Probabilistic model checking with PRISM

Dave Parker

University of Birmingham

September 2005
Overview

- **Probabilistic model checking**
  - probabilistic models, property specifications, algorithms

- **Tool support: PRISM**
  - functionality, PRISM modelling language

- **Case studies**
  - molecular reactions, cell cycle control in Eukaryotes, Fibroblast Growth Factor (FGF)
Model Checking

- Automatic **formal verification** of concurrent systems
  - construct model: all possible states/configurations, transitions
  - analysis based on **exhaustive exploration** of model

Labelled transition system

- System model
- Specification
- Model checker
- **Result**
  - "yes", "no", error trace

Temporal logic, e.g. CTL
Probabilistic Model Checking

- Automatic formal verification of probabilistic systems

- Probabilistic model
  - e.g. Markov chain

- Probabilistic specification
  - e.g. PRISM

- Temporal logic, e.g. CSL

- Prob. model checker
  - "yes", "no", probabilities, ...

- Result
Probabilistic Models

- **Discrete-time Markov chains (DTMCs)**
  - discrete state space $S$, time-steps and probabilities
  - transition probability matrix: $P : S \times S \rightarrow [0,1]$

- **Continuous-time Markov chains (CTMCs)**
  - discrete state space $S$, continuous (real) time
  - transition rate matrix: $R : S \times S \rightarrow [0,\infty)$
  - $R(s,s') = \text{parameter of negative exponential distribution}$
  - $P (\text{"transition } s \rightarrow s' \text{ occurs within } t \text{ time units") = } 1-e^{-R(s,s') \cdot t}$
  - suited to modelling component lifetimes, inter-arrival times, ...

- **Other possibilities:**
  - Markov decision processes: nondeterminism (concurrency) + probs
  - probabilistic timed automata: nondeterminism + real-time + probs
Specifying model properties

- Formulae in temporal logic – prob. extensions of CTL
  - for DTMCs/MDPs, use PCTL [Hansson & Jonsson]
  - for CTMCs, use CSL [Aziz et al.] [Baier et al.]
- Path-based properties
  - $\text{Prob}_s(\phi)$ - “prob. paths from state $s$ satisfy path formula $\phi$”
  - $\phi = “\text{eventually } X”, “X is always true”, “X by time } t\”,“X within time bound [t1,t2]”, “X until Y”, ...
- Transient versus steady-state properties
  - transient probabilities: $p_s(s',t)$ - “probability, from state $s$, of being in state $s'$ at time instant $t$”
  - steady-state probabilities: $p_s(s') = \lim_{t \to \infty} \{p_s(s',t)\} = p(s')$
Specifying model properties...

• **P operator** – analogue of CTL's A (for all) E (there exists)
  - $\text{EF event (CTL)} = P>0 \ [ F \text{ event }]$ (CSL)
  - $P<0.001 \ [ F \text{ shutdown }]$ - “shutdown eventually occurs with probability at most 0.001”
  - $P<0.2 \ [ F[t,t] \ (\text{conc}_x < \text{min}) ]$ “the probability that the concentration of reactant x has dropped below minimum at time t is less than 0.2”
  - $P\geq 0.95 \ [ \neg \text{repair} \ U\leq 200 \text{ done }]$ - “with probability 0.95 or greater, the process will successfully complete within 200 hours and without requiring any repairs”

• **S operator** - steady-state
  - $S>0.75 \ [ \text{num}_s\text{ensors} \geq \text{min} ]$ - “in the long-run, the probability that an adequate number of sensors are operational is greater than 0.75”
Computing actual values

- Determine **actual** probabilities
  - $P=? \quad [\ F \ shutdown \ ]$ - “what is the probability of shutdown eventually occurring?”
  - $S=? \quad [\ conc_x > \ text{safe} \ ]$ - “what is the long-run probability that the concentration of $x$ is above a safe level?”
  - $P=? \quad [\ F \ shutdown \ \{x=11 \ & \ y=17\} \ ]$
    “probability in a specific model state”
  - $P=? \quad [\ F \ shutdown \ \{"init"\}\{\text{max}\} \ ]$
    “worst-case probability from several states”

