PRISM Overview

• Automatic verification of probabilistic systems

DTMC, CTMC, MDP

Probabilistic model

Probabilistic specification

PRISM

Result

“yes”, “no”, prob.

PCTL, CSL
Probabilistic Models

- **Discrete-time Markov chains** (*DTMCs*)
  - Discrete time/probabilities
- **Continuous-time Markov chains** (*CTMCs*)
  - Real time (exponential distributions)
- **Markov decision processes** (*MDPs*)
  - Discrete time/probabilities + nondeterminism
PRISM Language

- Simple, state-based language for DTMCs/CTMCs/MDPs
  - based on Reactive Modules [Alur/Henzinger]
- Modules (system components, composed in parallel)
- Variables (local or global)
- Guarded commands (labelled with probabilities/rates)
- Action labellings (synchronisation between modules)
Language developments

- **Types and type checking** (ints, doubles, booleans)
- **Variable probabilities/rates/etc.**
  - e.g. \([x=0 \land n>0]\) \rightarrow \frac{1}{n} : (x'=1) + \frac{1-1}{n} : (x'=2)
- **Process algebra style constructions**
  - More flexible **parallel composition** of modules
    - \(P1 \parallel P2\)  \(P1 \parallel [a,b]\)  \(P2\)  \(P1 \parallel P2\)
  - Action hiding/renaming
  - Aim: translation from (probabilistic) CSP
Property Specifications

• **PCTL/CSL** - prob. extensions of CTL

• **P>p [ ◊ A ]**
  - “the probability that event A eventually occurs is > p”

• **Also: ◊≤T A “within time T”, A U B “until”**

• **S<p [ A ]**
  - “in the long-run, the probability that A is true is < p”
Property Specifications...

• Can now write “unbounded” formulae:
  - e.g. $P=? \left[ \diamond A \right]$ “what is the probability that...”

• Future:
  - Implement linear time (LTL) model checking
  - 2 new algorithms:
    • [Eliosoff/Panangaden]
    • [Couvreur/Saheb/Sutre]
Costs And Rewards

- Extend model with real-valued costs/rewards
  - State or transition based
  - e.g. “time”, “num. messages sent”, “power consumption”, “downtime”, ...

- Example properties:
  - “expected cost to reach a ?-state”
  - “expected cumulated cost by time T”
  - “expected cost at time instant t”
Graphical User Interface

- Complete redesign/implementation
- **Integrated editor** for PRISM language
- Support for "experiments"
  - e.g. check: $P \sim p[true \ U \leq T \ error]$ for $T=1..100$
- **Automatic graph** plotting
Screenshots

```plaintext
// constants
const int HEADS = 1;
const int TAILS = 2;

// a single module
dtmc
module coin

    // variable
    x : [0..3] init 0;

    // guarded commands
    (x=0) -> 0.5 : (x'=HEADS) + 0.5 : (x'=TAILS);
    (x>0) -> 1 : (x'=x);

endmodule
```
Screenshots...
GUI Prototypes

- **Graphical modelling language**
  - Based on UPPAAL

- **Simulator**
  - Manual exploration
    - e.g. counter examples?
  - Automatic simulations
Case Studies - Recent

- IPv4 ZeroConf protocol [FORMATS'03]
- Nanotechnology: multiplexing [VLSI'04]
- Wireless LAN: extension using costs
- Probabilistic fair exchange protocol
Case Studies - Ongoing

- **Bluetooth wireless protocol**
  - quality of service properties

- **Quantum cryptography**
  - BB84 key distribution protocol
External Uses of PRISM

- Model checking probabilistic extensions of UML state charts [Twente/Saarland]
- Performance analysis of PEPA nets (stochastic process algebra + Petri nets) [Edinburgh]
- Comparison with probabilistic extension of Murphi verifier [Rome/L'Aquila]
- Probabilistic model checking of fault-tolerant architectures [Monash]
Implementation

- “Symbolic” model checker (BDD-based)
  - Ongoing efficiency improvements
- Parallel, distributed versions
- Disk-based (“out-of-core”) version
- Alternative solution techniques
  - Sampling-based [Younes, Simmons]
  - Monte-Carlo approximations [Peyronnet et al.]
PRISM Research Directions

- Abstraction
  - Ordsets/scalar sets
  - Symmetry reduction
- Compositional approaches
- Mobility (pi calculus)
- Native support for PTAs
  - Add clocks to PRISM language