

| | | | | | | | | | |
|--|---|---------|---|--|-----------------------------------|------------------------------------|----------|-----------------|---|
| Applied Computational Engines: Solving Diverse Computational Problems in Practice | | | | | | | | Modulnummer: | |
| <i>Applied Computational Engines</i> | | | | | | | | | |
| Bachelor | | | | Schwerpunkt | | | | | |
| Pflicht <input type="checkbox"/> | | | | Computational Finance <input type="checkbox"/> | | | | | |
| Winf-Schwerpunkt-Pflicht <input type="checkbox"/> | | | | E-Business <input type="checkbox"/> | | | | | |
| Winf-Schwerpunkt-Wahlpflicht <input type="checkbox"/> | | | | IT-Management <input type="checkbox"/> | | | | | |
| Winf-Wahl <input type="checkbox"/> | | | | Logistik <input type="checkbox"/> | | | | | |
| Anzahl der SWS | V | UE | K | S | Prak. | Proj. | Σ | Kreditpunkte: 4 | Turnus Bei Interesse in jedem Sommersemester |
| | 2 | 1 | 0 | 0 | 0 | 0 | 3 | | |
| Formale Voraussetzungen: Keine | | | | | | | | | |
| Inhaltliche Voraussetzungen: Basic theoretical computer science and moderate proficiency of some programming language (for the practical exercises) | | | | | | | | | |
| Vorgesehenes Semester: ab 1. Semester | | | | | | | | | |
| Sprache: Englisch | | | | | | | | | |
| <p>Ziele: To be able to identify when difficult computational problems that can occur in the computer scientist's working life can be solved by standard computational engines.</p> <p>To know the strenghts and limits of a diverse set of computational engines, such as SAT solving, QBF solving, and linear programming.</p> <p>To be able to apply some commonly used computational engines to a wide variety of decision and optimization problems.</p> | | | | | | | | | |
| <p>Inhalte: Topics include:</p> <ul style="list-style-type: none"> • SAT Solving (Basic algorithms for SAT solving: unit propagation, backtracking, variable selection, and learning; Tseitin encoding and alternatives; SAT encodings in practice; Theory of tractability: "Backdoors") • Quantified Boolean Formula (QBF) solving • Integer Linear Programming (ILP) and Linear Programming (LP) as an "easy" subset (Definitions & encodings, Extension: Quadratic programming) • SMT solving (Basic idea and algorithms, SMT encodings of complex problems) • Supporting the encoding of difficult problems (Delta debugging & fuzz testing) • BDDs • Maximum flow algorithms & their applications • Automata for PSPACE-complete problems • Sub-engineering problems (clustering, ...) • Robust problem solving: games of infinite duration • Applied branch-and-bound | | | | | | | | | |
| <p>Unterlagen (Skripte, Literatur, Programme usw.):</p> <ul style="list-style-type: none"> • Armin Biere, Marijn Heule, Hans van Maaren, Toby Walsh (eds.): Handbook of Satisfiability, IOS Press, 2009 • Donald E. Knuth: The Art of Computer Programming (Volumes 1-4A), Addison Wesley, 2014 • Jon Kleinberg, Eva Tardos: Algorithm Design, 2006 | | | | | | | | | |
| <p>Form der Prüfung: i.d.R. Bearbeitung von Übungsaufgaben und Fachgespräch oder mündliche Prüfung</p> | | | | | | | | | |
| Arbeitsaufwand | | Präsenz | | 42 h | | Übungsbetrieb/Prüfungsvorbereitung | | 78 h | |
| | | Summe | | 120 h | | | | | |
| Lehrende: Rüdiger Ehlers | | | | | Verantwortlich: Rüdiger Ehlers | | | | |