Distributed LTL Model Checking of Probabilistic Systems

Jiří Barnat, Luboš Brim, Ivana Černá

ParaDiSe
Parallel & Distributed Systems Laboratory

Faculty of Informatics
Masaryk University
Brno

SENVA 2006

Overview

- probabilistic systems
- LTL model checking of probabilistic systems
- accepting end components
- sequential algorithms
- distributed algorithm for qualitative model checking

Probabilistic systems

Markov chain

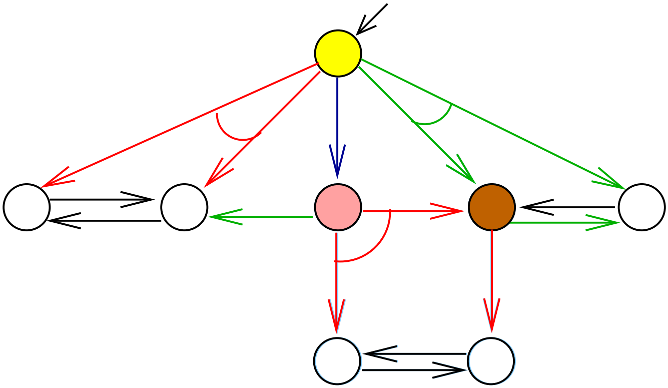
(finite state) sequential probabilistic program

Markov decision process

concurrent probabilistic program

- probability and nondeterminism
- each state is associated a set of possible actions
- choice of the action is nondeterministic
- the choosen action determines the transition probability disribution for the successor states

Markov decision process



Markov decision process - polices

Policy

- resolves the nondeterminism in states
- reduces the system to ordinary stochastic system
  (to reason about probability of events of interest)
- history dependent, deterministic polices

Qualitative model checking of LTL properties

Markov chain

program is correct if the specification is satisfied with probability one

Markov decision process

program is correct if meets the specification with probability one for all polices

Quantitative model checking of LTL properties

Markov chain

the exact probability that the program satisfies the specification

Markov decision process

maximal (resp. minimal) probability represents the probability that the program meets the specification provided that the nondeterministic choices are as favorable (resp. unfavorable) as possible

Qualitative verification - complexity

Given MDP M and LTL formula f

Markov chain

$$\mathcal{O}(|M| \cdot 2^{|\mathcal{O}(f)|})$$

Courcobetis, Yannakakis, 1995; Bustan, Rubin, Vardi, 2004

Markov decision process

$$\mathcal{O}(|M|^2 \cdot 2^{2^{|\mathcal{O}(f)|}})$$

Courcobetis, Yannakakis, 1995

Qualitative verification - algorithms

- transform $\neg f$ into a deterministic ω-automaton A
- product MDP $M \times A$
- calculate accepting end components (AEC) in $M \times A$
- existence of a reachable AEC implies the existence of a policy under which f holds with positive probability

- end component is a set of states that can be repeated infinitely often along a path with nonzero probability
- end component is accepting if the accepting condition of ω-automaton A holds

Accepting end component

Product MDP viewed as a graph

end component is a strongly connected component closed under probabilistic transitions

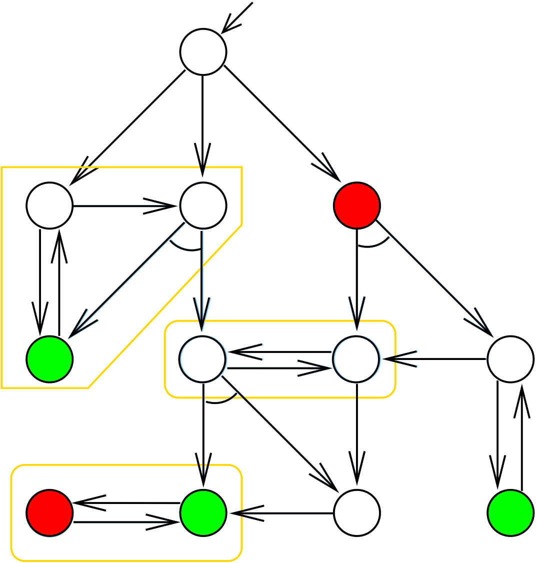
accepting condition for deterministic Rabin automaton is a collection of pairs of sets of states

