

<b>Modulcode</b> (1.)	<b>Modulbezeichnung</b> (2.)	<b>Zuordnung</b> (3.)
BAAI-1220	Theoretische Informatik 2 (THI2)	
	<b>Studiengang</b> (4.)	Bachelor Angewandte Informatik
	<b>Fakultät</b> (5.)	Gebäudetechnik und Informatik

<b>Modulverantwortlich</b> (6.)	Prof. Dr. Oksana Arnold
<b>Modulart</b> (7.)	Pflicht
<b>Angebotshäufigkeit</b> (8.)	SS
<b>Regelbelegung / Empf. Semester</b> (9.)	BA2
<b>Credits (ECTS)</b> (10.)	5 CP
<b>Leistungsnachweis</b> (11.)	PL (N)
<b>Unterrichtssprache</b> (12.)	Deutsch
<b>Voraussetzungen für dieses Modul</b> (13.)	-
<b>Modul ist Voraussetzung für</b> (14.)	BAAI-8610: Einführung in die KI MAAI-2130: Lernende Systeme MAAI-8130: Lernende Systeme
<b>Moduldauer</b> (15.)	1 Semester
<b>Notwendige Anmeldung</b> (16.)	-
<b>Verwendbarkeit des Moduls</b> (17.)	Informatik & technische Studiengänge, in denen grundlegende Modellierungskompetenzen benötigt werden; spez. auch für Compilerbau

	<b>Lehrveranstaltung</b> (18.)	<b>Dozent/in</b> (19.)	<b>Art</b> (20.)	<b>Teilnehmer (maximal)</b> (21.)	<b>Anzahl Gruppen</b> (22.)	<b>SWS</b> (23.)	<b>Workload</b>	
							<b>Präsenz</b> (24.)	<b>Selbststudium</b> (25.)
1	Theoretische Informatik 2	Arnold	V	100	1	2	30	35
2	Theoretische Informatik 2	Arnold	Ü	25	4	2	30	30
<b>Summe</b>						<b>4</b>	<b>60</b>	<b>65</b>
<b>Workload für das Modul</b> (26.)							<b>125</b>	

<b>Qualifikationsziele</b>	<p>27.</p> <p>Die Studierenden können</p> <ul style="list-style-type: none"> <li>• formale Sprachen in die Chomsky-Hierarchie einordnen,</li> <li>• deterministische endliche Automaten minimieren und auf Äquivalenz prüfen,</li> <li>• nichtdeterministische Akzeptoren in deterministische umwandeln,</li> <li>• kontextfreie Grammatiken entwerfen,</li> <li>• die Chomsky-NF erzeugen und das Wortproblem mit dem CYK-Algorithmus lösen,</li> <li>• die Greibach-NF erzeugen und einen Kellerautomaten realisieren,</li> <li>• Turingmaschinen für einstellige Funktionen entwerfen.</li> </ul> <p>Sie verstehen, was sich hinter Konzepten wie</p> <ul style="list-style-type: none"> <li>• die Universelle Turingmaschine und Gödelnumerierung</li> <li>• Entscheidbarkeit, Semientscheidbarkeit, Aufzählbarkeit verbirgt.</li> </ul> <p>Sie kennen verschiedene Berechenbarkeitsmodelle und deren Ausdruckfähigkeit.</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• modellieren primitiv-rekursive, allgemein-rekursive und partiell-rekursive Funktionen,</li> <li>• wissen, was das Halteproblem und das Äquivalenzproblem bedeutet, und können es erklären.</li> </ul> <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>
<b>Inhalte</b>	<p>28.</p> <ul style="list-style-type: none"> <li>• Ausdruckfähigkeit von Endlichen Automaten &amp; Algorithmen zur Umwandlung und Minimierung</li> <li>• Reguläre Ausdrücke</li> <li>• Kontextfreie Sprachen &amp; Grammatiken</li> <li>• Pumping-Lemma für reguläre Sprachen und kontextfreie</li> <li>• Chomsky-Normalform und CYK-Algorithmus (inkl. Konzept der Dynamischen Programmierung)</li> <li>• Greibach-Normalform und Kellerautomaten</li> <li>• Abschlusseigenschaften formaler Sprachen &amp; Konsequenzen</li> <li>• Kontextsensitive Sprachen &amp; Phrasenstruktursprachen</li> <li>• Turingmaschine und Universelle Turingmaschine</li> <li>• Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit</li> <li>• Berechenbarkeitsmodelle</li> <li>• Rekursive Funktionen und Gödelnumerierung</li> <li>• Unentscheidbare Probleme</li> <li>• Komplexitätstheorie und Komplexitätsklassen</li> <li>• np-vollständige Probleme und polynomielle Reduktion</li> <li>• Cooksche These</li> </ul>
<b>Vorleistungen und Modulprüfung</b>	<p>29.</p> <p>Vorleistungen:</p> <ul style="list-style-type: none"> <li>• keine</li> </ul> <p>Modulprüfung:</p> <ul style="list-style-type: none"> <li>• 100% Klausur über 90 min im Prüfungszeitraum</li> </ul>

## 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.