CS620: New Trends in Information Technology

Formal Modeling and Verification of Cyber-Physical Systems

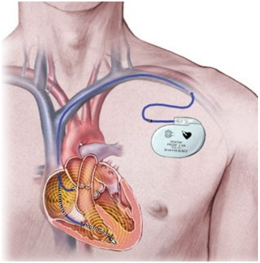
Krishna S. and Ashutosh Trivedi

Department of Computer Science and Engineering, IIT Bombay

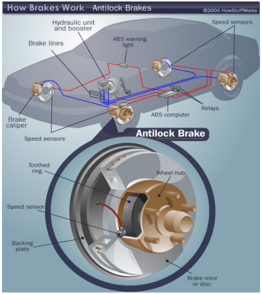
24 July 2013

Cyber-Physical Systems (CPS)

Medical Devices



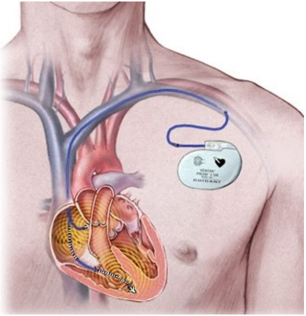
Avionics



Automobile

Energy

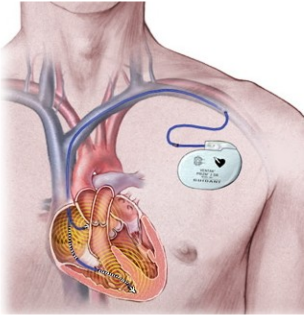
Cyber-Physical Systems: Medical Devices



1. Over half a million new implants every year
2. Failed implantable devices responsible for at least 212 deaths in US alone during 1997–2003
3. Percentage of software-related causes in medical device recalls grew from 10% to 21% from 1996 to 2006
4. In the first half of 2010 FDA issued 23 device recalls of defective devices, 6 out of which were software related defects

- Similar examples can be cited for CPS from other domains
- CPS are increasingly playing safety-critical role

Cyber-Physical Systems: Medical Devices



1. Over half a million new implants every year
2. Failed implantable devices responsible for at least 212 deaths in US alone during 1997–2003
3. Percentage of software-related causes in medical device recalls grew from 10% to 21% from 1996 to 2006
4. In the first half of 2010 FDA issued 23 device recalls of defective devices, 6 out of which were software related defects

- Similar examples can be cited for CPS from other domains
- CPS are increasingly playing safety-critical role

Challenge

How to guarantee the correctness/performance of Cyber-Physical Systems?

Model Based Design of Cyber-Physical Systems

Benefits:

- reduces time, risk, and costs
- permits evaluations of various design decision trade-offs
- enables design verification, validation, simulation, and testing
- allows automatic code-generation

Model Based Design of Cyber-Physical Systems

Benefits:

- reduces time, risk, and costs
- permits evaluations of various design decision trade-offs
- enables design verification, validation, simulation, and testing
- allows automatic code-generation

Current industrial practices:

- Norm in automotive and avionics industries
- Stateflow/Simulink is the most preferred tool (code-generation mechanism is highly trusted among control system designers)
- Rich functionality via interactive graphical environment
- Limited analysis (simulation)

Testing/Simulation Vs Formal Verification

Testing/Simulation

- traditional design verification techniques
- when to stop testing/simulation? (coverage criteria vs 70% of design time)
- detect presence of bugs but not absence
- inadequate for safety-critical systems

Testing/Simulation Vs Formal Verification

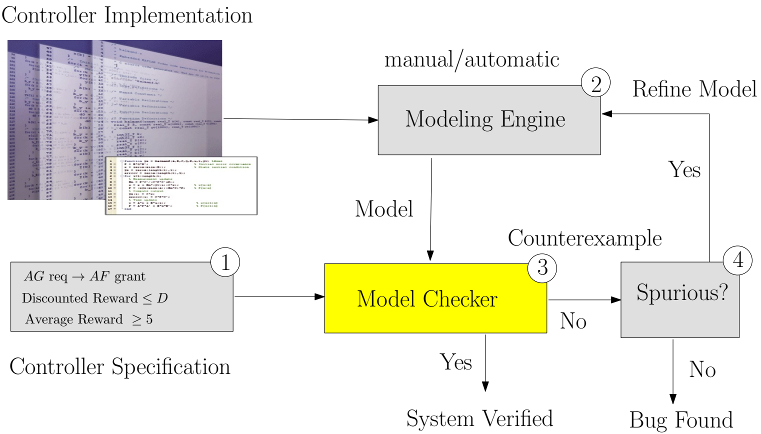
Testing/Simulation

– traditional design verification techniques
– when to stop testing/simulation? (coverage criteria vs 70% of design time)
– detect presence of bugs but not absence
– inadequate for safety-critical systems

