Formal Methods
Course description sheet

Basic information

Field of study
Computer Science

Major
Systems Modelling and Intelligent Data Analysis

Organisational unit
Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering

Study level
Second-cycle (engineer) programme

Form of study
Full-time studies

Profile
General academic

Didactic cycle
2019/2020

Course code
EAIIIBINFMSS.III1O.c16b27e40c89f8a46358b7431d6e22f d.19

Lecture languages
english

Mandatory
Obligatory

Course related to scientific research
Yes

Course coordinator
Tomasz Szmuc

Lecturer
Tomasz Szmuc, Wojciech Szmuc

Period
Semester 1

Method of verification of the learning outcomes
Exam

Activities and hours
Lectures: 28
Laboratory classes: 28

Number of ECTS credits
4

Goals

C1 The goal of the subject is to provide knowledge and to train skills in using formal methods for modelling and verification of hybrid systems and software. Petri nets, process algebras are used for modelling and temporal logics for verification - proving of the correctness.
## Course's learning outcomes

<table>
<thead>
<tr>
<th>Code</th>
<th>Outcomes in terms of Learning outcomes prescribed to a field of study</th>
<th>Methods of verification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Knowledge - Student knows and understands:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W1</td>
<td>Student has a deepened knowledge of the popular formal modelling languages.</td>
<td>INF2A_W01, INF2A_W04, INF2A_W07</td>
</tr>
<tr>
<td>W2</td>
<td>Student has a basic knowledge of formal verification of IT systems.</td>
<td>INF2A_W01, INF2A_W04, INF2A_W07</td>
</tr>
<tr>
<td>W3</td>
<td>Student has a basic knowledge of software tools for modelling and verification of IT systems with formal methods.</td>
<td>INF2A_W01, INF2A_W04, INF2A_W07</td>
</tr>
<tr>
<td><strong>Skills - Student can:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U1</td>
<td>Student is able to use formal methods the the analysis and design of IT systems.</td>
<td>INF2A_U01, INF2A_U05</td>
</tr>
<tr>
<td>U2</td>
<td>Student is able to use software tools supporting the formal modelling and verification</td>
<td>INF2A_U01, INF2A_U05</td>
</tr>
<tr>
<td><strong>Social competences - Student is ready to:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td>Student understands understands the need for providing the public with information and opinions on the role of formal methods and their impact on the software quality.</td>
<td>INF2A_K02</td>
</tr>
</tbody>
</table>

### Program content ensuring the achievement of the learning outcomes prescribed to the module

The aim the subject to provide knowledge and train skills in using formal methods for modelling and verification of systems/software. Petri nets, process algebras and temporal logics are used.

### Student workload

<table>
<thead>
<tr>
<th>Activity form</th>
<th>Average amount of hours* needed to complete each activity form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation for classes</td>
<td>24</td>
</tr>
<tr>
<td>Lectures</td>
<td>28</td>
</tr>
<tr>
<td>Laboratory classes</td>
<td>28</td>
</tr>
<tr>
<td>Realization of independently performed tasks</td>
<td>20</td>
</tr>
<tr>
<td>Examination or final test/colloquium</td>
<td>2</td>
</tr>
<tr>
<td>Preparation of project, presentation, essay, report</td>
<td>10</td>
</tr>
</tbody>
</table>

**Student workload**

<table>
<thead>
<tr>
<th></th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>112</td>
</tr>
</tbody>
</table>
Workload involving teacher

Hours
56

* hour means 45 minutes

Program content

<table>
<thead>
<tr>
<th>No.</th>
<th>Program content</th>
<th>Course's learning outcomes</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 1.  | 1. Formal methods in computer science: formal modelling, types of formal languages, application of formal methods.  
     2. Formal methods in software engineering: the role of formal methods, verification, validation, norms.  
     4. Process algebras: CCS (Calculus of Communicating Systems), equivalence relations and their applications, LOTOS language, modelling with LOTOS.  
     5. Modelling time properties in process algebras.  
| 2.  | Practical classes using software modelling tool for formal modelling and verification.  
     1. Place-Transition Petri nets - TINA  
     2. High-Level Coloured Petri nets - CPN Tools  
     3. Process algebras - CADP software. | U1, U2, K1 | Laboratory classes |

Extended information/Additional elements

Teaching methods and techniques:
Lectures, Case study, Design thinking, Problem based learning, Group work

<table>
<thead>
<tr>
<th>Activities</th>
<th>Methods of verification</th>
<th>Credit conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lectures</td>
<td>Test, Examination</td>
<td>Final positive grade of the laboratory classes and positive grade received from exam are required to complete the subject.</td>
</tr>
<tr>
<td>Lab. classes</td>
<td>Test, Examination</td>
<td>Positive grades of all laboratory tasks.</td>
</tr>
</tbody>
</table>

Conditions and the manner of completing each form of classes, including the rules of making retakes, as well as the conditions for admission to the exam

Presence on laboratory classes is obligatory according to the general study regulation. If the requirement is not met the chief lecturer may specify additional conditions for the positive final grade or issue the negative grade (negative
assessment). The student can take the exam only after obtaining a positive assessment of laboratory classes.

**Method of determining the final grade**

Course grade: The course grade is the average of laboratory and exam grades. In case of a correction exam, grades of all terms are taken into consideration. Laboratory classes: The final grade will be awarded based on the result of two tests. The credit will be based on a verification of both the theoretic knowledge and abilities, and the more practical skills of students. The activity during laboratory classes will be also taken into consideration as an additional evaluation criterion.

**Manner and mode of making up for the backlog caused by a student justified absence from classes**

All absences without leave may be made up by an individual agreement with the class instructor. The preferable form is a discussion on the topic, which should be made up by the student.

**Prerequisites and additional requirements**

Basic knowledge of:

1. Discrete mathematics, in particular relations and graphs.
2. Finite automata theory.
3. Propositional logic, first-order logic.

**Rules of participation in given classes, indicating whether student presence at the lecture is obligatory**

Lectures:

1. Attendance: Not obligatory.
2. Students are encouraged for active participation, asking questions and discussion. Video recording requires permission from the lecturer.

Laboratory classes:

1. Attendance: obligatory.
2. Students perform tasks according to guidance and tutorials delivered by teaching assistant. Interim verifications of knowledge by colloquiums are applied.

**Literature**

**Obligatory**


**Optional**


**Scientific research and publications**

**Research**

1. Rigorous Development of Cyber-Physical Systems (CPS) supported by mathematical modelling and formal verification.

**Publications**


<table>
<thead>
<tr>
<th>Code</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>INF2A_K02</td>
<td>ma świadomość roli społecznej absolwenta uczelni technicznej, rozumie potrzebę formułowania i przekazywania społeczeństwu informacji i opinii dotyczących osiągnięć informatyki, wagi profesjonalnego zachowania i przestrzegania zasad etyki zawodowej, prawidłowo identyfikuje i rozstrzyga dylematy związane z wykonywaniem zawodu</td>
</tr>
<tr>
<td>INF2A_U01</td>
<td>potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł, integrować uzyskane informacje, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, a także określić kierunki dalszego uczenia się i realizować proces samokształcenia</td>
</tr>
<tr>
<td>INF2A_U05</td>
<td>potrafi wykorzystać poznane metody i modele do tworzenia różnego rodzaju programów o charakterze użytkowym i naukowym, z uwzględnieniem specyfiki specjalności</td>
</tr>
<tr>
<td>INF2A_W01</td>
<td>ma pogłębioną wiedzę w zakresie przedmiotów ścisłych, pozwalającą na formułowanie i rozwiązywanie złożonych zadań z zakresu informatyki</td>
</tr>
<tr>
<td>INF2A_W04</td>
<td>ma podbudowaną teoretyczną wiedzę w zakresie inżynierii oprogramowania z uwzględnieniem specyfiki specjalności, w szczególności w zakresie budowy narzędzi i systemów informatycznych, etapów i metod projektowania, rozwoju i analizy oprogramowania, oraz stosowanych modeli procesu wytwarzania oprogramowania z zakresu specjalności</td>
</tr>
<tr>
<td>INF2A_W07</td>
<td>orientuje się w obecnym stanie oraz najnowszych osiągnięciach i trendach rozwojowych informatyki i dziedzin pokrewnych oraz ma wiedzę niezbędną do rozumienia pozatechnicznych uwarunkowań działalności inżynierskiej</td>
</tr>
</tbody>
</table>