inf450 - Correctness of Graph Programs

Module label: Correctness of Graph Programs
Module code: inf450
Credit points: 6.0 KP
Workload: 180 h

Used in course of study:
- Master's Programme Computing Science > Theoretische Informatik
- Master's Programme Embedded Systems and Microrobotics > Akzentsetzungsmodulle

Contact person:
Module responsibility:
- Annegret Habel
- Die im Modul Lehrenden

Authorized examiners:
- Andreas Hein
- Die im Modul Lehrenden

Entry requirements:
The objectives of this module are modelling of systems, system changes and system properties. Introduction to graph programs. Introduction into system correctness. Methods for proving system correctness.

Skills to be acquired in this module:
Professional competence:
The students:
- Describe the basics of graph programs and graph properties
- Describe verification procedures of system correctness

Methodological competence:
The students:
- Model systems, system changes and system properties
- Apply the formalism of graph programs

Social competence:
The students:
- Solve problems in a team
- Present and discuss their proposed solutions

Self-competence:
The students:
- Reflect upon their actions with regard to term rewriting systems and the methods of those

Module contents:
The module is an introduction to the modelling of systems, system changes and system properties by means of graphs, graph programs and graph conditions and presents a method for proving correctness of systems with respect to a pre- and a postcondition.

The basic structures used in this lecture are graphs; they are used in practically all domains of computing science for the representation of complex structures. Graph programs are constructed from the core constructs of nondeterministic rule application, sequential composition and iteration and they can effect programmatic changes of a graph structure. One well-known method for determining the correctness of programs with respect to a pre- and a postcondition is based on the construction of a weakest precondition of the postcondition with respect to the program and the attempt to decide whether the given precondition implies the computed weakest precondition.

Reader's advisory:
- K. Azab, A. Habel, K.-H. Pennemann, C. Zuckschwerdt. ENFORCe: A system for ensuring
formal correctness of high-level programs. In Electronic Communications of the EASST, Vol. 1. 82-93, 2007.

<table>
<thead>
<tr>
<th>Links</th>
<th>Language of instruction</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (semesters)</td>
<td></td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td></td>
<td>im 2-Jahres-Zyklus</td>
</tr>
<tr>
<td>Module capacity</td>
<td></td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td></td>
<td>AC (Aufbaucurriculum / Composition)</td>
</tr>
<tr>
<td>Modulart</td>
<td></td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-Lehrform / Type of program</td>
<td></td>
<td>- inf400 Theoretische Informatik I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- inf401 Theoretische Informatik II</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>Will be announced during the course</td>
<td>presentation or oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td></td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1.00</td>
<td></td>
<td>14 h</td>
</tr>
</tbody>
</table>

Total time of attendance for the module 56 h