Seminar
Concurrency Theory
Introduction

Joost-Pieter Katoen Thomas Noll

Software Modeling and Verification Group

RWTH AACHEN UNIVERSITY

noll@cs.rwth-aachen.de
http://www-i2.informatik.rwth-aachen.de/i2/ct14/

Summer Semester 2014; 20 February, 2014

Outline

Concurrency Theory

- Rigorous, mathematically based techniques for modelling and analysing concurrent systems
- Aim at improving correctness, reliability and robustness of such systems
- Important classification:

  interleaving: concurrency = non-deterministic merging of sequential executions (process algebras, ...)

  true concurrency: model parallel behaviour explicitly (Petri nets, ...)

Applications

- Concurrent programming
  - avoid errors such as deadlocks, memory inconsistencies due to violation of atomicity, ...
  - techniques: semaphores, locks, ...
  - bugs difficult to reproduce (non-deterministic behaviour, state-space explosion)
- Reactive systems
  - maintain ongoing interaction with environment
  - behaviour determined by concurrent execution, interaction, and mobility of non-terminating processes
  - examples: operating systems, control systems for production lines/power plants/vehicles, ...
  - often safety critical $\Longrightarrow$ require rigorous formal techniques for design/implementation/validation

Outline

Goals

Aims of this seminar

- Independent understanding of a scientific topic
- Acquiring, reading and understanding scientific literature
- Writing of your own report on this topic
- Oral presentation of your results

Requirements on Report

Your report

- Independent writing of a report of 15–20 pages
- Complete set of references to all consulted literature
- Correct citation of important literature
- Plagiarism: taking text blocks (from literature or web) without source indication causes immediate exclusion from this seminar
- Font size 12pt with "normal" page layout
- Language: German or English
- We expect the correct usage of spelling and grammar
  - $\geq$ 10 errors per page $\Longrightarrow$ abortion of correction

Requirements on Talk

Your talk

- Talk of about 45 minutes
- Focus your talk on the audience
- Descriptive slides:
  - $\leq$ 15 lines of text
  - use (base) colors in a useful manner
- Language: German or English
- No spelling mistakes please!
- Finish in time. Overtime is bad
- Ask for questions

Final Preparations

Preparation of your talk

- Setup laptop and projector ahead of time
- Use a (laser) pointer
- Number your slides
- Multiple copies: laptop, USB, web
- Have backup slides ready for expected questions

Outline

Important Dates

Talks

The seminar will be held as a weekly meeting on Tuesdays at 16:00 (?) starting end of April

- see `http://www-i2.informatik.rwth-aachen.de/i2/ct14/`

Important Dates

Talks

The seminar will be held as a weekly meeting on Tuesdays at 16:00 (?) starting end of April

- see `http://www-i2.informatik.rwth-aachen.de/i2/ct14/`

Deadlines

You are requested to adhere to the following firm deadlines:

- immediately: obtain the required literature from the web or library
- eight weeks before your talk: present a table of contents
- six weeks before your talk: preliminary version of your report
- four weeks before your talk: final version of your report
- two weeks before your talk: preliminary version of your slides
- one week before your talk: final version of your slides

Important Dates

Talks

The seminar will be held as a weekly meeting on Tuesdays at 16:00 (?) starting end of April

- see `http://www-i2.informatik.rwth-aachen.de/i2/ct14/`

Deadlines

You are requested to adhere to the following firm deadlines:

- immediately: obtain the required literature from the web or library
- eight weeks before your talk: present a table of contents
- six weeks before your talk: preliminary version of your report
- four weeks before your talk: final version of your report
- two weeks before your talk: preliminary version of your slides
- one week before your talk: final version of your slides

Missing a deadline causes immediate exclusion from the seminar

Outline

Selecting Your Topic

Procedure

- You obtain(ed) a list of topics of this seminar.
- Indicate the preference of your topics (first, second, third).
- We do our best to find an adequate topic-student distribution.
- Disclaimer: no guarantee for an optimal solution.
- Your topic will be published on our website by 25 February.
- Then also your supervisor will be indicated.
- Please give language preference
  - unsure $\Longrightarrow$ German

Selecting Your Topic

Procedure

- You obtain(ed) a list of topics of this seminar.
- Indicate the preference of your topics (first, second, third).
- We do our best to find an adequate topic-student distribution.
- Disclaimer: no guarantee for an optimal solution.
- Your topic will be published on our website by 25 February.
- Then also your supervisor will be indicated.
- Please give language preference
  - unsure $\Longrightarrow$ German

Withdrawal

- You have up to three weeks to refrain from participating in this seminar.
- Later cancellation (by you or by us) causes a not passed for this seminar and reduces your (three) possibilities by one.

