

Modulhandbuch für den Master- Studiengang Informatik

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

Herausgegeben vom
Ausschuss für die Master-Prüfung
im Fach Informatik

Aktualisiert am 16.02.2024

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.

Im Abschnitt Teilnahmevoraussetzungen unterscheiden wir zwischen formalen und inhaltlichen Voraussetzungen. Wenn Sie formale Voraussetzungen nicht erfüllen, dürfen Sie sich zu diesem Modul nicht anmelden. Module mit inhaltlichen Voraussetzungen, setzen bestimmte Kenntnisse voraus. Wenn Sie diese Kenntnisse nicht mitbringen, müssen Sie diese selbstständig erwerben. In der Regel empfehlen wir dazu, die entsprechenden Module zu absolvieren.

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

Düsseldorf, 16.02.2024

Der Prüfungsausschuss für den Master-Studiengang Informatik

Inhaltsverzeichnis

Vorwort	2
Qualifikationsziele des Studiengangs Master Informatik	5
Wissenschaftliche Befähigung und Befähigung zu einer qualifizierten Erwerbstätigkeit	5
Persönlichkeitsentwicklung	5
Studienverlaufspläne	6
Lehreinheiten für Wahlpflichtbereiche (en. Courses for elective areas)	7
Advanced topics in Bayesian Data Science (en. Advanced topics in Bayesian Data Science)	8
Algorithmen für Ad-hoc- und Sensornetzwerke (en. Algorithms for Ad-hoc and Sensor networks)	10
Algorithmen für schwere Probleme (en. Algorithms for Hard Problems)	12
Algorithmen und Datenstrukturen 2 (en. Algorithms and Data Structures 2)	14
Algorithmische Spieltheorie (en. Algorithmic Game Theory)	16
Algorithms for Sequence Analysis (en. Algorithms for Sequence Analysis)	18
Approximationsalgorithmen (en. Approximation Algorithms)	20
Approximationsalgorithmen für Clusteringprobleme (en. Approximation Algorithms for Clustering Problems)	22
Betriebssystem-Entwicklung (en. Operating System Development)	24
Create Your Tech Startup (en. Create Your Tech Startup)	26
Discrete Optimization: Theory and Applications (en. Discrete Optimization: Theory and Applications)	28
Dynamische Programmiersprachen (en. Dynamic Programming Languages)	30
Einführung in die statistische Analyse mittels Computersimulationen (en. Introduction to Statistical Analysis through Computer Simulations)	32
Gerechte Aufteilungsverfahren (en. Fair Division)	34
Graphenalgorithmen 2 (en. Algorithms for Graphs 2)	36
Growth Mechanics (en. Growth Mechanics)	38
Information Theory (en. Information Theory)	40
Isolation und Schutz in Betriebssystemen (en. Isolation and Protection in Operating Systems)	42
Knowledge Discovery in Databases - ausgewählte Themen (en. Knowledge Discovery in Databases - Selected Topics)	44
Knowledge Discovery in Databases (en. Knowledge Discovery in Databases)	46
Kryptokomplexität 2 (en. Cryptocomplexity 2)	48
Master-Seminar: Aktuelle Themen der theoretischen und computergestützten Biologie (en. Master's Seminar: Current Topics in Theoretical and Computational Biology)	50
Master-Seminar: Algorithmen für graphentheoretische Konzepte in der Informatik (en. Master's Seminar: Algorithms for Graph Theoretical Concepts in Computer Science)	51
Master-Seminar: Algorithmen für perfekte Graphen (en. Master's Seminar: Algorithms for Perfect Graphs)	53
Master-Seminar: Betriebssystem-Entwicklung in Rust (en. Master's Seminar: Operating System Development with Rust)	55
Master-Seminar: Combinatorial Optimization in Bioinformatics (en. Master's Seminar: Combinatorial Optimization in Bioinformatics)	57
Master-Seminar: Computational Multiomics (en. Master's Seminar: Computational Multiomics)	59
Master-Seminar: Datenbanken und Informationssysteme (en. Master's Seminar: Databases and Information Systems)	61
Master-Seminar: Deep Learning (en. Master's Seminar: Deep Learning)	63

Master-Seminar: Digital Innovation and Entrepreneurship (en. Master's Seminar: Digital Innovation and Entrepreneurship)	65
Master-Seminar: Kollektive Entscheidungsfindung (en. Master's-Seminar: Collective Decision-Making)	67
Master-Seminar: Komplexität und Kryptologie (en. Master's-Seminar: Complexity and Cryptology) . . .	69
Master-Seminar: NP-Schwere Probleme (en. Master's Seminar on NP-hard problems)	71
Master-Seminar: Property Testing (en. Master's Seminar: Property Testing)	73
Master-Seminar: Reproducibility in Bioinformatics Research (en. Master's Seminar: Reproducibility in Bioinformatics Research)	75
Master-Seminar: User Experience (UX) Design (en. Master's Seminar: User Experience (UX) Design)	77
Master-Seminar: Wissenschaftliche Methoden (en. Master's Seminar: Scientific Methods)	79
Methods for Population Genetics (en. Methods for Population Genetics)	81
Model Checking (en. Model Checking)	83
Multimedia-Datenbanksysteme (en. Multimedia Database Systems)	85
Natural Language Processing (en. Natural Language Processing)	87
Practical: Implementing Transformer Models (en. Practical: Implementing Transformer Models)	89
Representations in Natural Language Processing (en. Representations in Natural Language Processing)	91
Sicherheitskritische Systeme (en. Safety-Critical Systems)	93
Stochastic Models of Biological Systems (en. Stochastic Models of Biological Systems)	95
System-Software für Big-Data-Computing (en. System Software for Big Data Computing)	97
Transaktionsverwaltung (en. Transaction Management)	98
User Experience (UX) Design and Management (en. User Experience (UX) Design and Management)	100
Vertiefung Compilerbau (en. Advanced Compiler Construction)	102
Vertiefung Funktionale Programmierung: Clojure (en. Advanced Functional Programmierung: Clojure) .	104
Vertiefung Logische Programmierung (en. Advanced Topics in Logic Programming)	106
Lehrveranstaltungen anderer Studiengänge oder Institute (en. Courses from other programs or institutes)	108
Master AI and Data Science	108
Master Physik	109
Master Betriebswirtschaftslehre	110
Projektarbeit (en. Individual Research Project)	111
Modulbeschreibung (en. Module Description)	112
Master-Arbeit (en. Master's Thesis)	113
Modulbeschreibung	113
Nicht mehr angebotene Module	114
Big-Data-Systeme und Anwendungen (en. Big-Data Systems and Applications)	115
Introduction to Linear Optimization (en. Introduction to Linear Optimization)	117
Judgment Aggregation (en. Judgment Aggregation)	119
Master-Seminar: System-Software für Big-Data-Computing (en. Master's Seminar: System Software for Big Data Computing)	121

Qualifikationsziele des Studiengangs Master Informatik

Nach Abschluss des Master Studiums sollen die Studierenden folgende Qualifikationsziele erreicht haben.

Wissenschaftliche Befähigung und Befähigung zu einer qualifizierten Erwerbstätigkeit

Absolvent*innen

- können auch komplexere Methoden und Verfahren der Informatik selbständig beurteilen, vergleichen und anwenden.
- sind in der Lage, wissenschaftliche Methoden selbstständig zu analysieren, zu evaluieren und anzuwenden.
- haben sich mit aktuellen fachwissenschaftlichen Themen in der theoretischen und/oder praktischen Informatik auseinandergesetzt und können hier eigenständig Problemstellungen lösen und evaluieren.
- können eine umfangreiche Forschungsarbeit eigenständig durchführen, in der sie erlerntes Wissen unter Anwendung von Forschungsmethoden auf eine abgeleitete Forschungsfrage unter Berücksichtigung allgemein anerkannter Grundsätze guter wissenschaftlicher Praxis angewenden und deren Nutzen beurteilt haben.

Persönlichkeitsentwicklung

Absolvent*innen

- haben sich in ihren gewählten Schwerpunkten vertieftes Wissen angeeignet.
- können auch komplexe Methoden und Verfahren der Informatik anderen Personen in Wort und Schrift erläutern und berücksichtigen dabei auch die Sichtweisen Anderer und gehen auf diese ein.
- können ihre eigenen Fähigkeiten realistisch einschätzen, haben konkrete Vorstellungen entwickelt, wie sie sich beruflich weiterentwickeln möchten und können sich selbständig umfangreiches Fachwissen zielorientiert erschließen.
- kennen die Herangehensweisen an ethische Fragen und Herausforderungen aus Sicht der Informatik sowie die gesellschaftliche, kulturelle und politische Bedeutung ihrer Fachdisziplin.

Studienverlaufspläne

Auf den Webseiten des Instituts für Informatik finden Sie den [Studienverlaufsplan](#). Dieser dient der Orientierung und hilft Ihnen bei der Planung Ihres Studiums.

Lehreinheiten für Wahlpflichtbereiche (en. Courses for elective areas)

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.

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*. Im Folgenden zusammengefasst als *Voraussetzungen für den Vorgriff auf Mastermodule*.

Advanced topics in Bayesian Data Science (en. Advanced topics in Bayesian Data Science)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	English

Inhalte

- Introduction to Bayesian hierarchical models, Bayesian inference and empirical Bayesian inference, Bayesian Monte Carlo techniques and MCMC calculations
- Introduction to Bayesian meta-analysis, Bayesian meta-analysis for diagnostic test data, Bayesian evidence synthesis for deterministic models
- Exploratory data analysis of multilevel and longitudinal data, Bayesian hierarchical linear models, Bayesian hierarchical generalized linear models
- Bayesian models for statistical regression problems, Bayesian linear regression with a large number of covariates, hierarchical pre-distributions and variable selection
- Bayesian models for exploratory multivariate analysis, a priori distributions for variance-covariance matrices, multivariate comparisons of groups and other multivariate methods
- Bayesian methods to deal with missing data, Bayesian approaches to missing random data, Bayesian approaches to missing non-random data
- Bayesian methods of censoring and truncation, modeling survival and time to event data, handling truncated data
- Bayesian non-parametric models, introduction to mixture models, statistical modeling with Dirichlet process mixtures

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- describe the difference between Empirical Bayes (EB) and full Bayesian analysis in hierarchical models,
- compare EB and full Bayesian approaches in multi-level regression, logistic regression and multivariate analysis,
- perform data analysis for aggregated data using Bayesian hierarchical models,
- develop Bayesian hierarchical models for multi-level and longitudinal data,
- develop scripts in R, OpenBUGS and JAGS software for hierarchical Bayesian statistical modelling,
- describe different Bayesian hierarchical models for different types of outcomes (e.g. continuous, categorical, ordinal categorical, time-to-event),
- perform Bayesian non-parametric data analysis for different types of data outcomes,
- characterize different types of missing data (MCAR, MAR, MNAR) and different types of modeling missing data (e.g. not-ignorability missing mechanisms), and
- implement and develop Bayesian hierarchical models in new situations of data analysis.

Literatur

- Andrew Gelman, John B. Carlin, Hal S. Stern: Bayesian Data Analysis. Chapman & Hall. 2013.
- David Lunn, Chris Jackson, Nicky Best, Andrew Thomas, David Spiegelhalter. The BUGS Book (Chapman & Hall/CRC Texts in Statistical Science). 2012

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of module *Introduction to Statistical Analysis through Computer Simulations* or a very good background knowledge in the fields of probability theory, statistics and the use of the statistical software R and OpenBUGS or JAGS.

