Probabilistic Model Checking of Labelled Markov Processes via Finite Approximate Bisimulations

Marta Kwiatkowska

Oxford University Computing Laboratory

PrakashFest 2014, Oxford, 23-25 May 2014

Joint work with: Alessandro Abate, Dave Parker and Gethin Norman

UNIVERSITY OF OXFORD

Probabilistic model checking with PRISM meets Labelled Markov Processes

A historical interlude

- MFPS 1990

AD-A237 754

Final Project Report

Principal Investigator: Prof. Micheal W. Mislove
PI Institution: Tulane University
6823 St. Charles Ave.
New Orleans, LA 70118
(504) 865-5727

Date of this report: June 30, 1991
ONR Grant #N00014-90-J-1809, Computer Science Division
Award period: April 1, 1990 - March 31, 1991
Cummulative total: $5,700

6TH WORKSHOP ON THE MATHEMATICAL FOUNDATIONS OF PROGRAMMING SEMANTICS

This workshop was the sixth in this series, which dates back to 1985. It was held on the campus of Queen's University, Kingston, Canada from May 16 to May 19, 1990. Since this meeting was a workshop, as opposed to a full conference, the intention was to keep the number of participants small to enhance the chance for interaction among the attendees. There were 63 participants at the

A historical interlude

- MFPS 1990

Thursday, 17 May

9:00 am
P. Panangaden - Invited Speaker
Queen's University

10:00 am
S. Brookes
Carnegie Mellon University
Towards a theory of parallel algorithms on concrete data structures.

10:30 am
Coffee Break

11:00 am
R. Kent
University of Arkansas
Processes as 2-dimensional relational structures.

11:30 am
M. Kwiatkowska
University of Leicester
Causality and fairness properties.

- No probability, yet…

A few years later…

Probabilistic Methods in Verification

(PROBMIV'98)

A Pre-LICS'98 Workshop

19-20 June 1998, Indianapolis, Indiana, USA

Workshop description and aims

Scientific Justification: While there has been a steady current of research activity in probabilistic logics and systems for some years, little experimental work has been done up until now. This situation is beginning to change. Randomization has proved effective in deriving efficient distributed algorithms and is now widely used in practical applications, to mention computer networks and graphics. However, randomized algorithms are notoriously difficult to verify: the proofs of their correctness are complex, and therefore argued informally, and thus appropriate formal methods and tools are called for. These have to combine a variety of dissimilar techniques, from conventional proof theory and model checking, through systems modelling to linear algebra and probability theory.

- Questions asked in panel session
  - Randomization – is it really used widely?
  - Where are the tools? heuristics?
  - Did you find any bugs?
- In this talk
  - Some answers
  - As always, new challenges!

www.cs.bham.ac.uk/~mzk/probmiv98.html

Probabilistic model checking

- First algorithms proposed in 1980s
  - [Vardi, Courcoubetis, Yannakakis, …]
  - algorithms [Hansson, Jonsson, de Alfaro] & first implementations
- 2000: tools ETMCC (MRMC) & PRISM released
  - PRISM: efficient extensions of symbolic model checking
  - ETMCC (now MRMC): model checking for continuous-time Markov chains [Baier, Hermanns, Haverkort, Katoen, …]
- Selected advances in probabilistic model checking:
  - 1997 BBD-based symbolic methods [Baier, de Alfaro, K, Parker, …]
  - 2000 Uniformisation [Baier, Haverkort, Hermanns, Katoen, …]
  - 1999 Zone-based techniques [Sproston, Norman, Parker, K, …]
  - 2007 Multi-objective methods [Etessami, Vardi, K, Yannakakis, …]
  - 2010 Compositional methods [K, Norman, Parker, Qu, …]
  - 2012 Stochastic games [Simaitis, Forejt, Chen, Parker, K, …]

