Seminar on Concurrency Theory

Ori Lahav

February 27, 2019
Today

• What is this seminar about?

• Goals, requirements and logistics of the seminar

• Brief introduction to “weak memory concurrency”
Concurrency Theory

• Rigorous mathematical formalisms and techniques for **modelling** and **analysing** concurrent systems.

• Concurrent systems:
  • Concurrent programs
  • Reactive systems
Concurrent Programming

- Shared memory
- Message passing

- C / C++
- Erlang
Reactive systems

The classical view

• A program transforms an input into an output.

• Denotational semantics: the meaning of a program is a partial function:

  \[ \text{States} \rightarrow \text{States} \]

• Non-termination is bad.

• \textit{Is that what we need?}
Reactive systems

• What about: operating systems? communication protocol? control programs? database management systems? vending machines?

Reactive systems continuously reacts to the environment and influence the environment

• Key issue: communication and interaction.

• Non-determinism is often inevitable.

• Related and similar to parallelism.
Parallelism is here

Problem

- Concurrency is widespread, but it is also **error prone**.

- Unlike sequential programs, programmers need to take care of **synchronization**, **race conditions**, **deadlocks**, etc.

- Therac-25: Concurrent programming errors (in particular, race conditions) → accidents causing death and serious injury

- Mars Rover: Problems with interaction between concurrent tasks caused periodic software resets reducing availability for exploration
Verification

\[ \text{system} \not\models \text{specification} \]

Testing

- Hard to apply for concurrent systems
- Randomization

Formal verification

- Even short concurrent programs are hard to analyse
- Reasoning principles
- Automatic decision procedures
- Compositionality
Practice

• **Modelling** and simulating concurrent systems

• Concurrent **programming**: programming languages have different concurrency models in (e.g., C, Go, Rust, Erlang)

• **Verifying** correctness to avoid expensive bugs
This seminar
Goals

- Introduction different fundamental topics in concurrency (*basis for advanced studies*)
- Independent understanding of a scientific topic
- Understanding scientific literature
- Technical presentation skills
Requirements

• Attend all meetings.

• Present one subject in a **70-90 minute talk**, based on a research paper or a chapter from a book.

• Prepare slides and/or handouts (pdf, in English), and send them to me **a week before the lecture**.

• May work in pairs (present 2 subjects).

• Discuss presentations with me (a few days) before the lecture.

• May use other sources besides the one that will be suggested (with citations).

• **Grade**: meeting these requirements; understanding of the material; quality and clarity of presentation in class; quality of the slides/handouts.

• Optional: extra short lecture (e.g., introduce an alternative model of concurrency).
Your presentations

- This is an advanced seminar: the material is sometimes not easy and not self-contained.
- Identify and present the crux, rather than all details.
- Demonstrate with *clear and effective examples*.
- Be *precise*.
- May (and often should) *skip proof details*. 
Some tips

• Take your *time* to understand the material.

• Discuss the content with me and other students.

• Practice your talk.
Logistics

- Website:
  https://www.cs.tau.ac.il/~orilahav/seminar19/index.html

- By next week: subject assignments
  add your preferences in
  https://docs.google.com/document/d/1ZXvFFnu5ttNi0GwmYitdUoj3DOZSj88XEhbpt8IJLJk/edit

- Speaker for next week?
Topics

see website