

Formale Sprachen und Komplexität (IN5005)

Title	Formal Languages and Complexity Theory	
Typ	Vorlesung mit Übungen	
Credits	6	
Lehrform/SWS	3V + 2Ü	
Sprache	Deutsch	
Modulniveau	Bachelor	
Arbeitsaufwand	Präsenzstunden	75 Stunden
	Eigenstudium	105 Stunden
	Gesamtaufwand	180 Stunden
Angestrebte Lernergebnisse	<p>Die Studierenden kennen die theoretischen Grundlagen der Informatik und sind in der Lage, sie auf praktische Fragestellungen anzuwenden. Beispielsweise können sie ein vorgelegtes Problem als NP-vollständig identifizieren oder zustandsorientierte Spezifikationen als endliche Automaten erkennen und können Methoden wie Determinisierung und Minimierung auf endliche Automaten anwenden. Darüber hinaus verstehen die Studierenden abstrakte theoretische Inhalte und können mathematische Beweise nachvollziehen.</p>	
Intended Learning Outcomes	<p>The students are able to remember and reproduce theoretical foundations of computer science and are able to apply them to practical problems. For example they are able to identify a problem as NP-complete or to identify state-based specifications as finite automata as well as they are able to apply determinization and minimization methods to finite automata. In addition the students understand abstract theoretical topics and are able to follow mathematical proofs.</p>	
Inhalt	<p>Das Modul vermittelt Grundkenntnisse in den Gebieten Formale Sprachen, Berechenbarkeit und Komplexitätstheorie. Im Einzelnen werden vermittelt:</p> <ul style="list-style-type: none"> • Automatentheorie und Formale Sprachen: Chomsky-Hierarchie, reguläre Sprachen und endliche Automaten, kontextfreie Sprachen und Kellerautomaten, kontextsensitive Sprachen; • Berechenbarkeit: Turing-Maschinen und andere Berechnungsmodelle, Unentscheidbarkeit, Halteproblem, rekursiv aufzählbare Probleme; • Komplexitätstheorie, insbesondere die Klassen P und NP, Definition und Beweise für NP- 	

	Vollständigkeit.
Contents	<p>The module provides basic knowledge in the areas of formal languages, computability and complexity theory.</p> <p>The main topics of the module are as follows:</p> <ul style="list-style-type: none"> • automata and formal Languages: Chomsky hierarchy, regular languages and finite automata, context-free languages and pushdown automata, context-sensitive languages; • computability: Turing machines and other models of computation, undecidability, halting problem, recursively enumerable problems; • complexity theory, especially the classes P and NP, definition and proof of NP completeness.
Prüfung	<p>Prüfungsleistung (benotet): Klausur (90 min)</p> <p>Wiederholungsprüfung am Ende des Semesters oder im Folgesemester. Details werden zu Beginn des Moduls bekannt gegeben.</p> <p>In der Klausur weisen die Studierenden nach, dass sie die Grundkonzepte der Automatentheorie, formalen Sprachen, Berechenbarkeit und Komplexität wiedergeben und anwenden können. Konkret werden in der Klausur Aufgaben bearbeitet, die eine eigenständige Anwendung der Konzepte, Methoden und Techniken der Automatentheorie, formalen Sprachen, Berechenbarkeit und Komplexität auf eine gegebene Problemstellung erfordern. Beispielsweise können sie eine gegebene formale Sprache in die Chomsky-Hierarchie einordnen und sowohl eine generierende Grammatik als auch einen erkennenden Automaten hierfür angeben und umgekehrt. Für reguläre Sprachen können sie die zugehörigen endlichen Automaten sowohl determinisieren als auch minimieren. Sie können Probleme bzgl. der Komplexitätsklassen P und NP einordnen und Turing-Reduktionen für konkrete Probleme durchführen.</p>
Examination	<p>Examination requirements: written exam (90-120 min)</p> <p>A makeup exam will be offered at the end of the semester or in the following semester. Details will be announced at the beginning of the module.</p> <p>Within the written exam, students demonstrate that</p>

	<p>they are able to reproduce and to apply the fundamental concepts of automata theory, formal languages, computability, and complexity theory. The written exam consists of assignments, which require independent application of the concepts, methods, and techniques of automata theory, formal languages, computability and complexity to solve given problems. For instance, for a given formal language, students are able to classify it within the Chomsky hierarchy, to construct recognizing automata as well as generating grammars for it and vice versa. For regular languages, they are able to determinize and minimize the corresponding finite automaton. Students are able to classify a given problem within P and NP and are able to construct a Turing reduction for specific problems.</p>
Literatur/Literature	<p>Uwe Schöning Theoretische Informatik - kurz gefasst Spektrum Akademischer Verlag</p>
Medienformen	<p>Folienpräsentation, Tafelanschrieb</p>
Media	<p>slide show, blackboard</p>
Lehr- und Lernmethode	<p>Vorlesung, Tutorübung, Aufgaben zum Selbststudium.</p> <p>Das Modul besteht aus einer Vorlesung sowie Übungen in kleinen Gruppen. In den Hausaufgaben, die freiwillig abzugeben sind, wird das Verständnis der Konzepte, Automaten, Grammatiken und Reduktionen, die in der Vorlesung vorgestellt werden, anhand konkreter Beispiele vertieft. Mit Hilfe der in der Vorlesung vorgestellten Konzepte, können die Studierenden gegebene formale Sprachen in die Chomsky-Hierarchie einordnen, generative Grammatiken und erkennende Automaten hierfür entwerfen sowie eine Klassifizierung in die Klassen P und NP vornehmen und Turing-Reduktionen angeben. In den Hausaufgaben werden selbständig anspruchsvolle Übungsaufgaben bearbeitet, die ähnlich wie die Klausuraufgaben sind und daher zur Vorbereitung darauf dienen. In den Übungen werden mögliche Lösungsansätze der Aufgaben zum Selbststudium diskutiert.</p>
Teaching and Learning Methods	<p>Lecture, tutorial, assignments for individual study.</p> <p>The module consists of a lecture and tutorials in small groups. Within the assignments (submission is optional), concept, automata, grammars, and reductions (presented in the lecture) will be applied to real examples to deepen the understanding. Using the concepts presented in the lecture, students are able to</p>

	<p>classify given formal languages into the Chomsky hierarchy, construct generative grammars and accepting automata therefor. Further, they are able to classify a problem to the class P or NP and to construct Turing-reduction between languages. The assignments consist of demanding problems similar to the assignments in the written exam and thus serve as a preparation for the exam. Within the tutorials possible approaches for solutions of the assignments will be discussed.</p>
Turnus	Sommersemester
Modulverantwortlicher	Prof. Dr. Martin Hofmann
Dozenten	Professoren der Informatik (LMU)