$$[(L_1, U_1), \ldots, (L_k, U_k)]$$

End component C is accepting iff *for some i* we have

$$C \cap L_i \neq \emptyset \text{ and } C \cap U_i = \emptyset$$

Example



Reachability of AEC - sequential algorithm

For every pair (L, U)

- decompose G into maximal SCC
- iterate
  - If a component Q is not closed under probabilistic transitions then delete the bad states from G and recompute the decomposition.
  - If a component Q does not contain any L-state then delete all states in Q from G.
  - If a component Q contains both states from L and U then delete the U-states from G and recompute the decomposition.

The final decomposition consists of all AEC.

Complexity $O(n \cdot (n + m))$

Reachability of AEC - sequential vs distributed algorithm

Sequential setting

decomposition into strongly connected components

Distributed setting

reachability ??

Reachability of AEC

Fix a pair (L, U)

Elimination criterion

if

- no L-state is "safely" reachable from state

or

- out-degree of state is zero

then the state does not belong to AEC

Reachability of AEC - distributed algorithm

For every pair (L, U)

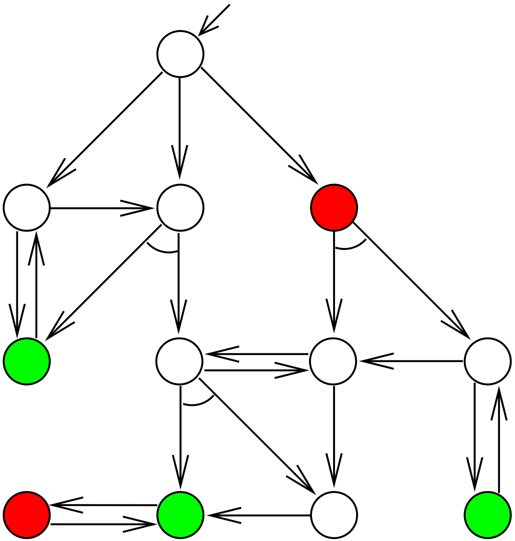
iterate

- mark all states from which an L-state is reachable along a path without any U-states
- eliminate all unmarked states
- recursively eliminate
  - states with zero out-degree
  - incomplete probabilistic transitions

