium, 180h

### **Inhalte**

- Prinzipien objektorientierter Software Entwicklung
- Prozesse in der professionellen Softwareentwicklung
- Grundlegende Softwarearchitekturen
- Werkzeuge der Softwareentwicklung

### **Lernergebnisse / Kompetenzen**

Studierende sollen nach erfolgreichem Abschluss des Moduls

- grundlegende Architekturen beschreiben können
- selbständig Problemstellungen analysieren können
- aus einfachen Problemstellungen ein objektorientiertes System zur Lösung entwerfen können
- eine Lösung hinsichtlich der Wartbarkeit analysieren und verbessern können
- mit den gängigen Werkzeugen (z.B. IDE) umgehen können

### **Literatur**

- Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008
- Robert C. Martin, The Clean Coder: A Code of Conduct for Professional Programmers, Prentice Hall, 2011

### **Verwendbarkeit des Moduls**

- Professionelle Softwareentwicklung
- Modul D2 im Studienbereich Informatik (D) im Integrativen Bachelor-Studiengang Sprachtechnologie und Informationswissenschaft

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss des Moduls:

- „Programmierung“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung in den Übungen (Projekten)
- Abschlusstest (unbenotet) zum Ende der Vorlesungszeit

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester.

### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Jens Bendisposto

## **1.6 Softwareentwicklung im Team (Programmierpraktikum II)**

### **Teamwork in Software Development (Programming Project II)**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

8 LP

#### **Lehrveranstaltungen**

- Vorlesung „Softwareentwicklung im Team“, 2 SWS
- Übung, 2 SWS
- Praktische Übung, 120h (Blockveranstaltung)
- Selbststudium, 60h

#### **Inhalte**

- Projektbezogene Gruppenarbeit
- Entwurf und Gliederung eines umfangreicheren Programms
- Dokumentation
- Diskussion über Softwareaufbau und Design im Team
- Werkzeuge zur Teamkoordination

#### **Lernergebnisse / Kompetenzen**

Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,

- grundlegende Entwicklungsprozesse beschreiben zu können,
- eine gestellte komplexere Aufgabe verstehen und in Teilaufgaben strukturieren zu können,
- die Teilaufgaben in einem Team entsprechend eines Entwicklungsprozesses implementieren zu können,
- die einzelnen Bestandteile sowie das Gesamtprogramms mit geeigneten Verfahren testen zu können,
- mit den Werkzeugen zur Softwareentwicklung im Team (Versionskontrolle, CI Systeme, Bugtracker) sicher umgehen zu können.

#### **Literatur**

- Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008
- Robert C. Martin, The Clean Coder: A Code of Conduct for Professional Programmers, Prentice Hall, 2011

#### **Verwendbarkeit des Moduls**

- Softwareentwicklung im Team

#### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss des Moduls:

- „Professionelle Softwareentwicklung (Programmierpraktikum I)“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung an dem Gruppenprojekt
- Abschlusstest (unbenotet) zum Ende der Vorlesungszeit
- Vortrag/Vorstellung der Ergebnisse im Plenum

#### **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester.

#### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Jens Bendisposto

## 1.7 Theoretische Informatik

### Theoretical Computer Science

#### Studiengang

Bachelor-Studiengang Informatik

#### Leistungspunkte

10 LP

#### Lehrveranstaltungen

- Vorlesung „Theoretische Informatik“, 4 SWS
- Übung, 2 SWS

#### Inhalte

- Formale Sprachen und Automaten
  - Grundbegriffe
    - Wörter, Sprachen und Grammatiken
    - Die Chomsky-Hierarchie
  - Reguläre Sprachen
    - Endliche Automaten
    - Reguläre Ausdrücke
    - Gleichungssysteme
    - Das Pumping-Lemma
    - Satz von Myhill und Nerode und Minimalautomaten
    - Abschlusseigenschaften regulärer Sprachen
    - Charakterisierungen regulärer Sprachen
  - Kontextfreie Sprachen
    - Normalformen
    - Das Pumping-Lemma
    - Der Satz von Parikh
    - Abschlusseigenschaften kontextfreier Sprachen
    - Der Algorithmus von Cocke, Younger und Kasami
    - Kellerautomaten
  - Deterministisch kontextfreie Sprachen
    - Deterministische Kellerautomaten
    - LR(k)- und LL(k)-Grammatiken
    - Anwendung: Syntaxanalyse durch LL(k)-Parser
  - Kontextsensitive und L0-Sprachen
    - Turingmaschinen
    - Linear beschränkte Automaten
    - Zusammenfassung
- Berechenbarkeit
  - Intuitiver Berechenbarkeitsbegriff und die These von Church
  - Turing-Berechenbarkeit
  - LOOP-, WHILE- und GOTO-Berechenbarkeit
    - LOOP-Berechenbarkeit
    - WHILE-Berechenbarkeit
    - GOTO-Berechenbarkeit
  - Primitiv rekursive und partiell rekursive Funktionen
    - Primitiv rekursive Funktionen
    - Die Ackermann-Funktion
    - Allgemein und partiell rekursive Funktionen
    - Der Hauptsatz der Berechenbarkeitstheorie
  - Entscheidbarkeit und Aufzählbarkeit
    - Einige grundlegende Sätze
    - Entscheidbarkeit
    - Rekursiv aufzählbare Mengen

- Unentscheidbarkeit
  - o Der Satz von Rice
  - o Reduzierbarkeit
  - o Das Postsche Korrespondenzproblem
  - o Unentscheidbarkeit in der Chomsky-Hierarchie
  - o Zusammenfassung

### **Lernergebnisse/Kompetenzen**

Ziel dieser Veranstaltung ist die Vermittlung von Grundlagenwissen aus den Bereichen Formale Sprachen und Automaten sowie Berechenbarkeitstheorie. Am Ende der Veranstaltung sollten Studierende in der Lage sein, formale Sprachen in die Chomsky-Hierarchie einzuordnen, verschiedene äquivalente Automatenmodelle ineinander bzw. in Grammatiken des entsprechenden Typs umzuformen, Argumente für die In-Äquivalenz von bestimmten Automatenmodellen bzw. Grammatiktypen zu geben, die algorithmische Entscheidbarkeit von Problemen einzuschätzen und Argumente für die Nichtentscheidbarkeit von Problemen zu geben. Auch sollten sie die Erkenntnis gewonnen haben, dass es nicht berechenbare Funktionen gibt, und eine Vorstellung vom Aufbau eines Compilers und von lexikalischer und Syntaxanalyse erworben haben. Neben diesen Kenntnissen sollten sie sich auch Fertigkeiten im Umgang mit formalen Begriffs- und Modellbildungen sowie mit formalen Argumentationsweisen sowie bestimmte Beweistechniken (wie etwa Diagonalisierung) angeeignet haben.

### **Empfohlene Literatur**

- Uwe Schöning: Theoretische Informatik - kurz gefasst, Spektrum Akademischer Verlag, 2. Auflage, 1995.
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium, 2. Auflage, 2002.
- Klaus W. Wagner: Theoretische Informatik. Eine kompakte Einführung, Springer-Verlag, 2. Auflage, Berlin, Heidelberg, 2003.

### **Ergänzende Literatur**

- Norbert Blum: Theoretische Informatik. Eine anwendungsorientierte Einführung, Oldenbourg, 2001.
- Alexander Asteroth, Christel Baier: Theoretische Informatik. Eine Einführung in Berechenbarkeit, Komplexität und formale Sprachen mit 101 Beispielen, Pearson Studium, 2002.

### **Verwendbarkeit des Moduls**

- Pflichtbereich
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur)

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel, Prof. Dr. Jörg Rothe, Jun.-Prof. Dr. Dorothea Baumeister



## Kapitel 2

### 2. Pflichtmodule der Mathematik (1.–4. Fachsemester) Compulsory Modules in Mathematics

#### 2.1 Lineare Algebra I Linear Algebra I

##### Studiengang

Bachelor-Studiengang Informatik

##### Leistungspunkte

10 LP

##### Lehrveranstaltungen

- Vorlesung „Lineare Algebra I“, 4 SWS
- Übung, 2 SWS

##### Inhalte

lineare Gleichungssysteme, Mengen, Gruppen, Ringe, Körper, komplexe Zahlen, endliche Primkörper, Vektorräume, Basen, Summenvektorräume, Äquivalenzrelationen, Quotientenvektorräume, lineare Abbildungen, Matrizen, Kern und Bild, Isomorphiesatz, Rang, Gauß-Algorithmus, Endomorphismen, Determinante, Laplace-Entwicklung, Eigenwerte und Eigenvektoren, charakteristisches Polynom, Diagonalisierbarkeit, Skalarprodukte, Länge und Winkel, Gram-Schmidt-Verfahren, orthogonale und unitäre Endomorphismen

##### Lernergebnisse/Kompetenzen

Die Studierenden bewältigen die Begriffsbildungen und grundlegenden Resultate der linearen Algebra. Sie argumentieren anhand der Definitionen und Sätze und können intuitive Vorstellungen mathematisch präzisieren. Sie sind in der Lage, Übungsaufgaben selbstständig zu lösen und diese Lösungen in den Übungsgruppen zu präsentieren sowie kritisch zu diskutieren. Sie verfügen über Methoden der systematischen und effizienten Wissensaneignung.

##### Literatur

- S. Bosch: Lineare Algebra.
- G. Fischer: Lineare Algebra.

##### Verwendbarkeit des Moduls

- Bachelor-Studiengang Informatik
- Bachelor-Studiengang Mathematik
- Bachelor-Studiengang Physik

##### Teilnahmevoraussetzungen

Voraussetzungen nach Prüfungsordnungen: Keine; Empfohlene Voraussetzung: Keine

##### Voraussetzungen für die Vergabe von Leistungspunkten

- Erfolgreiche Teilnahme an der Übungsgruppen
- Schriftliche Prüfung (Anmeldung in der Vorlesung)
- Die Klausur muss bestanden werden; das Modul wird nicht benotet.

##### Häufigkeit des Angebots, modulare Schiene

Jedes Semester

##### Modulbeauftragte und hauptamtliche Lehrende

Die Dozenten der Mathematik (Festlegung jeweils 1 Jahr im Voraus in einer Sitzung der Wissenschaftlichen Einrichtung)

## **2.2 Analysis I**

### **Calculus in One Variable**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

10 LP

#### **Lehrveranstaltungen**

- Vorlesung „Analysis I“, 4 SWS
- Übung, 2 SWS

#### **Inhalte**

Reelle und komplexe Zahlen, Folgen, Konvergenz, Cauchy-Folgen, Grenzwerte, Reihen, Stetigkeit, Kompaktheit, spezielle Funktionen, Differentialrechnung, Integralrechnung, Funktionenfolgen, Potenzreihen, Taylor-Entwicklung

#### **Lernergebnisse/Kompetenzen**

Die Studierenden bewältigen die Begriffsbildungen und grundlegenden Resultate der Analysis einer Veränderlichen. Sie argumentieren anhand der Definitionen und Sätze und können intuitive Vorstellungen mathematisch präzisieren. Sie sind in der Lage, Übungsaufgaben selbstständig zu lösen und diese Lösungen in den Übungsgruppen zu präsentieren sowie kritisch zu diskutieren. Sie verfügen über Methoden der systematischen und effizienten Wissensaneignung.

#### **Literatur**

- H. Amann, J. Escher: Analysis I.
- R. Denk, R. Racke: Kompendium der Analysis. Band 1.

#### **Verwendbarkeit des Moduls**

- Bachelor-Studiengang Informatik
- Bachelor-Studiengang Mathematik
- Bachelor-Studiengang Physik

#### **Teilnahmevoraussetzungen**

Voraussetzungen nach Prüfungsordnungen: Keine; Empfohlene Voraussetzung: Keine

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme an den Übungsgruppen
- Schriftliche Prüfung (Anmeldung in der Vorlesung)
- Die Klausur muss bestanden werden; das Modul wird nicht benotet.

#### **Häufigkeit des Angebots, modulare Schiene**

Jedes Semester

#### **Modulbeauftragte und hauptamtliche Lehrende**

Die Dozenten der Mathematik (Festlegung jeweils 1 Jahr im Voraus in einer Sitzung der Wissenschaftlichen Einrichtung)

## **2.3 Analysis II**

### **Calculus of Several Variables**

## **Studiengang**

Bachelor-Studiengang Informatik

## **Leistungspunkte**

10 LP

## **Lehrveranstaltungen**

- Vorlesung „Analysis II“, 4 SWS
- Übung, 2 SWS

## **Inhalte**

Differentialrechnung mehrerer Veränderlicher, Mittelwertsätze und Taylor-Formel in mehreren Veränderlichen, Satz über implizite Funktionen, Extremwerte mit und ohne Nebenbedingungen, normierte und metrische Räume, Banachscher Fixpunktsatz, gewöhnliche Differentialgleichungen, Existenz- und Eindeutigkeitssätze, spezielle Lösungsmethoden, lineare Differentialgleichungen, Systeme mit konstanten Koeffizienten, Stabilität

## **Lernergebnisse/Kompetenzen**

Die Studierenden bewältigen die Begriffsbildungen und grundlegenden Resultate der mehrdimensionalen Analysis sowie der gewöhnlichen Differentialgleichungen. Sie argumentieren anhand der Definitionen und Sätze und können intuitive Vorstellungen mathematisch präzisieren. Sie sind in der Lage, Übungsaufgaben selbstständig zu lösen und diese Lösungen in den Übungsgruppen zu präsentieren sowie kritisch zu diskutieren. Sie verfügen über Methoden der systematischen und effizienten Wissensaneignung.

## **Literatur**

- H. Amann, J. Escher: Analysis II.
- R. Denk, R. Racke: Kompendium der Analysis. Band 1.

## **Verwendbarkeit des Moduls**

- Bachelor-Studiengang Informatik
- Bachelor-Studiengang Mathematik
- Bachelor-Studiengang Physik

## **Teilnahmevoraussetzungen**

Voraussetzungen nach Prüfungsordnungen: Keine; Empfohlene Voraussetzung: Analysis I

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme an den Übungsgruppen
- Schriftliche Prüfung (Anmeldung in der Vorlesung)
- Die Klausuren zu beiden Lehrveranstaltungen müssen bestanden werden; das Modul wird nicht benotet.

## **Häufigkeit des Angebots, modulare Schiene**

Jedes Semester

## **Modulbeauftragte und hauptamtliche Lehrende**

Die Dozenten der Mathematik (Festlegung jeweils 1 Jahr im Voraus in einer Sitzung der Wissenschaftlichen Einrichtung)

## **2.4 Angewandte Mathematik Applied Mathematics**

### **Studiengang**



## Leistungspunkte

10 LP

## Lehrveranstaltungen

Zur Wahl:

- Vorlesung „Numerik I (Numerical Analysis I)“, 4 SWS
- Übung, 2 SWS
- Programmierübung, 1 SWS

oder

- Vorlesung „Modellbildung in der Stochastik (Modelling in Stochastics)“, 4 SWS
- Übung, 2 SWS

## Inhalte

Numerik I:

Interpolation und Approximation, Quadraturverfahren, direkte Verfahren zur Lösung linearer Gleichungssysteme, Iterative Verfahren zur Lösung nichtlinearer Gleichungssysteme, Fehleranalyse

Modellbildung in der Stochastik:

Grundlagen der Wahrscheinlichkeitsrechnung, Modelle für Zufallsexperimente, Anwendungsbeispiele in der Informatik und den Naturwissenschaften, Unabhängigkeit von Zufallsvariablen, erzeugende Funktionen, Gesetz der großen Zahlen, zentraler Grenzwertsatz, Maximum-Likelihood-Schätzer, Signifikanztests, lineare Regression

## Lernergebnisse/Kompetenzen

Numerik I:

Die Studierenden bewältigen die Begriffsbildungen und grundlegenden Resultate der numerischen Mathematik. Sie argumentieren anhand der Definitionen und Sätze und können intuitive Vorstellungen mathematisch präzisieren. Sie sind in der Lage, Übungsaufgaben selbstständig zu lösen und diese Lösungen in den Übungsgruppen zu präsentieren sowie kritisch zu diskutieren. Sie können abstrakte Algorithmen zu einem konkreten Programm umsetzen. Sie verfügen über Methoden der systematischen und effizienten Wissensa-neignung.

Modellbildung in der Stochastik:

Die Studierenden bewältigen die Begriffsbildungen und grundlegenden Resultate der Stochastik. Sie argumentieren anhand der Definitionen und Sätze und können intuitive Vorstellungen mathematisch präzisieren. Sie sind in der Lage, Übungsaufgaben selbstständig zu lösen und diese Lösungen in den Übungsgruppen zu präsentieren sowie kritisch zu diskutieren. Sie verfügen über Methoden der systematischen und effizienten Wissensaneignung.

## Literatur

Numerik I:

- P. Deuffhard, A. Hohmann: Numerische Mathematik 1.
- R. Freund, R. Hoppe: Stoer/Bulirsch: Numerische Mathematik 1.

Stochastik:

- N. Henze: Stochastik für Einsteiger.
- U. Krengel: Einführung in die Wahrscheinlichkeitstheorie und Statistik.

## Verwendbarkeit des Moduls

- Bachelor-Studiengang Informatik
- Bachelor-Studiengang Mathematik

## Teilnahmevoraussetzungen

Numerik:

Voraussetzungen nach Prüfungsordnungen: Keine; Empfohlene Voraussetzung: Analysis I-II, Lineare Algebra I, Computergestützte Mathematik zur linearen Algebra

Stochastik:

Voraussetzungen nach Prüfungsordnungen: Keine; Empfohlene Voraussetzung: Analysis I, Lineare Algebra I

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme an den Übungsgruppen
- Schriftliche Prüfung (Anmeldung in der Vorlesung)
- Die Klausur muss bestanden werden; das Modul wird nicht benotet.

#### **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester

#### **Modulbeauftragte und hauptamtliche Lehrende**

Die Dozenten der Mathematik (Festlegung jeweils 1 Jahr im Voraus in einer Sitzung der Wissenschaftlichen Einrichtung)



## Kapitel 3

### 3. Nebenfachmodule Minor module

Die Module im Bereich „Nebenfach“ sind abhängig vom gewählten Nebenfach (siehe Webseiten der Informatik: [Nebenfach](#)). Folgende Fächer stehen zur Auswahl: Biologie, Physik, Chemie, Mathematik und Psychologie (begrenzt auf 5 Plätze pro Studienjahr; Beginn immer im Wintersemester). Die Module, die im jeweiligen Nebenfach belegt werden können, werden vom Prüfungsausschuss auf der Webseite des Fachs bekannt gegeben. Andere Nebenfächer können auf schriftlichen Antrag vom Prüfungsausschuss genehmigt werden, sofern ein ausreichender Bezug zur Informatik vorhanden ist.

Die Festlegung des Nebenfaches erfolgt bei der Studierenden- und Prüfungsverwaltung in der Regel im dritten Semester, auf jeden Fall vor Ablegung der ersten Teilprüfung im Nebenfach. Ein Wechsel des Nebenfaches ist auf Antrag zulässig, solange keine Fachprüfung im Nebenfach endgültig nicht bestanden ist. Im gewählten Nebenfach sind 30 LP zu erbringen, die sich je nach Nebenfach unter Umständen auf mehr als drei Module verteilen.

Informationen zur Schwerpunktbildung im Nebenfach und in der Informatik finden Sie auf folgender Webseite: [Schwerpunkte](#)

## Kapitel 4

### 4. Praxis- und Berufsorientierung Professional Issues

#### 4.1 Praxis- und Berufsorientierung Professional Issues

##### Studiengang

Bachelor-Studiengang Informatik

##### Leistungspunkte

5 LP

##### Lehrveranstaltungen

Aus dem Angebot der Heinrich-Heine-Universität Düsseldorf sind Lehrveranstaltungen zur Praxis- und Berufsorientierung zu besuchen.

Dabei müssen zumindest zwei LP in Lehrveranstaltungen erworben werden, die Techniken des wissenschaftlichen Arbeitens oder Präsentationstechniken vermitteln. Hierzu zählen unter anderem:

- Rhetorikseminare / -kurse
- Seminare in der Informatik oder im gewählten Nebenfach
- Veranstaltungen des KUBUS-Programms zu Techniken des wissenschaftlichen Arbeitens oder zu Präsentationstechniken

Die übrigen Leistungspunkte (maximal 3 LP) können in einer oder mehreren der folgenden Veranstaltungsarten erworben werden:

- Praktika mit hohem Informatik-Anteil innerhalb oder außerhalb der Universität (jedoch nicht im Rahmen von Lehrveranstaltungen). Für die Anerkennung ist eine [Praktikumsbescheinigung](#) erforderlich, die den Zeitumfang ausweist und detailliert die Inhalte des Praktikums beschreibt; weiterhin muss durch die Studierenden anhand der vom Fachbereich bereitgestellten Vorlage ein aussagekräftiger [Praktikumsbericht](#) erstellt werden.
- Veranstaltungen rheinjug
- Bewerbungstraining

##### Inhalte und Kompetenzen

Die Studierenden sollen in diesem Modul die Grundlagen von wissenschaftlichen Arbeitstechniken bzw. die professionelle Präsentation von Ergebnissen erlernen.  
Die Studierenden können weiterhin im Rahmen eines Industriepraktikums im Umfeld der Informatik ihre theoretischen Kenntnisse praktisch anwenden und vertiefen. Ferner lernen Sie die betrieblichen Abläufe kennen und können ihre sozialen Kompetenzen weiter verbessern.

### **Verwendbarkeit des Moduls**

Bachelor-Studiengang Informatik

### **Teilnahmevoraussetzungen**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

Erfolgreiche Teilnahme an den ausgewählten Veranstaltungen; der Erwerb von Leistungspunkten richtet sich nach den jeweiligen Regelungen für die besuchten Veranstaltungen. Bei Praktika richtet sich die Vergabe von Leistungspunkten nach der Dauer des Praktikums: 1 LP entspricht dabei in der Regel 28 Stunden Praktikum, zusätzlich ca. 2 Stunden Nachbereitung.

### **Häufigkeit des Angebots, modulare Schiene**

regelmäßige Angebote der Universität

### **Modulbeauftragte und hauptamtliche Lehrende**

Lehrende der jeweiligen Fächer

## Kapitel 5

### 5. Lehreinheiten für Wahlpflichtbereiche (B.Sc.) Courses for elective areas (B.Sc.)

Zwecks verbesserter Studierbarkeit des Studiengangs und zur Erhöhung der Wahlmöglichkeiten der Studenten werden im Bachelor-Studiengang Informatik Module in unterschiedlichen Größen angeboten. Module müssen so kombiniert werden, dass die Gesamtsumme der LP den Erfordernissen der Prüfungsordnung entspricht. Für den Wahlpflichtbereich können Module frei kombiniert werden. Die Zusammensetzung des Schwerpunktfaches muss mit dem Mentor und Betreuer der Bachelor-Arbeit im Schwerpunkt besprochen werden (siehe Formular „Schwerpunktwahl“).

Prinzipiell ist die Unterrichtssprache Deutsch, Englischkenntnisse werden aber als Voraussetzung zum Studium verlangt. Dies ist in den jeweiligen Studienordnungen in Sektion 3, Punkt 2 dokumentiert: "Es wird darauf hingewiesen, dass das Informatikstudium Kenntnisse der englischen Sprache erfordert." Die Literatur für einige Veranstaltungen ist öfters (gezwungenermaßen) auf Englisch. Dies kann besonders für Seminare im Master-Studiengang erforderlich sein. Folien und Skripte von Vorlesungen sind auch manchmal auf Englisch verfasst.

Die schriftlichen Abschlussarbeiten müssen in deutscher oder englischer Sprache verfasst werden PO BSc, 16(1), "Die Bachelor-Arbeit kann auf Deutsch oder Englisch angefertigt werden.").

Formale Voraussetzung für die Teilnahme an Wahlpflicht-Lehrveranstaltungen für den Wahlpflicht- oder Schwerpunktbereich im Bachelor-Studiengang Informatik ist der erfolgreiche Abschluss der Module „Programmierung“ und „Rechnerarchitektur“.

Ein Modul für den Bachelor-Studiengang kann nur dann für die Individuelle Ergänzung im Master-Studiengang Informatik verwendet werden, wenn es nicht bereits für ein vorangegangenes Bachelor-Studium verwendet wurde.

#### 5.1 Algorithmen in der Bioinformatik Algorithms in bioinformatics

##### Studiengang

Bachelor-Studiengang Informatik

##### Leistungspunkte

5 LP ab PO 2013, alte PO's 7,5 LP (Bachelor, ohne Seminar)  
7,5 LP (Master, inkl. Seminar)

##### Lehrveranstaltungen

- Vorlesung „Algorithmen in der Bioinformatik“, 2 SWS
- Übung, 2 SWS
- Seminar, 2 SWS (nur Master)

##### Inhalte

- Biologische und algorithmische Grundlagen
- Exhaustive Suche: DNA-Motive
- Gierige Algorithmen: Genom-Umordnungen
- Dynamische Programmierung: Sequenz-Alignment
- Graphen-Algorithmen: Sequenzierung
- Kombinatorische Mustersuche, BLAST
- Hidden Markov Modelle
- Cluster & Cliques
- Phylogenetische Bäume und molekulare Evolution
- Maximum-Likelihood und Bayesianische Methoden

- Randomisierte Motivsuche

### **Lernergebnisse/Kompetenzen**

Nach der Veranstaltung werden die Studierende in der Lage sein

- verschiedene Klassen von Algorithmen zu differenzieren;
- klassische Algorithmen und spezielle bioinformatische Algorithmen auf biologische Probleme anzuwenden;
- gelernte Algorithmen exemplarisch in der Programmiersprache Python umzusetzen;
- bioinformatische Probleme mit den vorgestellten Algorithmen selbständig lösen zu können.

