

<b>Theoretische Informatik 1 (Kopie vom Mon Jun 22 14:28:09 +0200 2020) (deleted:Mon Jun 22 14:28:54 +0200 2020)</b> <i>Theoretical Computer Science 1</i>								Modulnummer:			
Bachelor Pflicht/Wahl <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/> Wahl <input type="checkbox"/> Sonderfall <input type="checkbox"/>				Modulbereich: Pflicht							
Anzahl der SWS		V	UE	K	S	Prak.	Proj.	$\Sigma$	Kreditpunkte: 9		Turnus angeboten in jedem WiSe
		4	2	0	0	0	0	6			
Formale Voraussetzungen: -											
Inhaltliche Voraussetzungen: -											
Vorgesehenes Semester: ab 1. Semester											
Sprache: Deutsch											
Ziele: <ul style="list-style-type: none"> <li>• Formale Grundlagen und elementare Fragestellungen der Informatik kennen und die fundamentale Rolle der Theorie in der Informatik verstehen.</li> <li>• Konzepte zur formalen Beschreibung und Analyse von Informatiksystemen kennen.</li> <li>• Beherrschung der grundlegenden Methoden aus den Bereichen der Automatentheorie, formalen Sprachen und Algorithmen.</li> <li>• Beherrschung elementarer Beweistechniken und Beweise selbst durchführen können.</li> <li>• Probleme analysieren, von spezifischen Gegebenheiten abstrahieren und formale Modelle in mathematischen Definitionen darstellen können.</li> <li>• Algorithmen für diese Probleme kennen und auf neue Problemvarianten anwenden können.</li> <li>• Korrektheit von Algorithmen beweisen und Eigenschaften von Algorithmen analysieren können.</li> <li>• Eigenständig und in Gruppen Lösungsstrategien für formale Problemstellungen entwickeln können und Lösungen verständlich präsentieren.</li> </ul>											

Inhalte: A) Automaten und formale Sprachen

1. Endliche Automaten und Reguläre Sprachen:

- Definition und Grundbegriffe
- Nichtdeterminismus
- Nichterkennbarkeit von Sprachen und Pumping-Lemma
- Abschlusseigenschaften
- Potenz- und Produktautomat
- Leerheits-, Wort- und Äquivalenzproblem
- regulare Ausdrücke
- Minimale Automaten und Nerode-Rechtskongruenz
- Rechtslineare Grammatiken

1. Kontextfreie Sprachen:

- Grammatiken und Chomsky-Hierarchie
- kontextfreie Grammatiken
- Chomsky Normalform
- Leerheits-, Wort- und Äquivalenzproblem
- CYK-Algorithmus
- Abschlusseigenschaften
- Pumping-Lemma
- Kellerautomaten

B) Algorithmentheorie

- Algorithmenbegriff
- Korrektheit und Komplexität von Algorithmen
- Suchen und Rekursionen
- Sortieren
- Graphen und elementare Graphenalgorithmen: Graphdurchläufe, MST und SP
- Algorithmen Paradigmen: Divide and Conquer, Greedy Algorithmen, Dynamische Programmierung

Lehrveranstaltung(en): Dieses Modul besteht aus zwei Veranstaltungen, jeweils im Format 2VL+1UE, die beide verpflichtend sind.

- 03-IBGT-AFS Automaten und formale Sprachen
- 03-IBGT-AT Algorithmentheorie

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 1, Skript
- D. Kozen: Automata and Computability, Springer, 2007
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein: Introduction to Algorithms, MIT Press, 2003
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein: Algorithmen - Eine Einführung, Walter de Gruyter GmbH & Co KG, 2017
- Martin Dietzfelbinger, Kurt Mehlhorn, Peter Sanders: Algorithmen und Datenstrukturen: Die Grundwerkzeuge, Springer-Verlag, 2014
- T. Ottmann and P. Widmayer. Algorithmen und Datenstrukturen. Spektrum Verlag, 2002.

Form der Prüfung:

TP, PL1: 50%, PL2: 50%, Klausur, Fachgespräch, ggf. Bonusprüfung

Arbeitsaufwand	Präsenz	84 h
	Übungsbetrieb/Prüfungsvorbereitung	186 h
	Summe	270 h

Lehrende:

Prof. Dr. C. Lutz, Prof. Dr. N. Megow, Prof. Dr. S. Siebertz

Verantwortlich:

Prof. Dr. C. Lutz