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

- Lectures in 2020: “distant education”
  - Wednesday: 14:15-16:00, MS Teams
  - Thursday: 14:15-16:00, MS Teams
- Official holidays
  - September 23 (BME Sport Day)
  - November 12 (TDK Day)
- 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
    - Homework presentations
    - Exam topics

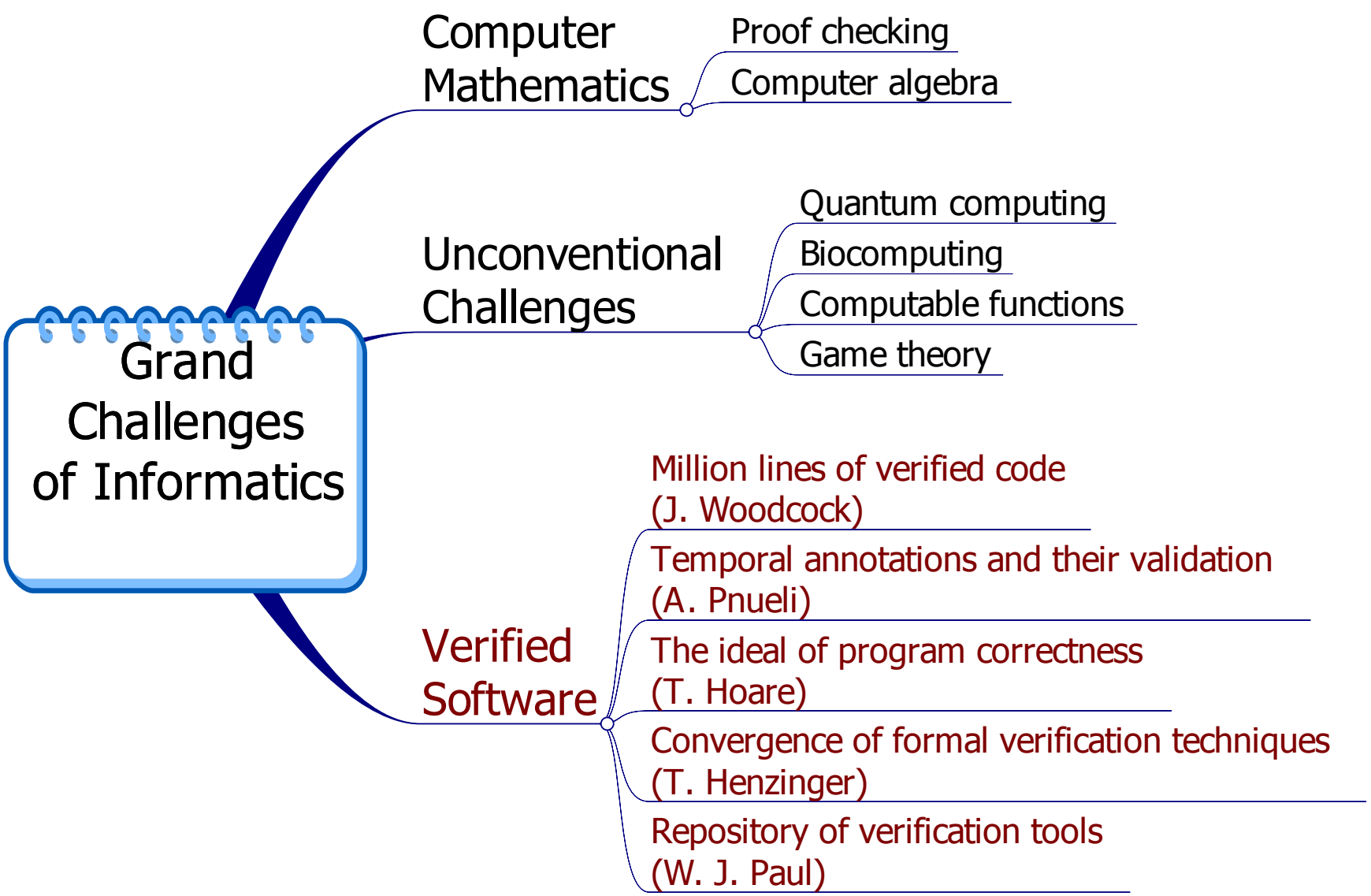
Assessment and requirements

- During the semester: Three options
  1. Oral presentation of a V&V case study based on your own work
  2. Oral presentation of a V&V case study based on selected literature (journal or conference paper presenting a case study or a tool)
  3. Submission of a short essay (4-5 pages) that summarizes a case study or capabilities of a tool based on selected literature
- In the examination period: Oral exam based on a list of topics
  - Topics are given in pairs (known in advance)
  - Exam is based on a pair of topics that is assigned randomly
  - In case of having oral presentation (option 1 or 2) during the semester, you are free to select for the exam one topic from the pair
- To do:
  - Negotiate the theme of your oral presentation / essay with the instructor in email <majzik@mit.bme.hu> by October 31, 2020
  - Presentations: in the last 2 weeks of the semester (to be negotiated)

Motivations and goals of the course

- Motivation: 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
- Goal: Systematic overview and assessment 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 previous 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)