### **Empfohlene Literatur**

- Neil C. Jones, Pavel A. Pevzner: An Introduction to Bioinformatics Algorithms. The MIT Press, 2004.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Teil eines Master-Moduls im Master-Studiengang Biologie

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“

Master-Studierende müssen zusätzlich folgende Module erfolgreich abgeschlossen haben:

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (i.d.R. schriftlich)

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Studienjahr, in der Regel im Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Gunnar W. Klau

## **5.2 Algorithmen zur Entscheidungsfindung unplugged Decision-making Algorithms Unplugged**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmen zur Entscheidungsfindung unplugged“, 4 SWS

## **Inhalte**

Diese Veranstaltung beschäftigt sich mit verschiedenen Bereichen zur Entscheidungsfindung aus Sicht der Informatik mit einem speziellen Fokus auf „Informatik unplugged“. Ziel von Informatik unplugged ist es, fundamentale Konzepte der Informatik anhand von einzelnen Aktivitäten zu erlernen. Dabei haben alle diese Aktivitäten gemeinsam, dass keine Computer benötigt werden. In dieser Veranstaltung sollen die Studierenden zunächst dieses Gesamtkonzept kennenlernen und sich danach in Kleingruppen intensiv mit einem Thema aus dem Bereich der Entscheidungsfindung auseinandersetzen. Im Anschluss daran sollen die Teilnehmer selber unplugged Aktivitäten zu einem gewählten Thema entwickeln, und die zugrundeliegenden Konzepte den anderen Teilnehmern vorstellen. Die unplugged Aktivität soll dabei den Zugang zu einem neuen Themengebiet erleichtern. In Übungsaufgaben soll der erlernte Inhalt angewendet werden. Beispiele für Themengebiete sind Algorithmen zur Entscheidungsfindung in den Bereichen Hedonic Games, Matching oder Majority graphs.

In dieser Veranstaltung müssen Vorträge ausgearbeitet und gehalten werden, außerdem muss eine schriftliche Zusammenfassung des gewählten Themas erfolgen. Durch gegenseitige Begutachtungen sollen die Studierenden ihre Fähigkeiten im Bereich der kritischen und fairen Begutachtung erweitern.

## **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- das Konzept von Informatik unplugged zu erläutern.
- Informatik unplugged Aktivitäten mit anderen Teilnehmern durchzuführen, und das dahintersteckende fundamentale Konzept anschaulich zu erklären.
- neue Informatik unplugged Aktivitäten zu Algorithmen zur Entscheidungsfindung selbst zu entwickeln.
- sich einen neuen Themenbereich selbstständig zu erarbeiten.
- ein selbstständig erarbeitetes Thema anschaulich zu präsentieren
- erlernte Inhalte anzuwenden

## **Empfohlene Literatur**

- Computer Science Unplugged. Tim Bell, Ian H. Witten, und Mike Fellows. Online verfügbar unter: <http://csunplugged.org>
- Abenteuer Informatik: IT zum Anfassen – von Routenplaner bis Online Banking. Jens Gallenbacher, Spektrum Akademischer Verlag, 2012.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

## **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Schriftliche Ausarbeitung einer neuen unplugged Aktivität
- Vorstellen der neu erarbeiteten unplugged Aktivität
- Vorstellen der theoretischen Grundlagen zu der erarbeiteten unplugged Aktivität
- erfolgreiche Bearbeitung der wöchentlichen Übungsaufgaben
- angemessene Begutachtung einer anderen schriftlichen Zusammenfassung
- Bestehen der Prüfung am Ende der Veranstaltung über alle Themengebiete der Veranstaltung (i.d.R. Klausur, 90 Minuten)

## **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul findet in unregelmäßigen Abständen statt.

## **Modulbeauftragte und hauptamtliche Lehrende**



### 5.3 Algorithmische Eigenschaften von Wahlsystemen I Algorithmic Properties of Voting Systems I

#### Studiengang

Bachelor-Studiengang Informatik

#### Leistungspunkte

5 LP ab PO 2013, alte PO's 7,5 LP

#### Lehrveranstaltungen

- Vorlesung „Algorithmische Eigenschaften von Wahlsystemen I“, 2 SWS
- Übung, 1 SWS

#### Inhalte

- Grundlagen der Social-Choice-Theorie
  - Wahlen
  - Wahlsysteme
  - Eigenschaften von Wahlsystemen
  - Einige Wahl-Paradoxa und Unmöglichkeitssätze
- Algorithmik und Komplexität von Wahlproblemen
  - Gewinnerbestimmung
  - Mögliche und notwendige Gewinner
  - Manipulation

#### Lernergebnisse/Kompetenzen

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem jungen, sich rasant entwickelnden, interdisziplinären Gebiet *Computational Social Choice* zu vermitteln, welches derzeit in der Abteilung „Komplexitätstheorie und Kryptologie“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird, und in dem bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende die wichtigsten Wahlverfahren und ihre „social-choice“-theoretischen Eigenschaften sowie die damit im Zusammenhang stehenden Entscheidungsprobleme (Gewinnerbestimmung, Manipulation usw.) und die zu ihrer Lösung entwickelten Algorithmen und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Ebenso sollten sie die Komplexität dieser Probleme formal bestimmen und beschreiben können.

#### Empfohlene Literatur

- Jörg Rothe (ed.): *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: *Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen*, Spektrum Akademischer Verlag (Springer), 2011.

#### Ergänzende Literatur

- Piotr Faliszewski, Lane A. Hemaspaandra, Edith Hemaspaandra, and Jörg Rothe: A Richer Understanding of the Complexity of Election Systems, Chapter 14 in: „Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz,” pp. 375-406, S. Ravi and S. Shukla (eds.), Springer, 2009.
- Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel Procaccia (eds.): *Handbook of Computational Social Choice*, Cambridge University Press, 2015.
- Steven J. Brams and Peter C. Fishburn: Voting Procedures, Chapter 4 in: „Handbook of Social Choice and Welfare,” Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura (eds.), North-Holland, 2002.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle vier Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **5.4 Algorithmische Geometrie Computational Geometry**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP für alle PO's

### **Lehrveranstaltungen**

- Vorlesung „Algorithmische Geometrie“, 2 SWS
- Übung, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 4 SWS

### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse zur Berechnung und Anwendung der folgenden Probleme und Methoden.

- Konvexe Hülle
- Distanzprobleme
- Schnittprobleme
- Voronoi Diagramme
- Polygon Triangulierung
- Delaunay Triangulierung
- Plane-Sweep Algorithmen
- Randomisierte inkrementelle Algorithmen

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein erweitertes Verständnis der theoretischen Grundlagen über den Entwurf von Algorithmen für Fragestellungen aus der algorithmischen Geometrie.

### **Literatur**

- de Berg et al: Computational Geometry, Algorithms and applications. Springer-Verlag, Berlin, 2. rev. ed., 2000.
- Preparata, Shamos: Computational geometry, an introduction. Springer-Verlag, New York, 1985.
- Edelsbrunner: Algorithms in Combinatorial Geometry, EATCS Monographs in Computer Science 10. Springer-Verlag, 1987.
- Matousek: Lectures on discrete geometry, Graduate Texts in Mathematics, 212. Springer-Verlag, New York, 2002.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 45 Minuten) am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Jedes zweite Sommersemester

### **Modulbeauftragte und hauptamtlich Lehrende**

Prof. Dr. Egon Wanke

## **5.5 Algorithmische Komplexitätstheorie Algorithmic Complexity Theory**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmische Komplexitätstheorie“, 2 SWS
- Übung, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 4 SWS

### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen.

- Methoden des Algorithmenentwurfs (Teile und Beherrsche, Dynamische Programmierung, ...),
- Einführung in die algorithmische Komplexität (Nichtdeterminismus, NP-Vollständigkeit, ...),
- Schaltkreiskomplexität
- Approximationsalgorithmen
- Parameterisierte Algorithmen
- Randomisierte Algorithmen
- Einführung in Online-Algorithmen

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein erweitertes Verständnis der theoretischen Grundlagen über den Entwurf von Algorithmen. Sie können algorithmische Probleme spezifizieren und sie bzgl. ihrer algorithmischen Komplexität analysieren.

### **Literatur**

- Christos H. Papadimitriou, Computational, Addison-Wesley, 1993
- Richard Johnsonbaugh, Marcus Schäfer: Algorithms, Pearson Education, 2004
- Jon Kleinberg, Eva Tardos: Algorithm Design, Addison Wesley, 2006

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 45 Minuten) am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester

### **Modulbeauftragte und hauptamtlich Lehrende**

Prof. Dr. Egon Wanke

## **5.6 Angewandte Algorithmik Applied Algorithmics**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Vorlesung „Angewandte Algorithmik“, 2 SWS
- Übung, 2 SWS

## Inhalte

"In theory, there is no difference between theory and practice. In practice, there is."

Algorithmen bilden die Grundlage jedes Computerprogramms. Traditionell liegt der Fokus des Algorithmen-Designs auf der Theorie effizienter Algorithmen und deren *worst case*-Analyse. In dieser Veranstaltung geht es jedoch um praktisch effiziente Algorithmen, oft auch für beweisbar schwere Probleme, mit dem Ziel, die Prinzipien der Optimalität nicht (vollständig) aufzugeben. Themengebiete sind:

- Pattern Matching und Textindizes (von Suffix-Trees zur Burrows-Wheeler-Transformation)
- Algorithmen für schwere Probleme: Branch-and-Bound, Ganzzahlige Lineare Programmierung, Approximationsalgorithmen
- Algorithmen zum Clustern großer Datenmengen
- Google's Pagerank-Algorithmus

Zudem wird ein ausgewähltes (fiktives) praktisches Problem (aus dem Bereich der Alchemie) im Rahmen einer Programmierchallenge in Gruppenarbeit gelöst.

## Lernergebnisse/Kompetenzen

Nach der Veranstaltung werden die Studierenden in der Lage sein

- verschiedene Techniken der Angewandten Algorithmen zu beherrschen und diese auf neue Probleme anzuwenden
- Probleme praktisch effizient zu lösen und diese Lösungen zu implementieren

## Empfohlene Literatur

- Teilweise Originalarbeiten

## Verwendbarkeit des Moduls

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Teil eines Master-Moduls im Master-Studiengang Biologie

## Teilnahmevoraussetzungen

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“

Master-Studierende müssen zusätzlich folgende Module erfolgreich abgeschlossen haben:

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (i.d.R. schriftlich)

## Häufigkeit des Angebots, modulare Schiene

Jedes Studienjahr, in der Regel im Wintersemester

## Modulbeauftragte und hauptamtliche Lehrende

Prof. Dr. Gunnar W. Klau

## **5.7 Approximative Algorithmen**

### **Approximate Algorithms**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Approximative Algorithmen“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

Dieses Modul befasst sich mit den folgenden schweren Optimierungsproblemen, für die es vermutlich keine effizienten Algorithmen gibt, und stellt verschiedene Näherungslösungen für diese Probleme vor.

- Metric Traveling Salesman
- Job Scheduling
- Knapsack
- Bin Packing

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen schweren Optimierungsprobleme zu erläutern und
- die behandelten Approximationsalgorithmen auf konkrete Eingaben anzuwenden

#### **Literatur**

- K. Jansen, M. Margraf: Approximative Algorithmen und Nichtapproximierbarkeit, de Gruyter Verlag, 2008.
- R. Wanka: Approximationsalgorithmen, Teubner Verlag, 2006.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

#### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Bestehen der Klausur zum Ende des Semesters

#### **Häufigkeit des Angebots, modulare Schiene**

Dieses Halb-Modul wird in unregelmäßigen Abständen angeboten.

#### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **5.8 Betriebssysteme Operating Systems**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Betriebssysteme“, 2 SWS
- Übung/Seminar, 2 SWS

### **Inhalte**

- Architekturformen: monolithisch, geschichtet, Mikrokern, Client/Server
- Speicher: Segmentierung, Paging, Garbage Collection
- Nebenläufigkeit: Schedulingstrategien, Prozesse, Threads, Interrupts
- Synchronisierung: Semaphore, klassische Problemstellungen, Verklemmungen
- Dateisysteme: FAT, UNIX, EXT, NTFS
- Kommunikation: Signale, Pipes, Sockets
- Sicherheit: HW-Schutz
- Fallstudien, u.a. Linux, Microsoft Windows

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Betriebssystembegriffe zu nennen und zu erläutern
- Speicherverwaltungstechniken (physikalisch, virtuell, Segmentierung und Paging) auf gegebene Beispiele anzuwenden und zu bewerten.
- Schedulingstrategien anzuwenden und zu bewerten.
- Synchronisierungsprobleme in parallelen Threads zu erkennen und eigene Synchronisierungslösungen zu konzipieren
- Interprozesskommunikation anzuwenden
- grundlegende Betriebssystemkonzepte in modernen Desktop-Betriebssystemen in eigenen Worten erklären zu können

### **Empfohlene Literatur**

- Andrew S. Tanenbaum: „Modern Operating Systems“, 4. Auflage, Prentice Hall, 2014.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

Erfolgreiche Teilnahme an der Prüfung am Ende der Veranstaltung.

## **Häufigkeit des Angebots, modulare Schiene**

Wird nicht mehr angeboten

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **5.9 Betriebssysteme und Systemprogrammierung Operating Systems and System Programming**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Kreditpunkte**

10 LP

### **Lehrveranstaltungen**

- Vorlesung „Betriebssysteme und Systemprogrammierung, 4 SWS
- Übung, 2 SWS
- Praktische Übung, 2 SWS

### **Inhalte**

- Grundlagen: C- und UNIX Shell-Programmierung
- Grundlegende Betriebssystemkonzepte, u.a. Speicher (Heap, Stack), Adressräume, Prozesse und Threads, Scheduling, Synchronisierung, Interrupts, Dateisysteme, Interprozess-Kommunikation, Schutzmechanismen
- Programmierung von Systemsoftware
- Praktische Übungen, direkt am Rechner

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Betriebssystemkonzepte zu erläutern
- Zusammenhänge zwischen der Ausführung von Programmen und dem Betriebssystemen zu verstehen und zu erklären
- Synchronisierungsprobleme in parallelen Threads zu erkennen und eigene Synchronisierungslösungen zu konzipieren
- einfache Systemprogramme auf Basis der Systemaufrufchnittstelle UNIX-ähnlicher Betriebssysteme in der Programmiersprache C zu entwickeln

### **Empfohlene Literatur**

- Andrew S. Tanenbaum: „Modern Operating Systems“, 4. Auflage, Prentice Hall, 2014.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“



### **Voraussetzungen für die Vergabe von Kreditpunkten**

Erfolgreiche Bearbeitung der Übungsaufgaben. Erfolgreiche Teilnahme an der Prüfung am Ende der Veranstaltung.

### **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **5.10 Compilerbau**

### **Compiler construction**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Compilerbau“, 2 SWS
- Übung, 2 SWS
- Praktische Übung, 2 SWS

#### **Inhalte**

Die Inhalte dieses Moduls sind wie folgt:

Grundlagen des Compilerbaus

- Lexikale Analyse (reguläre Ausdrücke und endliche Automaten)
- Syntaxanalyse (kontextfreie Grammatiken und deterministisches Parsing)
- Semantische Analyse
- Code-Generierung
- Benutzen von Werkzeugen zur automatischen Erzeugung von Compilern

#### **Lernergebnisse/Kompetenzen**

Nach erfolgreichem Abschluss des Moduls sollte der Student:

- verstehen wie Programmiersprachen übersetzt und implementiert werden
- Syntaxbeschreibungen einer Programmiersprache verstehen und anpassen können. Insbesondere soll der Student ermitteln können ob die Beschreibung für eine automatisierte Bearbeitung in einem Compiler geeignet ist
- in der Lage sein für eine neue Programmiersprache einen Parser, bzw. Compiler selbst zu entwickeln

#### **Literatur**

- Andrew W. Appel, Modern Compiler Implementation in Java, 2nd Edition, Cambridge University Press, (ISBN-13: 9780521820608 — ISBN-10: 052182060X).
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools (2nd Edition), Addison Wesley, ISBN-13: 978-0321486813.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik

- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

Erfolgreiche Bearbeitung der Pflichtübungen; Erfolgreiche Entwicklung eines eigenen Compilers; Bestehen der Klausur;

### **Häufigkeit des Angebots, modulare Schiene**

Jährlich im Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **5.11 Datenbanksysteme Database Systems**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

10 LP ab PO 2013, alte PO's 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Datenbanksysteme“, 4 SWS
- Übung, 2 SWS
- Praktische Übung bzw. Praktikum, 2 SWS

### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse zu Datenbanksystemen. Der Schwerpunkt liegt auf relationalen Datenbanksystemen. Neben den verschiedenen Phasen des Datenbankentwurfs werden ausführlich Anfragen und Anwendungsprogrammierung behandelt. Darüber hinaus werden weitere Themen (u.a. Architektur, Transaktionsverwaltung, Anfrageoptimierung) angesprochen. Im Einzelnen werden behandelt:

- Aufgaben eines Datenbanksystems
- Architektur von Datenbanksystemen
- Daten(bank)modelle; hierarchische Modelle, Netzwerkmodell, Relationales Modell
- Anfragesprachen für relationale Datenbanken und ihre Grundlagen; Relationale Algebra, Tupel- und Bereichskalkül, QBE, SQL
- konzeptioneller Datenbankentwurf
- logischer Datenbankentwurf; Normalisierung
- Datenbankanwendungsprogrammierung; Datenbankanbindung im Web
- Transaktionen und Grundlagen der Transaktionsverwaltung
- Anfrageverarbeitung und -optimierung

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Aufgaben und Funktionen eines Datenbanksystems zu nennen und zu erläutern
- selbstständig Datenbanken zu entwerfen,
- Datenbankanfragen in verschiedenen Formalismen (SQL, Relationalenalgebra, Tupelkalkül) zu formulieren und die Effizienz der Auswertung verschiedener Formulierungen zu bewerten,
- Datenbankanwendungen zu programmieren,

- die grundlegenden Aspekte der Transaktionsverwaltung Nebenläufigkeit und der Synchronisierung zu verstehen

### **Literatur**

- Heuer, A., Saake, G., Sattler, K.-U.: Datenbanken - Konzepte und Sprachen. 5. Aufl., mitp-Verlag, Bonn, 2013
- Vossen, G.: Datenbankmodelle, Datenbanksprachen und Datenbankmanagement-Systeme. 5. Aufl., Oldenbourg, München, 2008
- Kemper, A., Eickler, A.: Datenbanksysteme --Eine Einführung. 9. Aufl., Oldenbourg, München, 2013
- Elmasri, R.; Navathe, S.B.: Fundamentals of Database Systems, Addison-Wesley / Pearson, 2011
- Date, C.J.: An Introduction to Database Systems, 8/e; Addison-Wesley / Pearson, 2004
- Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book. 2/e, Prentice Hall / Pearson, 2009

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Modul D3 im Studienbereich Informatik (D) im Integrativen Bachelor-Studiengang Sprachtechnologie und Informationswissenschaft

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

Wünschenswert sind zudem fundierte Kenntnisse aus dem Modul Algorithmen und Datenstrukturen

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Teilnahme an den Übungen (in der Regel mit Hausaufgaben)
- erfolgreicher Abschluss der Praktischen Übungen durch ein kleines Projekt, in dem eine Aufgabe mit Datenbankentwurf und Datenbankanwendungsprogrammierung zu bearbeiten ist
- schriftliche Prüfung (Klausur, i.d.R. 90 Minuten)

### **Häufigkeit des Angebots, modulare Schiene**

Jährlich im Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

## **5.12 Einführung in die logische Programmierung Introduction to Logic Programming**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Einführung in die logische Programmierung“, 2 SWS

- Übung, 2 SWS
- Praktische Übung bzw. Praktikum, 2 SWS

## **Inhalte**

In der imperativen Programmierung werden Programme durch Sequenzen von Instruktionen dargestellt. In der logischen Programmierung beschreibt man statt einer Lösung für ein Problem in Form eines Algorithmus das Problem durch Fakten und Regeln und der Computer findet selbständig eine Lösung. Das eröffnet einen völlig neuen und radikal unterschiedlichen Blickwinkel auf das Programmieren der auch bei der alltäglichen Entwicklung mit C oder Java sehr nützlich ist.

Die Vorlesung behandelt die Themen:

- Aussagenlogik, Prädikatenlogik
- Resolution
- Programmieren mit Horn Klauseln
- Praktische Grundlagen von Prolog
- Suchalgorithmen und Künstliche Intelligenz mit Prolog
- Grundzüge der Constraint-Programmierung

## **Lernergebnisse/Kompetenzen**

Nach erfolgreichem Abschluss des Moduls sollte der Student

- die logischen Grundlagen von Prolog verstehen und logische Berechnungen in Aussagenlogik und Prädikatenlogik durchführen können
- die praktischen Datenstrukturen von Prolog benutzen können
- in der Lage sein kleinere Prolog Programme eigenständig entwickeln zu können
- wichtige Suchalgorithmen vergleichen können und praktisch in Prolog umsetzen
- kleinere Aufgaben der künstlichen Intelligenz in Prolog lösen können

## **Literatur**

- Nilsson, Maluszynski - Logic, Programming and Prolog (eBook)
- Blackburn, Bos, Striegnitz, Learn Prolog Now!, College Publications

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

## **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung in den Übungen
- Erfolgreiche Teilnahme an der Abschlussprüfung

## **Häufigkeit des Angebots, modulare Schiene**

Jährlich im Wintersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **5.13 Graphenalgorithmen I**

# **Algorithms for Graphs I**

## **Studiengang**

Bachelor-Studiengang Informatik

## **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

## **Lehrveranstaltungen**

- Vorlesung „Graphenalgorithmen I“, 2 SWS
- Übung, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 2 SWS

## **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen.

- Suchmethoden auf Graphen
- Topologische Anordnungen
- Zusammenhangsprobleme
- Kürzeste Wege
- Minimale Spannbäume
- Netzwerkfluss-Probleme
- Matching-Probleme

## **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein erweitertes Verständnis der theoretischen Grundlagen über den Entwurf von Graphalgorithmen.

## **Basisliteratur**

Verschiedene aktuelle Lehrbücher über Graphenalgorithmen

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

## **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 45 Minuten) am Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Egon Wanke

## **5.14 Grundlagen Verteilter Systeme**

### **Foundations of Distributed Systems**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Grundlagen Verteilter Systeme“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

- Architekturformen: Client/Server, P2P, hybride Ansätze
- Kommunikation Sockets (UDP, TCP), entfernter Methodenaufruf (RMI), verteilter gemeinsamer Speicher (DSM)
- Koordination: Uhrensynchronisierung, logische Zeit, wechselseitiger Ausschluss, globale Zustände, Wahlen, Terminierung, Verklemmungen
- Replikation und Konsistenz: daten- und klientenzentrierte Modelle, Aktualisierungsmethoden
- Fehlertoleranz: Fehlermodelle, Konsensus, Checkpointing
- Sicherheit: Grundlagen Verschlüsselung und Authentisierung

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Kommunikationsformen (Nachrichten, entfernte Aufrufe, gemeinsamer Speicher) in eigenen Programmen anzuwenden.
- Koordinationsalgorithmen zu verstehen und zu konzipieren.
- Konsistenzstrategien für Replikate zu vergleichen und zu bewerten
- Grundlegende Fehlertoleranz und Sicherheitsaspekte in verteilten Systemen zu erläutern

#### **Empfohlene Literatur**

- G. Coulouris et.al., „Distributed Systems: Concepts and Design“, Addison-Wesley, 5. Aufl. 2011
- A. Tanenbaum and M. van Steen: „Distributed Systems: Principles and Paradigms“, 3. Auflage, Prentice Hall, 2013.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

#### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme an der Prüfung am Ende der Veranstaltung.

#### **Häufigkeit des Angebots, modulare Schiene**

Wird nicht mehr angeboten

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **5.15 Knapsack Problems**

### **Knapsack Problems**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Knapsack Problems“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

Dieses Modul befasst sich mit verschiedenen Varianten des Rucksackproblems und stellt verschiedene Lösungsmethoden für diese Probleme vor.

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen Rucksackprobleme zu erläutern
- die Ideen der vorgestellten Algorithmen zu kennen und
- die behandelten Algorithmen auf konkrete Eingaben anzuwenden

#### **Literatur**

- H. Kellerer, U. Pferschy, D. Pisinger: Knapsack Problems, Springer, 2010.
- B. Korte, J. Vygen: Kombinatorische Optimierung, Springer, 2008.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

#### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Bestehen der Klausur zum Ende des Semesters

#### **Häufigkeit des Angebots, modulare Schiene**

Dieses Halb-Modul wird in unregelmäßigen Abständen angeboten.

## **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## 5.16 Kryptokomplexität I

### Cryptocomplexity I

#### Studiengang

Bachelor-Studiengang Informatik

#### Leistungspunkte

10 LP ab PO 2013, alte PO's 15 LP

#### Lehrveranstaltungen

- Vorlesung „Kryptokomplexität I“, 4 SWS
- Übung, 2 SWS
- Seminar, 2 SWS

#### Inhalte

- Einführung in die Kryptologie
  - Aufgaben und Ziele der Kryptologie
  - Einige klassische Kryptosysteme und ihre Kryptoanalyse
    - o Substitutions- und Permutationschiffren
    - o Affin Lineare Blockchiffren
    - o Block- und Stromchiffren
  - Perfekte Geheimhaltung
    - o Satz von Shannon und Vernams One-Time Pad
    - o Entropie und Schlüsselmehrdeutigkeit
  - RSA, Primzahltests und das Faktorisierungsproblem
    - o Das Public-Key Kryptosystem RSA
    - o Digitale Signaturen mit RSA
    - o Primzahltests und das Faktorisierungsproblem
    - o Sicherheit von RSA: Mögliche Angriffe und Gegenmaßnahmen
- Einführung in die Komplexitätstheorie
  - Grundlagen
    - o Komplexitätsmaße und -klassen
    - o Kompression und Beschleunigung
    - o Hierarchiesätze
    - o Eine untere Schranke
  - Zwischen L und PSPACE
    - o Einfache Inklusionen
    - o Komplexitätsbeschränkte Many-one-Reduktionen
    - o Vollständige Probleme in NL
    - o NP-vollständige Probleme

#### Lernergebnisse/Kompetenzen

Ziel des Moduls Kryptokomplexität I ist die Vermittlung von Kenntnissen über die wichtigsten Kryptosysteme (also effiziente Algorithmen zur Verschlüsselung und legitimierten Entschlüsselung); andererseits wird die Sicherheit solcher Systeme behandelt, die häufig auf der Komplexität geeigneter Probleme beruht. Deshalb wird auch in die Grundlagen der Komplexitätstheorie eingeführt, insbesondere mit dem Ziel, Methoden zum Nachweis unterer Schranken bezüglich der Komplexitätsmaße Rechenzeit und Speicherplatz zu verstehen und anwenden zu können. Dies ergänzt auch die Veranstaltungen zur Algorithmik, in denen es vorrangig um den Nachweis oberer Schranken (also die Entwicklung möglichst effizienter Algorithmen) für Probleme geht. Am Ende der Veranstaltung sollten Studierende in der Lage sein, die erlernten Modellbildungen und Techniken auf neue Probleme, Kryptosysteme und Algorithmen anwenden können. So sollen sie etwa die Komplexität von Problemen, die Sicherheit von kryptographischen Verfahren und die Laufzeit und den Speicherplatzbedarf von Algorithmen einschätzen können. Besonderer Wert wird darauf gelegt, dass die Studierenden die enge Verflechtung dieser Gebiete verstehen.