Voraussetzungen für die Vergabe von Leistungspunkten

- active participation in the exercises
- successful completion of the exercises
- passing the final exam (usually written)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Martin Lercher, Dr. Pablo Verde

Algorithmen für Ad-hoc- und Sensornetzwerke (en. Algorithms for Ad-hoc and Sensor networks)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen:

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

Zusätzliche Bemerkungen

- 15 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- verteilte Berechnungen in Ad-hoc-Netzwerken formulieren
- verschiedene Routing-Techniken in Ad-hoc-Netzwerken durchführen
- Topologiekontrolle auf Ad-hoc-Netzwerke anwenden
- Standorte und virtuelle Addressierungen in Ad-hoc-Netzwerke bestimmen

Literatur

- Dorothea Wagner und Roger Wattenhofer: Algorithms for Sensor and Ad Hoc Networks. Springer, LNCS 4621. Berlin, 2007.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof Dr. Egon Wanke

Algorithmen für schwere Probleme (en. Algorithmics for Hard Problems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

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 (Bäume und Co-Graphen)
- Parametrisierte Algorithmen (Standardparameter, Parametrisierung in der Baumweite, Parametrisierung in der Cliquesweite)
- Exakte Exponentialzeitlgorithmen

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die besprochenen schweren Probleme erläutern und formal definieren,
- die behandelten Algorithmen für schwere Probleme erklären und auf konkrete Eingaben anwenden,
- eine parametrisierte Laufzeitanalyse von Algorithmen durchführen und
- die betrachteten Baumstrukturen für gegebene Instanzen 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 die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Mitarbeit in den Übungen
- Abgabe der Übungsaufgaben
- Bestehen der Klausur zum Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Priv.-Doz. Dr. Frank Gurski

Algorithmen und Datenstrukturen 2 (en. Algorithms and Data Structures 2)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen:

- Verketteten und Aufteilen von balancierten Suchbäumen
- Median der Mediane
- Vorrangwarteschlangen (Linksbäume, Binomial Queues, Fibonacci Heaps, Amortisierte Laufzeitanalyse)
- Algorithmische Entwurfstechniken (greedy, teile und herrsche, dynamische Programmierung)
- Low-Memory Algorithmen (In Place Modelle, Read-Only Modelle, Streaming Modelle, Min Count Sketch Ansatz)
- Preprocessing (Minimum Bereichsanfragen, Kartesische Bäume, Fischer Heun Schema, 4 Russen Methode)
- Algorithmische Geometrie (flächenmaximale Dreiecke in Polygonen, kleinste Umschließende Kreise)
- Dynamische Algorithmen (Zusammenhang für Graphen, konvexe Hülle)
- Parallele Algorithmen (PRAM Modell, Netzwerkmodell, parallele Komplexitätsklassen, parallele Basisalgorithmen)
- Parameterisierte Algorithmen (Fixed Parameter Tractability, W-Hierarchie)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- fortgeschrittene theoretische und praktische Konzepte von komplexen Algorithmen und Datenstrukturen einsetzen,
- effiziente Algorithmen und Datenstrukturen für kombinatorische Probleme aus verschiedenen Bereichen selber entwerfen und analysieren, un
- eine breite Palette fortgeschrittener Techniken und Strategien zur Lösung algorithmischer Probleme anwenden.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof Dr. Egon Wanke

Algorithmische Spieltheorie (en. Algorithmic Game Theory)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	unregelmäßig	Bachelor-Studiengang Informatik	Deutsch

Inhalte

In diesem Modul werden die wichtigsten Grundlagen, Modelle, Methoden und Resultate aus dem Gebiet der algorithmischen Spieltheorie vermittelt. Dabei unterscheidet man zwischen nichtkooperativen und kooperativen Spielen und untersucht insbesondere Gleichgewichte und Stabilitätskonzepte, um den Ausgang von Spielen vorherzusagen. Die zugehörigen Probleme werden hinsichtlich ihrer algorithmischen und Komplexitätstheoretischen Eigenschaften untersucht.

Nichtkooperative Spiele: Gegeneinander spielen

- Grundlagen (Normalform, dominante Strategien und Gleichgewichte, Zwei-Personen-Spiele)
- Nash-Gleichgewichte in gemischten Strategien (Definition und Eigenschaften; Existenz)
- 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 und 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)

Zusätzliche Bemerkungen

- 15 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die behandelten Zwei-Personen-Spiele in Normalform und ihre Eigenschaften (dominante Strategien, Nash-Gleichgewichte, Pareto-Optimalität usw.) beschreiben bzw. begründen,
- Nash-Gleichgewichte in gemischten Strategien berechnen,
- die wichtigsten Schritte des (mit einem Nobelpreis gewürdigten) Satzes von Nash über die Existenz von Gleichgewichten in gemischten Strategien erläutern,
- Gleichgewichte in Spielbäumen bestimmen,
- die vorgestellte einfache Poker-Variante analysieren,
- die Komplexität von Problemen im Zusammenhang mit Nash-Gleichgewichten erläutern,
- die behandelten kooperativen Spiele und ihre Eigenschaften erläutern bzw. begründen,

- die Repräsentation und Eigenschaften (insbesondere Stabilitätskonzepte) von einfachen Spielen erläutern (z. B., ob ihr Kern leer ist),
- die Macht-Indizes in gewichteten Wahlspielen erläutern und berechnen und
- die Komplexität der behandelten Probleme für gewichtete Wahlspiele, Spiele auf Graphen und hedonische erläutern.

Literatur

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- 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). Heidelberg, 2011. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Algorithms for Sequence Analysis (en. Algorithms for Sequence Analysis)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Englisch

Inhalte

This lecture addresses classic as well as recent advanced algorithms for the analysis of sequences, with an emphasis on algorithms that are fast in practice. DNA sequencing data are one example that motivates this lecture, but the focus of this course is on algorithms and concepts that are not specific to bioinformatics.

- Full text search without index
- Approximate pattern matching
- Suffix trees, enhanced suffix arrays, and linear-time construction algorithms
- Burrows-Wheeler Transform (BWT) and the FM index
- De Bruijn graphs and overlap graphs
- Data compression
- Advanced pairwise alignment algorithms
- Multiple sequence alignment
- Positional Burrows-Wheeler Transform (PBWT)
- Locality sensitive hashing

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- state concepts for different data structures to index strings
- explain algorithms to build these data structures
- explain different algorithms for text search and sequence alignment
- design and implement advanced algorithms for sequence analysis
- analyze and discuss the merits of different algorithms both in terms of asymptotic runtime and practical performance

Literatur

- Ohlebusch: Bioinformatics Algorithms. Oldenbusch Verlag. 2013.
- Navarro and Raffinot: Flexible Pattern Matching in Strings. Cambridge University Press. 2002.
- Mäkinen, Belazzougui, Cunial, Tomescu: Genome-Scale Algorithm Design. Cambridge University Press. 2015.
- Gusfield: Algorithms on Strings, Trees, and Sequences. Cambridge University Press. 1997.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Wahlpflichtbereich Praktische oder Technische Informatik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Individuelle Ergänzung

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte der Module *Programmierung, Algorithmen und Datenstrukturen* und *Theoretische Informatik*

Voraussetzungen für die Vergabe von Leistungspunkten

- The weekly exercise sheets come with programming tasks and theory tasks. To be eligible to take the exam, students have to earn 50% of programming tasks and 50% of theory tasks.
- oral exam (usually 20 minutes)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Tobias Marschall

Approximationsalgorithmen (en. Approximation Algorithms)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Diese Vorlesung ist eine weiterführende Algorithmenvorlesung, die die grundlegenden Konzepte aus dem Bereich der Approximationsalgorithmen einführt. Approximationsalgorithmen sind polynomiell zeitbeschränkte Algorithmen für schwere Optimierungsprobleme, für die eine beweisbare Gütegarantie für die Lösungen gilt. Besprochen werden Techniken für den Entwurf und die Analyse von Approximationsalgorithmen, insbesondere:

- gierige Algorithmen
- Approximationsalgorithmen für Graphprobleme
- Probleme mit euklidischen vs metrischen vs nicht-metrischen Eingaben
- (I)LP-basierte Algorithmen
- polynomielle Approximationsschemata

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- das Konzept „Approximationsgüte“ erklären und Ergebnisse aus diesem Bereich einordnen,
- die besprochenen Optimierungsprobleme formal definieren,
- komplexe Beweise geeignet zusammenfassen und erklären,
- einfache Techniken für die Analyse von Approximationsgüten anwenden, und
- einfache Entwurfstechniken für Approximationsalgorithmen anwenden.

Literatur

- Vijay Vazirani: Approximation Algorithms. Springer-Verlag, Berlin, 2003. 1. Ausgabe
- David P. Williamson and David B. Shmoys: The Design of Approximation Algorithms, Cambridge University Press, Cambridge, 2011, 1. Ausgabe.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Melanie Schmidt, Dr. Daniel Schmidt

Approximationsalgorithmen für Clusteringprobleme (en. Approximation Algorithms for Clustering Problems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch Deutsch

Inhalte

Die Vorlesung ist eine fortgeschrittene Algorithmenvorlesung, die sich mit Clusteringalgorithmen beschäftigt. Der Fokus liegt auf Approximationsalgorithmen. Insbesondere werden folgende Themen behandelt:

- LP-basierte Techniken
- Lokale Suche
- Approximationsalgorithmen für das k-median Problem und für das k-means Problem
- Clustering im Euklidischen Raum
- Dimensionsreduktion und Datenstromalgorithmen für Clusteringprobleme

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- lineare Programme für bekannte und neue Clusteringprobleme aufstellen,
- fortgeschrittene Analysetechniken für bekannte Clusteringprobleme wiedergeben,
- die besprochenen algorithmischen Ideen für das Design von Clusteringalgorithmen erklären und
- die Approximationsgüte von bekannten und leicht abgewandelten Clusteringverfahren analysieren.

Die Unterrichtssprache des jeweiligen Semesters wird in HIS/LSF angekündigt.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.
- Eigenes Skript

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Es ist hilfreich, die Inhalte der Module *Approximationsalgorithmen* und *Kombinatorische Algorithmen für Clusteringprobleme* zu kennen.

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- abschließende Prüfung (schriftlich oder mündlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Melanie Schmidt

Betriebssystem-Entwicklung (en. Operating System Development)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (1 SWS) Übung (3 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

Inhalte

Die Vorlesung vermittelt grundlegende Konzepte, die für den Bau eines x86-basierten 64-Bit Betriebssystems in den Übungen erforderlich sind. Dadurch können auch Studierende teilnehmen, die nicht *Betriebssysteme und Systemprogrammierung* im Bachelor gehört haben.

Inhalte der Vorlesung: - Bootvorgang - x86_64 Programmiermodell - Interrupts (PIC und APIC) - Koroutinen und Threads - Scheduling - Synchronisierung - Treiberarchitekturen

Der Schwerpunkt des Moduls liegt auf den Übungen, in denen in Einzelarbeit ein einfaches x86-basiertes 64-Bit Betriebssystem von Grund auf entwickelt wird. Dies erfolgt wahlweise in C++ oder Rust und mithilfe aufeinander aufbauender Aufgaben.

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Programmiersprache C++ oder Rust für die hardwarenahe Programmierung anwenden,
- grundlegende Betriebssystemfunktionen selbst entwickeln, und
- die nebenläufigen Vorgänge (Interrupts und Threads) in einem Betriebssystem selbst programmieren und synchronisieren.

Literatur

- Intel Corporation, Intel Architecture Software Developer's Manual.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Prüfung über das eigene Betriebssystem, welches die Funktionen aus den Übungsaufgaben realisieren muss.

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Create Your Tech Startup (en. Create Your Tech Startup)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Englisch

Inhalte

Students explore the entrepreneurial process using a learning by doing methodology.

- The lecture and case study sessions provide and discuss tools and methods of creating, visualizing, and analyzing digital business models (e.g., business model canvas, lean startup, design thinking).
- In a group project, students transfer and apply those tools and methods to create, evaluate, plan, and pitch their own tech startup ideas.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- model, analyze, and discuss digital business models and its components,
- assess the specific opportunities for and challenges of technology-based businesses,
- create, plan, and implement novel tech startups,
- pitch their startup idea in front of peers and experts,
- present, assess, and give feedback to novel tech business models,
- assess their entrepreneurial skills and
- collaborate with interdisciplinary peers comprising various competences.

Literatur

- Blank, S.; Dorf, B.: The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company. K & S Ranch, 2012. 1st edition
- Osterwalder, A; Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, 2010. 1st edition
- Osterwalder, A; Pigneur, Y.; Bernarda, G; Smith, A.: Value Proposition Design: How to Create Products and Services Customers Want. John Wiley & Sons, 2014. 1st edition
- Ries, E.: The Lean Startup: How Constant Innovation Creates Radically Successful Businesses, Portfolio Penguin, London. 2011. 1st edition
- Further literature is provided during the course.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- (Gruppen-)Präsentationen zum entwickelten Tech Startup
- Schriftliche Ausarbeitung eines Geschäftsplans
- Aktive Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Steffi Haag

Discrete Optimization: Theory and Applications (en. Discrete Optimization: Theory and Applications)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	120 Stunden	180 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

The lecture covers the basic theory and applications of (integer) linear and discrete optimization. In the exercises the content of the lecture is applied and deepened. For that the exercises contain theoretical as well as practical elements. The students use linear, integer linear programming and other discrete optimization modeling software and solvers to solve applied programming exercises.

- Foundations of Linear Programming
- Linear Programs and their geometric interpretation
- Duality
- The Simplex method
- Integer Linear Programming
- Linear programming-based Branch-and-Bound
- Cutting planes and Branch-and-Cut
- Network flows
- Satisfiability: Modeling and SAT solvers
- Constraint Programming
- Applications: Selected applications of discrete optimization techniques
- Project work on selected discrete optimization problem

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- state the basic principles of linear and integer linear programming (LP and ILP) as well as SAT and constraint programming
- build practical LP and discrete optimization models
- design and implement algorithms that solve these models

Literatur

- Vašek Chvátal: Linear Programming. W.H. Freeman, 1983
- Dimitris Bertsimas, John N. Tsitsiklis: Introduction to Linear Optimization. Athena Scientific. Belmont, MA, USA, 1997.
- Laurence Wolsey: Integer Programming. Wiley. New York, NY, USA, 1998.
- Dan Gusfield: Integer Linear Programming in Computational and Systems Biology. Cambridge University Press, 2019

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Wahlpflichtbereich Praktische oder Technische Informatik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

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

Teilnahmevoraussetzungen

- Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Das Modul kann nicht belegt werden, wenn das Vorgängermodul „Introduction to Linear Programming“ erfolgreich absolviert wurde.

