PRISM 4.0

Verification of Probabilistic Real-time Systems

Dave Parker

University of Oxford

CAV'11, Snowbird, Utah, July 2011

Joint work with: Marta Kwiatkowska, Gethin Norman, …

PRISM – An overview

- PRISM is a probabilistic model checker
  - automatic verification of systems with stochastic behaviour
  - e.g. due to unreliability, uncertainty, randomisation, …
- Construction/analysis of probabilistic models…
  - discrete- and continuous-time Markov chains, Markov decision processes, probabilistic timed automata
  - modelling language, case study repository, benchmark suite
- Verification of properties in probabilistic temporal logics…
  - PCTL, CSL, LTL, PCTL*, quantitative extensions, costs/rewards
- Various model checking engines and techniques…
  - symbolic (multi-terminal BDDs), explicit-state data structures, symmetry reduction, quantitative abstraction refinement, simulation-based (approximate/statistical model checking), …

PRISM – Probabilistic models

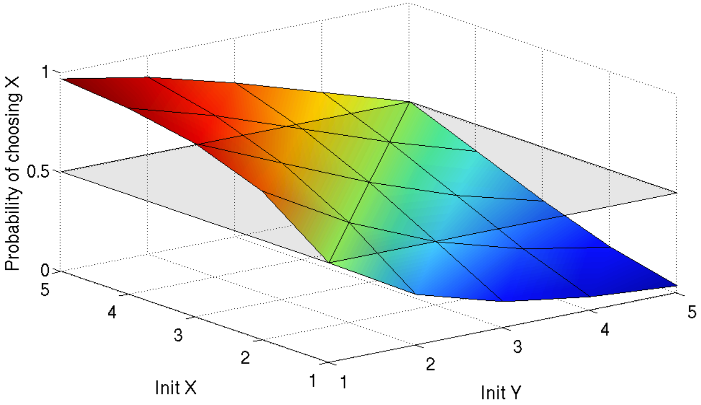
- Discrete-time Markov chains (DTMCs)
  - discrete states + probability
  - for: randomisation, unreliable communication media, …
- Continuous-time Markov chains (CTMCs)
  - discrete states + exponentially distributed delays
  - for: component failures, job arrivals, molecular reactions, …
- Markov decision processes (MDPs)
  - in fact: probabilistic automata [Segala]
  - probability + nondeterminism (e.g. for concurrency)
  - for: randomised distributed algorithms, security protocols, …
- Probabilistic timed automata (PTAs) [new in PRISM 4.0]
  - probability, nondeterminism + real-time
  - for wireless comm. protocols, embedded control systems, …

PRISM – Property specification

- Temporal logic-based property specification language
  - subsumes PCTL, CSL, probabilistic LTL, PCTL*, …

- Simple examples:
  - $P_{\leq 0.01}$ [F "crash"] – "the probability of a crash is at most 0.01"
  - $S_{>0.999}$ ["up"] – "long-run probability of availability is >0.999"

- Usually focus on quantitative (numerical) properties:
  - $P_{=?}$ [F "crash"]
    "what is the probability
    of a crash occurring?"
  - then analyse trends in
    quantitative properties
    as system parameters vary



PRISM – Property specification

- Properties can combine numerical + exhaustive aspects
  - $P_{max=?}$ [$F^{\leq 10}$ “fail”] – “worst-case probability of a failure occurring within 10 seconds, for any possible scheduling of system components”
  - $P_{=?}$ [$G^{\leq 0.02}$!“deploy” {“crash”}{max}] – “the maximum probability of an airbag failing to deploy within 0.02s, from any possible crash scenario”