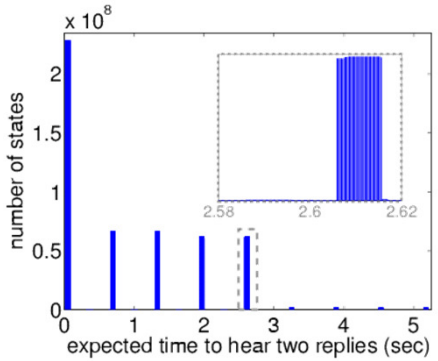
Probabilistic model checking

- What's involved
  - typically more expensive than the non-probabilistic case: need to build *and solve* model
  - algorithms involve often non-trivial combination of
    - graph-based analysis (typically symbolic)
    - and numerical solution, e.g. linear equations/linear programming
  - or simulation-based analysis (statistical model checking)

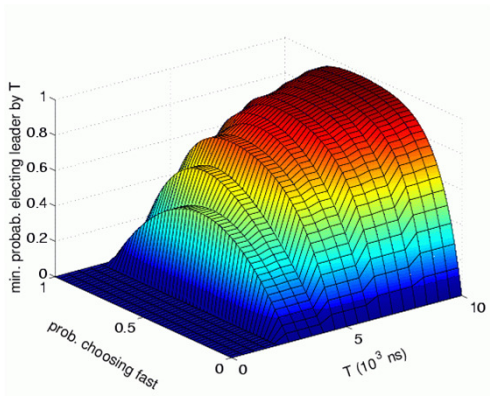
- The state of the art
  - fast/efficient techniques for a range of probabilistic models
  - feasible for models of up to 10^7 states (10^{10} with symbolic)
  - successfully applied to many application domains:
    - distributed randomised algorithms, communication protocols, security protocols, biological systems, quantum cryptography, …
  - beyond model checking: parametric methods, synthesis, …

Probabilistic model checking in action

- Bluetooth device discovery protocol
  - frequency hopping, randomised delays
  - low-level model in PRISM, based on detailed Bluetooth reference documentation
  - numerical solution of 32 Markov chains, each approximately 3 billion states
  - identified worst-case time to hear one message, 2.5 seconds

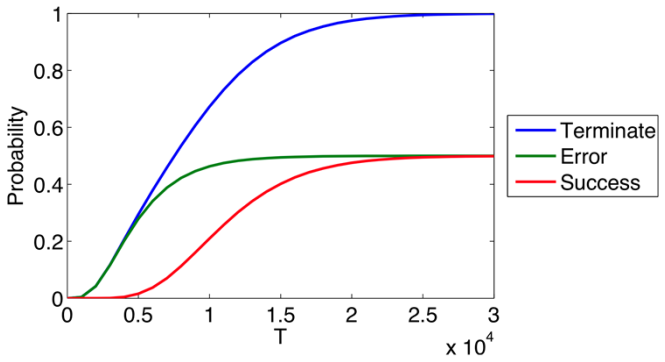


- FireWire root contention
  - wired protocol, uses randomisation
  - model checking using PRISM
  - optimum probability of leader election by time T for various coin biases
  - demonstrated that a biased coin can improve performance

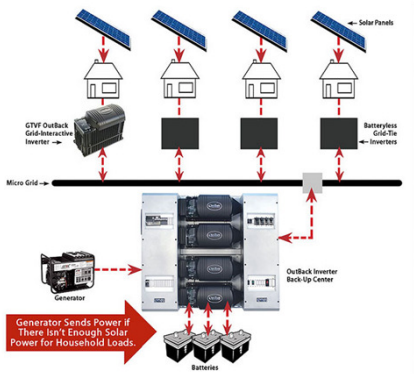


Probabilistic model checking in action

- DNA transducer gate [Lakin et al, 2012]
  - DNA computing with a restricted class of DNA strand displacement structures
  - transducer design due to Cardelli
  - automatically found and fixed design error, using Microsoft's DSD and PRISM



- Microgrid demand management protocol [TACAS12,FMSD13]
  - designed for households to actively manage demand while accessing a variety of energy sources
  - found and fixed a flaw in the protocol, due to lack of punishment for selfish behaviour
  - implemented in PRISM-games



But…

- Largely limited to discrete state, discrete probability models
  - Markov chains, Markov decision processes
  - continuous time: via uniformisation or zone abstractions
  - continuous distributions: via discretisation, no error bounds
  - continuous space: some early work on probabilistic hybrid automata, via reduction to MDPs [Hahn, Hermanns, …]
- At the same time
  - need more realistic models (real-time behaviour, continuous dynamics, stochastic hybrid systems, etc)
  - since 1997, much existing work on continuous space models [Panangaden, Desharnais, …]
  - separately, also in control and decision literature
- This talk: about extending probabilistic model checking for Labelled Markov Processes

