Imperial College London

From validating quantitative models to generating valid ones

Michael Huth

ERCIM Working Group MLQA
9 July 2010
Edinburgh, Scotland

Acknowledgments

- Nir Piterman
- Daniel Wagner
- Huaxin Wang

Outline of talk

Motivation

Design Synthesis

Quantitative Synthesis

Conclusions

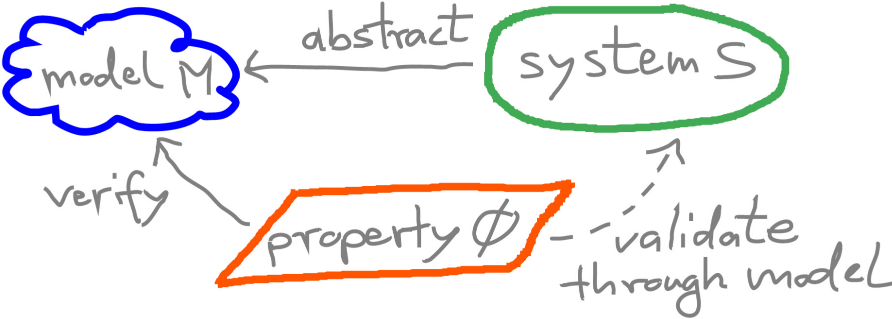
Motivation

Many kinds of quantitative models

- queueing networks
- timed trace sets
- randomized algorithms and protocols
- stochastic Petri nets
- stochastic games
- etc.

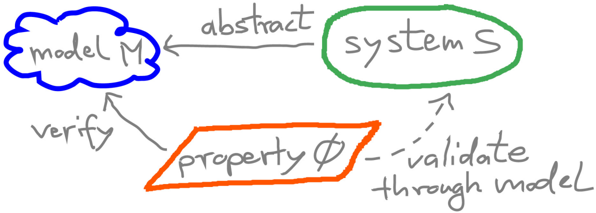
Talk won't commit to any one of these.
Talk focuses on formal verification of quantitative properties.

Abstraction-based (quantitative) model checking



If A satisfies ϕ and A abstracts S, then S satisfies ϕ.

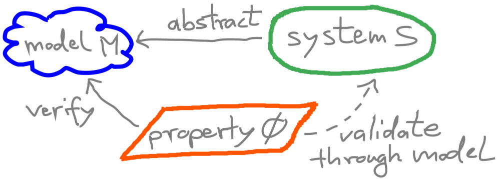
Technical problems with this approach



- System needs to be abstracted (sound?, precise enough?)
- Failed verification: have to remodify model or system
- Temporal logic may not have finite-model property (e.g. PCTL), so model may have to be infinite

Hard to automate, expensive, and may fail.

Conceptual problems with this approach



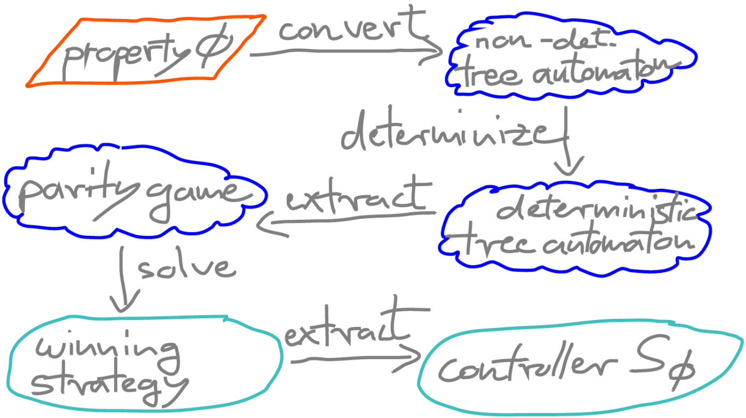
- Increased concurrency (cloud computing, multi-core platforms): harder to build desired systems manually
- Increased internet-based computing: systems no longer closed or no longer “real” (e.g. virtualization)
- Increased need for optimal tradeoffs: e.g. energy consumption vs. information security
- Increased need for systems as composed services