Topic 1: Timed CCS

- W. Yi: CCS + Time = an Interleaving Model for Real Time Systems, ICALP 1991
- CCS = Calculus of Communicating Systems [R. Milner]
- Describes processes by basic actions, choice, parallel composition
- Semantics = labelled transition system
- Timed CCS additionally supports specification of time delays

$$Light = press.Bright + \varepsilon(1.5).press.Off$$

Topic 2: Synchronous and Asynchronous CCS

- R. Milner: Calculi for synchrony and asynchrony, TCS 1983
- Parallel composition in CCS is asynchronous:

$$\text{(Par1)}\frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \qquad \text{(Par2)}\frac{Q \xrightarrow{\alpha} Q'}{P \parallel Q \xrightarrow{\alpha} P \parallel Q'}$$

- Synchronous CCS (SCCS):

$$\text{(Par)}\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\beta} Q'}{P \parallel Q \xrightarrow{\alpha\beta} P' \parallel Q}$$

Topic 3: Decidable Subsets of CCS

- S. Christensen, Y. Hirshfeld, F. Moller: Decidable Subsets of CCS
- Equivalence of CCS processes based on bisimulation
- But: generally undecidable for CCS (as CCS is universal, i.e., can encode Turing machines)
- Goal: identify syntactic fragments of CCS for which bisimulation is decidable
- Result: disallowing either of communication or both restriction and relabelling sufficient

Topic 4: ACP Process Algebra

- J.A. Bergstra, J.W. Klop: Process algebra for synchronous communication, IaC 1984
- Alternative approach to CCS
- More algebraic in nature (equational specifications)
- Essential operators:
  - $+$ (choice)
  - $\cdot$ (sequential composition)
  - $\|$ (parallel composition)
- Different semantics of parallel product (interleaving, communication, mutual exclusion, synchronous cooperation)

Topic 5: The Linear Time-Branching Time Spectrum

- R. van Glabbeek: The linear time - branching time spectrum, CONCUR 1990
- Considers domain of finitely branching, sequential processes (LTSs)
- Presents eleven different semantics
- Motivated by testing scenarios with "button pushing experiments"

Topic 6: Axiomatization of Bisimilarity

- R. van Glabbeek: A complete axiomatization for branching bisimulation congruence of finite-state behaviours, MFCS 1993
- "Standard" decision algorithm for bisimulation: state space partition refinement
- Here: two processes bisimilar if equivalent w.r.t. a set of equations
- Presents sound and complete inference system for bisimulation on a sublanguage of CCS

Topic 7: Reachability in Petri Nets

- E.W. Mayr: An Algorithm for the General Petri Net Reachability Problem, SIAM J. Comp. 1984
- Presents algorithm for reachability of markings in Petri nets
- Based on construction of reachability tree
- Uses finite automata for describing (approximations of) firing sequences

Topic 8: Liveness and Safeness of Petri Nets

- E. Best, P.S. Tiagarajan: Some Classes of Live and Safe Petri Nets, Concurrency and Nets 1987
- Liveness: transitions are (sometimes/infinitely often/always/...) enabled
- Safeness: in every (reachable) marking, every place contains at most one token
- Paper studies a series of structural restrictions under which liveness/safeness is guaranteed

Topic 9: Undecidability of Bisimilarity for Petri Nets

- P. Jancar: Undecidability of bisimilarity for Petri nets and some related problems, TCS 1995
- Considers labelled Petri nets and their labelled transition systems (states = markings)
- Shows undecidability of corresponding bisimulation problem
- Argument based on undecidability of halting problem for 2-counter machines

Topic 10: Efficient Net Unfolding

- J. Esparza, S. Rmer, W. Vogler: An Improvement of McMillan's Unfolding Algorithm, TACAS 1996
- Original technique proposed by McMillan to avoid state-space explosion problem in analysis of finite-state Petri nets
- Requires to construct finite initial part ("prefix") of net unfolding
- Prefixes can become larger than actually necessary (exponentially larger in worst case)
- Paper proposes refinement of algorithm to avoid this problem

Topic 11: Applications of Net Unfolding

- J. Esparza, C. Schrter: Unfolding Based Algorithms for the Reachability Problem, Fund. Inf. 2001
- Studies four solutions to the reachability problem for safe Petri nets (3 known, 1 new)
- Gives recommendations when to use which algorithm
- K. Heljanko: Model Checking with Finite Complete Prefixes Is PSPACE-Complete, CONCUR 2000
- Shows that model checking a formula of several temporal logics (LTL, CTL, CTL*) is PSPACE-complete in size of finite complete prefix of a safe Petri net

Topic 12: Timed Nets

- W.M. Zuberek: Timed Petri nets definitions, properties, and applications, Microelectronics Reliability 1991
- Associates (deterministic or random) firing time with each transition
- Provides performance analysis of timed nets based on stationary probabilities of states
- Determined from a set of simultaneous linear equilibrium equations

Topic 13: Generalized Stochastic Petri Nets

- M.A. Marsan, G. Conte, G. Balbo: A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems, TOCS 1984
- Introduces two types of transitions
  - timed with an exponentially distributed delay
  - immediate, with constant zero delay
  - immediate have priority over timed
- Application: performance evaluation of multiprocessor systems

Topic 14: Mazurkiewicz Traces

- A. Mazurkiewicz: Introduction to Trace Theory, Book of Traces 1995
- Goal: describing concurrent behaviour of systems via sequential observations
- Based on (in-)dependency of actions
- Establishes relation to net theory

Topic 15: Event Structures

- G. Winskel: An introduction to event structures, Logics and Models for Concurrency 1988
- Models concurrent processes by events constrained by relations of consistency and enabling
- Establishes relation to Petri nets
- Provides semantics to parallel programming languages

Topic 16: Modeling Concurrency with Partial Orders

- V. Pratt: Modeling Concurrency with Partial Orders, IJPP 1986
- Combines formal languages (traces), partial orders, and temporal logic
- Yields partially ordered multisets (pomsets)
- Introduce operations on pomsets for specifying concurrent processes

Outline

Some Final Hints

Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

Some Final Hints

Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

We wish you success and look forward to an enjoyable and high-quality seminar!