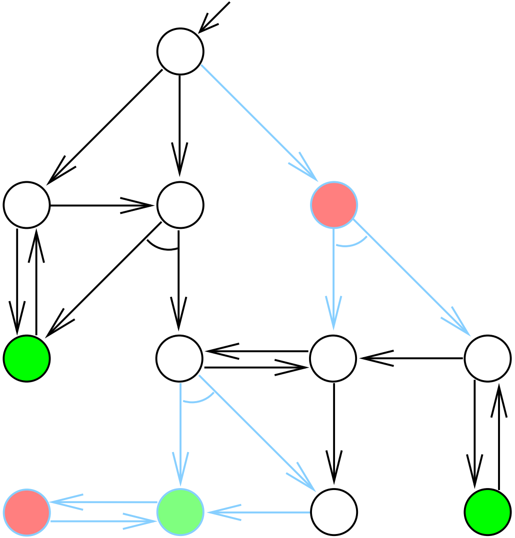
until stabilization

If the resulting graph is nonempty there is a reachable AEC

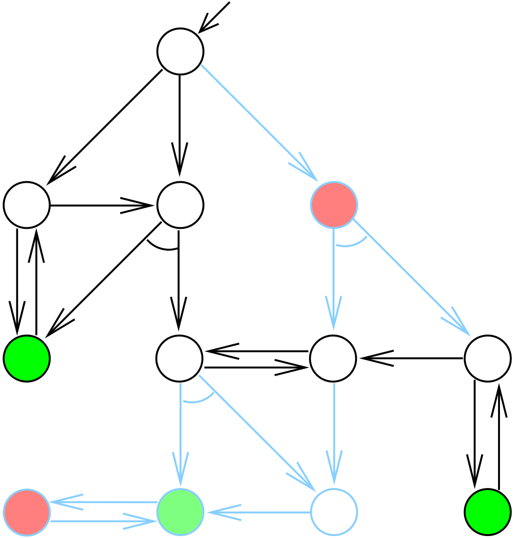
Example



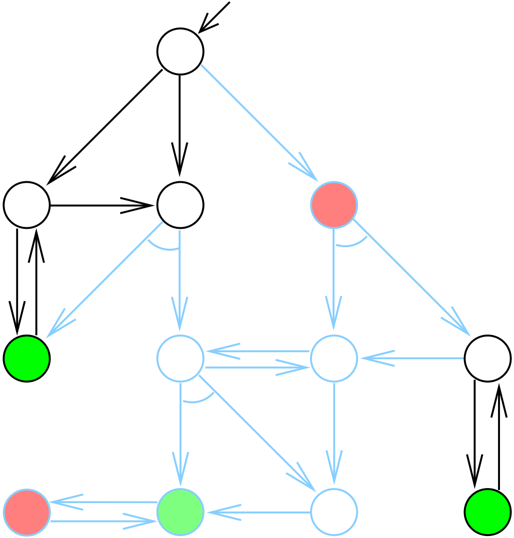
Example - cont.



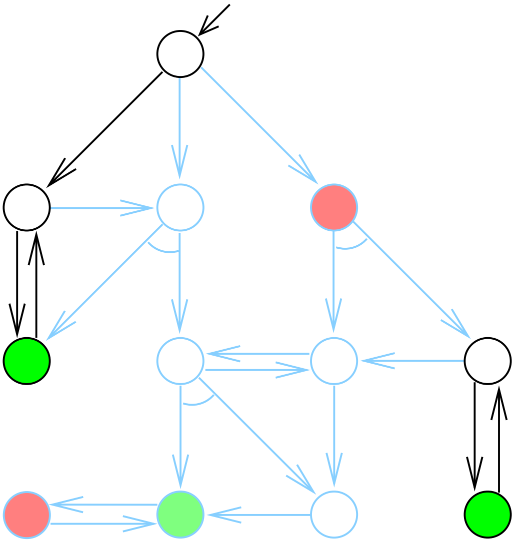
Example - cont.



Example - cont.



Example - cont.



Reachability of AEC - time complexity

A - property automaton, M - MDP, $M \otimes A$ - product automaton

For every pair (L, U) $\quad |A|$

iterate $\quad |M \otimes A|$

- mark all states from which an L-state is reachable along a path without any U-states $\quad |M \otimes A|$
- eliminate all unmarked states $\quad |M \otimes A|$
- recursively eliminate $\quad |M \otimes A|$
  - states with zero out-degree
  - incomplete probabilistic transitions

until stabilization

$$\mathcal{O}(|A| \cdot (|M \otimes A| \cdot |M \otimes A|) = \mathcal{O}(|M|^2 \cdot 2^{2^{\mathcal{O}(|f|)}})$$

Reachability of AEC - time complexity

Time complexity for Markov chains

$$\mathcal{O}(|M| \cdot 2^{2^{\mathcal{O}(|f|)}})$$

Reachability of AEC - space complexity

Space complexity

$$\mathcal{O}(|M \otimes A|)$$

reversed edges

Future work

- identification of *all* AEC based on reachability
- quantitative questions
- is nondeterminism unavoidable?
- implementation, DiVinE