Voraussetzungen für die Vergabe von Leistungspunkten

- active participation
- successful completion of exercise sheets and project work
- passing written exam

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Gunnar W. Klau

Dynamische Programmiersprachen (en. Dynamic Programming Languages)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

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 (Selbststudium) 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.

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

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

Literatur

- Robert Nystrom: *Crafting Interpreters*. Genever Benning. 2021. 1st Edition

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Grundlagen des Moduls *Compilerbau*

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Dr. John Witulski

Einführung in die statistische Analyse mittels Computersimulationen (en. Introduction to Statistical Analysis through Computer Simulations)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
5 LP	150 Stunden	60 Stunden	90 Stunden
Vorlesung (2 SWS) Übung (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

- Introduction into Statistical Modeling and Bayesian thinking (Probability and introduction to computer simulations, Introduction to Bootstrapping, Introduction to Bayesian Modeling, Introduction to multivariate probability distributions and multiple parameter models)
- Monte Carlo simulation methods (Monte Carlo method for calculating integrals, Rejection sampling, Importance sampling, Sampling importance re-sampling)
- Markov chain Monte Carlo methods (Introduction to Markov chains, Metropolis-Hastings algorithm, Directed acyclic graphs, Gibbs-sampling, MCMC output analysis)
- Statistic modeling (Regression modeling, Analysis of multiple contingency tables, Introduction to hierarchical models)

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- represent the role of Bayesian Analysis in a broad spectrum of data analysis,
- describe and use different prior distributions in Bayesian Analysis,
- delineate simple and advanced Monte Carlo simulation methods. In particular the application of Bootstrap methods in classical data analysis and Markov Chain Monte Carlo (MCMC) in Bayesian analysis,
- compare classical and Bayesian approaches in regression, logistic regression and multivariate analysis,
- apply computer simulations in classical and Bayesian data analysis,
- use R, OpenBUGS and JAGS software for Bayesian statistical modeling,
- apply simple Bayesian hierarchical models in data analysis, and
- implement and develop Bayesian models in new situations of data analysis.

Literatur

- Andrew Gelman, John B. Carlin, Hal S. Stern: Bayesian Data Analysis. Chapman & Hall. 2013.
- David Lunn, Chris Jackson, Nicky Best, Andrew Thomas, David Spiegelhalter. The BUGS Book (Chapman & Hall/CRC Texts in Statistical Science). 2012

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich

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

Teilnahmevoraussetzungen

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Voraussetzungen für die Vergabe von Leistungspunkten

- active participation in the exercises
- successful completion of the exercises
- passing the final exam (usually a project presentation)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Pablo Verde, Prof. Dr. Martin Lercher

Gerechte Aufteilungsverfahren (en. Fair Division)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	unregelmäßig	Bachelor-Studiengang Informatik	Deutsch

Inhalte

In diesem Modul werden die wichtigsten Grundlagen, Modelle, Methoden und Resultate aus dem Gebiet der gerechten Aufteilungsverfahren von Gütern oder Ressourcen vermittelt. Dabei unterscheidet man zwischen der Aufteilung von teilbaren und unteilbaren Gütern/Ressourcen und untersucht insbesondere Methoden, die bestimmte Konzepte der Fairness (wie Proportionalität oder Neidfreiheit) für jede Aufteilung garantieren, egal welche Bewertungen die beteiligten Agenten für welche Teile der aufzuteilenden Güter/Ressourcen individuell haben. Die Aufteilung einer teilbaren Ressource ist auch als "Cake-cutting" bekannt. Die zugehörigen Probleme werden auch hinsichtlich ihrer algorithmischen und Komplexitätstheoretischen Eigenschaften untersucht.

Cake-Cutting: Gerechte Aufteilung teilbarer Güter

- Grundlagen
- Bewertungskriterien (Fairness; Effizienz; Manipulierbarkeit; Laufzeit)
- Zwei neidfreie Cake-cutting-Protokolle für zwei Spieler
- Proportionale und superproportionale Cake-cutting-Protokolle für n Spieler
- Ungleiche Portionen
- Neidfreie Protokolle für drei und vier Spieler
- „Dirty-Work“-Protokolle
- Minimierung der Anzahl der Schnitte
- Grad der garantierten Neidfreiheit

Gerechte Aufteilung unteilbarer Güter

- Definition und Klassifikation von Allokationsproblemen
- Präferenzenerhebung und kompakte Darstellung
- Pareto-Effizienz und Neidfreiheit
- Relaxationen der Neidfreiheit
- Maximierung der sozialen Wohlfahrt: Das Weihnachtsmann-Problem
- Zentralisierte gerechte Aufteilung mit ordinalen Präferenzen
- Gerechte Aufteilung mit Geld und verwandte Probleme
- Dezentralisierte Allokationsprotokolle

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die behandelten Bewertungskriterien für die betrachteten Protokolle beschreiben und an Beispielen erklären,
- die behandelten Protokolle eigenständig anwenden und die entstehenden Aufteilungen sowie ihre Eigenschaften bestimmen,
- die Zusammenhänge zwischen den vorgestellten Eigenschaften (z. B. Pareto-Effizienz versus Neidfreiheit) erläutern,
- die wichtigsten Resultate dieser Vorlesung erläutern und die Argumentation in den Beweisen an Beispielen erklären,
- die behandelten Methoden der Präferenzenerhebung und kompakten Darstellung für Instanzen der

- Aufteilung unteilbarer Güter erläutern,
- Relaxationen der Neidfreiheit an Beispielen erläutern, und
 - die Komplexität von Problemen der Maximierung der sozialen Wohlfahrt (insbesondere des Weihnachtsmann-Problems) erläutern.

Literatur

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- 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). Heidelberg, 2011. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Graphenalgorithmen 2 (en. Algorithms for Graphs 2)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen:

- Planare Graphen (Erkennung, Dualität, Färbungsprobleme, ...)
- Chordale Graphen (lexikographische Breitensuche, perfekte Eliminationsordnung, ...)
- Baumweite beschränkte Graphen
- Cliquesweite, Rangweite, NLC-Weite
- Monadische Logik 2. Ordnung für relationale Graphenstrukturen
- Dynamische Programmierung auf Baumstrukturierte Graphen
- Minorensatz
- Extremale Graphentheorie
- Distanzerhaltende Graphen / Cographen

Zusätzliche Bemerkungen

- 15 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- verschiedene Graphklassen untereinander einordnen und ihre Erkennungskomplexität ermitteln
- Weiteparameter von Graphklassen vergleichen
- Graphenprobleme mit logischen Formeln für relationale Graphenstrukturen ausdrücken
- Graphklassen über Graphoperationen definieren

Literatur

- Krishnaiyan „KT“ Thulasiraman, Subramanian Arumugam , Andreas Brandstädt , Takao Nishizeki: Handbook of Graph Theory, Combinatorial Optimization, and Algorithms. Chapman & Hall/CRC, Computer & Information Science Series. 2012.
- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof Dr. Egon Wanke

Growth Mechanics (en. Growth Mechanics)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Semester	Master-Studiengang Informatik	Englisch

Inhalte

With the help of selected examples, the course describes the modeling and analysis of cellular reaction networks, combining multidisciplinary methods from Mathematics, Biology, Physics, Chemistry, and Computer Science.

- Introduction to the mathematical modeling and analysis of cellular reaction networks as an optimal resource allocation problem
- The main physical constraints on cellular growth and replication
- Growth modeling: self-replicator models and nonlinear optimization
- Growth Economy and Control Analysis: the marginal value of reactions and specific costs and benefits
- Growth Balance Analysis: the analytical conditions for optimal cellular growth
- Principles of nonlinear numerical optimization

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- describe the principles of mathematic modeling of cellular reaction networks,
- give various examples on how physical constraints on cellular growth and replication can be modeled,
- build and analyze self-replicator cellular models using the GNU R or Python programming languages,
- describe basic concepts behind linear and nonlinear numerical optimization and
- highlight the specific differences and computational challenges of these concepts.

Literatur

- Relevant literature is provided during the course.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Students should be comfortable with the fundamentals of multivariate Calculus and Linear Algebra, and should be proficient with a programming language.

Voraussetzungen für die Vergabe von Leistungspunkten

- at least 50% of the points from the exercises
- passing final exam (written exam, usually 90 minutes)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Hugo Dourado, Prof. Dr. Martin Lercher

Information Theory (en. Information Theory)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch

Inhalte

In this module students obtain a a better understanding of the concepts of information and randomness, as captured by the concepts of entropy, information loss and Kolmogorov complexity. The relation to coding, data compression and signal transmission is worked out. They are led to a principled view on general assumptions and techniques in machine learning and statistics, like the information bottleneck or the maximum likelihood method.

- What is information?
- Entropy, conditional, joint and relative entropy, mutual information and their interrelations,
- relation to coding, Source Coding Theorem
- characterizations of entropy and relative entropy
- Kolmogorov complexity and connection to entropy, applications of Kolmogorov complexity
- entropy of stochastic processes, channel capacity and Channel Coding Theorem
- differential entropy, Gaussian channel
- maximum entropy distributions
- information theory and statistics: maximum likelihood and exponential families
- applications

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- define and relate the basic notions of information theory,
- judge the plausibility of previously unseen relations between these notions and if true prove them,
- carry out coding algorithms and explain what they achieve and
- use information theoretic principles to analyze and justify methods of machine learning and statistics

Literatur

- Cover, Thomas: Elements of Information Theory. Wiley-Interscience. 2012. 2nd ed.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- successful and active participation in the exercise classes
- passing the exam (written or oral exam)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Peter Arndt

Isolation und Schutz in Betriebssystemen (en. Isolation and Protection in Operating Systems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (1 SWS) Übung (3 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Deutsch

Inhalte

In der Vorlesung liegt der Schwerpunkt auf Isolation und Schutz durch Prozessadressräume. Es werden Techniken zur Trennung logischer Adressräume (seiten- und segmentbasiert) sowie Systemaufrufe behandelt. Vor diesem Hintergrund werden auch verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen sind die Interprozesskommunikation durch Nachrichten und gemeinsamen Speicher.

Im Rahmen der Übungen werden ausgewählte Vorlesungsinhalte praktisch in Rust (oder C++) angewandt. Ausgehend von einem x86-basierten 64 Bit Einkern-Betriebssystem werden Privilegienisolation, räumliche Isolation und fortgeschrittene Adressraumkonzepte implementiert. Das Ausgangsbetriebssystem wird bereitgestellt, alternativ kann das eigene Betriebssystem aus dem Modul *Betriebssystem-Entwicklung* verwendet werden.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- die Programmiersprache Rust (oder C++) für die hardwarenahe Programmierung anwenden,
- Entwurfsprinzipien für Systemaufrufe erläutern und implementieren,
- grundlegende Mechanismen zur virtuellen und physikalischen Speicherverwaltung implementieren,
- Schutzmechanismen in einem Betriebssystem erläutern, vergleichen und entwickeln,
- Mechanismen und Abstraktionen zur Interprozesskommunikation vergleichen und programmieren sowie
- Betriebssystem-Architekturen erklären und vergleichen.

Literatur

- Intel Corporation, Intel Architecture Software Developer's Manual.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Prüfung über das eigene Betriebssystem, welches die Funktionen aus den Übungsaufgaben realisieren muss.

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Knowledge Discovery in Databases - ausgewählte Themen (en. Knowledge Discovery in Databases - Selected Topics)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Aufbauend auf dem Modul *Knowledge Discovery in Databases* werden hier weiterführende Verfahren behandelt, mit denen aktuelle Entwicklungen in der Forschung berücksichtigt werden sollen. Mögliche Themen sind zum Beispiel:

- Temporal Association Rule Mining
- Zeitreihen: Outlier Detection, Clustering und Finden wiederkehrender Muster
- weitere Clustering-Verfahren (z. B. Subspace-Clustering)
- Cluster Validity Indexes
- Clustering mit fehlenden Werten

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden - spezielle Verfahren in den behandelten Themengebieten bewerten und kritisch vergleichen und - anhand der Charakteristik des zu lösenden Problems geeignete Verfahren auszuwählen und, wenn erforderlich, kombinieren.

