

Modulcode <small>(1.)</small>	Modulbezeichnung <small>(2.)</small>	Zuordnung <small>(3.)</small>
BAI2020	Theoretische Informatik 2 (THI2)	
	Studiengang <small>(4.)</small>	Bachelor Angewandte Informatik
	Fakultät <small>(5.)</small>	Gebäudetechnik und Informatik

Modulverantwortlich <small>(6.)</small>	Prof. Dr. Oksana Arnold
Modular <small>(7.)</small>	Pflicht
Angebotshäufigkeit <small>(8.)</small>	SS
Regelbelegung / Empf. Semester <small>(9.)</small>	BA2
Credits (ECTS) <small>(10.)</small>	5 CP
Leistungsnachweis <small>(11.)</small>	PL (N)
Unterrichtssprache <small>(12.)</small>	Deutsch
Voraussetzungen für dieses Modul <small>(13.)</small>	-
Modul ist Voraussetzung für <small>(14.)</small>	BAI0302: Softwaretechnik 1 incl. PM Grundlagen BAI0402: Softwaretechnik 2 BAI0553: Einführung in die KI BAI0603: Algorithmen MAI0202: Lernende Systeme
Moduldauer <small>(15.)</small>	1 Semester
Notwendige Anmeldung <small>(16.)</small>	-
Verwendbarkeit des Moduls <small>(17.)</small>	Informatik & technische Studiengänge, in denen grundlegende Modellierungskompetenzen benötigt werden; spez. auch für Compilerbau

Lehrveranstaltung <small>(18.)</small>	Dozent/in <small>(19.)</small>	Art <small>(20.)</small>	Teilnehmer (maximal) <small>(21.)</small>	Anzahl Gruppen <small>(22.)</small>	SWS <small>(23.)</small>	Workload	
						Präsenz <small>(24.)</small>	Selbst-studium <small>(25.)</small>
1 Theoretische Informatik 2	Arnold	V	100	1	2	30	35
2 Theoretische Informatik 2	Arnold	Ü	25	4	2	30	30
Summe						4	60
Workload für das Modul						(26.)	125

Qualifikationsziele	<p>Die Studierenden können</p> <ul style="list-style-type: none"> • formale Sprachen in die Chomsky-Hierarchie einordnen, • deterministische endliche Automaten minimieren und auf Äquivalenz prüfen, • nichtdeterministische Akzeptoren in deterministische umwandeln, • kontextfreie Grammatiken entwerfen, • die Chomsky-NF erzeugen, das Wortproblem mit dem CYK-Algorithmus lösen und daran das Konzept der dynamischen Programmierung über hashtable & memoization in die Programmierung übertragen, • die Greibach-NF erzeugen und einen Kellerautomaten realisieren, • Turingmaschinen für einstellige Funktionen entwerfen. <p>Sie verstehen, was sich hinter Konzepten wie</p> <ul style="list-style-type: none"> • die Universelle Turingmaschine und Gödelnumerierung • Entscheidbarkeit, Semientscheidbarkeit, Aufzählbarkeit verbirgt. <p>Sie kennen verschiedene Berechenbarkeitsmodelle und deren Ausdrucksfähigkeit.</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> • modellieren primitiv-rekursive, allgemein-rekursive und partiell-rekursive Funktionen, • wissen, was das Halteproblem und das Äquivalenzproblem bedeuten, und können sie erklären. <p>Sie kennen den Unterschied zwischen unentscheidbaren Problemen und np-vollständigen. Sie können np-vollständige Probleme nennen und deren Komplexitätsklasse angeben. Sie sind darüber hinaus, mindestens ein np-vollständiges Problem polynomiell auf ein anderes np-vollständiges Problem reduzieren.</p>
Inhalte	<p>(28.)</p> <ul style="list-style-type: none"> • Ausdrucksfähigkeit von Endlichen Automaten & Algorithmen zur Umwandlung und Minimierung • Reguläre Ausdrücke • Kontextfreie Sprachen & Grammatiken • Pumping-Lemma für reguläre Sprachen und kontextfreie • Chomsky-Normalform und CYK-Algorithmus (inkl. Konzept der Dynamischen Programmierung und Memoisation) • Greibach-Normalform und Kellerautomaten • Abschlusseigenschaften formaler Sprachen & Konsequenzen • Kontextsensitive Sprachen & Phrasenstruktursprachen • Turingmaschine und Universelle Turingmaschine • Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit • Berechenbarkeitsmodelle • Rekursive Funktionen und Gödelnumerierung • Unentscheidbare Probleme • Komplexitätstheorie und Komplexitätsklassen • np-vollständige Probleme und polynomielle Reduktion • Cooksche These
Vorleistungen und Modulprüfung	<p>(29.)</p> <p>Vorleistungen:</p> <ul style="list-style-type: none"> • keine <p>Modulprüfung:</p> <ul style="list-style-type: none"> • 100% Klausur über 90 min im Prüfungszeitraum

Literatur

(30.)

- Dirk W. HOFFMANN: Theoretische Informatik.
2. aktualisierte Auflage, Hanser Verlag, München 2011.
- Uwe SCHÖNING: Theoretische Informatik – kurz gefasst.
5. Auflage, Spektrum Akadem. Verlag, Heidelberg 2009.
- John E. HOPCOFT, Rajeev MOTWANI, Jeffrey D. ULLMAN:
Einführung in die Automatentheorie, Formale Sprachen und
Komplexitätstheorie. 2. überarbeitete Auflage, Pearson
Studium Verlag, München 2002.
- Dirk W. HOFFMANN: Grenzen der Mathematik - Eine Reise
durch die Kerngebiete der mathematischen Logik.
Spektrum Akademischer Verlag, Heidelberg 2011.