- **Experiments**
  - e.g. $P=? \quad [\ F \leq T \ \text{stable}\ ]$ for $N=1..5$ and $T=1..100$
  - useful for spotting trends, anomalies
Cost- and reward-based properties

- Costs and rewards
  - real-valued measures assigned to states/transitions
- Instantaneous – assigned to states
  - queue size, reactant concentration, num. of molecules of...
  - “what is the expected conc. of reactant X at time t?”
  - “what is the long-run expected conc. of reactant X?”
- Cumulative – states (rates) or transitions
  - time, power consumption, messages lost, ...
  - “what is the expected power consumption during the first 2 hours of operation?”
  - “what is the expected time taken for the protocol to terminate?”
Probabilistic model checking involves...

- **Construction of model**, i.e. state space, rate matrix
  - translation from high-level modelling language
  - exploration of reachable state-space

- **Probabilistic model checking** algorithms
  - graph-theoretical algorithms – underlying transition relation
    - e.g. states which reach X with probability exactly 0 or 1
  - numerical computation – calculation of probabilities, expected values of costs/rewards, etc.
    - linear equation systems, linear optimisation problems
    - iterative methods
      - Jacobi, Gauss-Seidel, ...
      - dynamic programming (MDPs)
      - Uniformisation (CTMCs)
  - can also use **discrete-event simulation** (approximations)
Probabilistic model checking

• **Strengths**
  - **wide range** of quantitative measures
  - **exact answers** computed
  - **fully automatic**
  - **exhaustive analysis**, good for 'corner cases', e.g.
    • all possible initial configurations
    • all possible process schedulings (nondeterminism)
  - **efficient** algorithms and implementations

• **Weaknesses**
  - **state space explosion**: time and memory constraints
    • combine with simulation? abstractions?
  - **no counter examples**, as in non-prob. model checking
PRISM

- **PRISM**: Probabilistic Symbolic Model Checker
  - Tool developed at Birmingham Uni, approx. 7 years
  - Multiple platforms: Linux, Unix, Mac OS X, Windows
  - Freely available, open source (GPL)
  - [www.cs.bham.ac.uk/~dxp/prism](http://www.cs.bham.ac.uk/~dxp/prism)

- **Support for** construction/analysis of:
  - Models: CTMCs, DTMCs, MDPs
  - Logics: CSL, PCTL + custom extensions
  - Costs + rewards (ongoing)
PRISM

- **Features**
  - **PRISM language**: high-level model descriptions
  - **GUI**: PRISM language editor, graph plotting, ...
  - **Sophisticated implementation**
  - multiple solution techniques/engines
  - efficient data structures
    - binary decision diagram (BDD) based - “symbolic”
  - **Simulator** (ongoing)
    - manual exploration, debugging
    - Monte Carlo sampling
PRISM: Model editor
PRISM: Experiments, graphs
PRISM case studies

• Biological processes
  - Eukaryotic cell cycle control (based on Lecca & Priami)
  - RKIP-inhibited ERK pathway (by Calder et al) [CMSB 2005]
  - FGF: Fibroblast Growth Factor (ongoing)

• Communication/multimedia protocols
  - Bluetooth device discovery [ISOLA'04]
  - IEEE 1394 FireWire [FAC 2003, STTT 2004]
  - IPv4 Zeroconf dynamic configuration [FORMATS'03]
  - IEEE 802.11 (WiFi) wireless LAN MAC protocol [PROBMIV'02]

• Security protocols
  - Crowds anonymity protocol (by Shmatikov) [JSC 2003]
  - Probabilistic contract signing (with Shmatikov) [FASec'02]
PRISM case studies...

• Performance/reliability studies
  - NAND multiplexing for nanotechnology (with Shukla) [VLSI'04]
  - Dynamic power management (with Shukla and Gupta) [HLDVT'02]
  - Embedded controllers [INCOM'04]
  - Manufacturing systems
  - PC clusters, queueing systems

• Randomised distributed algorithms
  - leader election, self-stabilisation, consensus, mutual exclusion, ...