Literatur

- Keine

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte des Moduls *Knowledge Discovery in Databases*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiche Bearbeitung der Übungsaufgaben
- abschließende Prüfung (i. d. R. mündlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

Knowledge Discovery in Databases (en. Knowledge Discovery in Databases)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

In diesem Module werden unterschiedliche Formen des Knowledge Discovery (bzw. Data Mining und/oder maschinellen Lernens) sowie dafür grundlegende Algorithmen eingeführt und diskutiert. Schwerpunkte bilden Clustering-Verfahren, Klassifikationsverfahren sowie Verfahren zum Finden von Assoziationsregeln. Darüber hinaus werden exemplarisch Techniken zur effizienten Umsetzung solcher Verfahren auf großen Datenmengen behandelt. Die konkreten Inhalte sind:

- Grundlagen der Statistik
- Clustering (partitionierende, dichte-basierte und hierarchische Clustering-Verfahren)
- Klassifikation (Bewertung von Klassifikationsverfahren; Bayes, kNN, Entscheidungsbaum, Support-Vector-Machine)
- Assoziationsregeln (Frequent-Itemsets, Apriori-Algorithmus, effiziente Speicherung, hierarchische Assoziationsregeln)
- Leistungssteigerungstechniken für KDD-Algorithmen für große Datenmengen
- Weitere Mining-Verfahren und -Anwendungen im Überblick

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden: - die verschiedenen Arten des Knowledge Discovery erläutern, - konkrete Methoden vergleichen und bewerten und - entscheiden, welche Methode in welcher Situation sinnvoll eingesetzt werden kann.

Literatur

- M. Ester, J. Sander: Knowledge Discovery in Databases: Techniken und Anwendungen. Springer Verlag, 2000.
- M.H. Dunham: Data Mining – Introductory and Advanced Topics, Prentice Hall. 2003.
- P.-N. Tan, M. Steinbach, V. Kumar: Introduction to Data Mining. Addison Wesley. 2006.
- J. Han, M. Kamber: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. 2000.
- M. Berthold, D.J. Hand (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

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

Kryptokomplexität 2 (en. Cryptocomplexity 2)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
10 LP	300 Stunden	90 Stunden	210 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (4 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

In diesem Modul werden vertiefte Kenntnisse der Kryptologie vermittelt, einige zentrale Public-Key-Kryptosysteme und digitale Signaturschemata vorgestellt und ihre Sicherheit diskutiert, die häufig auf der Komplexität geeigneter Probleme beruht. Deshalb werden auch anspruchsvolle Themen der Komplexitätstheorie behandelt, insbesondere mit dem Ziel, Methoden zur komplexitätstheoretischen Klassifizierung von wichtigen Problemen verstehen und anwenden zu können. Besonderer Wert wird darauf gelegt, dass die Studierenden die enge Verflechtung dieser Gebiete verstehen.

Vertiefung der Kryptologie

- Wiederholung: Aufgaben und Ziele der Kryptologie, einige mathematische Grundlagen und RSA
- Diffie-Hellman und das Problem des diskreten Logarithmus
- Die Protokolle von ElGamal (das Public-Key-Kryptosystem und digitale Signaturen von ElGamal und ihre Sicherheit)
- Rabins Public-Key-Kryptosystem
- Arthur-Merlin-Spiele und Zero-Knowledge
- Weitere Protokolle

Vertiefung der Komplexitätstheorie

- Wiederholung: Aufgaben und Ziele der Komplexitätstheorie
- Randomisierte Algorithmen und Komplexitätsklassen (PP, RP und ZPP: Monte-Carlo- und Las-Vegas-Algorithmen; Quantoren und BPP; Graphisomorphie und die Arthur-Merlin-Hierarchie)
- Die Boolesche Hierarchie (Probleme in DP; Struktur und Eigenschaften der Booleschen Hierarchie über NP; exakte Graphfärbbarkeit)
- Die Polynomialzeit-Hierarchie (Orakel-Turingmaschinen; Struktur und Eigenschaften der Polynomialzeit-Hierarchie; vollständige Probleme; die Boolesche Hierarchie kollabiert die Polynomialzeit-Hierarchie)

Zusätzliche Bemerkungen

- 15 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Idee der Public-Key-Kryptographie erläutern,
- mathematische (insbesondere zahlentheoretische) Rechnungen, die bei der asymmetrischen Ver- und Entschlüsselung sowie bei digitalen Signaturen verwendet werden, eigenständig durchführen und erläutern,
- die Sicherheit von Public-Key-Kryptosystemen (RSA, ElGamal usw.) und digitalen Signaturen bewerten,
- begründen, weshalb welche der besprochenen Kryptosysteme unter welchen Bedingungen unsicher sind,
- erläutern, welche Gegenmaßnahmen es dagegen gibt,

- die Motivation und Definition von Komplexitätshierarchien erläutern,
- die Komplexität von natürlichen Problemen beschreiben und begründen,
- selbstständig Reduktionen zwischen Problemen zum Nachweis unterer Schranken entwerfen und ihre Korrektheit nachweisen und
- Zusammenhänge zwischen Hierarchien von Komplexitätsklassen sowie zu kryptologischen Anwendungen erkennen und erläutern.

Literatur

- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität. eXamen.Press. Springer-Verlag, Heidelberg, 2008. 1. Auflage.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity. EATCS Texts in Theoretical Computer Science, Springer-Verlag. Heidelberg, 2005. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Master-Seminar: Aktuelle Themen der theoretischen und computergestützten Biologie (en. Master's Seminar: Current Topics in Theoretical and Computational Biology)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
2 LP	60 Stunden	15 Stunden	45 Stunden
Seminar (1 SWS)	jedes Semester	Master-Studiengang Informatik	Englisch

Inhalte

- Modeling of biological systems
- Bioinformatics algorithms
- Methods for evaluating complex biological data

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- outline current topics in theoretical and computational biology,
- present scientific content in a lecture, and
- conduct an oral scientific discourse in English.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Erfolgreicher Vortrag eines Projektes
- Aktive Teilnahme an den Diskussionen während des Seminars

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Martin Lercher

Master-Seminar: Algorithmen für graphentheoretische Konzepte in der Informatik (en. Master's Seminar: Algorithms for Graph Theoretical Concepts in Computer Science)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Übung (2 SWS) Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

In diesem Seminar werden speziell ausgewählte Themen aus dem Bereich Algorithmen für graphentheoretische Konzepte in der Informatik behandelt. Die Auswahl variiert von Veranstaltung zu Veranstaltung und orientiert sich an den aktuellen Forschungsthemen der Arbeitsgruppe.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- aktuelle Themen im Bereich von Graphenalgorithmen überprüfen und bewerten
- wissenschaftliche Publikationen systematisch aufarbeiten
- Inhalte aus der Literatur strukturiert zusammenfassen und präsentieren
- wissenschaftliche Literatur kritisch analysieren

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen
- Inhaltlich: Inhalte der Module Graphenalgorithmen I oder Graphenalgorithmen II

Voraussetzungen für die Vergabe von Leistungspunkten

- schriftliche Ausarbeitung (Zusammenfassung) des gewählten Themas
- angemessene Präsentation des Themas
- aktive Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof Dr. Egon Wanke

Master-Seminar: Algorithmen für perfekte Graphen (en. Master's Seminar: Algorithms for Perfect Graphs)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

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

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbst Einblicke in einige der wichtigsten Themen, Resultate und Methoden aus dem Gebiet der perfekten Graphen erarbeiten,
- einen guten Vortrag vorbereiten und halten,
- die Vorträge der anderen Studierenden hören und kritisch diskutieren und
- eine gute schriftliche Ausarbeitung zu dem Vortragsthema anfertigen.

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 die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive Mitarbeit in den Vorträgen und Übungen
- Abgabe der Übungsaufgaben
- Bestehen der Klausur zum Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Priv.-Doz. Dr. Frank Gurski

Master-Seminar: Betriebssystem-Entwicklung in Rust (en. Master's Seminar: Operating System Development with Rust)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
5 LP	150 Stunden	30 Stunden	120 Stunden
Seminar (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Seminar beschäftigt sich mit Forschungsarbeiten im Bereich der Betriebssystem-Entwicklung mit der Programmiersprache Rust.

Die Sprache Rust gewährleistet, dass Programme sicher sind und dennoch schnell (nebenläufig) ausgeführt werden. Diese Ziele werden zur Übersetzungszeit durchgesetzt, wodurch Leistungseinbussen durch Laufzeit-Überprüfungen vermieden werden. Die Ausführungsgeschwindigkeit von Rust-Programmen ist vergleichbar zu C und C++ Anwendungen, wobei bekannte Probleme wie ungültige Zeiger, Pufferüberläufe und „Iterator-Invalidation“ verhindert werden. Diese Eigenschaften machen Rust interessant für die Betriebssystemen-Entwicklung.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- relevante Inhalte zum gegebenen Thema recherchieren und schriftlich und mündlich wiedergeben,
- Aufsätze von anderen Studierenden evaluieren hinsichtlich Leserbarkeit und technischer Qualität,
- eine Präsentation planen und durchführen,
- sinnvolle Fragen stellen, beantworten und
- konstruktive Kritik annehmen sowie geben.

Literatur

- Relevante Publikationen werden in der Lehrveranstaltung 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 die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zu einem gegebenen Thema
- Teilnahme am Peer-Reviewing

- Präsentation des Themas
- Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Master-Seminar: Combinatorial Optimization in Bioinformatics (en. Master's Seminar: Combinatorial Optimization in Bioinformatics)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Englisch

Inhalte

Many problems in bioinformatics can be formulated as combinatorial optimization problems. This module teaches advanced concepts to recognize, model and solve these problems. Topics are inspired by recent publications in important areas of bioinformatics including

- genomics, transcriptomics, proteomics and related -omics fields
- systems and network biology
- structural biology
- phylogenetics and clustering
- genotype/phenotype screens
- applications to cancer and other medical research as well as plant science and other areas of biology

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- extract significant contents of a given topic and to reproduce them in a written and oral form,
- prepare discussion points to enable a critical discussion on a topic,
- evaluate the readability and technical quality of scientific publications,
- plan and hold a presentation and create a written report,
- ask and answer adequate questions and to give and take constructive feedback.

Literatur

- Selected publications regarding the topic of the course.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Wahlpflichtbereich Praktische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen*, *Theoretische Informatik*, *Mathematik für Informatik 3* und *Wissenschaftliches Arbeiten*

Voraussetzungen für die Vergabe von Leistungspunkten

- Accepting a topic
- Active participation
- Presentation of a topic and preparation of discussion points
- Written report
- Peer Review

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Gunnar W. Klau

Master-Seminar: Computational Multiomics (en. Master's Seminar: Computational Multiomics)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
2 LP	60 Stunden	15 Stunden	45 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch

Inhalte

Modern laboratory equipment enables high-throughput experiments that provide detailed insights into molecular biology. These give rise to several branches of research including genomics, transcriptomics, proteomics, metabolomics, epigenomics, and metagenomics, with -omics meaning “the total of some sort”. Omics experiments produce massive amounts of data that are subject of a variety of challenging computational problems.

In this class,

- various multiomics experiments,
- methods for multiview learning, multimodal analysis, and deep clustering, and
- applications in multiomics data analysis presented and discussed.

Participants will be assigned individual topics. The selection and matching between topics and participants will be determined in the first meeting of the seminar.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- outline the current state of research on multiomics data integration,
- critically disseminate scientific papers,
- give oral presentations on scientific topics, and
- write a scientific review.

Literatur

- Selected publications regarding the topic of the course.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zum zugeteilten Thema

- Halten eines Vortrags

Die finale Note setzt sich zu 75% aus der Bewertung der schriftlichen Ausarbeitung und zu 25% aus der Bewertung des mündlichen Vortrags und der aktiver Mitarbeit zusammen.

Modulbeauftragte und hauptamtlich Lehrende

Jun.-Prof. Dr. Daniel Dörr

Master-Seminar: Datenbanken und Informationssysteme (en. Master's Seminar: Databases and Information Systems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch und Englisch

Inhalte

Im Seminar werden grundlegende und aktuelle Forschungsarbeiten aus den Gebieten „Information Retrieval“, „Natural Language Processing“, „Knowledge Discovery in Databases“ und „Machine Learning“ behandelt, die auf den in den entsprechenden Vorlesungen präsentierten Verfahren aufbauen, diese weiterentwickeln oder neue Verfahren einführen.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden: - aktuelle Methoden nachvollziehen und bewerten, - wissenschaftliche Publikationen systematisch aufarbeiten, - Inhalte aus der Literatur strukturiert zusammenfassen und präsentieren und - wissenschaftliche Literatur kritisch analysieren.

Literatur

- Keine

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte eines der Module *Knowledge Discovery in Databases*, *Knowledge Discovery in Databases - ausgewählte Themen*, *Multimedia-Datenbanksysteme*, *Natural Language Processing* (je nach thematischem Schwerpunkt des Seminars, welcher über HIS/LSF bekannt gegeben wird)

Voraussetzungen für die Vergabe von Leistungspunkten

- schriftliche Ausarbeitung (Zusammenfassung) des gewählten Themas
- angemessene Präsentation des Themas
- aktive Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

Master-Seminar: Deep Learning (en. {Master's Seminar: Deep Learning})

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Im Seminar sollen die Inhalte der Vorlesung *Machine Learning* weiter vertieft und erweitert werden, insbesondere im Bereich Deep Learning. Daher sollen in diesem Seminar verschiedene Themen aus dem Bereich Deep Learning vorgestellt und diskutiert werden. Die Themen beinhalten z. B. Deep Convolutional Networks, Deep Autoencoders, Generative Adversarial Networks, etc.

