

1.1.1 Modellierung und Analyse komplexer Systeme (MAX)/Bachelor

1.1.1.1 Ziele

Die Konstruktion und Beherrschung großer Software-Systeme ist nur durch ein systematisches Vorgehen möglich, das eine geeignete Modellbildung der Systeme sowie die Darlegung und Analyse ihrer Verhaltenseigenschaften umfasst. Dazu werden in dieser Vertiefungsrichtung

- fundierte Methoden zur Modellierung von Softwaresystemen eingeführt;
- Fallstudien zur Untersuchung von Verhaltenseigenschaften besprochen;
- erprobte Werkzeuge und effiziente Analyseverfahren vorgestellt.

Angesprochen werden

- mathematisch orientierte Studierende, die Zusammenhänge verstehen und herstellen wollen;
- algorithmisch orientierte Studierende, die Werkzeuge benutzen und entwickeln möchten.

1.1.1.2 Zu erwerbende Kenntnisse, Fertigkeiten und Kompetenzen

Kenntnisse:

- Kenntnisse von formalen Systemmodellen (Transitionssysteme, Petrinetze, Graphersetzungssysteme, Büchi-Automaten, Realzeitautomaten) und zugehörigen Äquivalenzbegriffen;
- Kenntnisse von Systemeigenschaften: verschiedene Konzepte der Sicherheit (mit den Aspekten „correctness“, „safety“ und „security“) sowie deren Formalisierung in Logiken wie Prädikatenlogik und temporaler Logik;
- Kenntnisse von Analysemethoden: Beweissysteme zur Programmverifikation, Methoden zur Programmanalyse, Algorithmen zum Model-Checking, linear-algebraische Methoden zur Korrektheitsanalyse; Umgang mit computerunterstützten Analysewerkzeugen.

Fertigkeiten:

- Umgang mit formalen Beschreibungssprachen;
- Einsatz adäquater Modellierungs- und Analysemethoden für konkrete Anwendungsszenarien;
- Methoden zur Reduktion großer Zustandsräume.

Kompetenzen:

- Verständnis der Konzepte „correctness“, „safety“ und „security“;
- Fähigkeit zur Anwendung von erprobten Methoden zum Systementwurf und zur Systemanalyse.

1.1.1.3 Studienanforderungen

Studierende der Vertiefungsrichtung MAX belegen fünf Wahlmodule aus folgender (nicht abschließender) Liste:

- Petri-Netze PN
- Kryptologie KRY
- Graphersetzungssysteme GES
- Praktikum Realzeitsysteme P-RZS
- Algorithmen zur Software-Verifikation ASV
- Formale Methoden eingebetteter Systeme FMES
- Programmverifikation PV
- Model-Checking MC.

- Algorithmische Graphentheorie

Die Studierenden erstellen eine BSc-Abschlussarbeit aus dem Bereich MAX.

1.1.1.4 Ziele-Matrix

Tabelle 8: Ziele-Matrix MAX

Übergeordnete Studienziele zusätzlich zu allgemeinen Zielen im BSc Informatik	Befähigung im Sinne von Lernergebnissen (learning outcomes) Kenntnisse (Wissen) Fertigkeiten Kompetenzen	Entsprechende Module / Modulziele
Konstruktion großer Softwaresysteme / Modellbildung	Kenntnis: Allgemeine Transitionssysteme und deren Äquivalenzen	PN, GES, MC
	Kenntnis: Automatenmodelle	P-RZS, FMES, MC
	Fertigkeit: Umgang mit formalen Beschreibungssprachen	Alle
	Kompetenz: Verständnis der Konzepte „Sicherheit“ und „Korrektheit“	Alle
Konstruktion und Beherrschung großer Softwaresysteme / Fallstudien, Verhaltenseigenschaften	„security“-Eigenschaften	KRY
	Kenntnis: „safety“-Eigenschaften / Logiken	ASV, FMES, MC
	Fertigkeit: Einsatz adäquater Methoden für konkrete Anwendungen	PN, ASV, FMES, MC
	Kompetenz: Fähigkeit zur Anwendung von erprobten Methoden	Alle
Beherrschung großer Softwaresysteme / Analysemethoden	Kenntnis: Methoden zur Programmverifikation und -analyse	ASV, PV
	Kenntnis: Algorithmen zum Model-Checking	ASV, FMES, MC
	Kenntnis: linear-algebraische Methoden	PN
	Kenntnis: Methoden zur Abstraktionsverfeinerung	ASV
	Kenntnis: Umgang mit Analyse-Werkzeugen	P-RZS, ASV, FMES
	Fertigkeit: Reduktion großer Zustandsräume	MC, ASV, FMES
	Kompetenz: Anwendung erprobter Analysemethoden	Alle