Formal Methods for Verification and Synthesis

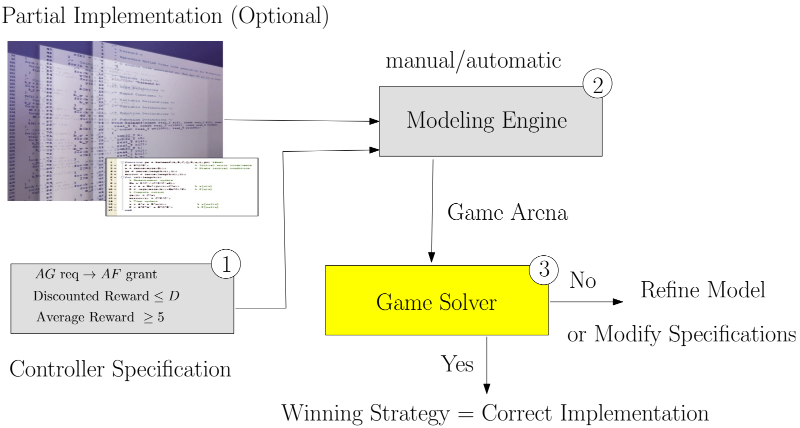
– employ rigorous mathematical reasoning to prove correctness of the systems or design provably correct systems
– based on exhaustive exploration of the state space
– formal verification/synthesis are computationally hard problem
– hence, scalability is one of the key challenges

Formal Methods: Verification and Synthesis



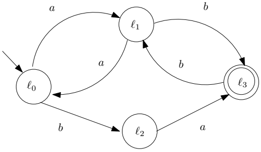
Formal Verification (Model Checking/ Performance Evaluation)

Formal Methods: Verification and Synthesis

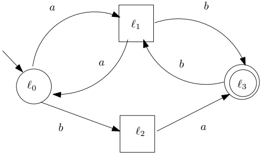


Controller Synthesis (Algorithmic Game Theory)

Verification/Synthesis with Finite State Machines



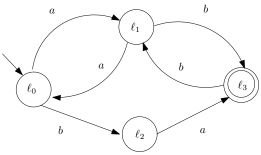
Finite Automata



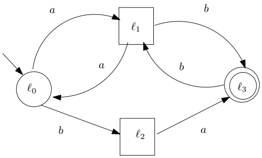
Finite Game Arena

- Backbone of both hardware and software verification (via abstraction)
- Quite influential both academically (two ACM Turing awards) and practically (two ACM Kanellakis awards)
- 30+ years of research efforts transformed design practices in both Hardware (Intel, Cadence) and Software (Microsoft, IBM) industries
- Extensive tool support:
  - Verification: NuSMV, SPIN, IFV (Cadence), SLAM (Microsoft)
  - Synthesis: Acacia, Lily, UnBeast, and Sketching.
- Current research focuses on improving scalability

Verification/Synthesis with Finite State Machines



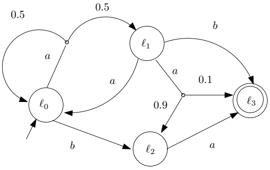
Finite Automata



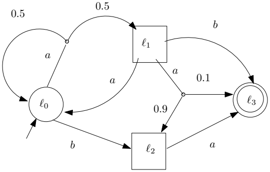
Finite Game Arena

- Backbone of both hardware and software verification (via abstraction)
- Quite influential both academically (two ACM Turing awards) and practically (two ACM Kanellakis awards)
- 30+ years of research efforts transformed design practices in both Hardware (Intel, Cadence) and Software (Microsoft, IBM) industries
- Extensive tool support:
  - Verification: NuSMV, SPIN, IFV (Cadence), SLAM (Microsoft)
  - Synthesis: Acacia, Lily, UnBeast, and Sketching.
- Current research focuses on improving scalability
- Inadequate for CPS Modeling:
  - stochastic modeling: faulty sensors/actuators, uncertainty in timing delays, random coin-flips, performance characteristics for third-party components
  - physical variables modeling

Verification/Synthesis with Stochastic Models



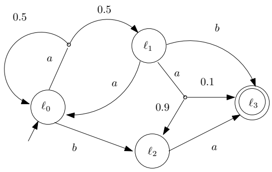
Markov Decision Process



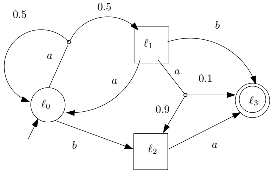
Stochastic Game Arena

- Applied in diverse fields such as Economics,control theory, OR, and AI
- A mature research field with wealth of results available (see excellent text from Puterman [Put94] and Filar & Vrieze and [FV97])
- Recent surge in research due to interest from verification community to model stochastic behavior
- Efficient tool support:
  - Verification: PRISM probabilistic model checker
  - Synthesis: GIST probabilistic game solver