Jede Woche werden ein oder zwei Themen durch die Teilnehmenden in Form eines Vortrages vorgestellt und diskutiert. Dabei geht es zum einen um die Inhalte, zum anderen darum, wie man einen guten wissenschaftlichen Vortrag hält. Zusätzlich erstellen die Teilnehmenden eine kurze schriftliche Ausarbeitung Ihres Vortrages.

Zusätzliche Bemerkungen

- Aus didaktischen Gründen können maximal 25 Studierende am Seminar teilnehmen.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

Sie verbessern Ihre Fähigkeit einen Vortrag über ein Thema aus diesem Bereich zu halten und

- Methoden und Techniken aus dem Bereich Deep Learning erklären,
- einen Vortrag über ein Thema aus diesem Bereich halten und
- wählen angemessene Methoden für spezifische Fragestellungen und Probleme aus.

Literatur

- Relevante Publikationen werden in der Lehrveranstaltung bekannt gegeben.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte der Module *Machine Learning* und *Mathematik für Informatik 1–3* (oder *Analysis I* und *Lineare Algebra I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- schriftliche Ausarbeitung zum gewählten Thema (2/3 der Note)
- Präsentation des eigenen Themas (1/3 der Note)
- aktive Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Dominik Heider

Master-Seminar: Digital Innovation and Entrepreneurship (en. Master's Seminar: Digital Innovation and Entrepreneurship)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
5 LP	150 Stunden	30 Stunden	120 Stunden
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch/Englisch

Inhalte

Das Seminar behandelt eine Auswahl aktueller Forschungsthemen innerhalb des Bereiches Digitale Innovation und Entrepreneurship.

- Die Studierenden lernen die wichtigsten Theorien, Konzepte, Prozesse, Werkzeuge und Methoden rund um IT-basierte Innovationen, Geschäftsmodelle, und Nutzungserlebnisse sowie IT-gestütztes Unternehmertum kennen.
- Die Themen werden in der ersten Sitzung vorgestellt und zugewiesen.
- Während des Semesters arbeiten die Studierenden an ihrer Seminararbeit.
- Die Ergebnisse werden in der Mitte und am Ende des Semesters präsentiert und diskutiert.
- Zudem erfolgt zu Beginn eine Einführung in wissenschaftliches Arbeiten.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbstständig die wichtigsten Theorien, Konzepte, Methoden und Resultate rund um das Gebiet Digitale Innovation und Entrepreneurship aufbereiten und interpretieren,
- wissenschaftliche Literatur analysieren, um eigenständig relevante Forschungsprobleme zu definieren,
- konzeptuelle, quantitative oder qualitative Forschungsmethoden zur Analyse dieser Fragestellungen anwenden,
- Seminararbeiten schreiben und begutachten,
- Thema, Forschungsdesign und Ergebnisse im Rahmen des Seminars kritisch reflektieren, präsentieren und diskutieren,
- angemessenes Feedback zu komplexen Herausforderungen geben und
- Teamfähigkeiten durch Zusammenarbeit mit Kommilitonen entwickeln.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen. Bei der Platzvergabe werden Master-Studierende bevorzugt.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zum bearbeiteten Thema (ggf. in Gruppen)
- Präsentation des Themas (ggf. in Gruppen)
- Aktive Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Steffi Haag

Master-Seminar: Kollektive Entscheidungsfindung (en. Master's-Seminar: Collective Decision-Making)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

In diesem Master-Seminar werden Vorträge zu ausgewählten Themen aus den folgenden Gebieten gehalten, die jeweils auch in Modulen der Theoretischen Informatik mit Vorlesungen und Übungen angeboten werden und alle unter den Oberbegriff der kollektiven Entscheidungsfindung fallen: Computational Social Choice, Algorithmic Game Theory und Fair Division. Thematisch ergänzt wird dieser Dreiklang aus „Wählen, Spielen, Teilen“ durch die verwandten Gebiete Judgment Aggregation, Participatory Budgeting und Argumentation Theory. Bei allen Vortragsthemen zu diesen Gebieten liegt der Fokus auf den algorithmischen und komplexitätstheoretischen Eigenschaften der jeweiligen Probleme.

- Präferenzaggregation durch Wählen
- Algorithmische Spieltheorie
- Gerechte Aufteilungsverfahren
- Gemeinsame Urteilsfindung
- Bürgerhaushalte
- Argumentationstheorie

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbstständig eine schriftliche Ausarbeitung zu einem wissenschaftlichen Thema aus der Originalliteratur erstellen,
- über dieses spezielle Thema einen Vortrag halten und dafür übersichtliche Folien erstellen,
- Fragen zu ihrem Thema korrekt beantworten,
- die Ausarbeitungen der anderen Studierenden im Rahmen eines *double-blind peer-reviewing* kritisch, aber auch fair begutachten und
- die Vorträge der anderen Studierenden kritisch und fair diskutieren sowie anspruchsvolle Fragen dazu stellen.

Literatur

Originalliteratur, die zum Beispiel in den folgenden Büchern zitiert wird:

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- 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). Heidelberg, 2011. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- schriftliche Ausarbeitung zum zugeteilten Thema
- Halten eines Vortrags
- Begutachtung einer anderen schriftlichen Ausarbeitung

Bemerkung: Dieses Modul wird nach zuvor bekannt gegebenen Kriterien benotet. Genauer ergibt sich die Gesamtnote gleichgewichtet durch die Bewertung der Qualität der Ausarbeitung und des Vortrags, sofern sie beide jeweils mit mindestens „bestanden“ bewertet wurden. Zusätzlich kann die Note durch Bonusleistungen — wie eine hilfreiche Begutachtung oder eine rege Teilnahme an der Diskussion zu den Vorträgen anderer — noch verbessert werden.

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Master-Seminar: Komplexität und Kryptologie (en. Master's-Seminar: Complexity and Cryptology)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

In diesem Master-Seminar werden Vorträge zu ausgewählten Themen Komplexitätstheorie und Kryptologie gehalten. Thematisch spielen bei den komplexitätstheoretischen Vorträgen auch, aber nicht nur Probleme aus der Computational Social Choice, Algorithmic Game Theory, Fair Division, Judgment Aggregation, Participatory Budgeting und Argumentation Theory eine Rolle. Bei den kryptologischen Themen liegt der Fokus auf der modernen Public-Key-Kryptologie.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbstständig eine schriftliche Ausarbeitung zu einem wissenschaftlichen Thema aus der Originalliteratur erstellen,
- über dieses spezielle Thema einen Vortrag halten und dafür übersichtliche Folien erstellen,
- Fragen zu ihrem Thema korrekt beantworten,
- die Ausarbeitungen der anderen Studierenden im Rahmen eines *double-blind peer-reviewing* kritisch, aber auch fair begutachten und
- die Vorträge der anderen Studierenden kritisch und fair diskutieren und anspruchsvolle Fragen dazu stellen.

Literatur

Originalliteratur, die zum Beispiel in den folgenden Büchern zitiert wird:

- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität. eXamen.Press. Springer-Verlag, Heidelberg, 2008. 1. Auflage.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity. EATCS Texts in Theoretical Computer Science, Springer-Verlag, Heidelberg, 2005. 1. Auflage.
- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag, Heidelberg, 2015. 1. Auflage.
- 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). Heidelberg, 2011. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- schriftliche Ausarbeitung zum zugeteilten Thema
- Halten eines Vortrags
- Begutachtung einer anderen schriftlichen Ausarbeitung

Bemerkung: Dieses Modul wird nach zuvor bekannt gegebenen Kriterien benotet. Genauer ergibt sich die Gesamtnote gleichgewichtet durch die Bewertung der Qualität der Ausarbeitung und des Vortrags, sofern sie beide jeweils mit mindestens „bestanden“ bewertet wurden. Zusätzlich kann die Note durch Bonusleistungen — wie eine hilfreiche Begutachtung oder eine rege Teilnahme an der Diskussion zu den Vorträgen anderer — noch verbessert werden.

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Master-Seminar: NP-Schwere Probleme (en. Master's Seminar on NP-hard problems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Die Härte mancher Probleme gibt Anlass dazu, kreative Methoden zu finden um sich der Lösung eines Problems zu nähern. Man kann beispielsweise Teilprobleme lösen oder das Problem auf eingeschränkte Klassen untersuchen, um Approximationsalgorithmen bzw. parametrisierte Algorithmen zu finden oder das Problem unter einer bestimmten Einschränkung effizient zu lösen. In diesem Modul werden verschiedene dieser Methoden untersucht. Ziel dieses Moduls ist es, den Studierenden Einblicke in einige wichtige oder aktuelle Forschungsthemen, Resultate und Methoden aus dem Gebiet der NP-schweren Probleme zu geben.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbst Einblicke in einige aktuelle Forschungsergebnisse verarbeiten und Resultate aus Forschungsergebnissen erklären und präsentieren,
- die besprochenen schweren Probleme erläutern und formal definieren,
- die behandelten Algorithmen und Methoden erklären und auf konkrete Eingaben anwenden,
- Laufzeitanalysen der Algorithmen durchführen,
- selbst Aufgaben stellen und
- anderen Studierenden hinreichend gute Materialien erstellen, um diese Aufgaben zu lösen.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive Mitarbeit in den Vorträgen und Übungen
- Abgabe der Übungsaufgaben
- Bestehen der Klausur zum Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Priv.-Doz. Dr. Frank Gurski

Master-Seminar: Property Testing (en. Master's Seminar: Property Testing)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch

Inhalte

We study algorithmic techniques that allow us to solve relaxed decision problems efficiently such as testing whether a sparse graph is connected or „far away“ from being connected in sublinear time. By sampling, we can obtain partial information of an input locally, which provides structural information about the global input. Individual topics usually deal with a specific problem setting, a specific testing algorithm, and a running time and accuracy analysis thereof.

- function properties (e.g., monotonicity, linearity)
- graph properties (e.g., connectivity in bounded-degree graphs, bipartiteness in dense graphs)

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- extract significant contents of a given topic and to reproduce them in a written and oral form,
- evaluate the readability and technical quality of other abstracts,
- plan and hold a presentation and
- ask and answer adequate questions and to give and take constructive feedback.

Literatur

- Oded Goldreich: Introduction to Property Testing. Cambridge University Press. 2017. 1st Edition
- Dana Ron: Algorithmic and Analysis Techniques in Property Testing. Foundations and Trends in Theoretical Computer Science. 2010. 5(2):73–205

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen*, *Theoretische Informatik*, *Mathematik für Informatik 3* und *Wissenschaftliches Arbeiten*

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zu einem gegebenen Thema
- Peer Review
- Präsentation des Themas
- Fragen stellen und beantworten

Modulbeauftragte und hauptamtlich Lehrende

Dr. Anja Rey

Master-Seminar: Reproducibility in Bioinformatics Research (en. Master's Seminar: Reproducibility in Bioinformatics Research)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
5 LP	150 Stunden	30 Stunden	120 Stunden
Seminar (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

Scientific data analysis is becoming increasingly important in research. The combination of huge data volumes, lack of standards and incentives, poor programming practices, and inadequately documented parameters and versions has led to a reproducibility crisis. Scientific software is often inadequately documented, inefficiently programmed, and not well-maintained. In this seminar, we will confront these issues by attempting to reproduce the data analysis of selected original papers from the area of bioinformatics. In addition, we will learn how to improve the analysis steps, e.g., by using workflow management tools such as Snakemake and depositing the pipelines on certified online storage services.

Students organize themselves in small groups and work together on a topic. They present the research and work together on its reproducibility, typically using a shared git repository. They present their results and write a report about the original research and their reproducibility efforts.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- explain the importance of reproducibility in science,
- formulate the pitfalls of reproducibility and know about the reproducibility crisis,
- apply techniques to ensure reproducibility in computational science,
- extract significant contents of a given topic and to reproduce them in a written and oral form,
- prepare discussion points to enable a critical discussion on a topic,
- evaluate the readability and technical quality of scientific publications,
- plan and hold presentations and create a written report, and
- ask and answer adequate questions and to give and take constructive feedback.

Students also have demonstrated to work as a group and to divide the work in a fair manner.

Literatur

- Selected publications regarding the topic of the course.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen*, *Theoretische Informatik*, *Mathematik für Informatik 3* und *Wissenschaftliches Arbeiten*

Voraussetzungen für die Vergabe von Leistungspunkten

- Accepting a topic
- Active participation
- Presentation of a topic and preparation of discussion points
- Written report
- Peer Review

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Gunnar W. Klau

Master-Seminar: User Experience (UX) Design (en.)

Master's Seminar: User Experience (UX) Design

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch

Inhalte

Das Seminar behandelt aktuelle Forschungsthemen rund um „User Experience (UX) Design“, das die Disziplinen der Informatik, insb. Human-Computer-Interaction (HCI), Psychologie und Wirtschaft integriert. UX Design befasst sich mit der Analyse, Kreation und Optimierung von Nutzungserfahrungen.

