

12 Theoretische Informatik Theoretical Informatics	
Semester	2
Credit Points	5
Pflicht/ Wahlpflicht	Pflicht
Häufigkeit des Angebotes/ Verwendbarkeit	Jedes Semester nach Bedarf der VFH-Hochschulen
Modulverantwortliche(r)	Prof. Dr. rer.nat. Friedhelm Seutter, Ostfalia Hochschule für angewandte Wissenschaften
Teilnahmevoraussetzungen	Grundlagen der Mathematik, Informatik, Programmierung
Lernergebnisse	<p>Die Studierenden...</p> <ul style="list-style-type: none"> • kennen grundlegende Modelle und Methoden der Theoretische Informatik und ihre Beziehungen untereinander. • verstehen formale Notationen und die ausgehend von Definitionen durch Sätze ausgedrückten Zusammenhänge und Beziehungen und die verwendeten Konstruktions- und Beweisideen. • verstehen Automatenmodelle und algebraische und generierende Konzepte zur Definition formaler Sprachen. • können die auf formaler Ebenen erworbenen Erkenntnisse auf Anwendungen in der Praxis, unter Berücksichtigung ihrer Beschränkungen, übertragen und anwenden. • können konkrete Probleme analysieren und eine Reduktion und Abstraktion des Problems durchführen, um das unbedingt Notwendige für die Lösung des Problems herauszustellen. • können ein Problem formal darstellen (mittels Modellen und Methoden der theoretischen Informatik), um es zu lösen. • verstehen Beschränkungen und Grenzen der Modelle und Methoden zur algorithmischen Berechnung von Lösungen und können diese in Bezug auf konkrete Anwendungen bewerten und auswählen.
Prüfungsvorleistung	Einsendaufgabe
Medien-/ Lernform	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung (E-Mail, Foren, Chat, Webkonferenzen, Einsendaufgaben u. a.) sowie ggf. Präsenzphasen
Arbeitsaufwand	Prüfung: 120 Minuten Selbststudium: 107 h Betreutes Lernen: 31 h Vorbereitung PVL: 12 h
Präsenzart	erfordert physische Anwesenheit

Präsenzinhalte	Zusammenfassung und Wiederholung ausgewählter Abschnitte aus dem Studienmodul, Klärung inhaltlicher Fragen, Besprechung von Übungsaufgaben, Klausurvorbereitung.
Prüfungsform	Klausur (120 min.)
Literatur	Sipser, M.: Introduction to the Theory of Computation. 3rd Edition. Sengage Learning, 2013. ISBN 13-978-1-133-18781-3 Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D.: Introduction to Automata Theory, Languages, and Computation. Third Edition. Boston, Addison-Wesley 2007. ISBN 0-321-47617-4
weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Studieninhalte
<p>Das Studienmodul gibt eine Einführung in einige grundlegenden Modelle und Methoden der Theoretischen Informatik. Anhand von Automatenmodellen und von diesen analysierbaren formalen Sprachen werden die grundsätzlichen Fähigkeiten und Beschränkungen von Computern und Softwaresystemen untersucht. Dabei stehen insbesondere die Beziehungen zwischen den Automatenmodellen als analysierende Konzepte und den beschreibenden bzw. generierenden Konzepten für formale Sprachen im Vordergrund. Darüber hinaus wird die Frage diskutiert und beantwortet, ob gewisse Probleme überhaupt durch einen Computer oder ein Softwaresystem lösbar sind oder sich einer algorithmischen Berechnung verschließen. Die Studierenden sollen diese Modelle, Methoden und Konzepte kennen lernen und verstehen, sie in ihren fachlichen Kontext einordnen und in konkreten Problemen anwenden können.</p> <p>Die Modelle, Methoden und Konzepte und ihre Beziehungen untereinander werden teils informell erläutert, teils formal definiert bzw. hergeleitet. Für das Studium (insbesondere die Programmierausbildung) und die Praxis (insbesondere die Softwareentwicklung) können diese theoretischen Modelle grundlegende Erkenntnisse und Hinweise zur Lösung diverser Probleme liefern. Computer und Softwaresysteme sind technische Systeme, die mit Hilfe mathematisch-formaler Modelle und Beschreibungen entwickelt und bedient werden. Auch neue Anwendungen sind auf dieser Basis zu konzipieren. Es ist deshalb unerlässlich, abstrakte Modelle und die darauf anzuwendenden Methoden mittels mathematisch-formaler Beschreibungen von Zuständen und Abläufen entwickeln, anpassen und anwenden zu können. Auch diese Kompetenzen sollen mit diesem Studienmodul eingeübt und vertieft werden.</p> <ol style="list-style-type: none"> 1. Formale Sprachen <ul style="list-style-type: none"> • Alphabete, Wörter und Sprachen • Zusammenhang mit Programmiersprachen 2. Endliche Automaten <ul style="list-style-type: none"> • Deterministische endliche Automaten • Nichtdeterministische endliche Automaten 3. Reguläre Sprachen (Arbeitsaufwand ca. 25h)

- Reguläre Sprachen und Operationen
 - Reguläre Ausdrücke
 - Eigenschaften regulärer Sprachen
4. Kontextfreie Sprachen
- Kontextfreie Grammatiken
 - Kellerautomaten
 - Eigenschaften kontextfreier Sprachen
5. Turingmaschinen und Berechenbarkeit
- Deterministische Turingmaschinen
 - Intuitiver Algorithmusbegriff
 - Turing-Berechenbarkeit
6. Entscheidbarkeit
- Entscheidbare Probleme
 - Das Halteproblem