Design Synthesis

Paradigm shift

- A work-flow for system validation (as regular expression): (build or obtain system; (model a little; verify a little)*)*
- Increased stress for such work-flows due to aforementioned conceptual problems of the approach
- Is there a way to cope with this increased stress?
- Design synthesis may be able to help:
  - Models interaction of system with unknown environment
  - System as finite-state controller, to be designed
  - Temporal-logic formula specifies desired system behavior
  - Automated process for turning formula into controller
- Synthesized controller is correct by construction

Design synthesis as linear work-flow



Turns ϕ into S satisfying ϕ. No model, no model check.

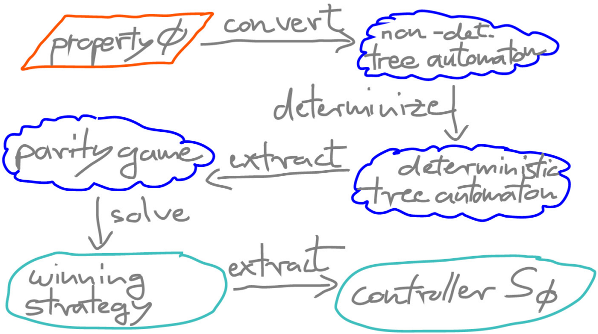
Quantitative Synthesis

Design Synthesis for Quantitative Systems?

A flavor of existing work:

- Bloem, Chatterjee, Henzinger, and Jobstmann 2009
  "Better Quality in Synthesis through Quantitative Objectives"
  - synthesize system S for property ϕ such that S is optimal with respect to some measure
  - e.g. preference of quick responses to requests in protocol
- Kwiatkowska, Norman, and Trivedi 2010
  "Quantitative Games on Probabilistic Timed Automata"
  - devise and solve quantitative, 2-player, 0-sum games
  - controller (winning strategy) optimizes time reach final state in probabilistic timed automaton

Quantitative synthesis: a wish list



- Adapt above process to quantitative systems.
- Support PCTL, regular path properties, counting, time, etc.
- Satisfiability and synthesis decidable for relevant fragments

Imperial College London

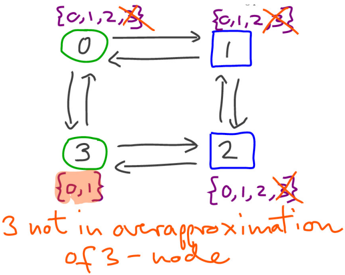
p-automata (QEST 2010 paper)

- accept language of Markov chains (DTMCs)
- can represent PCTL formulas & Markov chains
- support regular path properties and can count
- languages closed under bisimulation
- languages closed under Boolean operations
- acceptance of M by A (i.e. $M \in \mathcal{L}(A)$?) reduces to solving stochastic game
- complexity of $M \in \mathcal{L}(A_\phi)$ matches that of PCTL model checking $M \models \phi$

Q1. What is a good notion of non-deterministic p-automaton?
Q2. Is non-emptiness $\mathcal{L}(A) \neq \{\}$ decidable for such a notion?
Q3. How to do synthesis for a fragment of p-automata?

Static analysis for game-solving algorithms

- Solving quantitative or stochastic games is work horse of automata-based quantitative verification and synthesis.
- Static analysis can speed up such solvers, e.g. in over-approximations of optimal strategies in parity games:



Conclusions

Imperial College London

Daring predictions

- Today's and tomorrow's modeling challenges require more research on quantitative synthesis
- The boundaries between model checking and synthesis will become blurry
- (Quantitative) games and their solvers will become powerful back-ends of model validation tools
- Research in control theory, robust optimization, algorithmic game theory, and formal verification will converge more

Thank You for Your Kind Attention

Questions?