- Die Studierenden lernen Theorien, Konzepte, Prozesse, Werkzeuge und Methoden kennen zur Erforschung von Kognitionen, Emotionen und Verhaltensweisen von Nutzenden in Interaktion mit digitalen Technologien.
- Die Themen werden in der ersten Sitzung vorgestellt und zugewiesen.
- Während des Semesters arbeiten die Studierenden an ihrer Seminararbeit.
- Die Ergebnisse werden in der Mitte und am Ende des Semesters präsentiert und diskutiert.
- Zudem erfolgt zu Beginn eine Einführung in wissenschaftliches Arbeiten.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbstständig die wichtigsten Theorien, Konzepte, Methoden und Resultate rund um das Gebiet User Experience (UX) Design aufbereiten und interpretieren,
- wissenschaftliche Literatur analysieren, um eigenständig relevante Forschungsprobleme zu definieren,
- konzeptuelle, quantitative oder qualitative Forschungsmethoden zur Analyse dieser Fragestellungen anwenden,
- Seminararbeiten schreiben und begutachten,
- Thema, Forschungsdesign und Ergebnisse im Rahmen des Seminars kritisch reflektieren, präsentieren und diskutieren,
- angemessenes Feedback zu komplexen Herausforderungen geben und
- Teamfähigkeiten durch Zusammenarbeit mit Kommilitonen entwickeln.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Bei der Platzvergabe werden Master-Studierende bevorzugt.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zum bearbeiteten Thema (ggf. in Gruppen)
- Präsentation des Themas (ggf. in Gruppen)
- Aktive Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Steffi Haag

Master-Seminar: Wissenschaftliche Methoden (en. Master's Seminar: Scientific Methods)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Deutsch

Inhalte

Die Studierenden verschaffen sich selbst Einblicke in wesentliche wissenschaftliche Methoden und halten über ein spezielles Thema aus der Originalliteratur einen Vortrag im Seminar. Die Vorträge der anderen Studierenden werden unter allen Teilnehmern kritisch diskutiert. Zu jedem Vortrag gehört eine schriftliche Ausarbeitung und es findet eine gegenseitige Beurteilung der schriftlichen Ausarbeitungen der Studierenden untereinander statt. Dabei wird die Anonymität sowohl des Autors bzw. der Autorin als auch die der Gutachterin bzw. des Gutachters gewahrt („double-blind peer-reviewing“).

- Wie schreibt man einen wissenschaftlichen Artikel?
- Wie verfasst man einen Literaturreview?
- Wissenschaftliche Argumentation mit Diagrammen
- Wissenschaftliche Kreativität: Rolle von Fragen, Interdisziplinarität, Analogien und Anthropomorphismen
- Unvoreingenommene Datenanalyse
- Verhältnis zwischen Daten und Hypothesen
- Confirmation bias und Widersprüche

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- wesentliche wissenschaftliche Methoden beschreiben,
- eine wissenschaftliche Ausarbeitung anfertigen,
- einen guten Vortrag vorbereiten und halten,
- Präsentationen anderer Studierender und vorgetragene Inhalte kritisch diskutieren,
- fachbezogene Positionen zusammenfassen, kommunizieren und argumentativ verteidigen und
- wissenschaftliche Ausarbeitungen anderer Teilnehmer kritisch und fair begutachten.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zum gewählten Thema
- Halten eines Vortrags
- Begutachtung von zwei anderen schriftlichen Ausarbeitungen
- In die abschließende Benotung fließen alle Beiträge der Studierenden zum Modul zu gleichen Teilen ein, d. h. schriftliche Ausarbeitung, zwei Begutachtungen von Ausarbeitungen der Mitstudierenden, Seminarvortrag.

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Lercher, Dr. Mayo Röttger

Methods for Population Genetics (en. Methods for Population Genetics)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	45 Stunden	105 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (1 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

Population Genetics studies the distribution of genetic information across populations under varying conditions. This lecture provides the necessary methods to answer questions such as: Is there evidence for selection pressure on a certain gene? How can ancient migration patterns be retraced based on genetic information? How can quantitative traits such as disease susceptibility be attributed to genetic loci?

- Hardy-Weinberg principle
- Genetic drift and Wright-Fisher model
- Kolmogorov forward and backward equations
- Mutation and selection
- Linkage disequilibrium
- Population structure and inbreeding
- Haplotype phasing and imputation
- Genome-wide association studies and quantitative traits.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- explain the fundamental principles and terms used in population genetics,
- analyze and compare the effects of mutation, migration and selection on genotypes in a population,
- describe analyses of genetic variation in the context of phenotypic traits, and
- apply the algorithmic and statistical concepts of population genetics to modern sequencing data sets.

Literatur

- Hartl and Clark: Principles of Population Genetics. Sinauer. 2007.
- John Gillespie: Population Genetics - A Concise Guide. Johns Hopkins University Press. 2004.
- Matthew Hamilton: Population Genetics. Wiley-Blackwell. 2009.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science
- Anwendungsfach für den Ergänzungsbereich im Master-Studiengang Mathematik
- Individuelle Ergänzung

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- To be eligible to take the exam, students have to earn 50% of the points from the assignment sheets.
- Oral exam (usually 20 minutes)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Tobias Marschall

Model Checking (en. Model Checking)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Modul befasst sich mit der vollautomatischen Verifikation von Hardware und Software mit Hilfe von Model Checking Algorithmen und Werkzeugen.

- Model Checking Algorithmen zur Verifikation von temporalen (Safety, Liveness und Fairness) und regulären Eigenschaften
- Kenntnisse und Anwendung von temporalen Logiken (z. B. LTL, CTL, CTL*, ...)
- Verwendung von Model Checking Werkzeugen (z. B. SMV, Spin, ProB, ...)
- Grundlagen fortgeschrittener Methoden (z. B. Binary Decision Diagrams, Partial Order Reduction, Symmetrie, ...)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- verschiedene Verfahren zur Verifikation präsentieren, anwenden und miteinander vergleichen,
- einfache Spezifikationen erstellen und diese automatisiert mit Werkzeugen prüfen, \ die Spezifikationen sind für Systeme wie im Baier, Katoen Buch für Getränkeautomaten oder Peterson Mutual Exclusion
- temporale und reguläre Eigenschaften in geeigneten Formalismen formulieren, und
- einen eigenen Model Checker implementieren.

Literatur

- Christel Baier und Joost-Pieter Katoen. Principles of Model Checking. MIT Press, 2008.
- M. Clarke, O. Grumberg, D.A. Peled. Model Checking, Second Edition. MIT Press. 2018.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Seminarvortrag und erfolgreiche Bearbeitung einer Projektaufgabe zum Model Checking
- mündliche (20-30 Minuten) oder schriftliche (Klausur, 90 Minuten) Prüfung, in Absprache mit den Studierenden abhängig von der Anzahl der Teilnehmer

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Leuschel

Multimedia-Datenbanksysteme (en. Multimedia Database Systems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Diese Modul behandelt Grundbegriffe und Methoden für die Behandlung von Multimedia-Objekten in Retrievalsystemen am Beispiel von Bilddaten:

- Grundbegriffe des Information Retrieval (Definition; Bewertungsmaße)
- Multimedia Information Retrieval (mit Schwerpunkt auf Bilddatenbanken)
- Feature-Extraktion, -Selektion und -Transformation
- Distanzfunktionen; Fluch der Dimensionalität
- effiziente Algorithmen und Datenstrukturen zur Ähnlichkeitssuche in hochdimensionalen Feature-Räumen (mehrdimensionale Indexstrukturen, Ähnlichkeitssuche)

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die grundlegenden Fragestellungen des Multimedia Information Retrieval erklären,
- geeignete Distanzfunktionen auswählen bzw. entwerfen und
- ein eigenes Multimedia-Datenbanksystem entwerfen.

Literatur

- I. Schmitt: Ähnlichkeitssuche in Multimedia-Datenbanken. Oldenbourg-Verlag, München. 2005.
- K. Meyer-Wegener: Multimedia-Datenbanken – Einsatz von Datenbanktechnik in Multimedia-Systemen. Teubner, Stuttgart. 2003. 2. Auflage
- B. Prabhakaran: Multimedia Database Management Systems. Kluwer Academic Publishers, Massachusetts. 1997.
- R. Veltkamp, H. Burkhardt, H.-P. Kriegel (eds.): 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

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiche Bearbeitung der Übungsaufgaben
- abgeschlossene Prüfung (i. d. R. mündlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

Natural Language Processing (en. Natural Language Processing)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes zweite Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

- introduction to Natural Language Processing (NLP)
- NLP pipeline and methods (language identification, segmentation, POS tagger, stemming and lemmatization, stop word elimination, dependencies, thesaurus, ...)
- POS tagging in detail (rule-based; hidden markov model, ...)
- distributional semantics (vector representations, word embeddings, deep learning architectures for NLP)
- annotation process (guidelines, gold standard, measurements for inter-annotator agreement)

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- explain basic methods and algorithms for NLP,
- design a NLP pipeline for a dedicated task and implement it using adequate libraries,
- know how to evaluate NLP algorithms and whole pipelines, and
- interpret the results of such evaluations.

Literatur

- C. Manning, H. Schütze: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge. 1999.
- Further literature is provided during the course.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- active participation in the exercises

- homeworks
- final examination (usually as oral examination)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

Practical: Implementing Transformer Models (en. Practical: Implementing Transformer Models)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
praktische Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch

Inhalte

- Detailed study and implementation of the Transformer model, building an intuitive understanding of its unique architecture and the attention mechanism.
- Practical application of computational aspects of the Transformer model, including the scaling of dot products and shared parameter mechanisms, such as embedding vectors.
- Training a Transformer model tailored for machine translation, including the preparation and pre-processing of a translation dataset.
- Exploration of GPU utilisation, parallel training strategies and effective resource allocation for machine learning training.

Lernergebnisse/Kompetenzen

By the end of the module, students will be able to:

- interpret the architectural nuances and computational aspects of Transformer models, including the attention mechanism and multi-head attention layers.
- adopt scientific research skills, such as setting up independent projects, implementing models based on academic literature, and producing scientifically sound, reproducible results.
- apply knowledge of optimal GPU utilisation, including the setup and management of parallel training environments and effective resource allocation for machine learning training.

Literatur

- Attention is all you need. (<https://arxiv.org/pdf/1706.03762.pdf>)

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte des Moduls *Machine Learning*

Voraussetzungen für die Vergabe von Leistungspunkten

Active participation in the practical. Passing grades for presentation and report.

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Milica Gašić, Carel van Niekerk

Representations in Natural Language Processing (en. Representations in Natural Language Processing)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Englisch

Inhalte

Word embeddings model natural language by associating to each word in the vocabulary a vector in an ambient space, such that words which have similar meaning are close together. When data is encoded as points in a high-dimensional space, this is called an embedding.

Words which are used in similar contexts usually have similar meanings (this is also known as the distributional hypothesis). Thus it is possible to acquire meaningful word representations from unlabelled data, such as from Wikipedia paragraphs, books, or news headlines.

In this lecture, we study recently published techniques for building both static and contextual embeddings. In a static embedding, there is one fixed vector for each word in the vocabulary. In a contextual embedding, the vector we associate to a word is distinct if it is used in a different context. For example, the word 'bank' has separate representations depending on whether it refers to the side of a river, or to a financial institution.

Another important but challenging aspect is the evaluation of the quality of a word embedding, which form the basis of many natural language processing tasks such as document search, classification, information retrieval, language translation and sentiment analysis.

Overview:

- Static word embeddings: Frequency based methods, word2vec, GloVe, fastText
- Contextual word embeddings: Recurrent methods, Language Models, ELMo, Transformers, BERT and Masked Language Models, Contrastive Learning, Sentence Embeddings
- Intrinsic evaluation of word embeddings and applications in dialog systems
- Bias in word embeddings, Sentiment
- Multi-lingual word embeddings
- Hyperbolic word embeddings, Topological Data Analysis and its applications

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- compare methods for constructing word embeddings,
- implement and apply the underlying algorithms,
- name applications for word embeddings in natural language processing and dialog systems, and
- independently review and evaluate scientific publications.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.
- D. Jurafsky, J. Martin: Speech and Language Processing. Prentice Hall. Upper Saddle River, NJ, 2008. 2nd edition.
- J. Eisenstein: Introduction to Natural Language Processing. MIT Press. Cambridge, MA, 2019. 1st

edition.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Formal: Studierende, die das Master-Seminar über Worteinbettungen absolviert haben, dürfen an diesem Kurs nicht teilnehmen.
- Inhaltlich: Inhalte des Moduls *Machine Learning*

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive und erfolgreiche Mitwirkung in den Übungen
- Bestehen der Prüfung (Hausarbeit)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Milica Gašić, Benjamin Matthias Ruppik

Sicherheitskritische Systeme (en. Safety-Critical Systems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (1 SWS) praktische Übung (1 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

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 wurden.

