
Modulbezeichnung: **Praktische Semantik von Programmiersprachen (SemProg)** **7.5 ECTS**
(Practical Semantics of Programming Languages)

Modulverantwortliche/r: Tadeusz Litak
Lehrende: Tadeusz Litak

Startsemester: SS 2016	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 56 Std.	Eigenstudium: 169 Std.	Sprache: Deutsch und Englisch

Lehrveranstaltungen:

The dates are temporary. We are almost certainly **not** going to use Di 8:15 slot. The first meeting will take place 12:15 Wed, April 13. We agree on the dates then.

Praktische Semantik von Programmiersprachen (SS 2016, Vorlesung mit Übung, 4 SWS, Tadeusz Litak)

Inhalt:

We study the foundations of the imperative and functional languages, including semantics and type systems. The special feature of this course is that theory is done in a very practical and hands-on way: we not just prove, but program all the results from first-principles. The basic tool used in the course is Coq proof assistant, which can be regarded as a functional programming language in its own right. It has been used, for example, to verify correctness of Java Card technology, C compilers or, more recently, fragments of x86 architecture.

Our course is based on the online "Software Foundations" book: a successful model developed first at the Penn State University by ACM fellow Benjamin Pierce and tried since at numerous leading universities worldwide. At FAU, it was first run in SS 2013 with very satisfying effects.

Lernziele und Kompetenzen:

Fachkompetenz

Wissen

The students explain the basics of both programming semantics and proof assistants, in particular Coq.

Verstehen

The students prove theorems using a proof assistant.

Anwenden

The students transfer proofs into programs and programs into proofs.

Analysieren

The students examine behaviour of simple programs using formal semantics

Evaluiieren (Beurteilen)

The students evaluate the role played by logic and type theory in scientific approach to programming.

Erschaffen

The students provide formal semantics to a simple programming language.

Literatur:

Main reference: online book "Software Foundations" <http://www.cis.upenn.edu/~bcpierce/sf/>

Supplementary reading on the theory of programming: Types and Programming Languages Benjamin C. Pierce, The MIT Press

Supplementary reading on Coq: Interactive Theorem Proving and Program Development Coq'Art: The Calculus of Inductive Constructions Series: Texts in Theoretical Computer Science. An EATCS Series Bertot, Yves, Casteran, Pierre

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Informatik (Bachelor of Arts (2 Fächer))

(Po-Vers. 2010 | TechFak | Informatik (Bachelor of Arts (2 Fächer)) | Vertiefung Informatik I und II | Vertiefungsmodul Theoretische Informatik)

[2] Informatik (Bachelor of Arts (2 Fächer))

(Po-Vers. 2013 | TechFak | Informatik (Bachelor of Arts (2 Fächer)) | Vertiefung Informatik I und II | Vertiefungsmodul Theoretische Informatik)

[3] Informatik (Bachelor of Science)

(Po-Vers. 2009s | TechFak | Informatik (Bachelor of Science) | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsmodul Theoretische Informatik)

[4] Informatik (Bachelor of Science)

(Po-Vers. 2009w | TechFak | Informatik (Bachelor of Science) | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsmodul Theoretische Informatik)

[5] Informatik (Master of Science)

(Po-Vers. 2010 | TechFak | Informatik (Master of Science) | Wahlpflichtbereich | Säule der theoretisch orientierten Vertiefungsrichtungen | Vertiefungsmodul Theoretische Informatik)

[6] Mathematik (Bachelor of Science)

(Po-Vers. 2015w | NatFak | Mathematik (Bachelor of Science) | Module des Nebenfachs | Nebenfach Informatik | Vertiefungsmodule | Vertiefungsmodul Theoretische Informatik)

Studien-/Prüfungsleistungen:

Mündliche Prüfung zu Praktische Semantik von Programmiersprachen (Prüfungsnummer: 599478)

Prüfungsleistung, mehrteilige Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

In die Prüfungsnote gehen zu 50% die Leistungen aus der Bearbeitung von Übungsaufgaben ein.

Erstablegung: SS 2016, 1. Wdh.: WS 2016/2017 (nur für Wiederholer)

1. Prüfer: Lutz Schröder

1. Prüfer: Tadeusz Litak