#### Empfohlene Literatur



- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität, *eXamen.Press*, Springer-Verlag, 2008.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity, *EATCS Texts in Theoretical Computer Science*, Springer-Verlag, 2005.

### **Ergänzende Literatur**

- Douglas R. Stinson: Cryptography: Theory and Practice, Chapman & Hall/CRC, 2006.
- Johannes Buchmann: Einführung in die Kryptographie, Springer-Verlag, 2001.
- Christos Papadimitriou: Computational Complexity, Addison-Wesley, 1995.
- Gerd Wechsung: Vorlesungen zur Komplexitätstheorie, Teubner-Verlag, 2000.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle zwei Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **5.17 Machine Learning**

### **Machine Learning**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

10 LP ab PO 2013, alte PO's 15 LP

#### **Lehrveranstaltungen**

- Vorlesung „Machine Learning“, 4 SWS
- Übung, 2 SWS

#### **Inhalte**

Dieses Modul vermittelt grundlegendes Wissen zu folgenden Themen:

- Probability, frequentist statistics, Bayesian statistics
- Supervised learning, unsupervised learning
- Generative vs discriminative models

- Linear regression, linear discriminant analysis
- Gaussian processes
- Support vector machines
- Kernel trick, kernel PCA
- Graphical models
- Neural networks

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- die Grundlagen des Maschinellen Lernens zu beschreiben und anzuwenden
- die mathematische Beschreibung dieser Grundlagen zu erklären und einfache Sachverhalte zu beweisen
- die Grundlagen und ihre mathematischen Beschreibungen anzuwenden, um selbständig Datenanalyseprobleme zu bearbeiten

### **Empfohlene Literatur**

Es wird nicht ein bestimmtes Lehrbuch verwendet, jedoch sind folgende Bücher hilfreich:

- Murphy, Machine Learning: A Probabilistic Perspective
- MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge 2003
- Barber, Bayesian Reasoning and Machine Learning, Cambridge 2012
- Rasmussen/Williams, Gaussian Processes for Machine Learning, MIT 2006
- Bishop, Pattern Recognition and Machine Learning, Springer 2007
- Schölkopf/Smola, Learning with Kernels, MIT 2001
- Jaynes, Probability Theory – the Logic of Science, Cambridge 2003

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

Hilfreich ist erfolgreicher Abschluss der Mathematik Module:

- "Lineare Algebra", "Analysis", "Stochastik" und "Numerik"

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung in den Übungen;
- Prüfung zu Vorlesung und Übungen am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Harmeling

## **5.18 Matching**

### **Matching**

## **Studiengang**

Bachelor-Studiengang Informatik

## **Leistungspunkte**

10 LP

## **Lehrveranstaltungen**

- Vorlesung „Matching“, 4 SWS
- Übung, 2 SWS
- Seminar, 2 SWS

## **Inhalte**

Dieses Modul beschäftigt sich mit unterschiedlichen Arten von Matching bzw. Zuteilungsverfahren. Solche Verfahren können in unterschiedlichen Bereichen eingesetzt werden, Beispiele sind die Vergabe von Studienplätzen, die Verteilung von Assistenzärzten auf Krankenhäuser oder das Nierentransplantationsproblem. Inhalt der Veranstaltung sind verschiedene Verfahren mit ihren axiomatischen und algorithmischen Eigenschaften.

## **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- Zuteilungsverfahren in unterschiedlichen Situationen durchzuführen
- Herausforderung von praktischen Zuteilungsverfahren zu identifizieren
- neue Verfahren zur Zuteilung für spezielle Einsatzbereiche zu entwickeln und im Hinblick auf Ihre Eigenschaften zu untersuchen
- bekannte Verfahren zur Zuteilung auf neue Einsatzbereiche zu übertragen
- verschiedene Verfahren zur Zuteilung gegenüber zu stellen
- Empfehlungen für bestimmte Einsatzbereiche der Zuteilung zu geben

## **Literatur**

- The Stable Marriage Problem, Structure and Algorithms. Dan Gusfield and Robert W. Irving. MIT Press, 1989.
- Algorithmics of Matching under Preferences, David F. Manlove. World Scientific Publishing Company, 2013.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

## **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- aktive und erfolgreiche Mitwirkung in den Übungen
- bestehen der Prüfung am Ende der Veranstaltung (i.d.R. Klausur, 90 Minuten)

## **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul findet in unregelmäßigen Abständen statt.

## **Modulbeauftragte und hauptamtliche Lehrende**

## **5.19 Networks, Crowds, and Markets**

### **Networks, Crowds, and Markets**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Networks, Crowds, and Markets“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

##### **Inhalte**

Dieses Modul befasst sich mit folgenden theoretischen Aspekten von Netzwerken:

- Grundlagen
- Struktur von Netzwerken
- Teilgraphen in Netzwerken
- Zentralitätsmaße in Netzwerken
- Analysemethoden für Netzwerke
- Modellierung von Netzwerken

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen Anwendungsbereiche von Netzwerken zu erläutern
- die besprochenen Grundlagen von Netzwerken formal zu definieren
- die behandelten Analysemethoden für Netzwerke zu verstehen und auf konkrete Eingaben anzuwenden

#### **Literatur**

- D. Easley, J. Kleinberg: Networks, Crowds, and Markets, Cambridge University Press, 2010.
- E. Estrada: The Structure of Complex Networks: Theory and Applications, Oxford University Press, 2011.
- D. Jungnickel: Graphs, Networks and Algorithms, Springer, 2013
- S. Wassermann, K. Faust: Social Network Analysis: Methods and Applications, Cambridge University Press, 2009.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

#### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Abgabe der Übungsaufgaben, Bestehen der Klausur zum Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **5.20 Nichtkooperative Spieltheorie Noncooperative Game Theory**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Nichtkooperative Spieltheorie“, 2 SWS
- Übung, 1 SWS

### **Inhalte**

- Nichtkooperative Spiele: Gegeneinander spielen
  - Grundlagen
    - o Normalform, dominante Strategien und Gleichgewichte
    - o Weitere Zwei-Personen-Spiele
  - Nash-Gleichgewichte in gemischten Strategien
    - o Definition und Eigenschaften gemischter Nash-Gleichgewichte
    - o Existenz von Nash-Gleichgewichten in gemischten Strategien
  - Schachmatt: Spielbäume in Spielen mit perfekter Information
    - o Sequenzielle Zwei-Personen-Spiele
    - o Gleichgewichte in Spielbäumen
  - Full House: Spiele mit unvollkommener Information
    - o Das Ziegenproblem
    - o Analyse einer einfachen Poker-Variante
  - Wie schwer ist es, ein Nash-Gleichgewicht zu finden?
    - o Nash-Gleichgewichte in Nullsummenspielen
    - o Nash-Gleichgewichte in allgemeinen Normalform-Spielen

### **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem Gebiet der nichtkooperativen Spieltheorie zu vermitteln. Am Ende der Veranstaltung sollten Studierende wichtige Probleme und die zu ihrer Lösung entwickelten Algorithmen und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Insbesondere sollten sie in der Lage sein, „strategische Szenarien“ durch nichtkooperative Spiele zu beschreiben und Stabilitäts- und Gleichgewichtskonzepte in diesen Spielen formal zu charakterisieren. Ebenso sollten sie die Komplexität der für solche Spiele relevanten Entscheidungsprobleme (in einer geeigneten kompakten Darstellung) formal bestimmen und beschreiben können.

### **Empfohlene Literatur**

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen, Spektrum Akademischer Verlag (Springer), 2011.

### **Ergänzende Literatur**

- Martin J. Osborne and Ariel Rubinstein: A Course in Game Theory, MIT Press, 1994.
- Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani (eds.): Algorithmic Game Theory, Cambridge University Press, 2008.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle vier Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **5.21 Präferenzaggregation durch Wählen: Algorithmik und Komplexität Preferene Aggregation by Voting: Algorithmics and Complexity**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

10 LP ab PO 2013, alte PO's 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Präferenzaggregation durch Wählen: Algorithmik und Komplexität“, 4 SWS
- Übung, 2 SWS
- Seminar, 2 SWS

### **Inhalte**

- Grundlagen der Social-Choice-Theorie
  - Wählen
  - Wahlsysteme
  - Eigenschaften von Wahlsystemen
  - Einige Wahl-Paradoxa und Unmöglichkeitssätze
- Algorithmik und Komplexität von Wahlproblemen

- Gewinnerbestimmung
- Mögliche und notwendige Gewinner
- Manipulation
- Wahlkontrolle
- Bestechung

### **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem jungen, sich rasant entwickelnden, interdisziplinären Gebiet *Computational Social Choice* zu vermitteln, welches derzeit in der Abteilung „Komplexitätstheorie und Kryptologie“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird und in dem bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende die wichtigsten Wahlverfahren und ihre „social-choice“-theoretischen Eigenschaften sowie die damit im Zusammenhang stehenden Entscheidungsprobleme (Gewinnerbestimmung, Manipulation, Wahlkontrolle, Bestechung usw.) und die zu ihrer Lösung entwickelten Algorithmen und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Ebenso sollten sie die Komplexität dieser Probleme formal bestimmen und beschreiben können.

### **Empfohlene Literatur**

- Jörg Rothe (ed.): *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: *Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen*, Spektrum Akademischer Verlag (Springer), 2011.

### **Ergänzende Literatur**

- Piotr Faliszewski, Lane A. Hemaspaandra, Edith Hemaspaandra, and Jörg Rothe: A Richer Understanding of the Complexity of Election Systems, Chapter 14 in: „Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz,” pp. 375-406, S. Ravi and S. Shukla (eds.), Springer, 2009.
- Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel Procaccia (eds.): *Handbook of Computational Social Choice*, Cambridge University Press, 2015.
- Steven J. Brams and Peter C. Fishburn: Voting Procedures, Chapter 4 in: „Handbook of Social Choice and Welfare,” Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura (eds.), North-Holland, 2002.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle zwei Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

### **5.22 Rechnernetze Computer Networks**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

10 LP ab PO 2013, alte PO's 15 LP

#### **Lehrveranstaltungen**

- Vorlesung „Rechnernetze“, 4 SWS
- Übung, 2 SWS
- Praktikum, 2 SWS

#### **Inhalte**

Das Modul „Rechnernetze“ richtet sich an Studenten, die verstehen wollen, wie Rechnernetze aus technischer Sicht funktionieren und welche Entwicklungen in der Zukunft zu erwarten sind. Es werden einerseits die grundlegenden Fragestellungen des Gebietes untersucht und andererseits besprochen, wie diese Fragestellungen im Internet gelöst sind. Ziel des Moduls ist es, sowohl ein solides allgemeines Basiswissen im Bereich der Rechnernetze als auch praktisch einsetzbare Kenntnisse zu vermitteln. Damit die Lernziele eines Praktikumsversuchs erreicht werden, ist es erforderlich, dass sich die Studierenden inhaltlich vorbereiten. Um dies zu gewährleisten, werden Studierende nur dann zu einem Praktikumsversuch zugelassen, wenn sie vor Versuchsbeginn ein Antestat erfolgreich absolvieren.

#### **Themen der Vorlesung und Übung**

- Einleitung und Übersicht
- Anwendungsschicht
  - World Wide Web / HTTP
  - File Transfer / FTP
  - E-Mail / SMTP
  - Domain Name System / DNS
  - Peer-to-Peer Anwendungen (Guntella/KaZaA/Bittorrent)
  - Socketprogrammierung mit UDP und TCP
- Transportschicht
  - Adressierung
  - UDP
  - Zuverlässige Datenübertragung
  - Überlastkontrolle
  - TCP
- Netzwerkschicht
  - Virtuelle Leitungen und Datagrammnetzwerke
  - Funktionsweise und Aufbau von Routern
  - Adressierung / DHCP
  - Das Internetprotokoll / IP, ICMP
  - Link State Routing, Distance Vector Routing
  - RIP, OSPF, BGP
- Sicherungsschicht
  - Rahmenbildung
  - Fehlererkennung und -korrektur
  - Medienzugriff in Lokalen Netzen
  - Adressierung / ARP
  - Ethernet, Hubs, Switches
  - PPP



- IP over ATM und MPLS
- Multimediakommunikation
  - Anwendungen
  - Streaming
  - Multimedia über Best-Effort Netzwerke
  - Multimediaprotokolle / RTSP, RTP, SIP
  - Differentiated Services
  - Integrated Services
- Netzwerksicherheit
  - Grundlagen der Kryptographie
  - Nachrichtenintegrität
  - Authentifizierung
  - Absichern von E-Mail / PGP
  - Absichern von TCP / SSL, TLS
  - Absichern von IP / IPSec
  - Firewalls und Intrusion Detection Systeme

Themen des Praktikums:

- Netzwerkprogrammierung in Java
- Implementierung eines minimalen Webserver
- Aufsetzen eines komplexen Netzwerkes
- Konfiguration und Erprobung von Firewalls

### **Lernergebnisse/Kompetenzen**

Studierende verstehen nach erfolgreichem Abschluss dieses Moduls die grundlegende Funktionsweise moderner Computernetzwerke. Sie verfügen über grundlegende praktische Fähigkeiten in der Netzwerkprogrammierung und der Administration von Computernetzwerken.

### **Literatur**

Das primäre Lehrbuch zu dieser Veranstaltung ist:

- James F. Kurose und Keith W. Ross: Computer Networking – A Top-Down Approach Featuring the Internet; 6th Edition; Pearson, 2012.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen und im Praktikum
- schriftliche Prüfung (Klausur, i.d.R. 90 Minuten)

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester.

### **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr. Kalman Graffi, Prof. Dr. Martin Mauve

## **5.23 Statistische Datenanalyse**

### **Statistical data analysis**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Statistische Datenanalyse“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

- Grundlagen der schließenden Statistik
- Datenhandling mit R
- statistische Tests
- Regressionsanalyse
- parametrische und nichtparametrische Verfahren
- Planung empirischer Studien

#### **Lernergebnisse/Kompetenzen**

Die Studierenden werden nach der Veranstaltung in der Lage sein:

- grundlegende Konzepte der Datenanalyse zusammenzufassen;
- selbst Auswertung mit R zu planen und durchzuführen
- grafisch aussagekräftige Darstellungen von Daten zu erstellen.

#### **Literatur**

- Peter Dalgaard: Introductory Statistics with R. Springer Verlag, 2008.
- Fahrmeir, Künstler, Pigeot und Tutz: Statistik. Der Weg zur Datenanalyse. Springer Verlag, 2010
- Lothar Sachs und Jürgen Hedderich: Angewandte Statistik. Methodensammlung mit R. Springer Verlag, 2009

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (i.d.R. schriftlich)

#### **Häufigkeit des Angebots, modulare Schiene**

Jedes Studienjahr, in der Regel im Sommersemester

#### **Modulbeauftragte und hauptamtliche Lehrende**

## **5.24 Überblick Künstliche Intelligenz**

### **Overview of Artificial Intelligence**

#### **Studiengang**

Bachelor-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

#### **Lehrveranstaltungen**

- Seminar „Überblick Künstliche Intelligenz“, 2 SWS
- Übung / Vorbereitung, 2 SWS

#### **Inhalte**

Mögliche Inhalte des Seminars, die in Absprache mit den Teilnehmern behandelt werden können sind:

- Grundlagen künstlicher Intelligenz
- Rückblick / Ausblick künstlicher Intelligenz
- Grenzen von KI
- Algorithmen
- Constraint Logic Programming und Constraint Handling Rules
- Inductive / Abductive Logic Programming

#### **Lernergebnisse/Kompetenzen**

Nach erfolgreichem Abschluss des Moduls sollte der Student

- ein grundlegendes Verständnis für Fragestellungen und Probleme aus dem Bereich der künstlichen Intelligenz haben,
- Gängige Algorithmen und Techniken aus dem Bereich kennen,
- Mit Prolog und Erweiterungen wie CLP und CHR umgehen können,
- Techniken des “inductive” und “abductive logic programming” kennen und umsetzen können, und
- Aufgaben der künstlichen Intelligenz in Prolog lösen können.

#### **Literatur**

- Bratko, “Prolog Programming for Artificial Intelligence”, Addison Wesley
- Russel, Norvig, “Artificial Intelligence: A Modern Approach”, Prentice Hall

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

#### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“
- Hilfreich ist erfolgreicher Abschluss des Moduls: "Einführung in die logische Programmierung"

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung im Seminar
- Prüfung am Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **5.25 Verfahren zur kollektiven Entscheidungsfindung Methods for collective decision-making**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

10 LP ab PO 2013, alte PO's 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Verfahren zur kollektiven Entscheidungsfindung“, 4 SWS
- Übung, 2 SWS
- Seminar, 2 SWS

### **Inhalte**

Dieses Modul beschäftigt sich mit verschiedenen Verfahren zur kollektiven Entscheidungsfindung. Mechanismen zur kollektiven Entscheidungsfindung werden in vielen Bereichen der künstlichen Intelligenz, zum Beispiel bei der Interaktion von autonomen Agenten benötigt. Inhalt der Veranstaltung sind verschiedene Verfahren mit ihren axiomatischen und algorithmischen Eigenschaften. Zu den Verfahren gehören unter anderem:

- Judgment Aggregation
- Wahlen
- Argumentationsgraphen

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- kollektive Entscheidungsfindung in unterschiedlichen Situationen durchzuführen
- neue Verfahren zur kollektiven Entscheidungsfindung für spezielle Einsatzbereiche zu entwickeln und im Hinblick auf Ihre Eigenschaften zu untersuchen
- bekannte Verfahren zur kollektiven Entscheidungsfindung auf neue Einsatzbereiche zu übertragen
- verschiedene Verfahren zur kollektiven Entscheidungsfindung gegenüber zu stellen
- Empfehlungen für bestimmte Einsatzbereiche der kollektiven Entscheidungsfindung geben

### **Literatur**

- Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. J. Rothe (ed.). Unter Vertrag bei Springer. Voraussichtliche Publikation: 2015.

Eine kürzere deutsche Version dieses Buches erschien unter dem Titel:

- Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Jörg Rothe, Dorothea Baumeister, Claudia Lindner, Irene Rothe. Spektrum Akademischer Verlag, 2011.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete

- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- Prüfung am Ende der Veranstaltung (i.d.R. Klausur, 90 Minuten)

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul findet in unregelmäßigen Abständen statt.

### **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr. Dorothea Baumeister

## **5.26 Von NAND zu Tetris From NAND to Tetris**

### **Studiengang**

Bachelor-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2013, alte PO's 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Von NAND zu Tetris“, 2 SWS
- Übung, 2 SWS
- Projekt, 2 SWS

### **Inhalte**

Diese Veranstaltung bietet eine Reise durch verschiedene Gebiete der Informatik um dabei ein Gesamtbild über die prinzipielle Funktionsweise von Computern vermitteln. Ausgehend von dem NAND-Gatter werden alle wichtigen Hardware-Komponenten eines Computers in einer Simulation nachgebaut, ALU, CPU, RAM, etc. Für die im Rahmen der Vorlesung entwickelte CPU und die dazugehörige Maschinensprache werden schrittweise eine Assemblersprache, eine virtuelle Maschine und eine Programmiersprache sowie der dazugehörige Compiler entwickelt. Mit all diesen Werkzeugen werden schließlich ein einfaches Betriebssystem und Anwendungsprogramme entwickelt.

Die Vorlesung wird begleitet von Übungen. Die eigenständige praktische Anwendung des Gelernten soll im Fokus der Veranstaltung liegen. Die Studenten entwickeln in den Übungen die diversen in der Vorlesung vorgestellten Komponenten moderner Computer.

### **Lernergebnisse/Kompetenzen**

Der Student soll einen Gesamtüberblick über Funktionsweise eines Computers bekommen und auf jeder Ebene (von der untersten Hardwareebene bis zur höchsten Softwareebene)

- die Grundprinzipien erläutern und bewerten können
- eigenständig neue Funktionalitäten auf dieser Ebene entwickeln zu können.

### **Literatur**

- Noam Nisan, Shimon Schocken "The Elements of Computing Systems: Building a Modern Computer from First Principles", MIT Press.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

### **Teilnahmevoraussetzungen**

Erfolgreicher Abschluss folgender Module:

- „Programmierung“
- „Rechnerarchitektur“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i.d.R. schriftlich)

### **Häufigkeit des Angebots, modulare Schiene**

Jährlich, i.d.R. im Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel, Jens Bendisposto, John Witulski

## **Kapitel 6**

### **6. Bachelor-Arbeit**

#### **Bachelor Thesis**

#### **6.1 Bachelor-Arbeit**

##### **Bachelor Thesis**

##### **Studiengang**

Bachelor-Studiengang Informatik

##### **Leistungspunkte**

15 LP

##### **Lehrveranstaltungen**

—

##### **Inhalte**

Der Inhalt der Bachelor-Arbeit liegt im gewählten Schwerpunktfach. Die Bachelor-Arbeit muss in deutscher oder englischer Sprache verfasst und in einem mündlichen Vortrag präsentiert werden.

##### **Lernergebnisse/Kompetenzen**

Mit der schriftlichen Abschlussarbeit sollen die Studierenden nachweisen, dass sie in der Lage sind:

- innerhalb einer vorgegebenen Frist (von 3 Monaten)
- ein Thema selbstständig zu bearbeiten
- und dieses angemessen darzustellen.

##### **Verwendbarkeit des Moduls**

- Bachelor-Arbeit

##### **Teilnahmevoraussetzungen**

Für die Anmeldung zur Bachelor-Arbeit müssen mindestens 120 der im Rahmen des Bachelor-Studiums zu erwerbenden Leistungspunkte nachgewiesen werden. Das Thema der Bachelor-Arbeit wird aus dem Gebiet des gewählten Schwerpunktfachs vergeben. Dazu sollten üblicherweise alle Module im Schwerpunktfach erfolgreich abgeschlossen sein.

##### **Voraussetzungen für die Vergabe von Leistungspunkten**

Erfolgreiche Bearbeitung des Themas und Darstellung in einer fristgerecht abgegebenen Ausarbeitung (Bachelor-Arbeit); Präsentation in einem mündlichen Vortrag mit Diskussion

##### **Häufigkeit des Angebots, modulare Schiene**

Bei Vorliegen der Voraussetzungen ist der Beginn der Abschlussarbeit jederzeit möglich, d.h. die Vergabe von Themen ist nicht an bestimmte Zeiten im Semester gebunden.

##### **Modulbeauftragte und hauptamtliche Lehrende**

Dozenten der Informatik sowie der als Schwerpunktfach wählbaren mathematisch-naturwissenschaftlichen Fächer





## Kapitel 7

### 7. Lehreinheiten für Wahlpflichtbereiche (M.Sc.) Courses for elective areas (M.Sc.)

Im Master-Studiengang Informatik werden für die Wahlpflichtbereiche „Praktische oder Technische Informatik“, „Theoretische Informatik“, „Individuelle Vertiefung“, und „Schwerpunkt“ Lehreinheiten (Module) unterschiedlicher Größe angeboten. Die Anzahl der Leistungspunkte der Module wurde zum Wintersemester 2015/2016 angepasst. Diese Umstellung erfolgte im Rahmen der Anpassung der Studienordnung für den Master-Studiengang Informatik an die Rahmenstudienordnung für Master-Studiengänge der Mathematisch-Naturwissenschaftlichen Fakultät.

Der Studierende legt zur Prüfungsanmeldung fest, welchem Wahlpflichtbereich er das Modul zuordnen möchte. Die zur Auswahl stehenden Wahlpflichtbereiche eines Moduls sind unter „Verwendbarkeit des Moduls“ aufgeführt.

Die Auswahl der Module für den Wahlpflichtbereich „Schwerpunkt“ muss mit dem Mentor und Betreuer der Master-Arbeit abgesprochen werden (siehe Formular „Schwerpunktwahl“).

Der Schwerpunktbereich „Bioinformatik“ steht sowohl Studierenden offen, die bereits Vorkenntnisse in Bioinformatik mitbringen, als auch solchen, die Informatik im Bachelor-Studium mit einem anderen Schwerpunkt als Bioinformatik studiert haben. Daher wird die Lehrveranstaltung „Algorithmen in der Bioinformatik“ aus dem Bachelor-Studiengang auch für Master-Studierende ohne Vorkenntnisse in Bioinformatik angeboten. Eine Differenzierung zwischen Master- und Bachelor-Studierenden wird über eine gesonderte Seminaraufgabe vorgenommen.

Die Unterrichtssprache ist entweder Deutsch oder Englisch. Werden in den Modulbeschreibungen keine Angaben zur Unterrichtssprache gemacht, so ist die Unterrichtssprache Deutsch. Für ein erfolgreiches Studium sind Englischkenntnisse jedoch unumgänglich. Auch wenn Veranstaltungen in deutscher Sprache gehalten werden, sind die verwendeten schriftlichen Materialien häufig in englischer Sprache verfasst.

Mündliche Prüfungen werden in deutscher oder englischer Sprache durchgeführt. Englisch kann jedoch nur dann als Prüfungssprache gewählt werden, wenn alle an der Prüfung beteiligten Personen einverstanden sind.

Die schriftlichen Abschlussarbeiten werden in deutscher oder englischer Sprache verfasst.

Anmerkung: Der Prüfungsausschuss hat am 28.01.2008 folgende Regelung für Studierende des Bachelor-Studiengangs Informatik, die Module aus dem Master-Programm als Zusatzfach belegen wollen, beschlossen:

Formale Voraussetzung für die Teilnahme an Lehrveranstaltungen zu Modulen, die nur für den Master-Studiengang angeboten werden, ist für Bachelor-Studierende der erfolgreiche Abschluss der Module „Programmierung“, „Rechnerarchitektur“, „Algorithmen und Datenstrukturen“ sowie „Theoretische Informatik“.

