

Theoretische Informatik 2: Berechenbarkeitsmodelle und Komplexität <i>Theoretical Computer Science 2</i>								Modulnummer:											
Bachelor Pflicht <input type="checkbox"/> Winf-Schwerpunkt-Pflicht <input type="checkbox"/> Winf-Schwerpunkt-Wahlpflicht <input type="checkbox"/> Winf-Wahl <input type="checkbox"/>				Schwerpunkt Computational Finance <input type="checkbox"/> E-Business <input type="checkbox"/> IT-Management <input type="checkbox"/> Logistik <input type="checkbox"/>															
Anzahl der SWS	<table border="1"> <tr> <td>V</td> <td>UE</td> <td>K</td> <td>S</td> <td>Prak.</td> <td>Proj.</td> <td>Σ</td> </tr> <tr> <td>2</td> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>4</td> </tr> </table>	V	UE	K	S	Prak.	Proj.	Σ	2	2	0	0	0	0	4	Kreditpunkte: 6		Turnus angeboten in jedem SoSe	
V	UE	K	S	Prak.	Proj.	Σ													
2	2	0	0	0	0	4													
Formale Voraussetzungen: -																			
Inhaltliche Voraussetzungen: -																			
Vorgesehenes Semester: 4.Semester																			
Sprache: Deutsch																			
Ziele: <ul style="list-style-type: none"> • Fundamentale Konzepte und Ergebnisse aus den Gebieten Berechenbarkeit, Komplexität und Prädikatenlogik kennen und verinnerlicht haben. • Verschiedene Berechnungsmodelle kennen und die Grenzen der Berechenbarkeit einschätzen können. • Die Komplexität von typischen Informatik-Problemen einschätzen können und sensibilisiert sein für die Existenz schwieriger Probleme. • Induktionsbeweise über die Struktur von Zahlen, Wörtern, Berechnungssequenzen und/oder ähnliche Strukturen nachvollziehen und selbständig durchführen können. • Selbständig Algorithmen entwerfen und formal spezifizieren können. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. 																			

Inhalte: 1) Berechenbarkeit

- Turingmaschinen
- Linear beschränkte Automaten
- Grammatiken der Typen 0 und 1, Abschlusseigenschaften
- LOOP-Programme und WHILE-Programme
- Primitiv rekursive Funktionen und -rekursive Funktionen
- Unentscheidbarkeit
- Unentscheidbare Probleme für Turingmaschinen
- Satz von Rice
- Postisches Korrespondenzproblem
- Äquivalenzproblem kontextfreier Grammatiken
- Semi-Entscheidbarkeit und Rekursive Aufzählbarkeit
- Universelle Turingmaschinen
- Reduktionen

2) Komplexität

- Zeit- und Platzbeschränkte Turingsmaschinen
- Komplexitätsklassen P, NP, PSpace, ExpTime
- P vs NP-Problem
- NP-Vollständigkeit
- NP-vollständige Probleme aus verschiedenen Gebieten
- Komplemente und coNP
- Approximation NP-harter Probleme
- Satz von Savitch

Unterlagen (Skripte, Literatur, Programme usw.):

- J.E. Hopcroft, R. Motwani, J.D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium 2011
- J.E. Hopcroft, R. Motwani, J.D. Ullman: Introduction to Automata Theory, Languages, and Computation (3rd edition). Pearson Education, 2014
- C. Lutz: Theoretische Informatik 2, Skript

Form der Prüfung:

i. d. R. Bearbeitung von Übungsaufgaben und Fachgespräch

Arbeitsaufwand	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h

Lehrende:
Prof. Dr. C. Lutz u.a.

Verantwortlich:
Prof. Dr. C. Lutz