1.1.1.5 Beispielhafter Studienplan MAX

Erstes Jahr Semester 1	Algorithmen und Programmierung	Programmierkurs	Technische Grundlagen der Informatik	Diskrete Strukturen	Mathematik für Informatik I
Erstes Jahr Semester 2	Algorithmen und Datenstrukturen	Soft Skills	Technische Informatik	Theoretische Informatik I	Mathematik für Informatik II
Zweites Jahr Semester 3	Informationssysteme I	Software-Engineering	PB	Theoretische Informatik II	Mathematik speziell
Zweites Jahr Semester 4	Rechnernetze I	Software-Projekt	Praktikum Technische Informatik	Betriebssysteme I	PN

Drittes Jahr Semester 5	Informatik und Gesellschaft		KRY	GES	PB
Drittes Jahr Semester 6	Bachelorabschlussmodul mit Bezug zu MAX		PB: Seminar	ASV	FMES

Abbildung 3: Studienplan MAX

1.1.2 Vertiefungsrichtung Modellierung und Analyse komplexer Systeme/Master

1.1.2.1 Ziele

Die Konstruktion und Beherrschung großer Software-Systeme ist nur durch ein systematisches Vorgehen möglich, das eine geeignete Modellbildung der Systeme sowie die Darlegung und Analyse ihrer Verhaltenseigenschaften umfasst. Dazu werden in dieser Vertiefungsrichtung

- fundierte Methoden zur Modellierung von Softwaresystemen eingeführt;
- Fallstudien zur Untersuchung von Verhaltenseigenschaften besprochen;
- erprobte Werkzeuge und effiziente Analyseverfahren vorgestellt und weiterentwickelt.

Angesprochen werden

- mathematisch orientierte Studierende, die Zusammenhänge herstellen und Vermutungen beweisen wollen;
- algorithmisch orientierte Studierende, die Konzepte in Werkzeuge zur Modellierung und Analyse umsetzen möchten.

Absolventen der Vertiefungsrichtung MAX sind durch das intensive Training im Erstellen und Analysieren formaler Systeme hervorragend für anspruchsvolle Tätigkeiten und Leitungsfunktionen in der universitären und industriellen Forschung sowie in der Entwicklung komplexer Systeme qualifiziert.

1.1.2.2 Zu erwerbende Kenntnisse, Fertigkeiten und Kompetenzen

Kenntnisse:

- Kenntnisse von formalen Systemmodellen (Transitionssysteme, Petrinetze, Graphersetzungssysteme, Büchi-Automaten, Realzeitautomaten, hybride Automaten) und zugehörigen Äquivalenzbegriffen;
- Kenntnisse von Systemeigenschaften: verschiedene Konzepte der Sicherheit (mit den Aspekten „correctness“, „safety“ und „security“) sowie deren Formalisierung in Logiken wie Prädikatenlogik, temporaler Logik und monadischer Logik;
- Kenntnisse von Analysemethoden: Beweissysteme zur Programmverifikation, Methoden zur Programmanalyse, Algorithmen zum Model-Checking, linear-algebraische Methoden zur Korrektheitsanalyse, Methoden zur Abstraktionsverfeinerung; Umgang mit computerunterstützten Analysewerkzeugen des State-of-the-Art.