#### 7.1 Algebraische und Strukturelle Graphentheorie Algebraic and Structural Graph Theory

##### Studiengang

Master-Studiengang Informatik

##### Leistungspunkte

5 LP

##### Lehrveranstaltungen

- Vorlesung „Algebraische und Strukturelle Graphentheorie“, 2 SWS

- Übungen, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 2 SWS

### **Inhalte**

Dieses Modul vermittelt Kenntnisse aus folgenden Bereichen:

- Eigenschaften von Adjazenzmatrizen
- Graphhomomorphismen
- Strukturelle Ähnlichkeit von Graphen
- Normalformen von Graphen
- Graphen über algebraischen Strukturen
- Kombinatorische Grapheigenschaften

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend die Fähigkeit, Struktureigenschaften von Graphen und Graphklassen mittels kombinatorischer und algebraischer Methoden zu analysieren.

### **Literatur**

Aktuelle Lehrbücher über Graphentheorie sowie Originalliteratur

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Teil eines Master-Moduls im Master-Studiengang Biologie

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Graphenalgorithmen I“
- „Lineare Algebra I“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

schriftliche Prüfung (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 30 Minuten) am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Stefan Hoffmann

## **7.2 Algorithmen für Ad-hoc- und Sensornetzwerke Algorithms for Ad-hoc and Sensor networks**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

10 LP ab PO 2015, alte PO 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmen für Ad-hoc- und Sensornetzwerke“, 4 SWS
- Übung, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 4 SWS

### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen.

- Verteilte Algorithmen
- Leader Election
- Geographisches Routing
- Topologiekontrolle
- Standortbezogene Dienste
- Standortbestimmung
- Greedy-Einbettungen
- Beacon-Routing
- Interval-Routing
- Hop-Netzwerke
- Network Coding
- Dominierende Mengen
- Maximale unabhängige Mengen

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein breites und vertieftes Verständnis im Bereich „Algorithmen für Ad-hoc- und Sensornetzwerke“.

### **Basisliteratur**

Verschiedene aktuelle Lehrbücher über Ad-hoc- und Sensornetzwerke sowie Originalliteratur

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“
- „Algorithmische Komplexitätstheorie“
- „Graphenalgorithmen I“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- 10-minütiger Lehrvortrag über ein Thema der Vorlesung
- schriftliche Klausur (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 45 Minuten) am Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

Jedes zweite Sommersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Egon Wanke

## **7.3 Algorithmen für planare Graphen Algorithms for planar graphs**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

10 LP ab PO 2015, alte PO 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmen für planare Graphen“, 2 SWS
- Seminar, 2 SWS
- Übung, 2 SWS

### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen.

- Grundlagen zu planaren Graphen
- Planaritätstests
- Zeichnen von planaren Graphen
- Knoten- und Kantenfärbungen für planare Graphen
- Flüsse in planaren Graphen

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, sollen anschließend ein breites und vertieftes Verständnis im Bereich „Algorithmen für planare Graphen“ besitzen. Weiterhin sollen die Studierenden nach Absolvierung der Lehrveranstaltungen in der Lage sein, einen guten Vortrag vorzubereiten und zu halten und eine gute schriftliche Ausarbeitung zu dem Vortragsthema anzufertigen.

### **Literatur**

- T. Nishizeki, N. Chiba: Planar Graphs: Theory and Algorithms, Dover Publ Inc, 2008.
- T. Nishizeki, Md. S. Rahman: Planar Graph Drawing, World Scientific Publishing, 2004.
- R. Diestel: Graph Theory, Springer, 2010.
- G. Chartrand, L. Lesniak, P. Zhang: Graphs & Digraphs, CRC Press, 2010.
- R. Tamassia (Editor): Handbook of Graph Visualization, CRC Press, 2014.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung von Übungsaufgaben
- Aktive Mitarbeit im Seminar
- Schriftliche Ausarbeitung eines Seminarthemas
- Vortrag eines Seminarthemas
- Bestehen der Klausur

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **7.4 Algorithmen für schwere Probleme**

### **Algorithmics for hard problems**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Algorithmen für schwere Probleme“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

Dieses Modul befasst sich mit schweren Problemen, für die es vermutlich keine effizienten Algorithmen gibt, und stellt unter anderem folgende Lösungsansätze für solche Probleme vor.

- Pseudopolynomielle Algorithmen
- Algorithmen auf speziellen Graphen
- Parametrisierte Algorithmen
  - Standardparameter
  - Parametrisierung in der Baumweite
  - Parametrisierung in der Cliquesweite
- Exakte Exponentialzeit Algorithmen

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen schweren Probleme zu erläutern und formal zu definieren
- die behandelten Algorithmen für schwere Probleme zu verstehen und auf konkrete Eingaben anzuwenden
- eine Parametrisierte Laufzeitanalyse von Algorithmen durchzuführen
- die betrachteten Baumstrukturen für gegebene Instanzen zu konstruieren

#### **Literatur**

- F. Gurski, I. Rothe, J. Rothe, E. Wanke, Exakte Algorithmen für schwere Graphenprobleme, Springer Verlag, 2010.
- F.V. Fomin, D. Kratsch: Exact Exponential Algorithms, Springer Verlag, 2010.
- J. Hromkovic: Algorithmics for Hard Problems, Springer Verlag, 2003.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich

- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Abgabe der Übungsaufgaben, Bestehen der Klausur zum Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **7.5 Algorithmen in der Cheminformatik** **Algorithms in cheminformatics**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmen in der Cheminformatik“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

- Che
- mische & algorithmische Grundlagen
- Algorithmen auf molekularen Graphen
- Molekulare Deskriptoren
- Algorithmische Massenspektrometrie
- Monte-Carlo Simulationen
- Molekulardynamik

### **Lernergebnisse/Kompetenzen**

Nach der Veranstaltung werden die Studierenden ein vertieftes Verständnis ausgesuchter grundlegender Techniken der Informatik und Mathematik besitzen und in der Lage sein

- klassische Algorithmen und spezielle cheminformatische Algorithmen auf (bio-)chemische Probleme anzuwenden;
- gelernte Algorithmen exemplarisch in der Programmiersprache Python umzusetzen;
- cheminformatische Software-Libraries anzuwenden;
- cheminformatische Probleme mit den vorgestellten Algorithmen selbständig lösen zu können.

### **Empfohlene Literatur**

- Leach & Gillet: An Introduction to Chemoinformatics, Springer, 2007
- Leach: Molecular Modeling. Principles and Applications, Prentice Hall, 2nd ed., 2001

- Jensen: Introduction to Computational Chemistry, Wiley, 2007

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i.d.R. schriftlich)

### **Empfohlene Vorkenntnisse**

- „Algorithmen in der Bioinformatik“

### **Häufigkeit des Angebots, modulare Schiene**

Unregelmäßig

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Gunnar W. Klau

## **7.6 Algorithmische Eigenschaften von Wahlsystemen II Algorithmic Properties of Voting Systems II**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Algorithmische Eigenschaften von Wahlsystemen II“, 2 SWS
- Übung, 1 SWS

### **Inhalte**

- Grundlagen der Social-Choice-Theorie
  - Wiederholung: Wahlen und Wahlsysteme
  - Wiederholung: Eigenschaften von Wahlsystemen und einige Wahl-Paradoxa und Unmöglichkeitssätze
- Algorithmik und Komplexität von Wahlproblemen
  - Wiederholung: Gewinnerbestimmung, mögliche und notwendige Gewinner und Manipulation
  - Wahlkontrolle
  - Bestechung

### **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem jungen, sich rasant entwickelnden, interdisziplinären Gebiet *Computational Social Choice* zu vermitteln, welches derzeit in der Abteilung „Komplexitätstheorie und Kryptologie“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird und in dem bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende die wichtigsten Wahlverfahren und ihre „social-choice“-theoretischen Eigenschaften sowie die damit im Zusammenhang stehenden Entscheidungsprobleme (Wahlkontrolle, Bestechung usw.) und die zu ihrer Lösung entwickelten Algorithmen und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Ebenso sollten sie die Komplexität dieser Probleme formal bestimmen und beschreiben können.

### Empfohlene Literatur

- Jörg Rothe (ed.): *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: *Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen*, Spektrum Akademischer Verlag (Springer), 2011.

### Ergänzende Literatur

- Piotr Faliszewski, Lane A. Hemaspaandra, Edith Hemaspaandra, and Jörg Rothe: A Richer Understanding of the Complexity of Election Systems, Chapter 14 in: „Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz,” pp. 375-406, S. Ravi and S. Shukla (eds.), Springer, 2009.
- Piotr Faliszewski und Jörg Rothe: Control and Bribery in Voting, Chapter 7 in: „Handbook of Computational Social Choice,” Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel Procaccia (eds.), Cambridge University Press, 2015.
- Steven J. Brams and Peter C. Fishburn: Voting Procedures, Chapter 4 in: „Handbook of Social Choice and Welfare,” Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura (eds.), North-Holland, 2002.

### Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### Teilnahmevoraussetzungen

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### Voraussetzungen für die Vergabe von Leistungspunkten

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### Häufigkeit des Angebots, modulare Schiene

Etwa alle vier Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### Modulbeauftragte und hauptamtliche Lehrende

Prof. Dr. Jörg Rothe

## 7.7 Algorithmische Spieltheorie Algorithmic Game Theory



## Studiengang

Master-Studiengang Informatik

## Leistungspunkte

10 LP ab PO 2015, alte PO 15 LP

## Lehrveranstaltungen

- Vorlesung „Algorithmische Spieltheorie“, 4 SWS
- Übung, 2 SWS

## Inhalte

- Nichtkooperative Spiele: Gegeneinander spielen
  - Grundlagen
    - Normalform, dominante Strategien und Gleichgewichte
    - Weitere Zwei-Personen-Spiele
  - Nash-Gleichgewichte in gemischten Strategien
    - Definition und Eigenschaften gemischter Nash-Gleichgewichte
    - Existenz von Nash-Gleichgewichten in gemischten Strategien
  - Schachmatt: Spielbäume in Spielen mit perfekter Information
    - Sequenzielle Zwei-Personen-Spiele
    - Gleichgewichte in Spielbäumen
  - Full House: Spiele mit unvollkommener Information
    - Das Ziegenproblem
    - Analyse einer einfachen Poker-Variante
  - Wie schwer ist es, ein Nash-Gleichgewicht zu finden?
    - Nash-Gleichgewichte in Nullsummenspielen
    - Nash-Gleichgewichte in allgemeinen Normalform-Spielen
- Kooperative Spiele: Miteinander spielen
  - Grundlagen
    - Kooperative Spiele mit übertragbarem Gewinn
    - Superadditive Spiele
    - Stabilitätskonzepte für kooperative Spiele
  - Einfache Spiele
    - Der Kern einfacher Spiele
    - Darstellungen einfacher Spiele
    - Gewichtete Wahlspele
    - Dimensionalität
    - Machtindizes
    - Der Shapley-Shubik-Index und der Shapley-Wert
    - Die Banzhaf-Indizes
  - Komplexität von Problemen für kompakt darstellbare Spiele
    - Spiele auf Graphen
    - Gewichtete Wahlspele
    - Hedonische Spiele

## Lernergebnisse/Kompetenzen

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem Gebiet der algorithmischen Spieltheorie zu vermitteln, welches derzeit in der Abteilung „Komplexitätstheorie und Kryptologie“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird und in dem bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende wichtige Probleme und die zu ihrer Lösung entwickelten Algorithmen und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Insbesondere sollten sie in der Lage sein, „strategische Szenarien“ durch kooperative bzw. nichtkooperative Spiele zu beschreiben und Stabilitäts- und Gleichgewichtskonzepte in diesen Spielen formal zu charakterisieren. Ebenso sollten sie die Komplexität der für solche Spiele relevanten Entscheidungsprobleme (in einer geeigneten kompakten Darstellung) formal bestimmen und beschreiben können.

## Empfohlene Literatur

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen, Spektrum Akademischer Verlag (Springer), 2011.

### **Ergänzende Literatur**

- Bezalel Peleg and Peter Sudhölter: Introduction to the Theory of Cooperative Games, Kluwer Academic Publishers, 2003.
- Martin J. Osborne and Ariel Rubinstein: A Course in Game Theory, MIT Press, 1994.
- Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge: Computational Aspects of Cooperative Game Theory, Morgan and Claypool Publishers, 2011.
- Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani (eds.): Algorithmic Game Theory, Cambridge University Press, 2008.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle zwei Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **7.8 Android-Programmierung Android programming**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Android-Programmierung“, 2 SWS
- Seminar/Praktische Übung, 2 SWS

### **Inhalte**

- Java
- Grundgerüst einer Android-Anwendung
- Activities & Fragments
- Grafik & Animation
- Eingabeereignisse
- Persistenz und Speicherverwaltung
- Intents & Broadcast Receivers
- Netzwerk
- Native-C (Subkomponente?)
- Erarbeitung eines Anwendungskonzeptes
- Packages (.apk), Libraries (.aar)
- In-App-Purchases

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- mit Hilfe von Java (und Native-C) Android-Anwendungen zu programmieren
- Android Studio als Entwicklungsumgebung anzuwenden
- Anwendungskonzepte zu erarbeiten und umzusetzen
- grundlegende Netzwerkkommunikation zu programmieren
- mit den gewonnenen Kenntnissen das Software-Projekt zu konzipieren und zu realisieren

### **Literatur**

- Zigurd Mednieks et al., "Android Programmierung", O'Reilly, 2013
- Arno Becker, Marcus Pant, "Android - Grundlagen und Programmierung", dpunkt, 2010
- Android Developer Homepage

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung der Pflichtübungen
- Erfolgreiche Entwicklung einer Android-Anwendung
- Bestehen der Klausur

### **Häufigkeit des Angebots, modulare Schiene**

Alle 2 Jahre

### **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr. Kálmán Graffi und Mitarbeiter

## **7.9 Angewandte Bioinformatik**

### **Applied bioinformatics**

#### **Studiengang**

## **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

## **Lehrveranstaltungen**

- Vorlesung „Angewandte Bioinformatik“, 2 SWS
- Praktikum, 4 SWS

## **Inhalte**

- Python mit Fokus auf Anwendungen in bioinformatischer “Data Science”
- Workflowmanagement: Snakemake
- Analyse von High-Throughput-Sequencing Daten
- Genome Assembly
- Read Mapping
- Analyse biologischer Netzwerke
- Bestimmung evolutionär verwandter Gene (Orthologe)
- Berechnung von Stammbäumen
- Identifizierung von Genen, die Zeichen natürlicher Auslese zeigen (dN/dS)
- Modellierung der Evolution von DANN-Sequenzen

## **Lernergebnisse/Kompetenzen**

Die Studierenden verstehen die Grundlagen der vergleichenden Genomik und sind in der Lage, eigenständig Analysen von DNA-Sequenzdaten durchzuführen. Die Studierenden verstehen zugrundeliegende Konzepte fortgeschrittener bioinformatischer Analyse-Pipelines und können diese anwenden. Sie sind in der Lage, eigene Schritte dieser Pipelines zu implementieren wie zum Beispiel einen eigenen, einfachen Assembler. Sie verstehen die Herausforderungen der Anwendungen und die Grenzen der bestehenden Ansätze.

## **Basisliteratur**

Wird im Rahmen der Vorlesung vorgestellt

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich
- Schwerpunktbereich
- Individuelle Ergänzung im Master-Studiengang Informatik
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Regelmäßige und aktive Teilnahme am Praktikum
- Rechtzeitige Abgabe von erstellten Programmen / Protokollen, die den Anforderungen wissenschaftlicher Arbeit entsprechen
- Bestehen der Abschlussprüfung

## **Häufigkeit des Angebots, modulare Schiene**

Jährlich im Sommersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Gunnar W. Klau, Prof. Dr. Martin Lercher

## **7.10 Approximationsalgorithmen**

### **Approximation Algorithms**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Approximationsalgorithmen“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

Dieses Modul befasst sich mit den folgenden schweren Optimierungsproblemen, für die es vermutlich keine effizienten Algorithmen gibt, und stellt verschiedene Ansätze zum Finden von Näherungslösungen für diese Probleme vor.

- Grundlagen
- Metric Traveling Salesman
- Job Scheduling
- Knapsack
- Bin Packing
- Steiner Tree
- Weighted Vertex Cover

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen schweren Optimierungsprobleme zu erläutern und formal zu definieren
- die behandelten Approximationsalgorithmen zu verstehen und auf konkrete Eingaben anzuwenden
- die Optimierungsprobleme bekannten Komplexitätsklassen zuzuordnen
- Beweise für die Approximationsgüte von Approximationsalgorithmen zu durchschauen

#### **Literatur**

- G. Ausiello, P. Crescenzi, G. Gambosi, et al.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties, Springer Verlag, 1999.
- K. Jansen, M. Margraf: Approximative Algorithmen und Nichtapproximierbarkeit, de Gruyter Verlag, 2008.
- V. Vazirani: Approximation Algorithms, Springer Verlag, 2003.
- R. Wanka: Approximationsalgorithmen, Teubner Verlag, 2006.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Abgabe der Übungsaufgaben, Bestehen der Klausur zum Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **7.11 Architektur Verteilter Systeme Architecture of Distributed Systems**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Architektur Verteilter Systeme“, 2 SWS
- Seminar, 2 SWS

### **Inhalte**

- Cluster-, Grid- und Cloud-Computing
- P2P-Systeme: Gnutella
- DHTs: Chord, CAN, Pastry
- Multicast: atomarer Multicast, Scribe, SplitStream, Pub/Sub
- Konsensus: Flooding, Paxos, BFT, Dienste (Zookeeper, Chubby Lock)
- P2P-Dateisysteme: CFS, IVY, OceanStore
- Cloud-Storage: HDFS, GoogleFS, Big Table, RAMcloud

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein, folgende Inhalte in eigenen Worten zu erklären und zu bewerten:

- die Architektur von und die Algorithmen in skalierbaren und fehlertoleranten Cloud-Systemen
- P2P-Konzepte und deren Anwendung in Cloud-Systemen
- Replikation, Gruppenkommunikation und Konsensus in verteilten Systemen

### **Literatur**

- Coulouris et. al., „Distributed Systems“, 5. Auflage, 2012.
- K. Birman, „Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services“, Springer, 1. Auflage, 2012.
- Weitere relevante Publikationen werden in der Vorlesung bekannt gegeben

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Empfohlene Vorkenntnisse**

- „Grundlagen Verteilter Systeme“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme am Seminar
- Abschließende Prüfung am Ende des Semesters

#### **Häufigkeit des Angebots, modulare Schiene**

Wird nicht mehr angeboten

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **7.12 Betriebssystem-Entwicklung Operating System Development**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Betriebssystementwicklung“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

Die Lehrveranstaltung vermittelt grundlegende Konzepte, die für den Bau eines Betriebssystems erforderlich sind. In der vorlesungsbegleitenden Übung werden diese Kenntnisse praktisch angewendet, indem ein einfaches PC-Betriebssystem von Grund auf entwickelt wird. Hierzu sind fundierte Kenntnisse über Aufbau und Funktionsweise der PC-Hardware erforderlich, die ebenfalls in der Lehrveranstaltung vermittelt werden. Angesprochen werden zum Beispiel das Schutzkonzept der IA-32 Architektur, Speicherverwaltung, Interrupt-Verarbeitung und Multi-Threading mit Synchronisierung.

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Programmiersprache C++ anzuwenden
  - grundlegende Betriebssystemfunktionen selbst zu entwickeln
  - die Abläufe von Scheduler, Threads, Interrupts und Synchronisierung zu implementieren
- Jeder(e) Teilnehmer(in) entwickelt (stufenweise, mit Vorgaben) sein/ihr eigenes Betriebssystem.

### **Literatur**

- H. P. Messmer und K. Dembowski, PC-Hardwarebuch, Addison-Wesley 2006
- Intel Corporation, Intel Architecture Software Developer's Manual.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung der Übungen und Abgabe eines nativ lauffähigen Betriebssystems
- Abschließende Prüfung am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Circa alle zwei Jahre

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **7.13 Betriebssysteme Vertiefung Advanced Operating Systems**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

10 LP

### **Lehrveranstaltungen**

- Seminar „Betriebssysteme Vertiefung“, 2 SWS
- Praktische Übungen, 6 SWS

### **Inhalte**

Dieses Modul baut auf dem eigenen Betriebssystem HHUos auf, welches im Modul „Betriebssystem-Entwicklung“ entwickelt wird. HHUos ist ein natives Betriebssystem für die IA32-Architektur. Die Studierenden entwickeln eine oder mehrere fortführende Komponenten für HHUos in C++/C/Assembler. Diese Komponenten sollen am Ende in HHUos integriert und dokumentiert werden. Im Rahmen des wöchentlichen Seminars werden Konzepte und Entwicklungen diskutiert, sowohl aus der Literatur, als auch eigene Ideen.

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Betriebssystem-Konzepte in eigenen Worten zu erklären
- die Programmiersprache C++/C/Assembler für IA32(e) anzuwenden
- fortführende Betriebssystemfunktionen und Hardware-Treiber selbst zu entwickeln

### **Literatur**

- H. P. Messmer und K. Dembowski, PC-Hardwarebuch, Addison-Wesley 2006
- Intel Corporation, Intel Architecture Software Developer's Manual
- Weitere Literatur wird in der Veranstaltung bekannt gegeben

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung



- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Nur für Master-Studierende:

- „Betriebssystem-Entwicklung“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Entwicklung nativer Betriebssystem-Komponenten
- Aktive und regelmäßige Teilnahme am wöchentlichen Seminar
- Abschließende mündliche Prüfung am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Circa alle zwei Jahre

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **7.14 Big Data Techniques, Technologies and Trends**

### **Big Data Techniques, Technologies and Trends**

#### **Course of studies**

Master of Computer Science

#### **Credit points**

5 LP for all PO's

#### **Course**

- Lecture „Big Data Techniques, Technologies and Trends“, 2 SWS
- Hands-on exercises, 2 SWS

#### **Content description**

Course description: Big Data is one of the main buzz words nowadays, being a primary focus both for academic research and for industry. Big Data has emerged as a revolution driven by the continuous increasing volumes of data that are being collected at increasing velocities from various source: social networks, IoT, scientific simulations, finance, weather forecasting, etc. Tackling such challenges, commonly referred to as the V's of Big Data, has lead to the development of a plethora of technologies and concepts. Batch and stream processing are the main classes of dealing with the data, which can be either offline or in real time. Starting from these two categories, different programming models such as MapReduce or reactive programming have been recently proposed. Additionally multiple technologies have been, and are developed to facilitate the processing and the data management of Big Data scenarios: HDFS, MapReduce, Spark, Storm, Flink, Kafka, HBase, Hive, etc. All these form today the Hadoop ecosystem. This course aims to give an introduction to technologies and concepts that build the Hadoop ecosystem, both as lecture courses and practical sessions. From the point of view of the lecture courses the focus lays with giving the theoretical backgrounds of the concepts and mechanisms that enable Big Data processing. The course will present the different programming models, strategies to deal with large data sets or with data sets on the fly (e.g., MapReduce and MapReduce pipelines, Stream topologies, Windows, SQL and Hive Queries and interactive queries). From the point of view of the practical sessions the objective is to make the students familiar with the main Big Data processing tools used today in industry such as MapReduce, HDFS, Spark, Flink, HBase, Kafka. At the end of the course the students will have a good understanding of feasible approaches to address various Big Data scenarios as well as hands-on experience with some of the most commonly used Hadoop tools.

Course Topics to be addressed:

- Overview of Big Data: what it is, why it has emerged and future trends
- Data models and large scale infrastructures (cluster, grid, cloud, HPC)
- Batch processing

- Distributed storage systems concepts: GFC, HDFS and Cloud Public Storage (Azure Blobs and AWS S3)
  - NoSQL storage and distributed message queues
  - Google MapReduce programming model and Hadoop MapReduce
  - High level semantics processing tools for offline data: Spark, Hive, Pig, Flink
- Stream processing:
  - Stream overview: what it is and what are the main difference with respect to batch processing,
  - Stream concepts for data processing: operators, windows, sinks, ETLs
- Project topics

#### Evaluation:

- Project: A topic will be chosen from multiple available ones (sentiment analysis, twitter trends analysis, internet/social media search, ...)
- Solution: A software solution will be designed, built and delivered as the outcome of the project.
- Technology: The solution will be built using multiple advanced technologies covered in the course.
- Evaluation: The solution design will be presented together with a demo to show the specific use case.

#### Study results/competences

The result after completion of this course is that the students will:

- Have an overview of the principles of Big Data analytics
- Have an understanding of the data analytics ecosystem
- Have knowledge about the Big Data technologies most used in industry and research
- Have practical experience with Big Data tools from the Hadoop ecosystem, which will give competitive advantage for getting jobs in the domain
- Have a reference project in the area of Big Data that they can showcase in the future to prove their practical experience for industry

#### Literature

Literature will be given during the lecture

#### Module usage

- Compulsory module in the area of practical and technical computer science
- As focus module
- Individual complementary module
- Application module for the complementary area in the Master studies mathematics

#### Prerequisites

Bachelor students must have passed the following modules:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### Requirements for credit points

- Successful participation in hands-on exercises
- Submission of a final software project (base for grade)

#### Frequency of offering

Unregular

#### Lecturer

Dr. Radu Tudoran (Huawei, Munich)

## 7.15 Cake-cutting Algorithms

### Cake-cutting Algorithms

## Studiengang

Master-Studiengang Informatik

## Leistungspunkte

5 LP ab PO 2015, alte PO 7,5 LP

## Lehrveranstaltungen

- Vorlesung „Cake-cutting Algorithms“, 2 SWS
- Übung, 1 SWS

## Inhalte

- Grundlagen
- Bewertungskriterien
  - Fairness
  - Effizienz
  - Manipulation
  - Laufzeit
- Cake-cutting-Protokolle
  - Zwei neidfreie Protokolle für zwei Spieler
  - Proportionale Protokolle für beliebig viele Spieler
  - Überproportionale Protokolle für beliebig viele Spieler
  - Eine Hochzeitsfeier im Königshaus: Aufteilung in ungleiche Anteile
  - Neidfreie Protokolle für drei und vier Spieler
  - Versalzene Sahnetorte: Dirty-Work-Protokolle
  - Gekrümme vermeiden: Minimierung der Schnittanzahl
  - Der Grad der garantierten Neidfreiheit
  - Übersicht über einige Cake-cutting-Protokolle

## Lernergebnisse/Kompetenzen

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem Gebiet *Fair Division* zu vermitteln, welches derzeit in der Abteilung „Komplexitätstheorie und Kryptologie“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird und in dem bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende die wichtigsten Eigenschaften von gerechten Aufteilungsverfahren für eine teilbare Ressource (Proportionalität, Neidfreiheit usw.) kennen und die dafür entwickelten Protokolle und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Insbesondere sollten sie in der Lage sein, über die Eigenschaften von Verfahren zur gerechten Aufteilung formal zu argumentieren und selbstständig solche Verfahren zu entwerfen und zu bewerten.