• More...
  - www.cs.bham.ac.uk/prism/casestudies
PRISM modelling language

• Simple, state-based language for DTMCs/CTMCs/MDPs
  - based on Reactive Modules [Alur/Henzinger]

• Basic components:
  - modules (system components, parallel composition)
  - variables (finite-state, typed)
  - guarded commands (rate/probability, action-labelled)

\[ [\text{bind}] \ (N > 0) \rightarrow r^*M : (N' = N - 1) + r : (N' = N + 1); \]
PRISM modelling language...

• Other features:
  - synchronisation on action labellings
  - shared global variables
  - process algebra style specifications
    • parallel composition: \( P_1 ||| P_2, P_1 |[a,b]| P_2, P_1 || P_2 \)
    • action hiding/renaming: \( P/\{a\}, P\{a<-b\} \)
  - macros

• Connections:
  - import of PEPA models
  - import of CSP + probability models (ongoing)
  - planned connection to BioCHAM
Example: Molecular Reactions

- Model time until a reaction occurs
  - by an exponential distribution [Gillespie'77]
  - using continuous time Markov chains

- Rate of reaction determined by:
  - base rate (empirically determined constant)
  - concentration of reactants (number of each type of molecule that takes part in the reaction)

- This example: Na + Cl $\leftrightarrow$ Na$^+$ + Cl$^-$
  - forward base rate 100
  - backwards base rate 10
  - initially $N_1$ Na atoms and $N_2$ Cl atoms

- Based on examples for BioSPI [Regev & Shapiro]
const int N1; // number of Na atoms
const int N2 = N1; // number of Cl atoms

module na // Na and Na+ module

    na : [0..N1] init N1; // no of Na and Na+ atoms is fixed at N1

    [e1] na>0 -> na : (na'=na-1);
    [e2] na<N1 -> (N1-na) : (na'=na+1);

endmodule

module cl // Cl and Cl- module

    cl : [0..N2] init N2; no of Cl and Cl- atoms is fixed at N2

    [e1] cl>0 -> cl : (cl'=cl-1);
    [e2] cl<N2 -> (N2-cl) : (cl'=cl+1);

endmodule
const double e1rate = 100; // Na + Cl  -> Na+ + Cl-
const double e2rate = 10; // Na+ + Cl- -> Na + Cl

// module representing the base rates of reactions
module base_rates

dummy : bool; // dummy variable

[e1] true -> e1rate : true;
[e2] true -> e2rate : true;

endmodule

// reward: "percentage of Na atoms present (out of all Na/Na+)"
rewards
    true : 100*na/N1;
endrewards
Results: Molecular Reactions

\[ P = \text{F}[T,T] \text{na}=i \]

“probability of i Na atoms at time T”
Results: Molecular Reactions

\[ R = \frac{?}{I = T} \]

“expected percentage of Na atoms at time T”
Results: Molecular Reactions

R=? [ S ]

“expected percentage of Na atoms in the long run”
Case study: Cell cycle control

- Investigate physiology of cell proliferation in Eukaryotes
  - in terms of underlying proteins/molecules
  - cell cycle regulatory system: complex, concurrent reactions
- Most common approach: e.g. [Nasymth et al.]
  - ordinary differential equations (ODEs)
  - model variation of protein concentrations over time
  - some deterministic assumptions
- Alternative approach [Lecca & Priami, BIOCONCUR'03]
  - stochastic model using stochastic pi-calculus
  - yields a CTMC model, Monte-Carlo simulation with BioSPI
Case study: Cell cycle control...
Case study: Cell cycle control...

