

Modulcode (1.)	Modulbezeichnung (2.)	Zuordnung (3.)
MAAI-8130	Lernende Systeme (LS)	
	Studiengang (4.)	Master Angewandte Informatik
	Fakultät (5.)	Gebäudetechnik und Informatik

Modulverantwortlich (6.)	Prof. Dr.-Ing. Oksana Arnold
Modulart (7.)	Wahl
Angebotshäufigkeit (8.)	SS
Regelbelegung / Empf. Semester (9.)	MA1 (MA2 bei Immatrikulation im WS)
Credits (ECTS) (10.)	5 CP
Leistungsnachweis (11.)	SL (N)
Unterrichtssprache (12.)	Deutsch / Englisch
Voraussetzungen für dieses Modul (13.)	BAAI-1120: Theoretische Informatik 1 BAAI-1220: Theoretische Informatik 2 BAAI-1140: Grundkonzepte der Programmierung BAAI-1240: Softwaretechnik 1 BAAI-1420: Softwaretechnik 2 BAAI-1520: Algorithmen
Modul ist Voraussetzung für (14.)	-
Moduldauer (15.)	1 Semester
Notwendige Anmeldung (16.)	-
Verwendbarkeit des Moduls (17.)	Informatik & technische Studiengänge, in denen Programmsynthese von Interesse ist

Lehrveranstaltung (18.)	Dozent/in (19.)	Art (20.)	Teilnehmer (maximal) (21.)	Anzahl Gruppen (22.)	SWS (23.)	Workload	
						Präsenz (24.)	Selbst- studium (25.)
1 Lernende Systeme	Arnold	V	15	1	2	30	20
2 Lernende Systeme	Arnold	S	15	1	2	30	45
Summe					4	60	65
Workload für das Modul (26.)						125	

Qualifikationsziele

27.

Der ständige Umgang mit neuen Herausforderungen nötigt vor allem software-herstellende Unternehmen ihre Entwicklungsprozesse weiter zu optimieren und zu verbessern. Ziel dieser Lehrveranstaltung ist es in erster Linie, Abstraktions- und Denkvermögen weiter zu stärken und auszubauen, um sich systematisch mit dem Konzept der induktiven Inferenz auseinanderzusetzen. Dabei kommt man bei Begriffen wie Lernbarkeit, Umgang mit unvollständiger Information, Konvergenz im Lernprozess und den notwendigen Informationsangeboten für den Erfolg des Lernprozesses vorbei. Gleichzeitig werden bekannte Konzepte der Theoretischen Informatik wiederholt und Algorithmen zur Programmsynthese untersucht, bewertet und punktuell implementiert.

- **Fachkompetenz:** Die Studierenden beherrschen wichtige Denk- und Modellierungsstrukturen der Informatik. Insbesondere beherrschen sie die präzise und lösungsorientierte Beschreibung von Lernproblemen. Sie können Lernverfahren beschreiben, vergleichen und bewerten. Sie beherrschen Grundfähigkeiten für den Entwurf lernender Systeme. Sie verstehen die Relevanz von algorithmentheoretischen Kenntnissen und Methoden für bestimmte Anwendungen.
- **Analyse- und Designkompetenz:** Die Studierenden können Lernprobleme in Bezug auf ihre grundsätzliche Lösbarkeit abschätzen. Sie können einordnen, mit welchen wünschenswerten Eigenschaften praktische Lernverfahren ausgestattet werden müssen. Sie können in den Termini der Theoretischen Informatik die Aufgaben beim Design lernender Systeme formulieren.
- **Realisierungs- und Projektmanagementkompetenz:** Die Studierenden können typische Problemstellungen der theoretischen Informatik in einer Anwendungsaufgabe identifizieren und Projekte nach grundsätzlichen Schwierigkeitsgraden einstufen und zerlegen. Sie können neue Problemstellungen auf gut untersuchte Standardprobleme reduzieren und so geeignete effiziente Lösungen erarbeiten und verifizierbare Aussagen über die Güte einer Lösung ableiten.
- **Methodenkompetenz, Sozial- und Selbstkompetenz:** Die Studierenden können in Anwendungsgebieten auftretende Sachverhalte wissenschaftlich präzise ausdrücken und beschreiben. Sie sind umgekehrt in der Lage, das in der Literatur und wissenschaftlichen Community vorhandene Methodenwissen für eigene Anwendungsfälle produktiv zu erschließen.

Inhalte	<p style="text-align: right;">28</p> <ul style="list-style-type: none"> • Szenarien der Induktiven Inferenz • Formulierung von Lernproblemen für berechenbare Funktionen, Automaten, Formale Sprachen, Patternsprachen • Hierarchien von lösbaren Lernproblemen • Analyse der Eigenschaften von Lernaufgaben und -domänen • Beweismethoden der Algorithmentheorie • Aussagen über Möglichkeiten, Grenzen und Komplexität des Induktiven Lernens • Bezug der Induktiven Inferenz zu Informatik-Anwendungen und Methoden des Maschinellen Lernens • Nutzung von Algorithmen zur Programmsynthese
Vorleistungen und Modulprüfung	<p style="text-align: right;">29</p> <p>Vorleistungen:</p> <ul style="list-style-type: none"> • keine <p>Modulprüfung:</p> <ul style="list-style-type: none"> • 50% Projekt zu einem Lernverfahren • 50% Klausur (90 min) in der letzten Semesterveranstaltung
Literatur	<p style="text-align: right;">30</p> <ul style="list-style-type: none"> • ARNOLD, O.: Folien zur Vorlesungen • Sanjay JAIN, James S. ROYER, Arun Kumar SHARMA: Systems That Learn: An Introduction to Learning Theory (Learning, Development, & Conceptual Change). MIT Press, Cambridge, MA, USA 1999. • Boris Avraamovich TRAKHTENBROT, IAn Martynovich BARZDIN': Finite automata; behavior and synthesis. North-Holland Pub. Co. 1973. • Hartley ROGERS, Jr.: The Theory of Recursive Functions and Effective Computability. 3.Ausgabe, MIT Press, Cambridge, MA, USA 1992. • Ethem ALPAYDIN: Maschinelles Lernen. Oldenbourg Wissenschaftsverlag GmbH, München 2008. • 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. • Weitere Literatur in der Veranstaltung