## Empfohlene Literatur

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer-Verlag, 2015.  
*Eine kürzere deutsche Version dieses Buches erschien als:*
- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen, Spektrum Akademischer Verlag (Springer), 2011.

## Ergänzende Literatur

- Jack Robertson and William Webb: Cake-Cutting Algorithms: Be Fair if You Can, A K Peters, 1998.
- Steven J. Brams and Alan D. Taylor: Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.

## Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich

- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle zwei Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **7.16 Causality**

### **Causality**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Causality“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

Dieses Modul vermittelt grundlegendes Wissen zu folgenden Themen:

- Directed acyclic graphs, causal graphs
- Conditional independence
- PC algorithm
- Structural equation models
- Additive noise models
- Interventions
- Counterfactuals
- Markov equivalence
- Faithfulness
- Distinguishing cause and effect

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- die Grundlagen der kausalen Inferenz zu beschreiben und anzuwenden
- die mathematische Beschreibung dieser Grundlagen und verschiedene Algorithmen zu erklären und Sachverhalte darüber zu beweisen

- die Grundlagen und verschiedene Algorithmen anzuwenden, um selbstständig Problemstellungen der kausalen Inferenz zu bearbeiten.

### **Empfohlene Literatur**

Es wird nicht ein bestimmtes Lehrbuch verwendet, jedoch sind folgende Bücher hilfreich:

- Spirtes/Glymour/Scheines, Causation, Prediction, and Search, MIT 2000
- Pearl: Causality, Cambridge 2000

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung in den Übungen;
- Prüfung zu Vorlesung und Übungen am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Harmeling

## **7.17 Computational Systems Biology**

### **Computational Systems Biology**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Computational Systems Biology“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

- What is in a cell: Genome, proteome, metabolome.
- Cell growth - Mathematical modeling and numerical solution (Ingalls, Ch. 2).
- Cost-benefit analysis of protein expression (Alon, Ch. 10).
- Competition of energy and entropy in protein-DNA binding (Phillips, Sec. 6.1.2).
- Information content of DNA sequences recognized by proteins.
- Operator occupancy models of transcription regulation (Phillips, Sec. 19.2).
- Logic gates and Boolean network analysis of genetic circuits (Kauffman, Chs. 4 & 5).
- Kinetic modeling of gene regulatory networks (Alon, Chs. 2 & 3).

- Network motifs in genetic control circuits (Alon, Chs. 4 & 5).
- Genetic switches and oscillators (Ingalls, Ch. 7).
- First look at metabolic networks (Ingalls, Ch. 5).

### **Lernergebnisse/Kompetenzen**

After successfully completing the course students will be able to

- reason about fundamental biological processes, such as protein-DNA binding, gene expression, and metabolic transformations in terms of the relevant spatial, temporal and energetic scales;
- identify simple motifs of elementary transformations in given genetic and metabolic networks;
- model a given genetic control architecture as a Boolean logic gate;
- formulate simple physical/mathematical models of gene-expression and metabolic networks in terms of ordinary differential equations;
- numerically simulate and analyze these models using Python;
- relate the outcomes of these analyses to experimental measurements by generating meaningful scientific plots.

### **Literatur**

In addition to the original research literature that will be given in the lectures, some chapters of the following textbooks will be covered in detail.

- Uri Alon: An Introduction to Systems Biology - Design Principles of Biological
- Circuits, Chapman & Hall/CRC, 2007.
- Brian P. Ingalls: Mathematical Modeling in Systems Biology - An Introduction, MIT Press, 2013.
- Rob Phillips, Jane Kondev, Julie Theriot, Hernan G. Garcia: Physical Biology of the Cell, Garland Science, 2nd edition, 2016.
- Stuart Kauffman: At Home in the Universe, Oxford University Press, 1995.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung in den Übungen
- Midterm exam
- End-of-term presentation
- Final exam

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Studienjahr, in der Regel im Wintersemester.

### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Deniz Sezer

## **7.18 Computer Vision**

### **Computer Vision**

## **Studiengang**

Master-Studiengang Informatik

## **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

## **Lehrveranstaltungen**

- Vorlesung „Computer Vision“, 2 SWS
- Übung, 2 SWS

## **Inhalte**

Dieses Modul vermittelt grundlegendes Wissen zu folgenden Themen:

- Image formation
- Image processing
- Matting, segmentation
- Local features, e.g. SIFT
- Computational photography
- Face detection

## **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- die Grundlagen der Computer Vision zu beschreiben und anzuwenden
- die mathematische Beschreibung dieser Grundlagen und verschiedene Algorithmen zu erklären
- die Grundlagen und verschiedene Algorithmen anzuwenden, um selbstständig Problemstellungen der Computer Vision zu bearbeiten.

## **Empfohlene Literatur**

- Szeliski, Computer vision, Springer 2010

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive und erfolgreiche Mitwirkung in den Übungen;
- Prüfung zu Vorlesung und Übungen am Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich.

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Harmeling

## **7.19 Data Warehouses**

### **Data Warehouses**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Data Warehouses“, 2 SWS
- Theoretische und Praktische Übungen, 2 SWS

#### **Inhalte**

- Anwendungsbereiche
- Data Warehouse–Architektur
- Multidimensionales Datenmodell, Entwurf von Data Warehouses (MOLAP vs. ROLAP; Star- oder Snowflake-Schema)
- Anfragebearbeitung und -optimierung
- Materialisierung von Aggregationssichten
- multidimensionale Index- und Speicherstrukturen
- Extraktion, Transformation, Laden; Data Cleaning
- OLAP und Data Mining

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Architektur eines Data Warehouse-Systems und ihre Komponenten zu erläutern
- relationale Data Warehouses zu entwerfen
- für ein zu entwickelndes Data Warehouse zu entscheiden, welche Indexstrukturen genutzt werden können
- zu entscheiden, welche Sichten materialisiert werden sollten

#### **Empfohlene Literatur**

- W. Lehner: Datenbanktechnologie für Data-Warehouse-Systeme. dpunkt-verlag, 2003.
- H. Bauer, A. Günzel (Hrsg.): Data Warehouse Systeme. 4. Aufl.; dpunkt-verlag, 2013

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Empfohlene Vorkenntnisse**

- „Datenbanksysteme“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben



- abschließende Prüfung (i.d.R. mündlich)

#### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig
- wird im Wechsel mit den Modulen/Veranstaltungen „Knowledge Discovery in Databases“, „Multimedia-Datenbanksysteme“, „Knowledge Discovery in Databases – Ausgewählte Kapitel“, „Transaktionsverwaltung“ und „Information Retrieval und Natural Language Processing“ angeboten, so dass jedes Semester mindestens eine dieser Veranstaltungen angeboten wird.

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

## **7.20 Dynamische Programmiersprachen Dynamic Programming Languages**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Dynamische Programmiersprachen“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

Diese Vorlesung vermittelt typische Konzepte und Eigenschaften dynamischer Programmiersprachen, z.B. Typisierung und Duck-Typing, Metaprogrammierung u.a. Darüber hinaus wird behandelt wie ein Interpreter für eine dynamische Programmiersprache implementiert wird, um die zuvor behandelten Konzepte zu unterstützen.

In den praktischen Übungen sollen die Studierenden diverse Probleme mit Hilfe der Eigenschaften einer dynamischen Programmiersprache lösen, um ein tieferes Verständnis dieser Konzepte zu erlangen. In der zweiten Semesterhälfte besteht die Aufgabenstellung in der Übung darin, selbständig einen Interpreter für eine vorgegebene dynamische Programmiersprache zu schreiben.

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Prinzipien der dynamischen Programmierung bezeichnen können und klassischen, imperativen Programmiersprachen gegenüberstellen können
- bewerten für welche Einsatzbereiche die dynamische Programmierung vorteilhaft ist
- eigenständig funktionale Programme erstellen und testen können
- selbständig einen Interpreter für eine vorgegebene dynamische Programmiersprache zu schreiben

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“

- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

Grundlagen des Compilerbaus (zum Beispiel aus dem Modul "Compilerbau", ehemals "Softwaretechnik und Programmiersprachen").

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung der Pflichtübungen
- Erfolgreiche Entwicklung eines eigenen Interpreters
- Bestehen der Klausur

### **Häufigkeit des Angebots, modulare Schiene**

Jährlich, in der Regel jedes Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **7.21 Effiziente Algorithmen Efficient Algorithms**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Effiziente Algorithmen“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

In diesem Modul werden Algorithmen für effizient lösbare Probleme betrachtet. Anhand fundamentaler Verfahren werden die Prinzipien des Entwurfs und der Analyse von Algorithmen erklärt. Hierzu werden folgende Themen betrachtet.

- Algorithmenentwurfstechniken
- Algebraische und Zahlentheoretische Probleme
- Sortiernetze
- Sortieren auf Prozessorfeldern
- Lineare Programmierung

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen Probleme zu erläutern und formal zu definieren
- die behandelten Algorithmen zu verstehen und auf konkrete Eingaben anzuwenden
- die erlernten Methoden für den Entwurf und die Analyse von effizienten Algorithmen anzuwenden
- für vorgegebene Probleme eigenständig Vergleichernetze zu konstruieren
- zu Optimierungsproblemen eine Formulierung als Lineares Programm zu konzipieren

### **Literatur**

- N. Blum: Algorithmen und Datenstrukturen, Oldenbourg Verlag, 2004.
- T.H. Cormen, C.E. Leiserson, R.L. Rivest et al.: Algorithmen - Eine Einführung, Oldenbourg Verlag, 2007.
- V. Heun: Grundlegende Algorithmen, Teubner Verlag, 2003.
- H.-W. Lang: Algorithmen, Oldenbourg Verlag, 2006.

- U. Schöning: Algorithmik, Spektrum Akademischer Verlag, 2001.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Abgabe der Übungsaufgaben, Bestehen der Klausur zum Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **7.22 Einführung in die Geoinformatik**

### **Introduction to Geoinformatics**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Geoinformatik“, 2 SWS
- Übung, 2 SWS (im Block)

#### **Inhalte**

- Erfassung und Aufbereitung von Geodaten
- Datenstrukturen (Raster, Vektor usw.)
- Grundlagen der Bildverarbeitung
- Koordinatensysteme und –transformationen
- Georeferenzierung
- Räumliche Analysen
- Datenverschneidung
- Räumliche Interpolationsverfahren, Geländemodelle

#### **Lernergebnisse / Kompetenzen**

Die Geoinformatik befasst sich im weitesten Sinn mit der Erfassung, Aufbereitung, Analyse und Präsentation von räumlichen Daten. Der Raumbezug erfolgt dabei über Weltkoordinaten und kann beispielsweise per GPS hergestellt werden. Nach erfolgreichem Abschluss der Lehrveranstaltung sollen die Studierenden

- die wichtigsten Koordinatensysteme und Kartennetzentwürfe inklusive der mathematischen Hintergründe verstanden haben,

- die gebräuchlichen Datenstrukturen (Raster, Vektor, Attribute) mit ihren Anwendungsfällen und Vor- und Nachteilen kennen und erläutern können,
- das Georeferenzieren und Verschneiden von Vektor- und Rasterdaten beherrschen sowie
- grundlegende Verfahren der räumlichen Analyse anwenden können.

#### **Literatur**

- Norbert Bartelme: Geoinformatik. Modelle, Strukturen, Funktionen. Springer.
- Ralf Bill: Grundlagen der Geo-Informationssysteme. Wichmann.
- Norbert de Lange: Geoinformatik in Theorie und Praxis. Springer.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

Regelmäßige Teilnahme an den Übungen, Bearbeitung der Aufgaben, schriftliche Prüfung (Klausur, 90 Minuten).

#### **Häufigkeit des Angebots, modulare Schiene**

Jährlich, i.d.R. im Wintersemester

#### **Modulbeauftragte und hauptamtliche Lehrende**

PD Dr. Dr.-Ing. Wilfried Linder

## **7.23 Einführung in die statistische Analyse mittels Computersimulationen** **Introduction to statistical analysis through computer simulations**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Einführung in die statistische Analyse mittels Computersimulationen“, 2 SWS (auf Englisch)
- Theoretische und Praktische Übungen, 2 SWS

#### **Inhalte**

##### **Einführung in Bayesianisches Denken**

- Wahrscheinlichkeit und Einführung in Computersimulationen
- Einführung in Bootstrap
- Einführung in Bayesianische Modelle
- Einführung in multivariate Wahrscheinlichkeitsverteilungen und multiple Parameter Modelle

### **Monte Carlo Simulationsmethoden**

- Monte Carlo Methode zur Berechnung von Integralen
- Rejection Sampling
- Importance Sampling
- Sampling Importance Re-sampling

### **Markov Ketten Monte Carlo Methoden**

- Einführung in Markov Ketten
- Metropolis-Hastings Algorithmus
- Gerichtete azyklische Graphen
- Gibbs-sampling
- MCMC output analysis

### **Statistische Modellierung**

- Regression modelling
- Analysis of multiple contingency tables
- Einführung in hierarchische Modelle

### **Lernergebnisse/Kompetenzen**

Die Studierenden werden am Ende des Kurses in der Lage sein

- die Programmiersprache R sowie die Software OpenBUGS in Zusammenhang mit Bayesianischem Denken zu nutzen
- die Rolle von Computersimulationen und Bayesianischen Methoden für reale Modellierung zu erkennen
- verschiedene Simulationsmethoden zu klassifizieren

### **Literatur**

- Andrew Gelman, John B. Carlin, Hal S. Stern: Bayesian Data Analysis. Chapman & Hall, 2004

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Teil eines Master-Moduls im Master-Studiengang Biologie

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i.d.R. schriftlich)

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Studienjahr, in der Regel im Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Lercher, Dr. Pablo Verde

## **7.24 Entwurfs- und Analysetechniken für Algorithmen**

# **Design Principles and Analytic Methods for Algorithms**

## **Studiengang**

Master-Studiengang Informatik

## **Leistungspunkte**

5 LP

## **Lehrveranstaltungen**

- Vorlesung „Entwurfs- und Analysetechniken für Algorithmen“, 2 SWS
- Übungen, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 2 SWS

## **Inhalte**

Dieses Modul vertieft Kenntnisse aus folgenden Bereichen:

- Gierige Algorithmen
- Dynamische Programmierung
- Scanline
- Backtracking
- Korrektheitsbeweise
- (amortisierte) Aufwandsabschätzungen

## **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein vertieftes Verständnis in der systematischen Entwicklung und Analyse von Algorithmen zur Lösung von Entscheidungs- und Optimierungsproblemen.

## **Literatur**

Aktuelle Lehrbücher über Algorithmik sowie Originalliteratur

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Teil eines Master-Moduls im Master-Studiengang Biologie

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Empfohlene Vorkenntnisse**

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

schriftliche Prüfung (i.d.R. 90 Minuten) am Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

## **7.25 Funktionale Programmierung**

### **Functional Programming**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Funktionale Programmierung“, 2 SWS
- Übung, 2 SWS
- Praktische Übung, 2 SWS

#### **Inhalte**

Funktionale Programmierung hat in den letzten Jahren deutlich an Fahrt aufgenommen. Sprachen wie Haskell, Scala, Clojure, OCaml und F# gewinnen immer mehr an Bedeutung und auch Java hat in Version 8 mit Closures endlich Syntax für anonyme Funktionen bekommen.

Ausgehend von den Problemen, mit denen wir in objektorientierten Sprachen permanent konfrontiert sind, wollen wir in dem Kurs neue Ansätze erarbeiten, um so ein umfassenderes Verständnis von Programmierung zu erhalten. Wir werden uns damit befassen, wie wir Programme in funktionalen Sprachen strukturieren und unbeabsichtigte Komplexität loswerden können. Außerdem werden wir uns mit internen domänenspezifischen Sprachen beschäftigen. Die Sprache, die wir in dem Kurs verwenden werden ist Clojure, ein modernes Lisp, das auf der JVM läuft und viele aus softwaretechnischer Sicht interessante Konzepte umsetzt. Wir werden uns ebenfalls mit statischer Typisierung in der Programmiersprache Haskell und nebenläufiger Programmierung in Erlang befassen.

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Prinzipien der funktionalen Programmierung bezeichnen können und klassischen, imperativen Programmiersprachen gegenüberstellen können
- bewerten für welche Einsatzbereiche die funktionale Programmierung vorteilhaft ist
- eigenständig funktionale Programme erstellen und testen können

#### **Literatur**

- Abelson, Sussman: Structure and Interpretation of Computer Programs
- Moseley, Marks: Out of the tarpit
- Fogus, Houser: The Joy of Clojure
- Emerick, Carper, Grand: Programming Clojure
- Rathore: Clojure in Action
- Higginbotham: Clojure for the Brave and True

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i.d.R. schriftlich)

#### **Häufigkeit des Angebots, modulare Schiene**

Jährlich, i.d.R. im Wintersemester

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel, Jens Bendisposto

### **7.26 Fuzzy-Clusteranalyse Fuzzy Cluster Analysis**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Fuzzy-Clusteranalyse“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

- Fuzzy und possibilistische fuzzy Clusteringverfahren
- Bestimmung der Clusteranzahl
- Vergleich von Clusteringergebnissen
- Fuzzy Clustering von unvollständigen Daten
- Fuzzy Clustering hochdimensionaler Daten

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Nutzen von Konzepten der Fuzzy Logik für das Clustering zu erkennen
- die wichtigsten Verfahren, Herausforderungen und aktuellen Entwicklungen im Bereich Fuzzy-Clusteranalyse zu erläutern
- mit verschiedenen Anwendungsgebieten der Fuzzy Clustering Methoden vertraut zu werden und die entsprechenden Methoden selbstständig anzuwenden

#### **Empfohlene Literatur:**

- F. Höppner, F. Klawonn, R. Kruse: Fuzzy-Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse, Vieweg, Braunschweig/Wiesbaden, 1997
- R. Kruse, C. Borgelt, C. Braune, F. Klawonn, C. Moewes, M.
- Steinbrecher: Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze, Springer Vieweg, Wiesbaden, 2015

sowie ggfs. weiterführende Literatur (wird in der Veranstaltung bekanntgegeben)

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik



- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen bzw. dem Seminar
- erfolgreiches Bearbeiten der Übungsaufgaben bzw. Anfertigen einer angemessenen Seminausarbeitung und Halten eines Vortrags
- abschließende Prüfung (i.d.R. mündlich)

### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig, ggf. im Wechsel mit den Modulen/Veranstaltungen von Prof. Dr. Stefan Conrad.

### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Ludmila Himmelspach

## **7.27 Fuzzy Systeme**

### **Fuzzy Systems**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Fuzzy Systeme“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

- Eigenschaften und Operationen auf fuzzy Mengen
- Fuzzy Relationen
- Zugehörigkeitsfunktionen, Fuzzifizierung und Defuzzifizierung
- Fuzzy Logik und approximatives Schließen
- Fuzzy Regelsysteme
- Fuzzy Methoden in Pattern Recognition

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- theoretischen Grundlagen der fuzzy Mengen Theorie zu erklären
- Nutzen von Konzepten der fuzzy Logik in intelligenten Systemen zu erkennen
- mit verschiedenen Anwendungsgebieten der fuzzy Methoden vertraut zu werden und die entsprechenden Methoden selbstständig anzuwenden

#### **Empfohlene Literatur:**

- R. Kruse, C. Borgelt, C. Braune, F. Klawonn, C. Moewes, M. Steinbrecher: Computational Intelligence:

Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze, Springer Vieweg, Wiesbaden, 2015

- Timothy J. Ross: Fuzzy Logic with Engineering Applications, John Wiley & Sons Ltd, Chichester, 2004

sowie ggfs. weiterführende Literatur (wird in der Veranstaltung bekanntgegeben)

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen bzw. dem Seminar
- erfolgreiches Bearbeiten der Übungsaufgaben bzw. Anfertigen einer angemessenen Seminarausarbeitung und Halten eines Vortrags
- abschließende Prüfung (i.d.R. mündlich)

### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig, ggf. im Wechsel mit den Modulen/Veranstaltungen von Prof. Dr. Stefan Conrad.

### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Ludmila Himmelspach

## **7.28 Gerechte Aufteilungsverfahren Fair Division**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

10 LP

### **Lehrveranstaltungen**

- Vorlesung „Gerechte Aufteilungsverfahren“, 4 SWS
- Übung, 2 SWS

### **Inhalte**

- Cake-Cutting: Gerechte Aufteilung teilbarer Güter
  - Grundlagen
  - Bewertungskriterien
  - Cake-cutting-Protokolle
- Gerechte Aufteilung unteilbarer Güter
  - Definition und Klassifikation von Allokationsproblemen
  - Präferenzerhebung und kompakte Darstellung
  - Kriterien für Allokationen
  - Berechnung von Allokationen: Zentralisierte Mechanismen

- Dezentralisierte Allokationsprotokolle

## **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem Gebiet der gerechten Aufteilungsverfahren zu vermitteln, welches derzeit in den Abteilungen „Komplexitätstheorie und Kryptologie“ und „Computational Social Choice“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird und in denen bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende wichtige Probleme und die zu ihrer Lösung entwickelten Verfahren und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Insbesondere sollten sie in der Lage sein, verschiedene Fairness-Begriffe zu beschreiben und Aufteilungsprotokolle zu entwerfen und zu bewerten, die Fairness eines gegebenen Typs garantieren. Ebenso sollten sie die Komplexität der für solche Spiele relevanten Entscheidungsprobleme (in einer geeigneten kompakten Darstellung) formal bestimmen und beschreiben können.

## **Empfohlene Literatur**

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen, Spektrum Akademischer Verlag (Springer), 2011.

## **Ergänzende Literatur**

- Jack Robertson und William Webb: Cake-Cutting Algorithms: Be Fair if You Can, A K Peters, 1998.
- Steven J. Brams und Alan D. Taylor: Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

## **Häufigkeit des Angebots, modulare Schiene**

Etwa alle zwei Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **7.29 Graphenalgorithmen II** **Algorithms for Graphs II**

### **Studiengang**

### **Leistungspunkte**

10 LP ab PO 2015, alte PO 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Graphenalgorithmen II“, 4 SWS
- Übung, 2 SWS
- Die Bearbeitung der Hausaufgaben erfordern durchschnittlich weitere 4 SWS

### **Inhalte**

- Planare Graphen (Erkennung, Dualität, Färbungsprobleme, ...)
- Chordale Graphen (lexikographische Breitensuche, perfekte Eliminationsordnung, ...)
- Baumweite beschränkte Graphen
- Cliquesweite, Rangweite, NLC-Weite
- Baumautomaten
- Fixed Parameter Tractability
- Minorensatz
- Extremale Graphentheorie

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein breites und vertieftes Verständnis im Bereich „Graphenalgorithmen“.

### **Literatur**

Verschiedene aktuelle Lehrbücher über Ad-hoc- und Sensornetzwerke sowie Originalliteratur

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“
- „Algorithmische Komplexitätstheorie“
- „Graphenalgorithmen I“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- 10-minütiger Lehrvortrag über ein Thema der Vorlesung
- schriftliche Klausur (i.d.R. 90 Minuten) oder mündliche Prüfung (i.d.R. 45 Minuten) am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Jedes zweite Sommersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Egon Wanke

## **7.30 Grundlagen der Linearen Optimierung** **Introduction to Linear Optimization**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Vorlesung „Grundlagen der Linearen Optimierung“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

- Grundlagen der Linearen Programmierung
- Geometrische Interpretation und Dualität
- Simplex-Methode
- Ganzzahlige Lineare Programmierung
- Schnittebenenverfahren
- Branch-and-Bound
- Lagrange-Relaxierung
- Netzwerkflüsse: Theorie und Algorithmen
- Ausgewählte Anwendungen in der Bioinformatik. Z.b.: RNA-Strukturvorhersage, Strukturelles Proteinalignment, Peptide Sequencing, Seitenkettenplatzierung, Vergleich phylogenetischer Bäume, Fragment Assembly, Haplotype Assembly, Netzwerkanalyse, Netzwerkmodule, Netzwerkalignment
- Großes Programmierprojekt: für eine ausgewählte Fragestellung wird eine Lösung aufgrund der gelernten Konzepte entwickelt und implementiert.