Overview

- Probabilistic model checking & PRISM
  - background & history
- Labelled Markov processes (LMPs)
  - semantics of LMP models
  - formally relating LMPs and Markov decision processes (MDPs)
- Notions of (approximate) bisimulation for LMPs
- Probabilistic model checking of LMPs via PRISM
  - constructive derivation of finite abstraction (approximate bisimmulation) for LMP (MDP and thus for LMP)
  - case study: room temperature control

Labelled Markov processes

- Labelled Markov processes (LMPs) [Panangaden et al.]
  - uncountably infinite (continuous) state space
  - (we skip here details on measurability and topology)
  - evolve sequentially in discrete time-steps
  - here, we restrict our attention to a finite time interval [0,N]
- An LMP is a structure $(S, s_0, B(S), \{\tau_u | u \in U\})$ where:
  - S is the state space
  - $s_0 \in S$ is the initial state
  - $B(S)$ is a Borel σ-field on S
  - $\tau_u : S \times B(S) \to [0,1]$ is a sub-probability transition function
  - and U is a finite set of labels
- Evolution of LMP depends on the choice of label ("action") $u \in U$ at each time step, which may be accepted or rejected

Labelled Markov processes

- For the purposes of model checking, we also add:
  - a labelling $L : S \to 2^{AP}$ of states with atomic propositions
  - reward (or cost) structures of the form $r : S \times U \to \mathbb{R}_{\geq 0}$
- LMP semantics: two alternative views…
  - testing (à la [Larsen/Skou]) or decision processes (e.g. MDPs)
- Testing process
  - emphasis on observing labels ($u \in U$)
- Decision process
  - emphasis on controlling the system via labels and observing underlying system dynamics (variables, atomic propositions)

Semantics of LMPs

- Semantics of LMP $(S, s_0, B(S), \{\tau_u | u \in U\})$
- Model initialised at $k=0$ in state s_0
- At any $0 \leq k \leq N-1$, given $s_k \in S$ and selecting $u_k \in U$:
  - probability of successor s_{k+1} given by $\tau_u(s_k, \cdot)$
  - label u_k is accepted with probability $\int_S \tau_{u_k}(s_k, dx)$
  - otherwise, action u_k is rejected, with two possible behaviours:

  1. (testing process) the dynamics stops, s_{k+1} is undefined, process yields finite trace $(s_0,u_0),(s_1,u_1),\ldots,(s_k,u_k)$
  2. (decision process) some default action u is selected (either u_k or some extra $u' \notin U$), continues with sample $s_{k+1} \sim \tau_u(s_k, \cdot)$, yielding $(s_0,u_0),(s_1,u_1),\ldots,(s_k,u_k),(s_{k+1},u)\ldots,(s_{N-1},u),s_N)$

- For a policy (or strategy) σ that picks each u_k, we get a probability measure P^σ over the sample space S^{N+1}

Semantics: Examples

- Example (testing process)
  - slot/vending machine
  - internal state with finite-memory register
  - at a given (discrete) time, user pushes button (label)
  - machine (possibly – with some probability) outputs such label, and if so probabilistically updates state of register
- Example (decision process)
  - room temperature control (see later case study)
  - temperature in room evolves non-autonomously, i.e. is affected by controllable heater
  - at a given (discrete) time, user selects heater status (label) and the temperature is updated accordingly
  - heater can break down with some probability (label rejected)
  - temperature still updated according to default action

Exact probabilistic bisimulation

- Probabilistic bisimulation for LMPs [Desharnais/Edalat/Panangden'02]
  - extends notion for (discrete-state) Markov chains, MDPs
- An (exact) probabilistic bisimulation is an equivalence relation R on LMP states S such that, whenever s_1 R s_2:
  - $L(s_1) = L(s_2)$ and $r(s_1,u) = r(s_2,u)$ for all $u \in U$
  - $\tau_u(s_1,b) = \tau_u(s_2,b)$ for any $u \in U$ and set $b \in S/R$ (which is Borel measurable)
- As usual:
  - $s_1,s_2 \in S$ are called bisimilar if s_1 R s_2 for some such relation R
  - definition can be extended to relate two LMPs
- Questions/issues:
  - too conservative (i.e. fine) in practice?
  - is it feasible (or robust) to compute?