Diese Themen werden behandelt:

- Überblick über existierende Formale Methoden und deren praktischer Einsatz in Systementwicklung und in Entwicklungsnormen
- Grundlagen der B Sprache
- Invarianten und Beweisobligationen
- Verfeinerung
- Überblick über industrielle B Modelle
- Andere Korrektheitskriterien: Well-Definedness und temporale LTL Eigenschaften

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Grundzüge der B Sprache (Syntax und Semantik) mit klassischen Programmiersprachen vergleichen können,
- Typkorrektheit von B Ausdrücken, wie in den Übungen besprochen, prüfen können
- B Ausdrücke für konkrete Systemanfragen, wie in den Übungen besprochen, erstellen,
- formale Spezifikationen mit B erstellen und dabei praktische B Werkzeuge (hauptsächlich Rodin und ProB) benutzen,
- Spezifikationen auf Fehler bezüglich der Invariante und bezüglich von Verfeinerung untersuchen,
- die Korrektheit von einfachen Spezifikationen sowohl auf Papier als auch mit Hilfe einer automatisierten Beweisumgebung (Rodin) beweisen, und
- Verfeinerung nutzen um komplexe Systeme und komplexe Beweise zu strukturieren.

Literatur

- J.R. Abrial: Modeling in Event-B: System and Software Engineering. Cambridge University Press. 2010. ISBN 0521895561.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Wahlpflichtbereich Praktische oder Technische Informatik
- Schwerpunktbereich
- Individuelle Ergänzung

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Erfolgreiche Bearbeitung der Pflichtübungen
- Bestehen der Klausur

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Leuschel

Stochastic Models of Biological Systems (en. Stochastic Models of Biological Systems)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Semester	Master-Studiengang Informatik	Englisch

Inhalte

- Cell birth-and-death, moment-generating functions.
- Diffusion models and Brownian dynamics, comparison of deterministic versus stochastic approaches.
- Protein production, the Totally Asymmetric Simple Exclusion Process, its analytical solution in mean field.
- Cytoskeletal transport, directed transport properties of three dimensional structures.
- Polysaccharide synthesis, complex enzymatic processes.

The course describes the stochastic modelling (analytical and numerical) of the biological systems from cell biology above listed. Key algorithms used throughout the module are Monte Carlo and Gillespie, implemented in C. The module is highly interdisciplinary and includes knowledge and methods from Mathematics, Biology, Physics, and Chemistry, with a strong focus on algorithms and programming in the tutorials. The methods and tools learnt during this course also apply to other fields than Biology. Specifically, they are used in theoretical Physics of both dilute and condensed systems, and Engineering. The programming language C is widely spread in these fields since it is well suited for computer intensive numerical methods.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- distinguish and recognise key biological features that justify the use of the models studied,
- explain modelling approaches based on the underlying biology,
- develop and implement stochastic algorithms using the programming language C++,
- simulate the dynamics of the systems studied and critically evaluate the quality of their results,
- reproduce all analytical calculations executed in the course,
- explain physico-chemical parameters that influence the dynamics of the processes studied.

Literatur

- None

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- At least 50 percent of the points from the exercises
- Passing final exam (written exam, usually 90 minutes)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Adélaïde Raguin, Prof. Dr. Martin Lercher

System-Software für Big-Data-Computing (en. System Software for Big Data Computing)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	30 Stunden	120 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Seminar (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

Inhalte

- Big-Data-Systeme: Batch-, In-Memory, Stream-, Graph-Processing; Distributed Machine Learning, ...
- Cloud-Systeme: Storage, Virtualisierung, Koordinierungsdienste, Messaging, ...
- Betriebssysteme für Manycore, NVRAM (persistenter Speicher), RDMA (Remote Direct Memory Access), GPUs, ...

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- relevante Inhalte zum gegebenen Thema recherchieren und schriftlich und mündlich wiedergeben,
- Aufsätze von anderen Studierenden evaluieren hinsichtlich Leserbarkeit und technischer Qualität,
- eine Präsentation planen und durchführen,
- sinnvolle Fragen stellen, beantworten, und
- konstruktive Kritik annehmen sowie geben.

Literatur

- Relevante Publikationen werden in der Lehrveranstaltung 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 die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zu einem gegebenen Thema
- Teilnahme am Peer-Reviewing
- Präsentation des Themas
- Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Transaktionsverwaltung (en. Transaction Management)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Module behandelt Grundlagen der Transaktionsverwaltung. Hierzu werden formale Modelle eingeführt, ihre Eigenschaften analysiert sowie ihre Umsetzbarkeit in realen Systemen diskutiert. Im Einzelnen werden folgende Themen angesprochen:

- Transaktionseigenschaften und -anforderungen (ACID, ...)
- Serialisierbarkeit (Final-State-, Sicht-, Konfliktserialisierbarkeit und deren formale Eigenschaften)
- Abbrucheigenschaften (Rücksetzbarkeit, Vermeidung kaskadierender Abbrüche, Striktheit)
- Sperrverfahren und -protokolle (2-Phasen-Sperrprotokoll, Baumprotokoll, nicht sperrende Verfahren)
- Recovery (Undo/Redo, Wiederanlauf, ...)
- strukturierte Transaktionsmodelle
- formale Beschreibung von Transaktionsmodellen (ACTA, Transaktionshüllen)
- Transaktionen in verteilten Systemen (verteiltes Commit, 2PC-Protokoll)

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Probleme der Transaktionsverwaltung in Datenbanksystemen erläutern,
- Eigenschaften von Transaktionsschedules bestimmen,
- Recovery-Verfahren vergleichen und bewerten und
- die Einsatzmöglichkeiten strukturierter Transaktionsmodelle bewerten.

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

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte des Moduls *Datenbanken: Eine Einführung*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiche Bearbeitung der Übungsaufgaben
- abschließende Prüfung (i. d. R. mündlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

User Experience (UX) Design and Management (en. User Experience (UX) Design and Management)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) praktische Übung (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

Students explore the user perspective on interacting with digital technologies, thereby becoming more aware of users' needs during product development. Designing and maintaining great user experience (UX) is the best way for both startups and established companies to build trust, retention, and loyalty of staff and customers alike.

The lecture

- teaches the key concepts, methods, and approaches that help design, measure, and manage total UX across organizations and drive value propositions of digital business models.
- discusses established and new methods of UX research for (further) developing digital technologies.
- introduces frameworks to build and lead teams of UX researchers, designers, engineers, product managers.
- employs case studies to transfer and discuss the application of UX design, research, and management in practice.

In the practice sessions, (groups of) students - practically apply UX research methods and tools (e.g., user interviews, A/B testing, or emotion detection) to investigate users' experiences in interaction with state-of-the-art digital technology prototypes and to deduce implications for product and organizational strategy, development, and design. - present the results towards peers and experts from research and industry.

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- define, discuss, and apply the concepts, methods, and tools of analyzing and managing the experiences users perceive in interaction with new digital technologies of startups/established companies,
- measure and analyze user experiences of novel technologies and infer recommendations for technology and policy design and development,
- assess and reflect the social and ethical implications of designing, evaluating, and implementing digital technologies,
- present user research results towards peers, and
- develop skills in collaborative interaction with peers.

Literatur

- Relevant literature is provided during the course.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik

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

Teilnahmevoraussetzungen

- **Formal:** Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- **Projektbericht (pro Gruppe)**
- **Gruppen-Präsentationen**
- **Aktive Teilnahme an Diskussionen**

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Steffi Haag

Vertiefung Compilerbau (en. Advanced Compiler Construction)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Das Modul „Vertiefung Compilerbau“ behandelt weiterführende Themen, welche im Modul Compilerbau nicht behandelt werden.

Themen des Moduls sind: Erzeugung und Verwendung von SSA-Zwischencode, Schleifenanalyse, Dominatorbäume, Interprozedurale Analyse, Aliasanalyse, Codegenerierung, Compilerverifikation und Just-In-Time Compilation. Die Studierenden wenden erlerntes Wissen in einem Programmierprojekt an. Außerdem stellen die Teilnehmenden selbst ein Thema aus dem Bereich Compilerbau in einem Vortrag dar.

Voraussetzung ist ein Grundlagenwissen in diesem Bereich. Hierzu gehört: Parsing, kontextfreie Grammatiken, Typüberprüfung, Dreiadresscode und Datenflussanalyse wie z. B. die Liveness-Analyse.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- fortgeschrittene Konzepte und Probleme des Compilerbaus bezeichnen,
- bewerten wann und für welche Einsatzbereiche bestimmte Algorithmen und Verfahren vorteilhaft sind,
- eigenständig einen fortgeschrittenen Compiler erstellen und testen und
- einen Vortrag zu einem fortgeschrittenen Thema des Bereichs Compilerbau halten.

Literatur

- Andrew W. Appel: Modern Compiler Implementation in Java. Cambridge University Press. 2nd Edition
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: Compilers: Principles, Techniques, and Tools. Addison Wesley. 2nd Edition

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte des Moduls Compilerbau

Voraussetzungen für die Vergabe von Leistungspunkten

- erfolgreiche Bearbeitung der Pflichtübungen
- erfolgreiche Entwicklung und Erweiterung eines eigenen Compilers
- erfolgreicher Vortrag zu einem Thema aus dem Bereich Compilerbau
- bestehen der Prüfung, in der Regel mündlich

Modulbeauftragte und hauptamtlich Lehrende

Dr. John Witulski

Vertiefung Funktionale Programmierung: Clojure (en. Advanced Functional Programmierung: Clojure)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung oder Seminar (2 SWS) Übung (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

Inhalte

In diesem Modul werden fortgeschrittene Konzepte in Clojure vermittelt. Wir behandeln dabei folgende Themen:

- generatives Testing mit test.check
- optionales Typsystem mit clojure.spec
- Transducers
- asynchrones Programmieren mit core.async
- Datomic-ähnliche Datenbanken
- Clojurescript
- je nach Interesse weitere domänenspezifische Bibliotheken und Werkzeuge

Lernergebnisse/Kompetenzen

Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,

- komplexere theoretische Konzepte der funktionalen Programmierung zu erklären und zu verwenden,
- Webanwendungen mit Clojure und Clojurescript zu schreiben, und
- eigenständig sowie in Gruppen größere funktionale Programme zu erstellen und zu testen.

Literatur

- Abelson, Sussman: Structure and Interpretation of Computer Programs. MIT Press. 1996. 2nd edition.
- Baldrige: Communicating sequential processes with core.async. Infinite Skills. 2015. 1st edition.
- Fogus, Houser: The Joy of Clojure. Manning. 2014. 2nd edition.
- Emerick, Carper, Grand: Programming Clojure. O'Reilly. 2012. 1st edition.
- Rathore, Avila: Clojure in Action. Manning. 2016. 2nd edition.
- Higginbotham: Clojure for the Brave and True. No Starch Press. 2015. 1st edition.
- Miller, Hallway, Bedra: Programming Clojure. Pragmatic Programmers. 2018. 3rd edition.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.
- Inhaltlich: Inhalte des Moduls *Einführung in die Funktionale Programmierung*

Voraussetzungen für die Vergabe von Leistungspunkten

- Je nach Durchführung: aktive Teilnahme am Seminar
- Je nach Teilnehmendenzahl:
- bevorzugt mündliche Prüfung (i. d. R. 15-30 Minuten),
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten).

Modulbeauftragte und hauptamtlich Lehrende

Philipp Körner, Dr. Jens Bendisposto, Prof. Dr. Michael Leuschel

Vertiefung Logische Programmierung (en. Advanced Topics in Logic Programming)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (1 SWS) praktische Übung (1 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

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 Programmier Techniken in Prolog
- Metaprogrammierung in Prolog und Anwendungen
- Anwendung von Constraint Logic Programming
- Entwicklung eigener Constraint Solver
- Interpreterentwicklung in Prolog mit Illustration von Konzepten wie operationeller Semantik
- Programmanalyse mit abstrakter Interpretation und Datenflussanalyse
- partielle Auswertung und automatische Programmoptimierung

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Problemstellungen mit Hilfe von Constraint Solvern lösen,
- eigene Constraint Solver entwickeln,
- Inferenzregeln für operationale Semantik in Prolog ausdrücken,
- für eine neue Programmiersprache einen Interpreter in Prolog selber entwickeln,
- einen Interpreter für Prolog in Prolog schreiben,
- die Grundzüge der abstrakten Interpretation und ihrer Anwendung zur Programmverifikation erläutern und vergleichen,
- einfache Analyse- und Verifikationswerkzeuge in Prolog entwickeln,
- einen partiellen Auswerter anwenden und selber entwickeln.

Literatur

- Eigenes Skript

Verwendbarkeit des Moduls

- Wahlpflichtbereich Praktische oder Technische Informatik

- **Schwerpunktbereich**
- **Individuelle Ergänzung**

Teilnahmevoraussetzungen

- **Formal:** Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- **Erfolgreiche Bearbeitung der Pflichtübungen**
- **Bestehen der Klausur**

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Leuschel

Lehrveranstaltungen anderer Studiengänge oder Institute (en. Courses from other programs or institutes)

Einige Lehrveranstaltungen anderer Studiengänge und/oder Institute können durch Studierende im Master-Studiengang Informatik ebenfalls belegt werden. Diese Kurse können in den Wahlbereichen *Praktische oder Technische Informatik*, *Theoretische Informatik*, *Schwerpunkt* und *Individuelle Ergänzung* anerkannt werden. In der Spalte *Bereich* ist angegeben, ob das entsprechende Modul dem Bereich *Theoretische Informatik* (TH) oder *Praktische und Technische Informatik* (PR) zugeordnet werden kann; eine Verwendung in dem jeweils anderen Bereich ist nicht möglich.