### **Lernergebnisse/Kompetenzen**

Nach der Veranstaltung werden die Studierenden in der Lage sein

- die Grundlagen der Linearen und Ganzzahligen Optimierung zu beherrschen;
- praktische Probleme mit Hilfe der Linearen (ganzz.) Modellierung zu beschreiben;
- Algorithmen zu entwerfen, die diese Modelle lösen

### **Empfohlene Literatur**

- Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. 1997.
- Laurence Wolsey: Integer Programming. 1998.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktisch/Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Teil eines Master-Moduls im Master-Studiengang Biologie

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“

- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- erfolgreiches Bearbeiten eines Programmierprojekts
- abschließende Prüfung (i.d.R. schriftlich)

#### **Häufigkeit des Angebots, modulare Schiene**

Unregelmäßig

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Gunnar W. Klau

### **7.31 Information Retrieval und Natural Language Processing Information Retrieval and Natural Language Processing**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Information Retrieval und Natural Language Processing“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

- Evaluation von IR- bzw. NLP-Systemen
- Grundlegende NLP-Werkzeuge
- Grundlegende Formale Modelle
- Retrieval Modelle
- Das Vektorraummodell
- Fortgeschrittene NLP-Werkzeuge
- Anwendungsgebiete

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die grundlegenden Information Retrieval-Modelle zu erläutern
- NLP-Techniken zu verstehen und ihre Einsatzmöglichkeiten zu bewerten
- Information Retrieval- und NLP-Systeme zu konzipieren

#### **Empfohlene Literatur:**

- A. Henrich: Information Retrieval 1 (Grundlagen, Modelle und Anwendungen), eBook (Creative Commons), <http://www.uni-bamberg.de/?id=23516>, 2008
- R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval, Addison-Wesley, Boston, 1999
- C. Manning, H. Schütze: Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, 1999

sowie ggfs. weiterführende Literatur (wird in der Veranstaltung bekanntgegeben)

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik

- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen bzw. dem Seminar
- erfolgreiches Bearbeiten der Übungsaufgaben bzw. Anfertigen einer angemessenen Seminausarbeitung und Halten eines Vortrags
- abschließende Prüfung (i.d.R. mündlich)

### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig
- wird im Wechsel mit den Modulen/Veranstaltungen „Knowledge Discovery in Databases“, „Multimedia-Datenbanksysteme“, „Knowledge Discovery in Databases – Ausgewählte Kapitel“, „Transaktionsverwaltung“ und „Data Warehouses“ angeboten, so dass jedes Semester mindestens eine dieser Veranstaltungen angeboten wird.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

## **7.32 iOS-Programmierung**

### **iOS programming**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „iOS-Programmierung“, 2 SWS
- Übung, 1 SWS
- Projekt, 1 SWS

#### **Inhalte**

- Objective C
- Grundgerüst einer iOS-Anwendung
- Views
- Grafik & Animation
- Eingabeereignisse
- ViewController
- Persistenz
- Netzwerk

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Programmiersprache Objective C und das Entwicklungswerkzeug Xcode anzuwenden

- Benutzerschnittstellen mit Cocoa Touch zu konzipieren
- Datenspeicherung mit CoreData und iCloud anzuwenden
- grundlegende Netzwerkkommunikation selbst zu programmieren
- mit den erworbenen Kenntnissen das Software-Projekt zu konzipieren und zu realisieren

#### **Literatur**

- Neuburg, M., "Programming iOS 8", O'Reilly, 2014
- iOS Dev Center

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Software-Projekt (benotet)

#### **Häufigkeit des Angebots, modulare Schiene**

Alle 2 Jahre

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

### **7.33 Judgment Aggregation Judgment Aggregation**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Judgment Aggregation“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

Dieses Modul vermittelt grundlegende Kenntnisse im Bereich der Aggregation von individuellen Urteilen über möglicherweise logisch miteinander verknüpfte Aussagen. Es werden die folgenden Themen behandelt:

- Grundlagen, Einführung in die Social Choice Theorie
- Judgment Aggregation Prozeduren
  - Grundbegriffe
  - Eigenschaften von Urteilmengen
  - Mehrheitsregel, Voraussetzungs/Folgerungsbasierte Mehrheitsregel, Voraussetzungs-basierte Quotenregeln, Quotenregel, Sequential Majority Priority Regel, Distanzbasierte Prozeduren
- Eigenschaften von Judgment Aggregation Prozeduren



- Unmöglichkeitstheorem von List und Pettit
- Charakterisierung von Quotenregeln
- Agendacharakterisierungen und Agendasicherheit, Möglichkeitsresultate
- Berechnungskomplexität in der gemeinsamen Urteilsfindung
  - Grundlagen Komplexitätstheorie
  - Bestimmung der gemeinsamen Urteilsmenge
  - Manipulation
  - Bestechung
  - Kontrolle
  - Safety of the Agenda

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- die Aggregation von individuellen Urteilen an praktischen Beispiel durchzuführen
- spezifische Probleme im Bereiche der gemeinsamen Urteilsfindung zu identifizieren
- neue Prozeduren zur gemeinsamen Urteilsfindung zu entwickeln und im Hinblick auf Ihre Eigenschaften zu untersuchen
- Empfehlung für bestimmte Einsatzbereiche der gemeinsamen Urteilsfindung geben

### **Literatur**

- Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. J. Rothe (ed.). Unter Vertrag bei Springer. Voraussichtliche Publikation: 2015.

Eine kürzere deutsche Version dieses Buches erschien unter dem Titel:

- Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Jörg Rothe, Dorothea Baumeister, Claudia Lindner, Irene Rothe. Spektrum Akademischer Verlag, 2011.

### **Ergänzende Literatur**

- C. List. The Theory of Judgment Aggregation: An Introductory Review. Synthese, 187(1):179-207, 2012.
- C. List and C. Puppe. Judgment Aggregation: A Survey. In P. Anand, P. Pattanaik and C. Puppe (eds.), Handbook of Rational and Social Choice, Oxford University Press, 2009.
- D. Grossi and G. Pigozzi. Introduction to Judgment Aggregation. In N. Bezhanishvili and V. Goranko (eds.), Lectures on Logic and Computation, Springer-Verlag, 2012.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- Prüfung am Ende der Veranstaltung (i.d.R. Klausur, 90 Minuten)

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul findet in unregelmäßigen Abständen statt.

## **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr. Dorothea Baumeister

## **7.34 Knowledge Discovery in Databases**

### **Knowledge Discovery in Databases**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Knowledge Discovery in Databases“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

- Grundlagen der Statistik
- Clustering (partitionierten, dichte basiert, hierarchisch, ...)
- Klassifikation (Entscheidungsbaum, Support-Vektor-Maschine, ...)
- Assoziationsregeln (Frequent-Itemsets, ...)
- Weitere Mining-Verfahren und -Anwendungen

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die verschiedenen Methoden des Knowledge Discovery zu erläutern
- die Methoden vergleichend zu bewerten
- zu entscheiden, welche Methode in welcher Situation sinnvoll eingesetzt werden kann

#### **Empfohlene Literatur**

- Ester M., Sander J.: Knowledge Discovery in Databases: Techniken und Anwendungen, Springer Verlag, 2000.
- Dunham, M.H.: Data Mining – Introductory and Advanced Topics, Prentice Hall, 2003.
- Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining, Addison Wesley, 2006.
- Han J., Kamber M.: Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, August 2000.
- Berthold M., Hand D. J. (eds.): Intelligent Data Analysis: An Introduction. Springer Verlag, Heidelberg, 1999.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen bzw. dem Seminar
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i.d.R. mündlich)

#### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig
- wird im Wechsel mit den Modulen/Veranstaltungen „Data Warehouses“, „Multimedia-Datenbanksysteme“, „Knowledge Discovery in Databases – Ausgewählte Kapitel“, „Transaktionsverwaltung“ und „Information Retrieval und Natural Language Processing“ angeboten, so dass jedes Semester mindestens eine dieser Veranstaltungen angeboten wird.

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

### **7.35 Knowledge Discovery in Databases - Ausgewählte Themen** **Knowledge Discovery in Databases - selected topics**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Knowledge Discovery in Databases – Ausgewählte Themen“, 2 SWS
- Übung/Seminar, 2 SWS

#### **Inhalte**

Es werden zu dem Modul „Knowledge Discovery in Databases“ weiterführende Verfahren behandelt, wobei aktuelle Entwicklungen berücksichtigt werden sollen. Mögliche Themenschwerpunkte sind z.B.

- Temporal Association Rule Mining
- Zeitreihenanalyse und -vorhersage
- Erweiterungen des Clustering z.B. für fehlende Werte
- Outlier Detection

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- spezielle Verfahren in den behandelten Themengebieten zu bewerten und kritisch zu vergleichen
- anhand der Charakteristik des zu lösenden Problems geeignete Verfahren auszuwählen und, wenn erforderlich, zu kombinieren

#### **Empfohlene Literatur**

- Ester M., Sander J.: Knowledge Discovery in Databases: Techniken und Anwendungen, Springer Verlag, 2000.
- Dunham, M.H.: Data Mining – Introductory and Advanced Topics, Prentice Hall, 2003.
- Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining, Addison Wesley, 2006.
- Han J., Kamber M.: Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, August 2000.
- Berthold M., Hand D. J. (eds.): Intelligent Data Analysis: An Introduction. Springer Verlag, Heidelberg, 1999.

sowie weiterführende Literatur in Abhängigkeit von den behandelten Themen (wird in der Veranstaltung bekanntgegeben)

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Knowledge Discovery in Databases“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen bzw. dem Seminar
- erfolgreiches Bearbeiten der Übungsaufgaben bzw. Anfertigen einer angemessenen Seminarausarbeitung und Halten eines Vortrags
- abschließende Prüfung (i.d.R. mündlich)

### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig
- wird im Wechsel mit den Modulen/Veranstaltungen „Data Warehouses“, „Multimedia-Datenbanksysteme“, „Knowledge Discovery in Databases“, „Transaktionsverwaltung“ und „Information Retrieval und Natural Language Processing“ angeboten, so dass jedes Semester mindestens eine dieser Veranstaltungen angeboten wird.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

## **7.36 Kooperative Spieltheorie Cooperative Game Theory**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Kooperative Spieltheorie“, 2 SWS
- Übung, 1 SWS

### **Inhalte**

- Kooperative Spiele: Miteinander spielen
  - Grundlagen
    - Kooperative Spiele mit übertragbarem Gewinn
    - Superadditive Spiele
    - Stabilitätskonzepte für kooperative Spiele
  - Einfache Spiele
    - Der Kern einfacher Spiele
    - Darstellungen einfacher Spiele
    - Gewichtete Wahlspele

- Dimensionalität
- Machtindizes
- Der Shapley-Shubik-Index und der Shapley-Wert
- Die Banzhaf-Indizes
- Komplexität von Problemen für kompakt darstellbare Spiele
  - Spiele auf Graphen
  - Gewichtete Wahlspele
  - Hedonische Spiele

### **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem Gebiet der kooperativen Spieltheorie zu vermitteln, welches derzeit in der Abteilung „Komplexitätstheorie und Kryptologie“ (Prof. Jörg Rothe und Jun.-Prof. Dorothea Baumeister) intensiv in der Forschung bearbeitet wird und in dem bevorzugt Themen für Bachelor- und Master-Arbeiten ausgegeben werden. Am Ende der Veranstaltung sollten Studierende wichtige Probleme und die zu ihrer Lösung entwickelten Algorithmen und algorithmischen Techniken nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können. Insbesondere sollten sie in der Lage sein, „strategische Szenarien“ durch kooperative Spiele zu beschreiben und Stabilitätskonzepte in diesen Spielen formal zu charakterisieren. Ebenso sollten sie die Komplexität der für solche Spiele relevanten Entscheidungsprobleme (in einer geeigneten kompakten Darstellung) formal bestimmen und beschreiben können.

### **Empfohlene Literatur**

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen, Spektrum Akademischer Verlag (Springer), 2011.

### **Ergänzende Literatur**

- Bezalel Peleg and Peter Sudhölter: Introduction to the Theory of Cooperative Games, Kluwer Academic Publishers, 2003.
- Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge: Computational Aspects of Cooperative Game Theory, Morgan and Claypool Publishers, 2011.
- Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani (eds.): Algorithmic Game Theory, Cambridge University Press, 2008.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle vier Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

### **7.37 Kryptokomplexität II Cryptocomplexity II**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

10 LP ab PO 2015, alte PO 15 LP

#### **Lehrveranstaltungen**

- Vorlesung „Kryptokomplexität II, 4 SWS
- Übung, 2 SWS

#### **Inhalte**

- Vertiefung der Kryptologie
  - Wiederholung: Aufgaben und Ziele der Kryptologie, einige mathematische Grundlagen und RSA
  - Diffie-Hellman und das Problem des diskreten Logarithmus
    - o Das Schlüsseltauschprotokoll von Diffie und Hellman
    - o Diskrete Logarithmen und das Diffie-Hellman-Problem
  - Die Protokolle von ElGamal
    - o Das Public-Key-Kryptosystem und Digitale Signaturen von ElGamal
    - o Sicherheit der Protokolle von ElGamal
  - Rabins Public-Key Kryptosystem
    - o Rabins Kryptosystem
    - o Sicherheit von Rabins System
  - Arthur-Merlin-Spiele und Zero-Knowledge
  - Weitere Protokolle
- Vertiefung der Komplexitätstheorie
  - Wiederholung: Aufgaben und Ziele der Komplexitätstheorie
  - Randomisierte Algorithmen und Komplexitätsklassen
    - o PP, RP und ZPP: Monte-Carlo- und Las-Vegas-Algorithmen
    - o Quantoren und BPP
    - o Graphisomorphie und die Arthur-Merlin-Hierarchie
  - Die Boolesche Hierarchie
    - o Probleme in DP
    - o Struktur und Eigenschaften der Booleschen Hierarchie über NP
    - o Exakte Graphfärbbarkeit
  - Die Polynomialzeit-Hierarchie
    - o Orakel-Turingmaschinen
    - o Struktur und Eigenschaften der Polynomialzeit-Hierarchie
    - o Vollständige Probleme in den Stufen der Polynomialzeit-Hierarchie
    - o Die Boolesche Hierarchie kollabiert die Polynomialzeit-Hierarchie

#### **Lernergebnisse/Kompetenzen**

Ziel des Moduls Kryptokomplexität II ist die vertiefte Vermittlung von Kenntnissen über die wichtigsten asymmetrischen Kryptosysteme (also effiziente Algorithmen zur Verschlüsselung und legitimen Entschlüsselung); andererseits wird die Sicherheit solcher Systeme behandelt, die häufig auf der Komplexität geeigneter Probleme beruht. Deshalb werden auch anspruchsvolle Themen der Komplexitätstheorie behandelt, insbesondere mit dem Ziel, Methoden zum Nachweis der genauen Komplexität wichtiger Probleme – z.B. durch ihre Klassifizierung als „vollständige“ Probleme in bestimmten Komplexitätshierarchien – zu verstehen und anwenden zu können. Dies ergänzt auch die Veranstaltungen zur Algorithmik, in denen es vorrangig um den Nachweis oberer Schranken (also die Entwicklung möglichst effizienter Algorithmen) für Probleme geht. Am Ende der Veranstaltung sollten Studierende in der Lage sein,

die erlernten Modellbildungen und Techniken auf neue Probleme, Kryptosysteme und Algorithmen anwenden können. So sollen sie etwa die Komplexität von Problemen, die Sicherheit von kryptographischen Verfahren und die Laufzeit und den Speicherplatzbedarf von Algorithmen einschätzen können und in der Lage sein, strukturelle Eigenschaften von Komplexitätshierarchien zu erkennen und zu bewerten. Besonderer Wert wird darauf gelegt, dass die Studierenden die enge Verflechtung dieser Gebiete verstehen.

### **Empfohlene Literatur**

- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität, *eXamen.Press*, Springer-Verlag, 2008.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity, *EATCS Texts in Theoretical Computer Science*, Springer-Verlag, 2005.

### **Ergänzende Literatur**

- Douglas R. Stinson: Cryptography: Theory and Practice, Chapman & Hall/CRC, 2006.
- Johannes Buchmann: Einführung in die Kryptographie, Springer-Verlag, 2001.
- Christos Papadimitriou: Computational Complexity, Addison-Wesley, 1995.
- Gerd Wechsung: Vorlesungen zur Komplexitätstheorie, Teubner-Verlag, 2000.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Bearbeitung der Übungsaufgaben
- Bestehen der schriftlichen Prüfung (Klausur) oder der mündlichen Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle zwei Jahre, im Wechsel mit den anderen WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **7.38 Machine Learning – Deep Learning**

### **Machine Learning – Deep Learning**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Machine Learning – Deep Learning“, 2 SWS

- Übung, 2 SWS

## Inhalte

In diesem Modul werden die in dem Modul Machine Learning behandelten Verfahren vertieft und weiterentwickelt. Der Schwerpunkt liegt hierbei im Bereich Neuronale Netze und Deep Learning.

- Machine learning & neural networks
- Challenges of deep neural networks
- Deep belief networks
- Convolutional networks
- Auto-Encoders
- Training & Optimization techniques (dropout technique, batch normalization, distributed training, Hyper-parameter search)
- Anwendungen von Deep learning in
  - Computer Vision,
  - Speech Recognition,
  - Language Modeling

## Lernergebnisse/Kompetenzen

Ziel dieser Lehrveranstaltung ist es, den Studierenden Einblicke in einige der wichtigsten Themen, Herausforderungen und aktuellen Entwicklungen im Bereich maschinelles Lernen zu vermitteln. Am Ende der Veranstaltung sollten Studierende komplexe Lernprobleme und die zu ihrer Lösung entwickelten Deep Learning Algorithmen nicht nur kennen, sondern sie auch selbstständig modifizieren und anwenden können.

## Empfohlene Literatur

TBD

## Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## Teilnahmevoraussetzungen

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

Hilfreich ist erfolgreicher Abschluss der Mathematik Module:

"Lineare Algebra", "Analysis", "Stochastik", "Numerik" und „Knowledge Discovery in Databases“

## Empfohlene Vorkenntnisse

- „Machine Learning“ „Computer Vision“

## Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive Teilnahme an den Übungen
- Erfolgreiches Bearbeiten der Übungsaufgaben
- Abschließende Prüfung (i.d.R. mündlich)

## Häufigkeit des Angebots, modulare Schiene

Unregelmäßig

## Modulbeauftragte und hauptamtliche Lehrende

Prof. Dr. Stefan Harmeling

## 7.39 Master-Seminar über Algorithmen für perfekte Graphen



# **Master's Seminar on Algorithms for perfect Graphs**

## **Studiengang**

Master-Studiengang Informatik

## **Leistungspunkte**

5 LP

## **Lehrveranstaltungen**

- Seminar „Algorithmen für perfekte Graphen“, 2 SWS
- Übung, 2 SWS

## **Inhalte**

Perfekte Graphen sind eine der am besten untersuchten Graphenklassen. Zahlreiche Probleme lassen sich auf perfekten Graphen einfacher oder schneller lösen als auf allgemeinen Graphen. In diesem Seminar sollen verschiedene Typen perfekter Graphen, z.B. Bäume, bipartite Graphen, Intervallgraphen, Vergleichbarkeitsgraphen, Thresholdgraphen und Chordale Graphen vorgestellt und einige klassische Probleme auf diesen betrachtet werden.

## **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- selbst Einblicke in einige der wichtigsten Themen, Resultate und Methoden aus dem Gebiet der perfekten Graphen zu erarbeiten,
- einen guten Vortrag vorzubereiten und zu halten,
- die Vorträge der anderen Studierenden hören und kritisch zu diskutieren und
- eine gute schriftliche Ausarbeitung zu dem Vortragsthema anzufertigen.

## **Literatur**

- M. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Elsevier, 2004.
- F. Gurski, I. Rothe, J. Rothe, E. Wanke, Exakte Algorithmen für schwere Graphenprobleme, Springer Verlag, 2010.
- J.L. Ramirez-Alfonsin, Bruce A. Reed (Editors), Perfect Graphs, Wiley, 2010.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Empfohlene Vorkenntnisse**

- „Algorithmen für planare Graphen“ oder
- „Algorithmen für schwere Probleme“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive Mitarbeit in den Seminaren und Übungen
- schriftliche Ausarbeitung zu einem Thema des Seminars
- Vortrag des Themas
- Bestehen der Klausur zum Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

## **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **7.40 Master-Seminar über Graphparameter für gerichtete Graphen Master's Seminar on Graph parameters for directed Graphs**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Seminar „Graphparameter für gerichtete Graphen“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

In diesem Seminar beschäftigen wir uns mit der Analyse von Graphparametern für gerichtete Graphen. Hierzu betrachten wir gerichtete Formen der Cliquenweite, NLC-Weite und Rangweite sowie verschiedene Ansätze für eine gerichtete Form der Baumweite. Es sollen Beziehungen zwischen diesen Parametern und die Weite spezieller Graphklassen bzgl. dieser Parameter vorgestellt werden.

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- selbst Einblicke in einige der wichtigsten Themen, Resultate und Methoden aus dem Gebiet der Graphparameter für gerichtete Graphen zu erarbeiten,
- einen guten Vortrag vorzubereiten und zu halten,
- die Vorträge der anderen Studierenden hören und kritisch zu diskutieren und
- eine gute schriftliche Ausarbeitung/Vortragsfolien zu dem Vortragsthema anzufertigen.

### **Literatur**

- J. Bang-Jensen, G. Gutin, Digraphs: Theory, Algorithms and Applications, Springer, 2009.
- A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications. SIAM, 1999.
- M.C. Golumbic. Algorithmic Graph Theory and Perfect Graphs. Academic Press, 1980.
- N.V.R. Mahadev and U.N. Peled. Threshold Graphs and Related Topics. Annals of Discrete Math. 56. Elsevier, North-Holland, 1995.
- aktuelle Forschungsarbeiten

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“

- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Algorithmen für schwere Probleme“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Aktive Mitarbeit in den Seminaren und Übungen
- schriftliche Ausarbeitung/Vortragsfolien zu einem Thema des Seminars
- Vortrag des Themas
- Bestehen der Prüfung zum Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## **7.41 Master-Seminar über Kollektive Entscheidungsfindung Master's Seminar on Collective Decision Making**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Seminar „Collective Decision Making“, 2 SWS

### **Inhalte**

- Präferenzaggregation durch Wählen: Algorithmik und Komplexität
- Algorithmische Spieltheorie
- Gerechte Aufteilungsverfahren: Algorithmik und Komplexität

### **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, dass sich die Studierenden selbst Einblicke in einige der wichtigsten Themen, Resultate, Modellbildungen und Methoden aus dem Gebiet *Collective Decision Making* erarbeiten, über ein spezielles Thema aus der Originalliteratur einen Vortrag im Seminar halten und die Vorträge der anderen Studierenden hören und kritisch diskutieren. Weiter gehört eine schriftliche Ausarbeitung zu jedem Vortrag und es findet eine gegenseitige Beurteilung der schriftlichen Ausarbeitungen der Studierenden untereinander statt, wobei die Anonymität sowohl des Autors bzw. der Autorin als auch die des Gutachters bzw. der Gutachterin gewahrt bleibt („double-blind peer-reviewing“). Zu den in diesem Seminar erworbenen Kompetenzen gehört also sowohl die Fähigkeit, einen guten Vortrag vorzubereiten und zu halten, als auch die Fähigkeit der kritischen und fairen Begutachtung.

*Dieses Seminar ist kombinierbar mit dem Modul*

- Algorithmische Spieltheorie
- und mit den folgenden Modulen:*
- Algorithmische Eigenschaften von Wahlsystemen II
  - Kooperative Spieltheorie
  - Cake-cutting Algorithms

### **Empfohlene Literatur**

*Unter anderem Originalarbeiten, die in den folgenden Büchern zitiert werden:*

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer-Verlag, 2015.

*Eine kürzere deutsche Version dieses Buches erschien als:*

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen, Spektrum Akademischer Verlag (Springer), 2011.

### **Ergänzende Literatur**

*Unter anderem Originalarbeiten, die in den folgenden Büchern zitiert werden:*

- Piotr Faliszewski, Lane A. Hemaspaandra, Edith Hemaspaandra, and Jörg Rothe: A Richer Understanding of the Complexity of Election Systems, Chapter 14 in: „Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz,” pp. 375-406, S. Ravi and S. Shukla (eds.), Springer, 2009.
- Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel Procaccia (eds.): Handbook of Computational Social Choice, Cambridge University Press, 2015.
- Steven J. Brams and Peter C. Fishburn: Voting Procedures, Chapter 4 in: „Handbook of Social Choice and Welfare,” Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura (eds.), North-Holland, 2002.
- Bezalel Peleg and Peter Sudhölter: Introduction to the Theory of Cooperative Games, Kluwer Academic Publishers, 2003.
- Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge: Computational Aspects of Cooperative Game Theory, Morgan and Claypool Publishers, 2011.
- Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani (eds.): Algorithmic Game Theory, Cambridge University Press, 2008.
- Jack Robertson and William Webb: Cake-Cutting Algorithms: Be Fair if You Can, A K Peters, 1998.
- Steven J. Brams and Alan D. Taylor: Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- schriftliche Ausarbeitung zum zugeteilten Thema
- Halten eines Vortrags
- Begutachtung einer anderen schriftlichen Ausarbeitung

### **Häufigkeit des Angebots, modulare Schiene**

Etwa alle ein bis zwei Jahre, jeweils in Kombination mit den o.g. WP/SP-Modulen von Prof. Rothe.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Jörg Rothe

## **7.42 Master-Seminar über Modellierung biologischer Zellen**

### **Master's Seminar on modeling biological cells**

#### **Studiengang**

## **Leistungspunkte**

5 LP

## **Lehrveranstaltungen**

- Seminar, „Modeling biological cells“, 2 SWS

## **Inhalte**

- Balancierte Wachstumsmodelle für Bakterien
- Resource-Balance-Analyse
- ME-Modelle (Metabolismus und Genexpression)
- Phänomenologische Wachstumsmodelle
- Kinetische Stoffwechselmodelle

## **Lernergebnisse/Kompetenzen**

Ziel in diesem Modul ist es, dass sich die Studierenden selbst Einblicke in einige der wichtigsten Themen, Resultate und Methoden aus der Modellierung biologischer Prozesse mit Hilfe von Computern erarbeiten, über ein spezielles Thema aus der Originalliteratur einen Vortrag im Seminar halten und die Vorträge der anderen Studierenden hören und kritisch diskutieren. Weiter gehört eine schriftliche Ausarbeitung zu jedem Vortrag und es findet eine gegenseitige Beurteilung der schriftlichen Ausarbeitungen der Studierenden untereinander statt, wobei die Anonymität sowohl des Autors bzw. der Autorin als auch die des Gutachters bzw. der Gutachterin gewahrt bleibt („double-blind peer-reviewing“). Zu den in diesem Seminar erworbenen Kompetenzen gehören also sowohl die Fähigkeit, einen guten Vortrag vorzubereiten und zu halten, als auch die Fähigkeit der kritischen und fairen Begutachtung.

