Presentations

Advanced Computing Concepts for Engineering. MUN. Theodore Norvell

February 27, 2020

1 Presentations

1.1 Presentation requirements

Presentations will be 12 minutes + 2 minutes for questions and answers.

You may pick any topic approved by the instructor. Below I list some suitable topics. You do not have to pick a topic on this list; it is only a list of suggestions. Only one group per topic. Groups will consist of one or two students.

You should pick a topic by March 2. Send D2L (Brightspace) mail with your topic. Each group will have a different topic. Chosen topics will be posted on the course web page, but if you send me email with your choice, please include a second choice, in case your first choice is taken.

Presentations will be held between March 20 and April 9.

Written requirements. Due April 9. Please submit on paper.

- A copy of your slides. The usual standards of academic integrity and citation apply to your slide set. Each slide set should include a bibliography (in IEEE format). Slides should include citations.
- In addition to the presentation, you will be required to hand in a discussion of the literature that you used to prepare your presentation. This should include a one to three paragraph summary of each source and a bibliography in IEEE format.

1.2 Suggested topics

You don't need to pick one of the topics below. They are suggesions. You are welcome to pick any approved topic that is broadly within the themes of the course: Theories of programming. Models of computing. Computational Complexity.

- Program correctness and specification
  - Monitors and the correctness of parallel programs
  - Process algebra's: CSP and/or CCS and/or the pi-calculus.

  - Real-time systems specification and verification.
  - Hybrid systems specification and verification.
  - Bird and de Moor's relational calculus.
  - Hoare logic.
  - Owiki and Gries's method verifying concurrent programs

- Engineering applications of correctness
  - Verification of procedures with Dafny
  - Verification of classes with Dafny
  - Verification of concurrent code with Chalice
  - Lamport's TLA+
  - Separation Logic (nonconcurrent)
  - Concurrent Separation Logic
  - Symbolic model checking of hardware designs
  - The Alloy model checker
  - The Z software specification notation
  - The B software specification method
  - Morgan's refinement calculus
  - The Clean-Room method.
  - The Java Modeling Language (JML)
  - The Object Constraint Language (OCL)
  - Proof carrying code
  - Test oracle generation from specifications.
  - Other formal methods of hardware or software correctness.
  - Etherium

- Complexity and efficiency
  - Parallel complexity: PRAM models and algorithms.
  - Parallel complexity: Parallel sorting with sorting networks.
  - Public key cryptography.
  - NP-hard and NP-easy problems in the area of <fill in the blank>

- Models of computation
  - Timed automata
  - Hybrid automata
  - Quantum computing
  - Genetic Algorithms
  - Neural networks
  - Hierarchies of unsolvability