Fertigkeiten:

- Umgang mit formalen Beschreibungssprachen;
- Auswahl und Einsatz adäquater Modellierungs- und Analysemethoden für konkrete Anwendungsszenarien;
- Methoden zur Reduktion großer Zustandsräume und Abstraktionen zur Behandlung zustandsunendlicher Systeme.

Kompetenzen:

- Tiefgehendes Verständnis der Konzepte „correctness“, „safety“ und „security“;

- Fähigkeit zur systematischen Anwendung von effektiven Methoden zum Systementwurf und zur Systemanalyse;
- Fähigkeit zur Weiterentwicklung bestehender Modellierungs- und Analysemethoden und deren Werkzeugunterstützung.

1.1.2.3 Studienanforderungen

Es sind weitere Module bis zur Summe von fünf Modulen aus der nachstehenden Liste reiner Mastermodule zu wählen

- Realzeitsysteme (RZS)
- Hybride Systeme (HS)
- Automaten und Logik (AL)
- Kryptographische Protokolle (KRYP)
- Kommunizierende und mobile Systeme (KMS)
- Kombination von Spezifikationstechniken (KSPEZ)
- Korrektheit von Graphprogrammen (KGP).
- Komplexitätstheorie

Drittens erstellen die Studierenden eine Master-Abschlussarbeit mit Bezug zu MAX.

1.1.2.4 Ziele-Matrix

Tabelle 15: Ziele-Matrix Max

Übergeordnete Studienziele zusätzlich zu allgemeinen Zielen im MSc-Informatik	Befähigung im Sinne von Lernergebnissen (learning outcomes) Kenntnisse (Wissen) Fertigkeiten Kompetenzen	Entsprechende Module / Modulziele
Konstruktion großer Softwaresysteme / Modellbildung	Kenntnis: Allgemeine Transitionssysteme und deren Äquivalenzen	PN, GES, MC, AL, KSPEZ, KMS
	Kenntnis: Automatenmodelle	P-RZS, FMES, MC, RZS, HS
	Fertigkeit: Umgang mit formalen Beschreibungssprachen	Alle
	Kompetenz: Verständnis der Konzepte „Sicherheit“ und „Korrektheit“	Alle
Konstruktion und Beherrschung großer Softwaresysteme / Fallstudien, Verhaltenseigenschaften	„security“-Eigenschaften	KRY, KRYP
	Kenntnis: „safety“-Eigenschaften / Logiken	ASV, FMES, MC, AL, KGP
	Fertigkeit: Auswahl adäquater Methoden für konkrete Anwendungen	PN, ASV, FMES, MC
	Kompetenz: Fähigkeit zur systematischen Anwendung von Methoden	Alle
	Beherrschung großer Softwaresysteme / Analysemethoden	Kenntnis: Methoden zur Programmverifikation und -analyse
	Kenntnis: Algorithmen zum Model-Checking	ASV, FMES, MC

	Kenntnis: linear-algebraische Methoden	PN
	Kenntnis: Methoden zur Abstraktionsverfeinerung	ASV, HS
	Kenntnis: Umgang mit Analyse-Werkzeugen	P-RZS, ASV, FMES
	Fertigkeit: Reduktion großer Zustandsräume und Behandlung zustandsunendlicher Systeme	MC, ASV, FMES, RZS, HS
	Kompetenz: Weiterentwicklung bestehender Methoden	Alle

1.1.2.5 Beispielhafter Studienplan

Erstes Jahr Semester 1	Angleichungs-Modul 1	Angleichungs-Modul 2	Angleichungs-Modul 3	Angleichungs-Modul 4	Akzent-Wahl 1: Programmverifikation
Erstes Jahr Semester 2	Projektgruppe		Akzent-Wahl 2: Kryptographische Protokolle	Akzent-Wahl 3: Model-Checking	Akzent-Wahl 4: Formale Methoden eingebetteter Systeme
Zweites Jahr Semester 3			Akzent-Wahl 5: Korrektheit von Graphprogrammen	Nicht-Informatik 1	Nicht-Informatik 2
Zweites Jahr Semester 4	Masterarbeit, einschließlich begleitendem Seminar und Abschlusspräsentation				

Abbildung 9: Studienplan MAX