## **Empfohlene Literatur**

Originalarbeiten

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Schriftliche Ausarbeitung zum zugeteilten Thema
- Halten eines Vortrags
- Begutachtung von zwei anderen schriftlichen Ausarbeitungen

## **Häufigkeit des Angebots, modulare Schiene**

Jährlich im Wintersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Lercher

## **7.43 Master-Seminar zu aktuellen Themen der theoretischen und computergestützten Biologie**

### **Master's Seminar on Current Topics in Theoretical and Computational Biology**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

1 LP

#### **Lehrveranstaltungen**

- Seminar, 1 SWS (englischsprachig)

#### **Inhalte**

- Modellierung biologischer Systeme
- Bioinformatische Algorithmen
- Methoden zur Auswertung komplexer biologischer Daten

#### **Lernergebnisse/Kompetenzen**

Die Studierenden sollen einerseits in aktuelle Themenfelder der theoretischen und quantitativen Biologie und Bioinformatik eingeführt werden. Andererseits sollen sie die vortragshafte Darstellung wissenschaftlicher Inhalte sowie den mündlichen wissenschaftlichen Diskurs in englischer Sprache einüben.

#### **Empfohlene Literatur**

Aktuelle Arbeiten aus einschlägigen Fachzeitschriften

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreicher Vortrag eines Projektes
- Aktive Teilnahme an den Diskussionen während des Seminars

#### **Häufigkeit des Angebots, modulare Schiene**

Ganzjährig

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Lercher

## **7.44 Master-Seminar zu Informatik unplugged**

### **Master's Seminar on Computer Science Unplugged**

#### **Studiengang**

## Leistungspunkte

5 LP

## Lehrveranstaltungen

- Seminar „Informatik unplugged“, 2 SWS

## Inhalte

Dieses Modul beschäftigt sich mit verschiedenen Themen aus der Informatik unter dem Gesichtspunkt „Informatik unplugged“. Ziel von Informatik unplugged ist es, fundamentale Konzepte der Informatik, anhand von einzelnen Aktivitäten zu erlernen. Dabei haben alle diese Aktivitäten gemeinsam, dass keine Computer benötigt werden. In diesem Seminar sollen die Studierenden zunächst dieses Gesamtkonzept kennenlernen und sich danach in Kleingruppen intensiv mit bestimmten Themen auseinandersetzen. Im Anschluss daran, sollen die Teilnehmer selber unplugged Aktivitäten zu unterschiedlichen Bereichen der Informatik entwickeln. In diesem Seminar müssen mehrere Vorträge ausgearbeitet und gehalten werden, außerdem muss eine schriftliche Ausarbeitung erfolgen, und die Studierenden sollen ihre Fähigkeiten im Bereich der kritischen und fairen Begutachtung erweitern.

Dieses Seminar ist kombinierbar mit den Modulen:

- Judgment Aggregation
  - Mathematische Hintergründe der Wahlmanipulation
- und mit den Modulen (von Prof. Dr. Jörg Rothe):
- Algorithmische Spieltheorie
  - Algorithmische Eigenschaften von Wahlsystemen II
  - Kooperative Spieltheorie
  - Cake-cutting Algorithms
  - Kryptokomplexität II

## Lernergebnisse/Kompetenzen

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- das Konzept von Informatik unplugged zu erläutern.
- Informatik unplugged Aktivitäten mit anderen Teilnehmern durchführen, und das dahintersteckende fundamentale Konzept anschaulich erklären.
- neue Informatik unplugged Aktivitäten zu ganz unterschiedlichen Bereichen der Informatik selbst zu entwickeln.

## Empfohlene Literatur

- Computer Science Unplugged. Tim Bell, Ian H. Witten und Mike Fellows. Online verfügbar unter: <http://csunplugged.org>
- Abenteuer Informatik: IT zum Anfassen – von Routenplaner bis online Banking. Jens Gallenbacher, Spektrum Akademischer Verlag, 2012

## Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## Teilnahmevoraussetzungen

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## Voraussetzungen für die Vergabe von Leistungspunkten

- schriftliche Ausarbeitung einer neuen unplugged Aktivität
- Vorstellen einer vorhandenen und der neu erarbeiteten unplugged Aktivität
- Begutachtung einer anderen schriftlichen Ausarbeitung

### Häufigkeit des Angebots, modulare Schiene

Dieses Modul findet in unregelmäßigen Abständen statt.

### Modulbeauftragte und hauptamtliche Lehrende

Jun.-Prof. Dr. Dorothea Baumeister

## 7.45 Master-Seminar zu Kombinatorische Optimierung in der Bioinformatik Master's Seminar on Combinatorial optimization in bioinformatics

### Studiengang

Master-Studiengang Informatik

### Leistungspunkte

5 LP

### Lehrveranstaltungen

- Seminar „Kombinatorische Optimierung in der Bioinformatik“, 2 SWS

### Inhalte

Viele bioinformatische Probleme lassen sich als kombinatorische Optimierungsprobleme formulieren. Dieses Modul vermittelt fortgeschrittene Kenntnisse zum Erkennen, Formulieren und Lösen solcher Probleme. Anhand einer Auswahl aus den folgenden aktuellen bioinformatischen Themengebieten werden wichtige Techniken wie Dynamische Programmierung, Graphenalgorithmen, Lineare und Ganzzahlige Lineare Programmierung, Lagrange-Relaxierung und Parameterisierte Algorithmen eingeführt oder wiederholt und vertieft und Lösungsansätze entwickelt.

- Genomik: *De Novo*-Genomassemblierung: *fragment assembly*; Haplotype-Phasing; Finden von struktureller Variation
- Strukturbioogie: RNA-Sekundärstruktur: Vorhersage, Vergleich; Proteinstruktur: Vorhersage, *protein threading*, Seitenkettenplatzierung, Vergleich
- Transkriptomik: Transkript-Assembly
- Proteomik: *De Novo*-Peptidsequenzierung; Quantifizierung von Polymeren
- Phylogenetik: Perfekte Phylogenie, Vergleich von Bäumen
- Evolution: Finden von Gen-Clustern
- Netzwerkbiologie: Dichte Teilgraphen (Proteinkomplexe); Netzwerkmodule; Netzwerkalignment; Rekonstruktion Boolescher Signalwege
- Krebsgenomik: Finden von *driver mutations*

### Lernergebnisse/Kompetenzen

Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein breites und vertieftes Verständnis im Bereich „Algorithmische Bioinformatik“ und können neue Probleme einordnen und mit modernen Optimierungsmethoden lösen.

### Literatur

Originalliteratur.

### Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik



### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Algorithmen in der Bioinformatik“.

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Schriftliche Ausarbeitung zum gewählten Thema
- Begutachtung von anderen Ausarbeitungen
- Präsentation des eigenen Themas
- Aktive Teilnahme an Diskussionen

### **Häufigkeit des Angebots, modulare Schiene**

Unregelmäßig

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Gunnar W. Klau

## **7.46 Master-Seminar zu Machine Learning Master's Seminar on Machine Learning**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Seminar „Machine Learning“, 2 SWS

### **Inhalte**

Machine Learning hat als Oberbegriff für Methoden zur Erkennung von Mustern und Gesetzmäßigkeiten in Daten und der Erzeugung daraus abgeleiteter Entscheidungen und Vorhersagen eine enorme Bedeutung in vielen Fachdisziplinen und Branchen erlangt. Insbesondere die stetig wachsende Menge an digitalen Daten in Unternehmen, Wissenschaft und Internet hat einen massiven Bedarf an Experten auf diesem Gebiet geführt. Da die Entwicklung neuer Methoden in diesem Bereich sehr schnell verläuft, ist eine fortlaufende Auseinandersetzung mit den neuesten Erkenntnissen für künftige Experten unerlässlich. Ziel der Veranstaltung ist es, das Grundwissen aus der Vorlesung Machine Learning durch ein intensives Studium wichtiger wissenschaftlicher Veröffentlichungen zu vertiefen, und die Teilnehmer auf die selbständige Auseinandersetzung mit neuen wissenschaftlichen Entwicklungen in diesem Feld vorzubereiten. Um dieses Lernziel zu erreichen, wird den Teilnehmern wöchentlich eine wissenschaftliche Veröffentlichung in englischer Sprache genannt. Die Teilnehmer arbeiten diese Veröffentlichung selbständig durch, so dass sie in der Lage sind, deren Inhalt vor allen anderen Teilnehmern an der Tafel zu präsentieren. Jede Woche findet eine Diskussionsveranstaltung statt, bei der ausgewählte Teilnehmer die Veröffentlichung vorstellen. Anschließend diskutieren alle Teilnehmer die Veröffentlichung.

### **Lernergebnisse/Kompetenzen**

Studierende haben nach erfolgreichem Abschluss dieses Moduls ein vertieftes Verständnis aktueller Methoden und typischer Anwendungen im Bereich des Machine Learning. Sie können wissenschaftliche Veröffentlichungen systematisch aufarbeiten, deren Inhalte prägnant zusammenfassen und kritisch bewerten.

### **Empfohlene Literatur**

Zu dieser Veranstaltung gibt es kein Lehrbuch.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

Hilfreich ist erfolgreicher Abschluss des Moduls "Lineare Algebra"

### **Empfohlene Vorkenntnisse**

- „Machine Learning“, „Computer Vision“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Schriftliche Ausarbeitung zum gewählten Thema
- Begutachtung von anderen Ausarbeitungen
- Präsentation des eigenen Themas
- Aktive Teilnahme an Diskussionen

### **Häufigkeit des Angebots, modulare Schiene**

Unregelmäßig

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Harmeling

## **7.47 Master-Seminar zu Reproduzierbarkeit bioinformatischer Forschungsergebnisse**

### **Master's Seminar on Reproducibility in bioinformatics research**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Seminar „Reproduzierbarkeit bioinformatischer Forschungsergebnisse“, 2 SWS

#### **Inhalte**

Die wissenschaftliche Datenanalyse nimmt einen immer größer werdenden Stellenwert in der Forschung ein. Insbesondere in der Bioinformatik hat die Kombination von enorm wachsenden Datenmengen, fehlenden Standards und Anreizen, schlechten Programmierpraktiken und unzureichend dokumentierten Parametern und Versionen zu einer Reproduzierbarkeitskrise geführt. Wissenschaftliche Software wird allzu häufig unzu-

reichend dokumentiert, ineffizient programmiert und nicht gewartet. In diesem Seminar zeigen wir diese Fehler auf, in dem wir versuchen, die Datenanalyse ausgewählter Originalarbeiten zu reproduzieren. Zudem lernen wir, wie man es besser machen kann, indem diese Analysen mit Hilfe des Workflow-Management-Tools Snakemake realisiert werden und, bei Erfolg, auf zertifizierten Online-Speicherdiensten deponiert werden.

### **Lernergebnisse/Kompetenzen**

Studierende, die das Modul erfolgreich absolviert haben, kennen anschließend Wege aus der Reproduzierbarkeitskrise und können diese in Arbeit vermeiden.

### **Literatur**

Originalliteratur.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Algorithmen in der Bioinformatik“.

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Schriftliche Ausarbeitung zum gewählten Thema
- Begutachtung von anderen Ausarbeitungen
- Präsentation des eigenen Themas
- Aktive Teilnahme an Diskussionen
- Gruppenarbeit (2-4 Personen): Datenanalyse mit Snakemake

### **Häufigkeit des Angebots, modulare Schiene**

Unregelmäßig

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Gunnar W. Klau

## **7.48 Master-Seminar zu Verteilte und Parallele Systeme Master's Seminar on Distributed and Parallel Systems**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP

### **Lehrveranstaltungen**

- Seminar „Verteilte und Parallele Systeme“, 2 SWS

## **Inhalte**

In vielen Bereichen der Forschung und der Industrie werden immer größere Datenmengen (Schlagwort Big-Data) erfasst und gespeichert. Um diese Daten möglichst effizient verarbeiten zu können sind verteilte und parallele Systeme unabdingbar. Neben der Skalierbarkeit bezüglich der Anzahl an Ressourcen sind auch Fehlertoleranzaspekte sehr wichtig. Bei interaktiven Systemen die Millionen oder Milliarden von Benutzern bedienen ist zusätzlich die Latenz eine besondere Herausforderung.

Dieses Seminar behandelt aktuelle Forschungs-Themen in diesem Themen-Umfeld, u.a. Cloud, Batch- und Stream-Processing, Graph-Verarbeitung, Storage-Systeme (key-value, NoSQL, in-memory)

## **Lernergebnisse/Kompetenzen**

Studierende haben nach erfolgreichem Abschluss dieses Moduls ein vertieftes Verständnis von aktuellen Hard- und Software-Techniken im Themenumfeld verteilte und parallele Systeme.

Sie können wissenschaftliche Veröffentlichungen systematisch aufarbeiten, deren Inhalte prägnant zusammenfassen, kritisch bewerten und präsentieren.

## **Empfohlene Literatur**

Aktuelle wissenschaftliche Publikationen zu den oben genannten Themen.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Empfohlene Vorkenntnisse**

- „Verteilte Systeme“.

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme am Seminar
- Abschließende Prüfung am Ende des Semesters

## **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **7.49 Mathematische Hintergründe der Wahlmanipulation Mathematical Background of Manipulation**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

10LP ab PO 2015, alte PO 15 LP

### **Lehrveranstaltungen**

- Vorlesung „Mathematische Hintergründe der Wahlmanipulation“, 4 SWS
- Übung, 2 SWS
- Seminar, 2 SWS

## Inhalte

In vielen Situationen profitieren Wähler davon, wenn sie eine unehrliche Stimme abgeben. Gibbard und Satterthwaite haben sogar gezeigt, dass es bei mindestens drei zur Wahl stehenden Alternativen kein nicht manipulierbares Wahlsystem geben kann, welches eine Reihe von sinnvollen Eigenschaften erfüllt. Diese Veranstaltung behandelt die systematische mathematische Analyse von verschiedenen Formen der Manipulation in Wahlen.

- Einführung in die Social Choice Theorie
  - Formale Grundlagen
  - 20 Wahlsystemen
  - Arrows Theorem
- Grundlagen der Manipulation
  - Präferenzen über Mengen und Manipulierbarkeit
  - Beispiele für Manipulation
  - Resultate
  - Agenda Manipulierbarkeit und Transitivität
- Resolute Wahlregeln
  - Das Gibbard-Satterthwaite Theorem
  - Gleichstände in Stimmen
  - Äquivalenz von Arrows Theorem und dem Gibbard-Satterthwaite Theorem
  - Betrachtung des Beweises des Gibbard-Satterthwaite Theorems
- Umgang mit den Resultaten
  - Einführung
  - Manipulation ist zwar möglich aber schwer
  - Einschränkung der Annahmen
  - Variation der formalen Annahmen

## Lernergebnisse/Kompetenzen

Studierende sollen nach Absolvierung der Lehrveranstaltung in der Lage sein,

- verschiedene Formen der Manipulation formal zu definieren und zu vergleichen
- bekannte Beweise aus der Literatur wiederholen und erläutern
- Manipulation in unterschiedliche Formen einteilen
- Empfehlungen für die Verwendung von Wahlsystemen in unterschiedlichen Einsatzbereichen entwerfen

## Literatur

- A. Taylor: Social Choice and the Mathematics of Manipulation, Cambridge University Press, 2005.

## Ergänzende Literatur

- Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. J. Rothe (ed.). Unter Vertrag bei Springer. Voraussichtliche Publikation: 2015.

Eine kürzere deutsche Version dieses Buches erschien unter dem Titel:

- Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Jörg Rothe, Dorothea Baumeister, Claudia Lindner, Irene Rothe. Spektrum Akademischer Verlag, 2011.

## Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## Teilnahmevoraussetzungen

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- Prüfung am Ende der Veranstaltung (i.d.R. Klausur, 90 Minuten)

#### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul findet in unregelmäßigen Abständen statt.

#### **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr. Dorothea Baumeister

### **7.50 Mobilkommunikation Mobile Communications**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Mobilkommunikation“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

Das Modul richtet sich an Studenten, die bereits elementare Kenntnisse im Bereich Rechnernetze besitzen und die technischen Grundlagen von mobilen Netzen verstehen wollen. Ziel des Moduls ist das Erlernen allgemeiner Grundkenntnisse, das Verständnis von praxisrelevanten Systemen und das Erwerben praktischer Fertigkeiten.

- Grundlagen der Mobilkommunikation (Geschichte, Frequenzen und Regulierungen, Signale, Antennen, Signalausbreitung)
- Medienzugriff (Space-, Frequency-, Time-, Code Division Multiple Access)
- Drahtlose Lokale Netze (IEEE 802.11, Bluetooth)
- Mobilkommunikationssysteme (GSM, GPRS, UMTS, LTE)
- Ad-Hoc Netzwerke (Anwendungen, Wegewahl, DSR, AODV, LSR)

#### **Lernergebnisse/Kompetenzen**

Studierende verstehen nach erfolgreichem Abschluss dieses Moduls die grundlegende Funktionsweise von Mobilfunksystemen und mobilen Netzwerken. Sie verstehen die physikalischen Effekte der Funkübertragung, die Herausforderungen und Lösungen für den Mehrfachzugriff sowie die konkreten Protokolle und Standards für die Mobilkommunikation, nämlich IEEE 802.11, 2G-5G und weitere.

#### **Empfohlene Literatur**

- Jochen Schiller: Mobilkommunikation, 2. Auflage, Addison-Wesley/Pearson Studium, 2003.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich

- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- Modul „Rechnernetze“.

### **Voraussetzungen für die Vergabe von Kreditpunkten**

- aktive Mitarbeit in Vorlesung
- erfolgreiche Teilnahme an der praktischen Übung
- mündliche Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

unregelmäßig

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Mauve, Jun.-Prof. Dr.-Ing. Kalman Graffi

## **7.51 Model Checking**

### **Model Checking**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Model Checking“ in Seminarform, 2 SWS
- Übung, Praktische Übung, Eigenarbeit, 2 SWS

#### **Inhalte**

Dieses Modul befasst sich mit der vollautomatischen Verifikation von Hardware und Software mit Hilfe von Model Checking Algorithmen und Werkzeugen.

- Kenntnisse der temporalen Logiken LTL, CTL und CTL\*
- Spezifikationen in LTL, CTL, CTL\* lesen und schreiben können
- Kenntnisse der Model Checking Algorithmen (CTL, LTL) und der theoretischen Grundlagen (Büchi Automaten)
- Kenntnisse der Model Checking Werkzeuge (hauptsächlich SMV und Spin)
- Grundlagen fortgeschrittener Methoden (Binary Decision Diagrams, Partial Order Reduction, Symmetrie, ...)
- Entwicklung eigener Spezifikationen.

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- verschiedenen Verfahren der Programmverifikation präsentieren und miteinander vergleichen zu können
- ausgewählte wissenschaftliche Literatur auf dem Gebiet der Programmverifikation zusammenfassen, zuordnen und kritisieren können

- eigen Spezifikationen erstellen können und dann automatisiert mit Werkzeugen prüfen können
- geeignete Formalismen und Werkzeuge für Verifikationsaufgaben auswählen können

#### **Literatur**

- M. Clarke, O. Grumberg, D.A. Peled: Model Checking, MIT Press.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Empfohlene Vorkenntnisse**

Grundlagen in mathematischer Logik (zum Beispiel aus dem Modul „Softwaretechnik und Programmiersprachen“)

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreicher Vortrag eines Kapitels des Lehrbuches
- Eigenständige Entwicklung einer Spezifikation und Überprüfung mit Hilfe von Spin und SMV
- Mündliche Prüfung

#### **Häufigkeit des Angebots, modulare Schiene**

Normalerweise jedes Jahr im Sommersemester

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **7.52 Modellierung metabolischer Netzwerke**

### **Modelling metabolic networks**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Modellierung metabolischer Netzwerke“, 2 SWS
- Übung, 2 SWS

#### **Inhalte**

- Einführung in die statistische Programmiersprache R
- Grundlegende Elemente der Linearen Algebra für die Bioinformatik
- Grundeigenschaften und Rekonstruktion der stöchiometrischen Matrix
- Topologie und fundamentale Unterräume der stöchiometrischen Matrix
- Extreme Pathways & Elementarmoden



- Eigenschaften des Lösungsraums
- Flux-Balance Analyse
- Flux-Variability, Flux-Coupling
- Modellierung von Knock-out Mutanten
- Berücksichtigung weiterer biologischer Beschränkungen

### **Lernergebnisse/Kompetenzen**

Die Studierenden können nach Abschluss dieser Vorlesung;

- wichtige Methoden der constraint-based Modellierung von metabolischen Netzwerken zusammenfassen und anwenden;
- biologische Systeme anhand möglicher biochemischer Reaktionen beschreiben;
- lineare Optimierungsprobleme mit Hilfe der Programmiersprache R formulieren und lösen;
- Stoffwechselmodule im Zusammenhang betrachten und deren Verhalten unter verschiedenen Bedingungen simulieren.

### **Literatur**

- Bernhard Ø. Palsson: Systems Biology: Properties of Reconstructed Networks. Cambridge University Press, 2006.
- Uwe Ligges: Programmieren mit R. Springer Verlag, 2007
- Günther Gramlich: Lineare Algebra. Eine Einführung für Ingenieure. Fachbuchverlag Leipzig, 2003

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (i.d.R. schriftlich)

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Studienjahr, in der Regel im Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Lercher

## **7.53 Multimedia-Datenbanksysteme Multimedia-Databasesystems**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

## **Lehrveranstaltungen**

- Vorlesung „Multimedia-Datenbanksysteme“, 2 SWS
- Übung/Seminar, 2 SWS

## **Inhalte**

- Information Retrieval
- Multimedia Information Retrieval
- Feature-Transformationsverfahren
- Distanzfunktionen
- Effiziente Algorithmen und Datenstrukturen

## **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die grundlegenden Fragestellungen des Multimedia Information Retrieval verstehen
- geeignete Distanzfunktionen auszuwählen bzw. zu entwerfen
- ein eigenes Multimedia-Datenbanksystem prinzipiell zu konzipieren

## **Empfohlene Literatur**

- Ingo Schmitt: Ähnlichkeitssuche in Multimedia-Datenbanken, Oldenbourg-Verlag München, 2005.
- K. Meyer-Wegener: Multimedia-Datenbanken – Einsatz von Datenbanktechnik in Multimedia-Systemen (2. Auflage), Teubner, Stuttgart, 2003.
- B. Prabhakaran: Multimedia Database Management Systems, Kluwer Academic Publishers, Massachusetts, 1997.
- R. Veltkamp, H. Burkhardt, H.-P. Kriegel (Hrsg.): State-of-the-Art in Content-Based Image and Video Retrieval, Kluwer Academic Publishers, 2001.
- R. Băza-Yates, B. Ribeiro-Neto: Modern Information Retrieval, Pearson / Addison Wesley, 1999.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Empfohlene Vorkenntnisse**

- „Datenbanksysteme“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen bzw. dem Seminar
- erfolgreiches Bearbeiten der Übungsaufgaben bzw. Anfertigen einer angemessenen Seminarausarbeitung und Halten eines Vortrags
- abschließende Prüfung (i.d.R. mündlich)

## **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig
- wird im Wechsel mit den Modulen/Veranstaltungen „Data Warehouses“, „Knowledge Discovery in Databases“, „Knowledge Discovery in Databases“, „Transaktionsverwaltung“ und „Information Retrieval und Natural Language Processing“ angeboten, so dass jedes Semester mindestens eine dieser Veranstaltungen angeboten wird.

## **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

### **7.54 Netzwerksicherheit**

#### **Security in Networked Systems**

##### **Studiengang**

Master-Studiengang Informatik

##### **Leistungspunkte**

10 LP ab PO 2015, alte PO 15 LP

##### **Lehrveranstaltungen**

- Vorlesung „Netzwerksicherheit“, 4 SWS
- Übungen, 2 SWS
- Praktische Übung, 2 SWS

##### **Inhalte**

Moderne Computersysteme sind einer Vielzahl von Angriffen ausgesetzt.

Ziel der Vorlesung ist es ein fundamentales Verständnis dafür zu entwickeln, wie man sich vor diesen Angriffen schützen kann. Hierzu werden sowohl die theoretischen Grundlagen diskutiert als auch konkrete Angriffe und Abwehrstrategien praktisch erprobt.

Insbesondere werden die folgenden Themen behandelt:

- Sicherheitslücken und Exploits (Buffer Overflows und Stack Smashing, Shellcode, ...)
- Malware (Viren, Würmer, Trojaner, Botnetze, ...)
- Absichern von Netzwerken (Architekturen, DMZ, Firewalls, IDS, VPN, Denial of Service, ...)
- Sicherheitsprotokolle (PKI, Web of Trust, SSL/TLS, PGP, DNSSec) und Zufallszahlen
- Anonymität im Netz (TOR, ...)
- Passwortspeicherung und Single Sign-on
- Virtualisierung
- Ausfallsicherheit (RAID, UPS)
- Web- und Anwendungssicherheit (CSRF, XSS, SQL Injection, MAC Length Extension, Schwachstellenscanner, ...)
- Vermeiden von Spam
- Digitale Währungen (Bitcoin)

##### **Lernergebnisse/Kompetenzen**

Studierende verstehen nach erfolgreichem Abschluss dieses Moduls, wie aktuelle Angriffe in Computernetzwerken funktionieren, welche grundsätzlichen Schwachstellen häufig ausgenutzt werden und welche Gegenmaßnahmen gegen welche Angriffsarten schützen können.

Studenten werden mit dem Abschluss des Moduls Konzepte für die Absicherung von Netzwerken entwickeln und ein einfaches, abgesichertes Netzwerk aufsetzen können.

##### **Literatur**

Zu dieser Veranstaltung gibt es kein Lehrbuch.

##### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Rechnernetze“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in Vorlesung
- erfolgreiche Teilnahme an der praktischen Übung
- mündliche Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Mauve

## **7.55 Opportunistische und P2P-basierte Netzwerke Opportunistic and P2P Networks**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Seminar/Übung „Opportunistische und P2P-basierte Netzwerke“, 2 SWS (auf Englisch)
- Praktikum, 2 SWS

### **Inhalte**

Das Modul richtet sich an Studenten, die bereits elementare Kenntnisse im Bereich Rechnernetze besitzen und die das wissenschaftliche Arbeiten an neuster Forschung im Rahmen von opportunistischen und P2P-basierten Netzwerken kennenlernen wollen. Nach erfolgreichem Bestehen des Moduls verstehen die Studierenden methodische und fachbezogene Grundkenntnisse in ein aktuelles Forschungsthemenfeld, erlernen Methodiken für die Konzeption und Evaluation von neuen Forschungsansätzen in diesem Themenfeld sowie das Schreiben und das gegenseitige Begutachten von wissenschaftlichen Artikeln.