Master AI and Data Science

Die hier aufgeführten Module finden Sie im [Modulhandbuch des Studiengangs Master AI and Data Science](#). Beachten Sie die dort angegebenen inhaltlichen Voraussetzungen der jeweiligen Module. Wenn ein Modul hier aufgelistet ist, bedeutet dies *nicht* automatisch, dass das Modul allein mit den Vorkenntnissen aus dem Bachelor Informatik sinnvoll absolvierbar ist.

Titel	Verantwortliche Personen	Turnus	LP	Sprache	Bereich
Applications of Transformer Networks in Bio- and Cheminformatics	Dr. A. Kroll	unregelmäßig	5 LP	EN	PR
Computational Argumentation	Jun. Prof. Dr. G. Lapesa	unregelmäßig	5 LP	EN	PR
Causality	Prof. Dr. D. Heider	unregelmäßig	5 LP	EN	TH
Clinical Decision Support Systems (CDSS)	Prof. Dr. D. Heider	unregelmäßig	5 LP	EN	PR
Computer Vision	Prof. Dr. P. Swoboda	jedes Sommersemester	5 LP	EN	PR
Data & Knowledge Engineering (DKE)	Prof. Dr. S. Dietze	unregelmäßig	5 LP	EN	PR
Deep Learning	Prof. Dr. T. Dickscheid Prof. Dr. M. Kollmann	jedes Wintersemester	5 LP	EN	PR
Deep Learning: Generative Models	Prof. Dr. T. Dickscheid Prof. Dr. M. Kollmann	jedes Wintersemester	5 LP	EN	PR
Deep Learning: Representation Learning	Prof. Dr. T. Dickscheid Prof. Dr. M. Kollmann	jedes Sommersemester	5 LP	EN	PR
Master's Seminar on Ethics in Natural Language Processing	Jun. Prof. Dr. G. Lapesa	unregelmäßig	5 LP	EN	PR

Information Theory	Dr. P. Arndt	unregelmäßig	5 LP	EN	TH
Master's Seminar Advances in Data Science	Prof. Dr. S. Dietze	unregelmäßig	5 LP	EN	PR
Master's Seminar: Computational Argumentation	Jun. Prof. Dr. G. Lapesa	unregelmäßig	5 LP	EN	PR
Master's Seminar on Limits of Computation	A. Lahiri	unregelmäßig	5 LP	EN	TH
Master's Seminar on Word Embedding Spaces	Dr. B. Ruppik Prof. Dr. M. Gasic	unregelmäßig	5 LP	EN	PR
Neuroimaging and Precision Medicine	Prof. Dr. S. B. Eickhoff Prof. Dr. S. Caspers PD Dr. S. Weis	jedes Wintersemester	5 LP	EN	PR
Probabilistic Machine Learning	Prof. Dr. P. Swoboda	unregelmäßig	10 LP	EN	PR
Reinforcement Learning	Prof. Dr. S. Harmeling	unregelmäßig	5 LP	EN	PR
Relational Databases and Data Analysis	Prof. Dr. S. Conrad	unregelmäßig	5 LP	EN	PR
Spectral Graph Theory and Graph Signal Processing	Dr. P. Arndt	unregelmäßig	5 LP	EN	PR
Spoken Dialogue Systems	Prof. Dr. M. Gasic	unregelmäßig	10 LP	EN	PR
Unsupervised Deep Learning with Applications in Image Analysis	Prof. Dr. T. Dickscheid Prof. Dr. M. Kollmann	unregelmäßig	5 LP	EN	PR

Master Physik

Die hier aufgeführten Module finden Sie im [Modulhandbuch des Studiengangs Master Physik](#). Beachten Sie die dort angegebenen inhaltlichen Voraussetzungen der jeweiligen Module. Wenn ein Modul hier aufgelistet ist, bedeutet dies *nicht* automatisch, dass das Modul allein mit den Vorkenntnissen aus dem Bachelor Informatik sinnvoll absolvierbar ist.

Titel	Verantwortliche Personen	Turnus	LP	Sprache	Bereich
Advanced Quantum Information Theory	PD Dr. H. Kampermann	unregelmäßig	4 LP	EN	TH
Journal Club on Quantum Information Theory	Prof. Dr. D. Bruß	jedes Semester	3 LP	EN	TH
Quantum Cryptography	PD Dr. H. Kampermann	unregelmäßig	4 LP	EN	TH
Surface Code Quantum Computation	PD Dr. H. Kampermann	unregelmäßig	6 LP	EN	TH

Master Betriebswirtschaftslehre

Beachten Sie, dass diese Module *ausschließlich* im Bereich „Individuelle Ergänzung“ anerkannt werden können. Beachten Sie weiterhin die inhaltlichen Voraussetzungen der Kurse.

Titel	Verantwortliche Personen	Turnus	LP	Sprache
Start-up Werkstatt	Dr. M. Henn	jedes Sommersemester	8 LP	DE

Projektarbeit (en. Individual Research Project)

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

Modulbeschreibung (en. Module Description)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
20 LP	600 Stunden	nach Vereinbarung	nach Vereinbarung (überwiegender Anteil)
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
-	je nach Arbeitsgruppe, mindestens jährlich	Master-Studiengang Informatik	Deutsch Englisch

Inhalte

In diesem Modul sollen Studierende das selbständige wissenschaftliche Arbeiten erwerben und sich auf die Master-Arbeit vorbereiten. Die Inhalte hängen deshalb sehr stark vom jeweiligen Fachgebiet und den Interessensgebieten der Person ab.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- 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

Literatur

- In Absprache mit den Betreuungspersonen.

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.

Modulbeauftragte und hauptamtlich Lehrende

Die Dozierenden der Informatik sowie gegebenenfalls der Mathematisch-Naturwissenschaftlichen Schwerpunktfächer

Master-Arbeit (en. Master's Thesis)

Modulbeschreibung

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
30 LP	900 Stunden	in Absprache mit Betreuungsperson	in Absprache mit Betreuungsperson
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
-	jederzeit	Master-Studiengang Informatik	Deutsch Englisch

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.

Literatur

- In Absprache mit den Betreuungspersonen.

Verwendbarkeit des Moduls

- Masterarbeit

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 alle 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

Modulbeauftragte und hauptamtlich Lehrende

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

Nicht mehr angebotene Module

In diesem Kapitel finden Sie alle Module, die wir nicht mehr anbieten.

Big-Data-Systeme und Anwendungen (en. Big-Data Systems and Applications)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Deutsch

Inhalte

- Infrastruktur (Cloud, Edge, Fog)
- Grundlagen der Skalierbarkeit (Replikation, Konsistenz, CAP)
- Verteilte Hashtabellen (Chord und Pastry)
- Konsensusdienste (Paxos, Zookeeper)
- Key-Value Speichersysteme (Dynamo, Redis, DXRAM)
- Verteilte Dateisysteme (Google File System)
- NoSQL-Datenbanken (Google BigTable)
- MapReduce (Architektur und Beispiele)
- In-Memory-Verarbeitung (Spark-Architektur und Beispiele)
- Stream-Verarbeitung (Grundlagen, Spark, Flink)
- Graph-Verarbeitung (Pregel, GraphX)
- Messaging (PubSub, Kafka)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Funktionsweise und Konzepte von Big-Data-Systemen in eigenen Worten erklären
- für gegebene Problemstellungen passende Big-Data-Frameworks auswählen und anwenden
- Datenanalysen entwerfen und auf Datenströmen, mithilfe der in den Übungen verwendeten Werkzeuge, durchführen

Literatur

- Relevante Publikationen werden in der Lehrveranstaltung 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 die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Erfolgreiche Teilnahme an den Übungen, welche aus drei Übungsblättern und einem Projekt (Umfang

- ca. 9 Übungsblätter) bestehen
- Bestehen der Prüfung

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Introduction to Linear Optimization (en. Introduction to Linear Optimization)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	jedes Wintersemester	Master-Studiengang Informatik	Englisch

Inhalte

The lecture covers the basic theory and applications of (integer) linear optimization. In the exercises the content of the lecture is applied and deepened. For that the exercises contain theoretical as well as practical elements. The students use linear and integer linear programming modeling software and solvers to solve applied programming exercises.

- Foundations of Linear Programming
- Linear Programs and their geometric interpretation
- Duality
- The Simplex method
- Integer Linear Programming
- Linear programming-based Branch-and-Bound
- Cutting planes and Branch-and-Cut
- Network flows
- Applications: Selected applications of linear optimization techniques

Lernergebnisse/Kompetenzen

After completing the course, students are able to

- state the basic principles of linear and integer linear programming (LP and ILP)
- build practical LP and ILP models
- design and implement algorithms that solve these models

Literatur

- Dimitris Bertsimas, John N. Tsitsiklis: Introduction to Linear Optimization. Athena Scientific. Belmont, MA, USA, 1997.
- Laurence Wolsey: Integer Programming. Wiley. New York, NY, USA, 1998.

Verwendbarkeit des Moduls

- Wahlpflichtbereich Theoretische Informatik
- Wahlpflichtbereich Praktische oder Technische Informatik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- active participation
- successful completion of exercise sheets
- passing written exam

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Gunnar W. Klau

Judgment Aggregation (en. Judgment Aggregation)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
5 LP	150 Stunden	60 Stunden	90 Stunden
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
Vorlesung (2 SWS) Übung (2 SWS)	unregelmäßig	Master-Studiengang Informatik	Deutsch

Inhalte

Dieses Modul beschäftigt sich mit der Aggregation von individuellen Urteilen über möglicherweise logisch miteinander verknüpfte Aussagen. Inhalt der Veranstaltung sind Unmöglichkeitsergebnisse und die Betrachtung von verschiedenen Verfahren mit ihren axiomatischen und algorithmischen Eigenschaften.

- Judgment Aggregation Prozeduren (z. B.: voraussetzungs-basierte, folgerungs-basierte, distanz-basierte)
- Axiomatische Eigenschaften und Unmöglichkeitsergebnisse
- Agendacharakterisierungen
- Verschiedene Formen der Einflussnahme (Manipulation, Kontrolle, Bestechung)

Zusätzliche Bemerkungen

- 7,5 LP für Studierende der PO 2005.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Verfahren zur Aggregation von individuellen Urteilen durchführen,
- theoretische Beweise zu Unmöglichkeitsergebnissen aus der Literatur nachvollziehen,
- Folgerungen der Unmöglichkeitsergebnisse in unterschiedlichen Bereichen benennen,
- neue Verfahren zur Aggregation von individuellen Urteilen entwickeln und auf ihre axiomatischen Eigenschaften untersuchen und
- Empfehlungen für bestimmte Einsatzbereiche geben.

Literatur

- 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. 2011.

Verwendbarkeit des Moduls

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

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Theoretische Informatik*
- Formal: Bachelor-Studierende müssen die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

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

Modulbeauftragte und hauptamtlich Lehrende

apl. Prof. Dr. Dorothea Baumeister

Master-Seminar: System-Software für Big-Data-Computing (en. Master's Seminar: System Software for Big Data Computing)

Leistungspunkte	Arbeitsaufwand	Kontaktzeit	Selbststudium
Lehrveranstaltungen	Turnus	Studiengang	Unterrichtssprache
5 LP	150 Stunden	30 Stunden	120 Stunden
Seminar (2 SWS)	jedes Sommersemester	Master-Studiengang Informatik	Deutsch

Inhalte

- Big-Data-Systeme: Batch-, In-Memory, Stream-, Graph-Processing; Distributed Machine Learning, ...
- Cloud-Systeme: Storage, Virtualisierung, Koordinierungsdienste, Messaging, ...
- Betriebssysteme für Manycore, NVRAM (persistenter Speicher), RDMA (Remote Direct Memory Access), GPUs, ...

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- relevante Inhalte zum gegebenen Thema recherchieren und schriftlich und mündlich wiedergeben,
- Aufsätze von anderen Studierenden evaluieren hinsichtlich Lesbarkeit und technischer Qualität,
- eine Präsentation planen und durchführen,
- sinnvolle Fragen stellen, beantworten und
- konstruktive Kritik annehmen sowie geben.

Literatur

- Relevante Publikationen werden in der Lehrveranstaltung 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 die Voraussetzungen für den Vorgriff auf Mastermodule erfüllen.

Voraussetzungen für die Vergabe von Leistungspunkten

- Schriftliche Ausarbeitung zu einem gegebenen Thema
- Teilnahme am Peer-Reviewing
- Präsentation des Themas
- Teilnahme an Diskussionen

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner