

# Software Verification and Validation: Course overview

Istvan Majzik  
majzik@mit.bme.hu

**Budapest University of Technology and Economics**  
**Dept. of Measurement and Information Systems**

# Administrative data

- Class schedule
  - Wednesday: 14:15-16:00, room I.L.405.
  - Thursday: 16:15-18:00, room I.L.405.
- Official holidays (no classes)
  - Sept. 20., Sept. 28., Nov. 1., Nov. 16.
- Web page of the course
  - <https://inf.mit.bme.hu/en/edu/courses/swvv>
  - Uploaded as the course advances:
    - Course material (slides, background materials)
    - News and announcements
    - Previous homework presentations
    - Exam topics

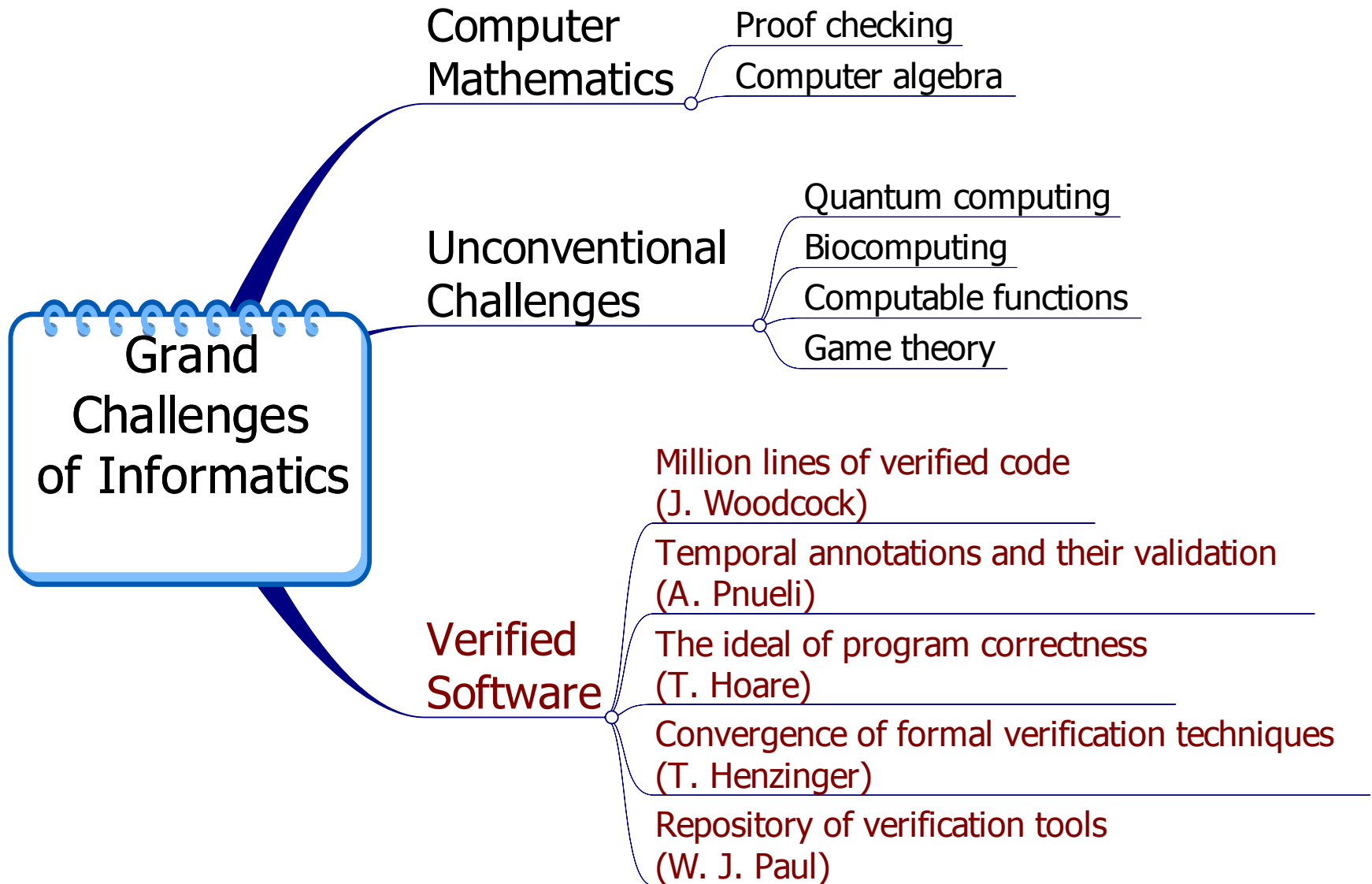
# Assessment

- During the semester: Three options for homework assignment
  1. Oral presentation of a V&V case study based on own work
  2. Oral presentation of a V&V case study based on selected literature (journal or conference paper presenting a case study)
  3. Submission of an abstract (4-5 pages) that summarizes a case study based on selected literature
- In the examination period: Oral exam based on a list of predefined exam topics
  - Topics are given in pairs (known in advance)
  - In case of having oral presentation (option 1 or 2) during the semester, only one of the topics shall be answered
- To do:
  - Discuss the topic of the presentation / abstract with the instructor (in email: majzikmit.bme.hu)
  - Deadline: October 31, 2017
  - Presentations will be scheduled in the last weeks of the semester

# Motivations and goals of the course

- Increasing costs of **design faults** in software; this way it is important to
  - **verify** that the design and implementation are correct w.r.t. the specification
  - perform verification **in each phase** of the development (not only testing the implementation)
  - **validate** the product or prototype w.r.t. user requirements and expectations
- Overview of verification and validation methods and techniques
  - Classic methods: review, testing, ...
  - Formal methods: mathematically precise techniques based on design models, to prove design correctness

# SW verification – Grand Challenges of Informatics



# Knowledge to be obtained

- Participants of the course will be able to
  - Understand the **role, advantages and limitations** of various verification and validation (V&V) techniques
    - **Select proper techniques** for each development phase
  - Plan and integrate **V&V processes** that support development processes
  - Understand the **mathematical background** of emerging techniques (e.g., formal verification, static source code analysis, model based test case generation)
  - Know **typical tools** that support V&V techniques
- Background included from courses:
  - Software and System Verification (testing techniques)
  - Formal Methods (temporal logic, model checking)

# Synopsis

- Introduction
- Verification in the **requirement specification phase**
- **Architecture** verification and evaluation
- Verification of the **detailed design**
  - Classic techniques
  - Formal methods: model checking, equivalence checking
  - Advanced methods: formal verification of extra-functional properties and timed behavior, handling complex designs (large state spaces)
- Verification of the **source code**
  - Code review, abstract interpretation, symbolic execution
  - Classic techniques of proving program correctness
- Testing and **test design**
  - Test case generation at unit level
  - Integration and system testing
  - Model based testing and test case generation
- **Validation** and assessment
- V&V in the **maintenance** phases
- Integrated approaches (esp. formal frameworks)