Organisation des Moduls:

- Einführungsinhalte:
  - Einführung in die Methodik der wissenschaftlichen Arbeitsweise an einem (kleinen) Beispiel im Bereich Peer-to-Peer-Protokolle oder Protokolle für opportunistische Netzwerke
  - Kennenlernen gängiger Evaluationstools für Netzwerkprotokolle in diesen Forschungsbereichen
  - Einführung in die Konzeption und Begutachtung von wissenschaftlichen Artikeln
- Praktikum/Übung in Einzelarbeit oder kleinen Teams:
  - Einarbeitung in ein ausgewähltes Themenfeld aus einer Menge von Vorschlägen
  - Design eines Netzwerkprotokolls
  - Umsetzung und Evaluation in einem gängigen Evaluationstool
  - Schreiben eines wissenschaftlichen Artikels zu der Funktionsweise und Evaluation des Protokolls
  - Vorstellung der Ergebnisse auf einer universitätsinternen Konferenz

## **Lernergebnisse/Kompetenzen**

Studierende verstehen nach erfolgreichem Abschluss dieses Moduls, wie die Konzeption, Evaluation und Präsentation von neuen Forschungsergebnissen im Bereich der Peer-to-Peer-Forschung und im Bereich der opportunistischen Netzwerke stattfindet.

Studenten werden mit dem Abschluss des Moduls die Methodiken zur Erstellung, Evaluation und Beschreibung einer neuen Lösung in den genannten Forschungsbereichen anwenden können.

## **Empfohlene Literatur**

Zu dieser Veranstaltung gibt es kein Lehrbuch

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Empfohlene Vorkenntnisse**

- „Rechnernetze“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- erfolgreiche Teilnahme an der praktischen Übung
- mündliche Prüfung / Präsentation

## **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr.-Ing. Kalman Graffi

## **7.56 Paralleles Rechnen mit Grafikkarten Parallel Computing On Graphic Cards**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Paralleles Rechnen mit Grafikkarten“, 2 SWS
- Übungen, 1 SWS

### **Inhalte**

- Aufbau von nVidia-basierten GPGPU-Karten
- Kernelprogrammierung in CUDA C

- Parallele Algorithmen
- Speicherarchitektur von Grafikkarten
- Streams
- Interoperabilität von CUDA und OpenGL

### **Lernergebnisse/Kompetenzen**

Studierenden sollen nach Absolvieren der Lehrveranstaltung massiv parallele Algorithmen für nVidia-Karten in CUDA C entwickeln und optimieren können. Dafür soll auf eine solide Kenntnis der Hardware-Architektur zurückgegriffen werden können.

### **Literatur**

- J. Sanders, E. Kandrot, „CUDA by Example“, Addison Wesley (2010)
- J. Cheng, M. Grossman, T. McKercher, „Professional CUDA C Programming“, John Wiley & Sons (2014)
- „CUDA Toolkit Documentation“, <http://docs.nvidia.com/cuda/index.html>

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

Solide Kenntnisse in C/C++, Linux und/oder Windows, Eclipse und/oder VisualStudio und teilweise OpenGL und GLUT.

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Teilnahme an den Übungen
- Abschließende Prüfung am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich.

### **Modulbeauftragte und hauptamtliche Lehrende**

Dr. Stephan Raub

## **7.57 Peer-to-Peer Systeme** **Peer-to-Peer Systems**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Peer-to-Peer Systeme“, 2 SWS (auf Englisch)

- Übung, 2 SWS

## **Inhalte**

Das Modul richtet sich an Studenten, die bereits elementare Kenntnisse im Bereich Rechnernetze besitzen und die Grundlagen von Peer-to-Peer Systemen verstehen wollen. Peer-to-Peer Systeme kennzeichnen sich durch Algorithmen und Mechanismen, in denen die beteiligten Knoten sowohl Anbieter als auch Konsumenten der zur Verfügung gestellten Ressourcen sind.

Themen der Vorlesung und Übung

- Strukturierte Overlaynetzwerke (Chord, Can, Pastry, ...)
- Unstrukturierte Overlaynetzwerke (Gnutella, BubbleStorm, ...)
- Lokationsbasierte Overlaynetzwerke
- Erweiterungen von Overlaynetzwerken (Heterogenität, Anonymität, NAT ...)
- Sicherheit in Peer-to-Peer-Systemen
- BitTorrent
- Monitoring in Peer-to-Peer-Systemen

## **Lernergebnisse/Kompetenzen**

Studierende verstehen nach erfolgreichem Abschluss dieses Moduls die grundlegende Funktionsweise von Peer-to-Peer-Netzwerken und dezentralen Datenhaltungsansätzen. Sie verfügen über grundlegende praktische Kenntnisse zu der Gestaltung von Peer-to-Peer-Mechanismen, ferner lernen Sie den Umgang mit einem aktuellen Peer-to-Peer-Simulator, um selbstständig derartige Mechanismen evaluieren und tiefer verstehen zu können.

## **Empfohlene Literatur**

- Ralf Steinmetz, Klaus Wehrle (Hrsg.): Peer-to-Peer Systems and Applications; 629 Seiten, Springer, ISBN 3-540-29192-X, 2005.
- Peter Mahlmann, Christian Schindelhauer: Peer-to-Peer-Netzwerke; 293 Seiten, Springer, ISBN 978-3-540-33991-5, 2007.

## **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

## **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

## **Empfohlene Vorkenntnisse**

- „Rechnernetze“

## **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in Vorlesung
- erfolgreiche Teilnahme an der praktischen Übung
- mündliche Prüfung

## **Häufigkeit des Angebots, modulare Schiene**

Jedes Wintersemester

## **Modulbeauftragte und hauptamtliche Lehrende**

Jun.-Prof. Dr.-Ing. Kalman Graffi

## **7.58 Programmsynthese und Transformation**

### **Program Synthesis and Transformation**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Programmsynthese und Transformation“, 2 SWS
- Übung, 1 SWS
- Praktische Übung, 1 SWS

#### **Inhalte**

Dieses Modul beschäftigt sich mit Synthese und Transformation von Programmen und Softwaresystemen. Dazu gehört sowohl die Anpassung und Optimierung bestehender Software durch partielle Auswertung, als auch die automatische Generierung individueller Algorithmen.

Die Themen sind:

- Transformation und Optimierung von Programmen
- Partielle Auswertung und Automatische Compilergenerierung
- Techniken zur Programmsynthese
- Synthese von Programmen basierend auf
  - Spezifikationen
  - Programmskizzen
  - Beispieldaten

#### **Lernergebnisse/Kompetenzen**

Nach erfolgreichem Abschluss des Moduls sollten die Studierenden:

- Vorteile und Grenzen partieller Auswertung benennen können
- verstehen wie Programme mit Hilfe partieller Auswertung spezialisiert werden
- in der Lage sein einen partiellen Auswerter für eine gegebene (simple) Programmiersprache selber zu entwickeln
- Techniken der Programmsynthese benennen und vergleichen können
- einfache Synthesewerkzeuge implementieren können

#### **Literatur**

Als Lehrbuch wird ein eigenes Skript verwendet.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

#### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

#### **Empfohlene Vorkenntnisse**



Grundlagen der logischen Programmierung und von Prolog (zum Beispiel aus dem Modul „Einführung in die logische Programmierung“) sind empfohlen, aber nicht zwingend notwendig.

#### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung der Pflichtübungen
- Bestehen der Klausur

#### **Häufigkeit des Angebots, modulare Schiene**

Normalerweise jedes Jahr im Sommersemester

#### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

### **7.59 Sicherheitskritische Systeme Safety-Critical Systems**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Sicherheitskritische Systeme“, 2 SWS
- Übung, 1 SWS
- Praktische Übung, 1 SWS

#### **Inhalte**

- Dieses Modul befasst sich mit der Entwicklung möglichst fehlerfreier Software mit Hilfe der formalen B- Methode.
- Die B-Methode wurde in den 80er Jahren von Jean-Raymond Abrial entworfen und hat eine sehr gute Werkzeugunterstützung (AtelierB, Rodin, ProB). Sie ist außerdem eine der am weitesten verbreiteten formalen Methoden in Europa. Ein Paradebeispiel der formalen Entwicklungsweise in B ist das erfolgreiche METEOR Projekt (Steuersystem für die automatischen Züge der Linie 14 der Pariser U-Bahn), bei dem 100,000 Zeilen von B mit Hilfe von 28,000 Beweisen in eine robuste Implementierung übersetzt wurde.

#### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Die Grundzüge der B Sprache (Syntax und Semantik) zusammenfassen können und mit klassischen Programmiersprachen vergleichen können
- neue formale Spezifikationen mit B erstellen können und dabei praktische B Werkzeuge (hauptsächlich Rodin und ProB) benutzen können
- Spezifikationen auf Fehler untersuchen zu können
- Einfache Spezifikationen auf Papier und mit Hilfe einer automatisierten Beweisumgebung (Rodin) beweisen können
- Grundlagen der Verfeinerung nutzen zu können

#### **Literatur**

- J.R. Abrial: Modeling in Event-B. System and Software Engineering.
- Steve Schneider: The B-Method: An Introduction.

#### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

Grundlagen in mathematischer Logik (zum Beispiel aus dem Modul „Softwaretechnik und Programmiersprachen“)

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung der Pflichtübungen,
- Bestehen der Klausur.

### **Häufigkeit des Angebots, modulare Schiene**

Normalerweise jedes Jahr im Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **7.60 Transaktionsverwaltung Transactionmanagement**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Transaktionsverwaltung“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

- Transaktionseigenschaften und -anforderungen (ACID, ...)
- Serialisierbarkeit (Final-State-, Sicht-, Konfliktserialisierbarkeit)
- Abbrucheigenschaften (Rücksetzbarkeit, Vermeidung kaskadierender Abbrüche, Striktheit)
- Sperrverfahren und -protokolle
- Recovery
- Strukturierte Transaktionsmodelle
- ACTA

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die Probleme der Transaktionsverwaltung in Datenbanksystemen zu erläutern
- Eigenschaften von Transaktionsschedules zu bestimmen
- Recovery-Verfahren zu vergleichen und zu bewerten

- die Einsatzmöglichkeiten strukturierter Transaktionsmodelle zu durchschauen

### **Empfohlene Literatur**

- P. Bernstein, V. Hadzilacos, N. Goodman: Concurrency Control and Recovery in Database Systems, Addison-Wesley, Reading, MA, 1987.
- J. Gray, A. Reuter: Transaction Processing: Concepts and Techniques, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- G. Weikum, G. Vossen: Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery, Morgan Kaufmann Publishers, 2001.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“
- „Datenbanksysteme“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i.d.R. mündlich)

### **Häufigkeit des Angebots, modulare Schiene**

- unregelmäßig
- wird im Wechsel mit den Modulen/Veranstaltungen „Data Warehouses“, „Knowledge Discovery in Databases“, „Knowledge Discovery in Databases – Ausgewählte Themen“, „Multimedia-Datenbanksysteme“ und „Information Retrieval und Natural Language Processing“ angeboten, so dass jedes Semester mindestens eine dieser Veranstaltungen angeboten wird.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Stefan Conrad

## **7.61 Verteilte Systeme Distributed Systems**

### **Studiengang**

Master-Studiengang Informatik

### **Kreditpunkte**

10 LP

### **Lehrveranstaltungen**

- Vorlesung „Verteilte Systeme“, 4 SWS
- Übung, 2 SWS

### **Inhalte**

*Teil 1: Grundlagen*

- Architekturformen
- Kommunikation: Nachrichten, entfernte Methodenaufrufe und gemeinsamer Speicher
- Koordination: Zeit, wechselseitiger Ausschluss und globale Zustände
- Replikation und Konsistenz
- Fehlertoleranz: Fehlermodelle, Konsensus, Checkpointing
- Sicherheit: Grundlagen Verschlüsselung und Authentisierung

#### *Teil 2: Peer-to-Peer- und Cloud-Systeme*

- P2P-Systeme: Gnutella, DHTs: Chord, CAN, Pastry
- P2P-Dateisysteme: CFS, IVY, OceanStore
- Multicast und Pub/Sub-Dienste
- Cloud-Systeme (Storage, Batch- und Stream-Processing)
- Graphverarbeitungssysteme

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- Kommunikationsformen in eigenen Programmen anzuwenden.
- Koordinationsalgorithmen zu verstehen und zu konzipieren.
- Konsistenzstrategien für Replikate zu vergleichen und zu bewerten
- Grundlegende Fehlertoleranz und Sicherheitsaspekte in verteilten Systemen zu erläutern

Ferner sollen sie Konzepte in den behandelten Themen in eigenen Worten erklären und bewerten können:

- die Architektur von und die Algorithmen in skalierbaren und fehlertoleranten Verteilten Systemen
- P2P-Konzepte (Overlays und Dateisysteme)
- Cloud-Storage-Lösungen
- Graph-, Batch- und Stream-Processing in Cloud-Systemen

### **Empfohlene Literatur**

- G. Coulouris et.al., „Distributed Systems: Concepts and Design“, Addison-Wesley, 5. Aufl. 2011
- A. Tanenbaum and M. van Steen: „Distributed Systems: Principles and Paradigms“, 3. Auflage, Prentice Hall, 2013.
- K. Birman, „Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services“, Springer, 1. Auflage, 2012.
- Weitere relevante Publikationen werden in der Vorlesung bekannt gegeben

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Kreditpunkten**

Erfolgreiche Bearbeitung der Übungsaufgaben. Erfolgreiche Teilnahme an der Prüfung am Ende der Veranstaltung.

### **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich

### **Modulbeauftragte und hauptamtliche Lehrende**

## **7.62 Verteilte und Parallele Programmierung** **Distributed and Parallel Programming**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Verteilte und Parallele Programmierung“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

- Grundlagen der Parallelisierung
  - Prozesse, Threads, Synchronisierung (mit und ohne Sperren)
  - Transactional Memory
  - Sockets
  - Infiniband
  - Fallstudien
  - OpenMP
  - Message Passing Interface (MPI)
- In den Übungen werden die Programmiersprachen Java und C verwendet.

### **Lernergebnisse/Kompetenzen**

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- einfache Algorithmen zu parallelisieren
- die besprochenen Konzepte anzuwenden
- parallele Problemstellung korrekt zu synchronisieren und gegebene Lösungen zu bewerten

### **Literatur**

Wird in der Vorlesung bekannt gegeben.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

Keine

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung von Übungsaufgaben

- Abschließende Prüfung am Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

I.d.R. jährlich.

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Schöttner

## **7.63 Vertiefung Logische Programmierung Advanced Logic Programming**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Vertiefung Logische Programmierung“, 2 SWS
- Übung, 1 SWS
- Praktische Übung, 1 SWS

### **Inhalte**

Dieses Modul beschäftigt sich mit fortgeschrittenen Themen der logischen Programmierung. Dazu gehören vertiefende Themen aus dem Constraint Logic Programming. Weiterhin befasst sich das Modul mit der Entwicklung von Interpretern sowie Programmanalyse- und Verifikationswerkzeugen in Prolog.

Die Themen sind:

- Fortgeschrittene Programmiertechniken in Prolog
- Metaprogrammierung in Prolog und Anwendungen
- Anwendung von Constraint Logic Programming
- Entwicklung eigener Constraint Solver
- Interpreterentwicklung in Prolog
- Programmanalyse mit abstrakter Interpretation und Datenflussanalyse

### **Lernergebnisse/Kompetenzen**

Nach erfolgreichem Abschluss des Moduls sollten die Studierenden:

- Problemstellungen mit Hilfe von Constraint Solvern lösen können
- gängige Techniken für die Implementierung von Constraint Solvern kennen und umsetzen können
- eigene Constraint Solver entwickeln können
- verstehen wie Programmiersprachen mit Interpretern implementiert werden können
- in der Lage sein für eine neue Programmiersprache einen Interpreter in Prolog selber zu entwickeln
- in der Lage sein einen Interpreter für Prolog in Prolog zu schreiben
- die Grundzüge der abstrakten Interpretation und ihrer Anwendung zur Programmverifikation erläutern und vergleichen können
- einfache Analyse und Verifikationswerkzeuge in Prolog entwickeln können

### **Literatur**

Als Lehrbuch wird ein eigenes Skript verwendet.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

Grundlagen der logischen Programmierung und von Prolog (zum Beispiel aus dem Modul „Einführung in die logische Programmierung“)

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- Erfolgreiche Bearbeitung der Pflichtübungen
- Bestehen der Klausur

### **Häufigkeit des Angebots, modulare Schiene**

Normalerweise jedes Jahr im Wintersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Michael Leuschel

## **7.64 Vertiefung Rechnernetze**

### **Advanced Computer Networks**

#### **Studiengang**

Master-Studiengang Informatik

#### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

#### **Lehrveranstaltungen**

- Vorlesung „Vertiefung Rechnernetze“, 2 SWS
- Durcharbeiten von wissenschaftlichen Veröffentlichungen, 2 SWS

#### **Inhalte**

Rechnernetze im Allgemeinen und das Internet im Besonderen wurden seit den frühen Anfängen des ARPANET maßgeblich durch wissenschaftliche Arbeiten geprägt. Ziel der Veranstaltung ist es, das Grundwissen aus der Vorlesung Rechnernetze durch ein intensives Studium der richtungsweisenden Veröffentlichungen im Bereich Rechnernetze zu vertiefen. Um dieses Lernziel zu erreichen wird folgende Lehrform verwendet: jede Woche wird den Teilnehmern eine wissenschaftliche Veröffentlichung genannt. Diese ist prinzipiell in englischer Sprache verfasst. Die Teilnehmer arbeiten diese Veröffentlichung selbständig durch, so dass sie in der Lage sind, deren Inhalt vor allen anderen Teilnehmern an der Tafel zu präsentieren. Jede Woche findet eine Diskussionsveranstaltung statt, bei der zufällig ausgewählte Teilnehmer die Veröffentlichung vorstellen. Anschließend diskutieren alle Teilnehmer die Veröffentlichung.

#### **Lernergebnisse/Kompetenzen**

Studierende haben nach erfolgreichem Abschluss dieses Moduls ein vertieftes Verständnis der Funktionsweise von Computernetzwerken. Sie können wissenschaftliche Veröffentlichungen systematisch aufarbeiten, deren Inhalte prägnant zusammenfassen und kritisch bewerten.

#### **Literatur**

Zu dieser Veranstaltung gibt es kein Lehrbuch.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Empfohlene Vorkenntnisse**

- „Rechnernetze“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

- aktive Mitarbeit in Vorlesung
- aktive Vorbereitung der Veröffentlichungen
- mündliche Prüfung

### **Häufigkeit des Angebots, modulare Schiene**

Jedes Sommersemester

### **Modulbeauftragte und hauptamtliche Lehrende**

Prof. Dr. Martin Mauve

## **7.65 Zeichnen von Graphen Graph Drawing**

### **Studiengang**

Master-Studiengang Informatik

### **Leistungspunkte**

5 LP ab PO 2015, alte PO 7,5 LP

### **Lehrveranstaltungen**

- Vorlesung „Zeichnen von Graphen“, 2 SWS
- Übung, 2 SWS

### **Inhalte**

Beim Zeichnen von Graphen betrachtet man das Problem der Visualisierung struktureller Informationen, welche sich als Graphen darstellen lassen. Das automatische Zeichnen von Graphen hat wichtige Anwendungen in vielen Bereichen der Informatik, wie z.B. Datenbanken, VLSI- und Netzwerk-Design, Software Engineering und visuelle Benutzerschnittstellen. Dieses Modul befasst sich mit verschiedenen Arten zum Zeichnen von Graphen und Algorithmen, welche diese Zeichnungen konstruieren.

- Grundlagen
- Zeichnen von Bäumen
- Zeichnen von planaren Graphen
- Flüsse und orthogonale Zeichnungen
- Hierarchische Zeichenverfahren
- Kräftebasierte Zeichenverfahren

### **Lernergebnisse/Kompetenzen**



Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- die besprochenen Zeichenvarianten für Graphen zu erläutern und formal zu definieren
- die behandelten Algorithmen zu verstehen und für konkrete Eingaben Zeichnungen zu konstruieren
- für die Algorithmen zum Zeichnen auf Gittern den Platzbedarf von Zeichnungen zu ermitteln
- Vor- und Nachteile spezieller Grapheigenschaften beim Zeichnen zu hinterfragen

### **Literatur**

- G. Di Battista, P. Eades, I.G. Tollis, R. Tamassia: Graph Drawing: Algorithms for the Visualization of Graphs, Prentice Hall, 1999.
- M. Jünger, P. Mutzel: Graph Drawing Software, Springer Verlag, 2004.
- M. Kaufmann, D. Wagner: Drawing Graphs: Methods and Models, Springer Verlag, 2001.
- T. Nishizeki, MD S. Rahman: Planar Graph Drawing, World Scientific Pub Co, 2004.

### **Verwendbarkeit des Moduls**

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik

### **Teilnahmevoraussetzungen**

Bachelor-Studierende müssen folgende Module erfolgreich abgeschlossen haben:

- „Programmierung“
- „Rechnerarchitektur“
- „Algorithmen und Datenstrukturen“
- „Theoretische Informatik“

### **Voraussetzungen für die Vergabe von Leistungspunkten**

Aktive Mitarbeit in den Übungen, Abgabe der Übungsaufgaben, Bestehen der Klausur zum Ende des Semesters

### **Häufigkeit des Angebots, modulare Schiene**

Dieses Modul wird in unregelmäßigen Abständen angeboten.

### **Modulbeauftragte und hauptamtliche Lehrende**

Priv.-Doz. Dr. Frank Gurski

## Kapitel 8

### 8. Projektarbeit

#### Individual Research Project

##### 8.1 Projektarbeit

##### Individual Research Project

###### Studiengang

Master-Studiengang Informatik

###### Leistungspunkte

20 LP

###### Lehrveranstaltungen

Die Projektarbeit wird in der Regel in der Arbeitsgruppe durchgeführt, in der später dann auch die Master-Arbeit geschrieben wird. Jede Arbeitsgruppe baut dieses Modul so auf, dass es am besten auf eine Master-Arbeit vorbereitet. Die Studierenden werden auch in die Forschungsarbeiten der Arbeitsgruppe eingebunden. Die Projektarbeit kann auch als Gruppenarbeit durchgeführt werden, sofern die individuelle Leistung hinreichend festgestellt werden kann.

Folgendes sind typische Bestandteile der Projektarbeit:

- Seminare/Projektpräsentationen
- individuelle Entwurfs- und Programmieraufgaben
- Durchführen von Experimenten oder Evaluierungen
- Lesen und Verarbeiten von Fachliteratur
- Mitwirken in Forschungsprojekten

###### Inhalte

In diesem Modul soll der Student das selbständige wissenschaftliche Arbeiten erwerben und sich auf die Master-Arbeit vorbereiten. Die Inhalte hängen deshalb sehr stark vom jeweiligen Fachgebiet und den Interessensgebieten des Studenten ab.

###### Lernergebnisse/Kompetenzen

Nach erfolgreichem Abschluss des Moduls sollte der Student

- die wichtigen Konzepte des Fachgebiets der Projekt- bzw. zukünftigen Master-Arbeit zusammenfassen und veranschaulichen können
- wichtige Artikel und Bücher für spezifische Problemstellungen des Fachgebiets ermitteln können
- Konzepte des Fachgebiets hinterfragen können und entscheiden können welche für seine zukünftige Master-Arbeit in Betracht kommen
- die wissenschaftlichen Methoden des Fachbereichs beschreiben und anwenden können

###### Verwendbarkeit des Moduls

- Projektarbeit

###### Teilnahmevoraussetzungen

Keine

###### Voraussetzungen für die Vergabe von Leistungspunkten

Die konkreten Anforderungen hängen von der zu bearbeitenden Aufgabenstellung ab. Daher sind die Kriterien zum Erwerb der Leistungspunkte zu Beginn der Projektarbeit individuell festzulegen.

###### Häufigkeit des Angebots, modulare Schiene

Jedes Semester (je Arbeitsgruppe i.d.R. mindestens jährlich)

###### Modulbeauftragte und hauptamtliche Lehrende

Die Dozenten der Informatik sowie gegebenenfalls der Mathematisch-Naturwissenschaftlichen Schwerpunktfächer

## Kapitel 9

### 9. Master-Arbeit Master's Thesis

#### 9.1 Master-Arbeit Master's Thesis

##### Studiengang

Master-Studiengang Informatik

##### Leistungspunkte

30 LP

##### Lehrveranstaltungen

—

##### Inhalte

Der Inhalt der Master-Arbeit liegt im ausgewählten Schwerpunktfach

##### Lernergebnisse/Kompetenzen

Mit der schriftlichen Abschlussarbeit sollen die Studierenden nachweisen, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist (von 6 Monaten) ein Thema aus dem von ihnen gewählten Schwerpunktfach selbstständig mit wissenschaftlichen Methoden zu analysieren sowie die Erkenntnisse prägnant zu entwickeln und kompetent zu bewerten bzw. zu interpretieren. Die Master-Arbeit muss in deutscher oder englischer Sprache verfasst und in einem mündlichen Vortrag präsentiert werden.

##### Verwendbarkeit des Moduls

- Master-Arbeit

##### Teilnahmevoraussetzungen

Für die Anmeldung zur Master-Arbeit müssen mindestens 60 der im Rahmen des Master-Studiums zu erwerbenden Leistungspunkte nachgewiesen werden.

Das Thema der Master-Arbeit wird aus dem Gebiet des gewählten Schwerpunktfachs vergeben. Dazu sollten üblicherweise beide Module im Schwerpunktfach erfolgreich abgeschlossen sein.

##### Voraussetzungen für die Vergabe von Leistungspunkten

Erfolgreiche Bearbeitung des Themas und Darstellung in einer fristgerecht abgegebenen Ausarbeitung (Master-Arbeit); Präsentation in einem mündlichen Vortrag mit Diskussion

##### Häufigkeit des Angebots, modulare Schiene

Bei Vorliegen der Voraussetzungen ist der Beginn der Abschlussarbeit jederzeit möglich, d.h. die Vergabe von Themen ist nicht an bestimmte Zeiten im Semester gebunden.

##### Modulbeauftragte und hauptamtliche Lehrende

Dozenten der Informatik sowie der als Schwerpunktfach wählbaren mathematisch-naturwissenschaftlichen Fächer