Approximate probabilistic bisimulation

- Approximate probabilistic bisimulation [Desharnais/Laviolette/Tracol'08]
  - for some precision ϵ, i.e. ϵ-probabilistic bisimulation
- An ϵ-probabilistic bisimulation is a relation R on LMP states S such that, whenever s_1 R s_2:
  - $L(s_1) = L(s_2)$ and $r(s_1,u) = r(s_2,u)$ for all $u \in U$
  - $|\tau_u(s_1,b) - \tau_u(s_2,b)| \leq \epsilon$ for any $u \in U$ and R_ϵ-closed set $b \subseteq S$
- In general, R_ϵ is not an equivalence relation (not transitive)
  - so induces a covering, not a partition, of S
- Like before:
  - $s_1, s_2 \in S$ are ϵ-bisimilar if s_1 R_ϵ s_2 for some such relation R_ϵ
  - definition can be extended to relate two LMPs
- And again: may be difficult to compute in practice

Finite-state approximations of LMPs

- We construct an abstraction of a (continuous-state) LMP as a (discrete-state) Markov decision process (MDP)
  - strictly speaking, abstraction is a labelled Markov chain (possibly containing sub-distributions)
- Induced by discretisation of state space S
  - i.e. a finite partition $S_1 \cup \ldots \cup S_Q = S$
  - that preserves both state labels and rewards
  - partition blocks correspond to abstract states
  - transition probabilities approximated by piecewise constant f.
- Finite abstraction (MDP) is ϵ-probabilistically bisimilar to concrete model where, over any possible label, ϵ
  - depends on max partition diameters and volumes
  - depends on Lipschitz constant
  - … straightforward to compute

Probabilistic model checking LMPs

- Time-bounded (finite-horizon) fragment of PRISM's property specification language (PCTL + rewards)
  - ϕ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | $P_{\sim p}$ [ϕ $U^{\leq k}$ ϕ]| $R_{\sim x}$ [$C^{\leq k}$]
  - where a is an atomic proposition, $p \in [0,1]$ is a probability bound, $x \in \mathbb{R}_{\geq 0}$ is a reward bound, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$
- Semantics defined wrt policies σ mapping state/time to labels
  - verification: M,s satisfies ϕ for all policies
  - synthesis: find optimal policy that satisfies ϕ
- Examples
  - $P_{\sim p}$ [ϕ $U^{\leq k}$ ϕ] – probability of satisfying until formula is ~ p
  - $P_{max=?}$ [ϕ $U^{\leq k}$ ϕ] – maximum probability of satisfying until
  - $R_{\sim x}$ [$C^{\leq k}$] – expected reward cumulated up to k steps is ~ x

Main theorem

- Theorem (approx. preservation of logic)
- For ϵ-bisimilar states $s \in M$, $s^d \in M^d$ and until property $\phi\ U^{\leq K}\ \phi$
  - For any (measurable) policy σ of M there exists a policy σ^d on M^d such that

    $$| P_\sigma(s, \phi\ U^{\leq K}\ \phi) - P_\sigma{}^d(s^d, \phi\ U^{\leq K}\ \phi) | \leq \epsilon K$$
  - where K is the step bound of until
  - For any policy σ^d on M^d there exists a (measurable) policy σ of M a such that

    $$| P_\sigma{}^d(s^d, \phi\ U^{\leq K}\ \phi) - P_\sigma(s, \phi\ U^{\leq K}\ \phi) | \leq \epsilon K$$
- Similarly for rewards
- (Approx) verification and synthesis thus
  - reduce to the computation of min/max probability (reward) on the discretised finite LMP
  - can reuse existing techniques for MDPs

