

03-IMVP-PROSY	Programmsynthese
	<i>Program Synthesis</i>

Lehrform (*teaching format*) / **SWS** (*hours per week*): 2VL + 2UE

Kreditpunkte (*credit points*): 6

Turnus (*frequency*): unregelmäßig (*infrequently*)

Inhaltliche Voraussetzungen (*content-related prior knowledge/skills*): recommended: aufgefrischte Grundkenntnisse in Automaten and Formale Sprachen, useful: Embedded Software Engineering

Sprache (*language*): Deutsch / Englisch

Lehrende (*teaching staff*): Dr. Mario Gleirscher

Studiengang (<i>degree program</i>)	Module	Semester
Informatik (Master)	IMVP, IMVP-SQ, IMVP-AI	ab 1. Sem.
Systems Engineering I/II (Master)	M07-VT-ESS	ab 1. Sem.
Informatik (Bachelor)	(nur <i>Freie Wahl</i>)	ab 4. Sem.

Lernergebnisse:

- Erlernen des Paradigmas der algorithmischen Konstruktion korrekter Software.
- Stärkung der Fähigkeit zur Modellierung und Spezifikation unter Nutzung entsprechender von KI-Techniken.
- Entwicklung eines Grundverständnisses moderner Programmsynthesetechniken sowie deren Algorithmen und Datenstrukturen.
- Erlernen von Ansätzen der Synthese garantiert korrekter Programme.
- Anwenden aktueller Synthesewerkzeuge (z.B. Prism, Scots, Github Copilot).
- Sammlung von Anwendungserfahrung anhand praktischer Beispiele, vorzugsweise im Bereich intelligenter, cyber-physikalischer Systeme und eingebetteter Regelungssoftware.

Learning Outcome:

- Learn to employ the paradigm of the algorithmic construction of correct software.
- Strengthen your ability to model and specify using appropriate AI techniques.
- Firmly understand important program synthesis techniques as well as their algorithms and data structures.
- Understand the correct-by-construction principle, that is, how program correctness can be guaranteed by synthesis algorithms.
- Learn how to use state-of-the-art synthesis tools (z.B. Prism, Github Copilot).
- Learn how to apply the discussed techniques and tools to practical examples from different domains; with a focus on intelligent, cyber-physical systems and embedded control software.

Inhalte:

- Ausgewählte Techniken und Algorithmen der automatischen Generierung von Computerprogrammen aus Spezifikationen (z.B. Schnittstellenanforderungen, Entwurfsvorlagen, Nutzerinteraktion, Beobachtungen existierender Systeme).
- Diskutiert werden bspw. reaktive Synthese (d.h. Herleiten eines Programs in Form eines endlichen

Automaten aus einer temporallogischen Schnittstellenspezifikation), suche-basierte Programmierung (d.h. die Auswahl eines optimalen Programmes aus einem Entwurfsraum) und interaktive Synthese (z.B. das halbautomatische Konstruieren von Programmen insb. mit Hilfe sogenannter “code recommender”).

- Einsatz von Techniken der künstlichen Intelligenz (insb. LLMs wie GitHub Copilot, ChatGPT) in der interaktiven Synthese.
- Diskussion von Korrektheitsgarantien in der Programmsynthese.

Contents:

- Selected state-of-the-art techniques and algorithms for automatically deriving computer programs from specifications (e.g. requirements, designs, user interaction, or existing systems).
 - We discuss, in particular, reactive synthesis (i.e. deriving finite automata from logical property specifications), search-based programming (i.e. selecting optimal solutions from program design spaces), and interactive synthesis (i.e. the semi-automated program design using modern code recommenders).
 - Recent approaches using artificial intelligence techniques (particularly, LLMs such as GitHub Copilot and ChatGPT) to aid in the interactive synthesis.
 - Correct-by-construction techniques to provide guarantees for synthesised programs.
-

Hinweise (remarks): The table lists only the primary / most specific modules to which this course is assigned.