Verification/Synthesis with Stochastic Models



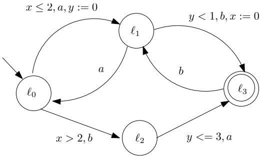
Markov Decision Process



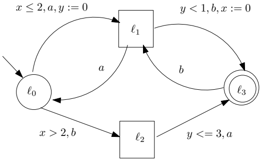
Stochastic Game Arena

- Applied in diverse fields such as Economics,control theory, OR, and AI
- A mature research field with wealth of results available (see excellent text from Puterman [Put94] and Filar & Vrieze and [FV97])
- Recent surge in research due to interest from verification community to model stochastic behavior
- Efficient tool support:
  - Verification: PRISM probabilistic model checker
  - Synthesis: GIST probabilistic game solver

Verification/Synthesis with Real-Time Models



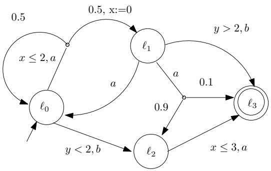
Timed Automata



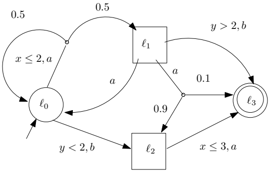
Timed Game Arena

- Introduced by Alur and Dill [AD94] to model real-time systems
- Verification
  - Reachability is decidable (in fact PSPACE-complete).
  - Language inclusion is undecidable.
- Synthesis
  - Reachability games are decidable (EXPTIME-complete)
  - A number of interesting open problems regarding games on weighted extensions of timed automata!
- Tool Support:
  - Verification : UPPAAL and Kronos
  - Sythesis: UPPAAL-Tiga

Models with Stochastic and Real-Time Behaviors



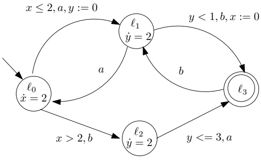
Probabilistic Timed Automata



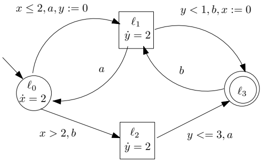
Probabilistic Timed Game Arena

- Introduced by [KNSS02] to model real-time and probabilistic systems
- Verification
  - Decidability of Qualitative model checking [KNSS02]
- Synthesis
  - Decidability of reachability-game [KNT10]
  - Number of open problems!
- Tool support: real-time extension of PRISM

Verification/Synthesis with Hybrid Automata



Hybrid Automata



Hybrid Game Arena

- Introduced by Alur et al. [ACHH93] to model hybrid systems
- Dynamics of physical variables are gives as ordinary differential equations
- Quite expressive, but undecidable verification (reachability) problems
- Decidable subclasses exists, e.g.
  - Initialized Rectangular Hybrid automata [HKPV98],
  - Hybrid Automata with Strong Resets [BBJ+08],
  - Piecewise constant derivative systems [AMP95],
  - Multi-Mode Systems [ATW12]
- Tool support: HyTECH, PHAVer
- A number of interesting open problems.

What we will study in this course?

- Language-theoretic properties of automata capable of modeling cyber-physical systems, e.g. timed and hybrid automata
- Modeling practical CPS using timed and hybrid automata via hands-on experience with hybrid system verification tools like UPPAAL and HyTech
- Expressiveness and decidability issues for various subclasses of hybrid automata
- Quantitative design (optimal controller-synthesis) and analysis (optimization) of CPSs using timed and hybrid automata
- Real-time extensions of temporal logics capable of specifying properties of CPS, e.g. beautiful theory of Metric temporal logic (LTL + timing constraints on operators)
- The quest for the automata-logic connection in the world of timed/hybrid automata

R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho.
Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems.
In *Hybrid Systems I*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer-Verlag, 1993.

R. Alur and D. Dill.
A theory of timed automata.
Theoretical Computer Science, 126(2):183–235, 1994.

Eugene Asarin, Oded Maler, and Amir Pnueli.
Reachability analysis of dynamical systems having piecewise-constant derivatives.
Theoretical Computer Science, 138:35–66, 1995.

R. Alur, A. Trivedi, and D. Wojtczak.
Optimal scheduling for constant-rate mulit-mode systems.
Technical Report MS-CIS-12-01, CIS, UPenn, 2012.
Accepted for publication in HSCC 2012.

P. Bouyer, T. Brihaye, M. Jurdzinski, R. Lazic, and M. Rutkowski.
Average-price and reachability-price games on hybrid automata with strong resets.

In *Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 5215 of *LNCS*, pages 63–77. 2008.

J. Filar and K. Vrieze.
Competitive Markov Decision Processes.
1997.

T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya.
What's decidable about hybrid automata?
Journal of Comp. and Sys. Sciences, 57:94–124, 1998.

M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston.
Automatic verification of real-time systems with discrete probability distributions.
Theoretical Computer Science, 282:101–150, June 2002.

M. Kwiatkowska, G. Norman, and A. Trivedi.
Quantitative games on probabilistic timed automata.
Computing Research Repository (CoRR), abs/1001.1933, 2010.

M. L. Puterman.
Markov Decision Processes: Discrete Stochastic Dynamic Programming.
Wiley, 1994.