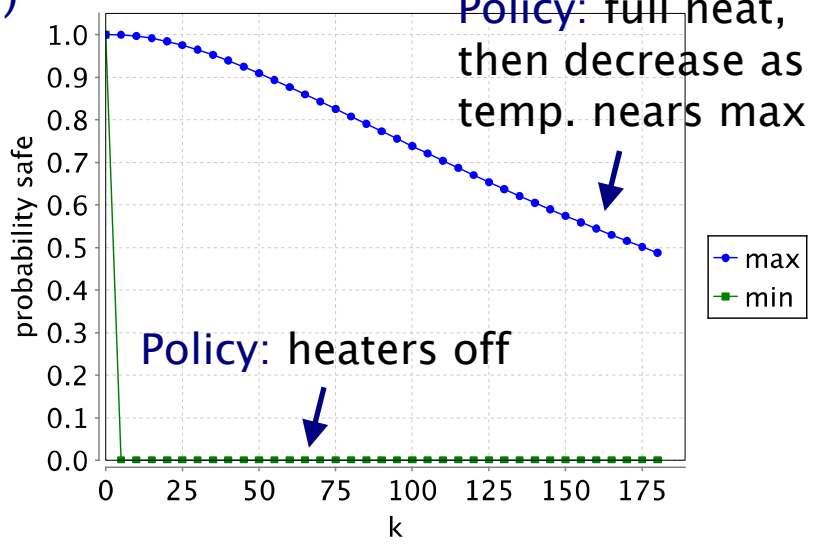
Case study

- Case study: multi-room heating system
  - 2 adjacent rooms, each with a heater and one shared control
  - control switches both heaters between 10 heating levels
- Modelled as an LMP with state space $S = \mathbb{R}^2$
  - state $(t_1,t_2) \in S$ gives temperature t_i in room i
  - average temperature evolves according to a stochastic difference equation; also model heat transfer between rooms
  - labels correspond to heating level changes
  - goal: keep temperature in "safe" interval [17.5, 22.5]°C
  - 0-1 reward structured added to count time in "safe"
  - fixed time horizon of N=180 steps
- Abstraction
  - temperature range partitioned into 5 sub-intervals

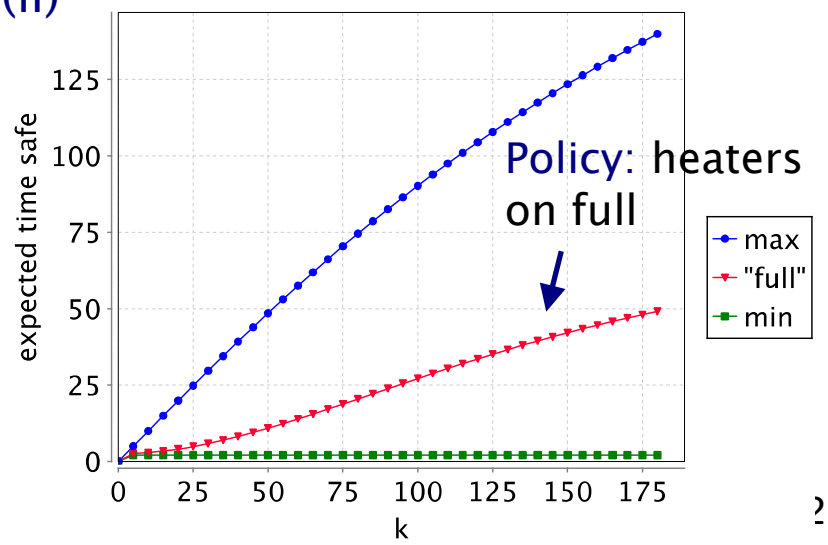
Case study

- Use probabilistic model checking to find:
  - (i) min/max probability of staying in "safe" over k steps
  - (ii) min/max expected time spent in "safe" over k steps
  - also synthesise optimal strategy (policy) to achieve these
- Tools used:
  - MATLAB to build abstraction; PRISM for prob. model checking

(i)



(ii)



Conclusions

- Developed automated methods to compute approximations of LMPs
  - enabled (approx) verification and strategy synthesis for LMPs
  - guaranteed error bounds: $K\epsilon$, where K is the step bound
  - extends to bounded LTL
- Efficient/symbolic implementation?
  - e.g. combination of numerical solution and simulation
- More expressive properties?
  - e.g. multiobjective
- More expressive models?
  - e.g. incentivise behaviour
- More case studies?
  - e.g. automotive controllers