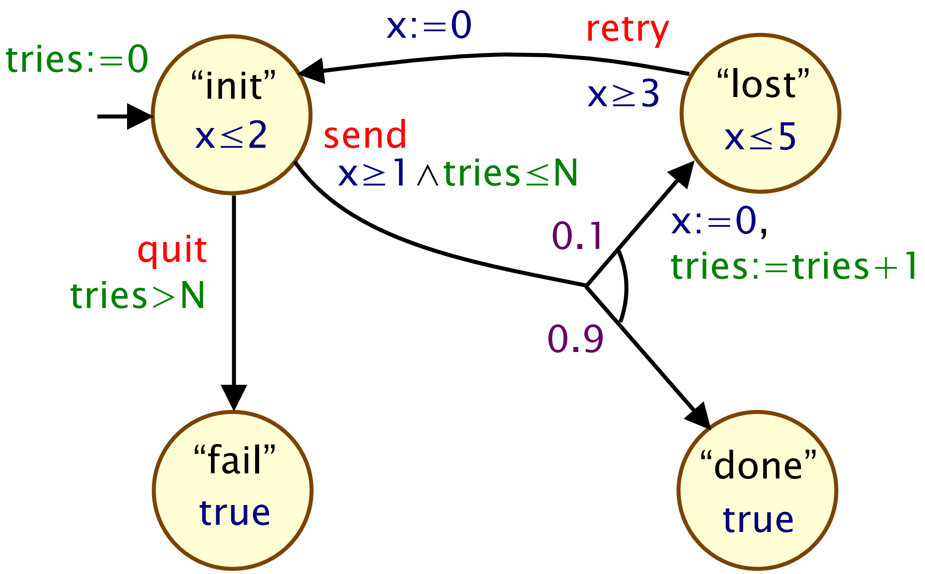
- Reward-based properties (rewards = costs = prices)
  - $R_{\{\text{“time”}\}=?}$ [F “end”] – “expected algorithm execution time”
  - $R_{\{\text{“energy”}\}max=?}$ [$C^{\leq 7200}$] – “worst-case expected energy consumption during the first 2 hours”

- Properties can be combined with e.g. arithmetic operators
  - e.g. $P_{=?}$ [F $fail_1$] / $P_{=?}$ [F $fail_{any}$] – “conditional failure prob.”

Probabilistic timed automata (PTAs)

- Probability + nondeterminism + real-time
  - timed automata + discrete probabilistic choice, or…
  - probabilistic automata + real-valued clocks

- PTA example: message transmission over faulty channel



States
- locations + data variables

Transitions
- guards and action labels

Real-valued clocks
- state invariants, guards, resets

Probability
- discrete probabilistic choice

Modelling PTAs in PRISM

- PRISM modelling language
  - textual language, based on guarded commands

```
pta
const int N;
module transmitter
    s : [0..3] init 0;
    tries : [0..N+1] init 0;
    x : clock;
    invariant (s=0 ⇒ x≤2) & (s=1 ⇒ x≤5) endinvariant
    [send] s=0 & tries≤N & x≥1
          → 0.9 : (s'=3)
          + 0.1 : (s'=1) & (tries'=tries+1) & (x'=0);
    [retry] s=1 & x≥3 → (s' =0) & (x' =0);
    [quit]  s=0 & tries>N → (s' =2);
endmodule
rewards "energy" (s=0) : 2.5; endrewards
```

Modelling PTAs in PRISM

- PRISM modelling language
  - textual language, based on guarded commands

```
pta
const int N;
module transmitter
    s : [0..3] init 0;
    tries : [0..N+1] init 0;
    x : clock;
    invariant (s=0 ⇒ x≤2) & (s=1 ⇒ x≤5) endinvariant
    [send] s=0 & tries≤N & x≥1
          → 0.9 : (s'=3)
          + 0.1 : (s'=1) & (tries'=tries+1) & (x'=0);
    [retry] s=1 & x≥3 → (s' =0) & (x' =0);
    [quit]  s=0 & tries>N → (s' =2);
endmodule
rewards "energy" (s=0) : 2.5; endrewards
```

Basic ingredients:

- modules
- variables
- commands

Modelling PTAs in PRISM

- PRISM modelling language
  - textual language, based on guarded commands

```
pta
const int N;
module transmitter
    s : [0..3] init 0;
    tries : [0..N+1] init 0;
    x : clock;
    invariant (s=0 ⇒ x≤2) & (s=1 ⇒ x≤5) endinvariant
    [send] s=0 & tries≤N & x≥1
          → 0.9 : (s'=3)
          + 0.1 : (s'=1) & (tries'=tries+1) & (x'=0);
    [retry] s=1 & x≥3 → (s' =0) & (x' =0);
    [quit]  s=0 & tries>N → (s' =2);
endmodule
rewards "energy" (s=0) : 2.5; endrewards
```

Basic ingredients:

- modules
- variables
- commands

New for PTAs:

- clocks
- invariants
- guards/resets

Modelling PTAs in PRISM

- PRISM modelling language
  - textual language, based on guarded commands

```
pta

const int N;

module transmitter

    s : [0..3] init 0;
    tries : [0..N+1] init 0;

    x : clock;

    invariant (s=0 ⇒ x≤2) & (s=1 ⇒ x≤5) endinvariant

    [send] s=0 & tries≤N & x≥1
          → 0.9 : (s'=3)
          + 0.1 : (s'=1) & (tries'=tries+1) & (x'=0);
    [retry] s=1 & x≥3 → (s' =0) & (x' =0);
    [quit]  s=0 & tries>N → (s' =2);

endmodule

rewards "energy" (s=0) : 2.5; endrewards
```

Basic ingredients:

- modules
- variables
- commands

New for PTAs:

- clocks
- invariants
- guards/resets

Also:

- rewards (i.e. costs, prices)
- parallel composition

Model checking PTAs in PRISM

- Properties for PTAs:
  - min/max probability of reaching X (within time T)
  - min/max expected cost/reward to reach X
    (for "linearly-priced" PTAs, i.e. reward gain linear with time)

- PRISM has two different PTA model checking techniques…

- "Digital clocks" – conversion to finite-state MDP
  - preserves min/max probability + expected cost/reward/price
  - (for PTAs with closed, diagonal-free constraints)
  - efficient, in combination with PRISM's symbolic engines

- Quantitative abstraction refinement
  - zone-based abstractions of PTAs using stochastic games
  - provide lower/upper bounds on quantitative properties
  - automatic iterative abstraction refinement

Also new in PRISM 4.0

- Discrete-event simulation engine
  - newly rewritten for PRISM 4.0
- Approximate/statistical model checking
  - approximate results (and confidence interval) for e.g. $P_{=?}$ […]
  - acceptance sampling (SPRT) for approximating e.g. $P_{<p}$ […]
  - offers improved scalability for fully-probabilistic models
- Generation of optimal strategies (schedulers, adversaries)
  - for MDPs (and, via digital clocks, for PTAs)
- New components for developers
  - explicit-state probabilistic model checking library
  - quantitative abstraction refinement component
  - discrete-event simulation engine

The PRISM benchmark suite

- PRISM models are widely used for testing/benchmarking
  - but there are many case studies in several locations
  - can be hard to find the right type of examples for testing
- The PRISM benchmark suite
  - collection of probabilistic model checking benchmarks
  - designed to make it easy to test/evaluate/compare tools
  - currently, approx. 20 models, of various types and sizes
  - wide range of model checking properties, grouped by type
  - PRISM can also export built models in various formats
- See: www.prismmodelchecker.org/benchmarks

More information…

- More info and resources at: www.prismmodelchecker.org
  - download PRISM (free, open source, runs on all major OSs)
  - documentation, tutorials, case studies
  - related papers, teaching material, benchmarks

- Tool demo session: Tue pm
  - or just ask any time…

- Coming soon:
  - probabilistic counterexample generation
  - multi-objective probabilistic model checking
  - assume-guarantee model checking
  - and more…