- Modelled in **PRISM** to evaluate the probabilistic model checking technology for such applications
  - based on stochastic pi-calculus formulation
  - parameters taken from [Nasmyth et al.]
  - exact answers computed

- Questions
  - effort required?
  - resulting model size?
  - expressiveness of temporal logic?
Results: Cell cycle control

\[ P = \text{[ F}[T,T] \text{ cyclin}=k \text{]} \]

“quantity of cyclin bound at time T equals k”
Results: Cell cycle control

\[ R=? [ I=T ] \]

“expected quantities at time T”
Case study: FGF pathway

- **FGF: Fibroblast Growth Factor**
  - receptor mediated signalling process
  - e.g. regulation of skeletal development

- **Modelling**
  - expensive experimental scenarios, can we prioritise them through modelling?
  - test model against experiments by removing key components, predicting outcome

- **Challenges** - proteins can:
  - change concentration - degradation/synthesis
  - associate/disassociate with partner proteins
  - change properties, e.g. phosphorylation/dephosphorylation
  - change location within cell
Case study: FGF pathway
Case study: FGF pathway

- Study **GRB2/SOS binding to FRS2** as a measure of Ras activation
- Evaluate role of three different types of signal attenuation mechanisms:
  - phosphorylation/dephosphorylation - modelled by SHP2
  - competition between and active/inactive forms of GRB2:SOS - modelled by GRB2:SOS bound to SPRY (inactive) or FRS2 (active)
  - proteolytic destruction of signalling component - modelled by CBL bound to SPRY
FGF pathway: PRISM/BioSPI

• **PRISM study** - ongoing work
  - collaboration with John Heath, Biosciences, Birmingham
  - aims to evaluate probabilistic model checking approach on “real-life” examples

• **Early results using simulation** (very large models)
  - BioSPI and PRISM
  - “percentage of GRB2 bound to FRS2 at time T”?
  - “how does this vary depending on model parameters?”
  - obtained testable predictions
Results: FGF

- “Percentage of GRB2 bound to FRS2 at time T”, for different scenarios
Results: FGF

- “Percentage of GRB2 bound to FRS2 at time T”, for different scenarios and longer timescales
Current and future work

- **Ongoing**
  - Case studies: FGF, mobile ad-hoc network protocols, quantum cryptographic protocols, ...
  - PRISM: simulator, graphical modelling language, ...
  - Efficiency: parallel/distributed implementations

- **Future directions:**
  - model reductions: symmetry, abstraction, compositionality
  - more expressive models:
    - continuous time MDPs
    - mobility, e.g. pi calculus
  - more expressive specifications
    - prob. LTL/PCTL*/mu-calculus
Case study: RKIP-inhibited ERK pathway

• Cell signalling
  – Need to understand complex signalling dynamics
  – Important in cancer research

• Models of variants of ERK pathway
  – ODE-based [Kolch et al], using Gepasi, [Cho et al, CMSB’03]
  – Stochastic [Calder et al, BIOCONCUR’04], using stochastic process algebra PEPA, and PRISM [Calder et al, CMSB’05]

• Contribution of this study
  – model checking of “if the concentration reaches a certain level, will it remain at that level thereafter?”
  – abstraction based on ‘levels’ of quantities, used instead of continuous values
  – aligned with ODE model, 5-6 levels suffices
Case study: RKIP-inhibited ERK pathway

- Small sized, but still practical
  - 11 species
  - 11 reactions

- Important for cancer research
  - connection to FGF

- Allows evaluation of semi-quantitative properties
  - the probabilities of an activation precedence
  - stable state analysis
  - range of parameter values
Results: RKIP-inhibited ERK pathway

\[ P=? (\text{red\_level}=2 \ U[T,T] \ \text{green\_level}=5) \]
“probability that the “red” protein will reach level 2 before the “green” one reaches level 5”

\[ S = 2.3 \quad S = 0.1 \]

**Conclusion:**
Over 98%, this activation sequence is very stable.
FGF pathway: the model

• The stochastic pi-calculus fragment

\[
\begin{align*}
GRB2 & := (bind1, R1).GRB2.SOS \\
SOS & := (bind1, R1).0 \\
GRB2.SOS & := (bind2, R2).GRB2.SOS.RAF \\
& + (release1, R3).(GRB2 | SOS) \\
GRB2.SOS.RAF & := (release2, R4).(GRB2.SOS | RAF) \\
Release & := (release1, R3).Release \\
& (release2, R4).Release
\end{align*}
\]

• Encodes

\[
\begin{align*}
GRB2 + SOS & = GRB2:SOS \\
GRB2:SOS + RAF & = GRB2:SOS:RAF
\end{align*}
\]