On Model Checking Techniques for Randomized Distributed Systems

Christel Baier
Technische Universität Dresden

joint work with Nathalie Bertrand
Frank Ciesinski
Marcus Größer

Probability elsewhere

- randomized algorithms [Rabin 1960]
  breaking symmetry, fingerprints, input sampling, ...

- stochastic control theory [Bellman 1957]
  operations research

- performance modeling [Markov, Erlang, Kolm., ∼ 1900]
  emphasis on steady-state and transient measures

- biological systems, resilient systems, security protocols

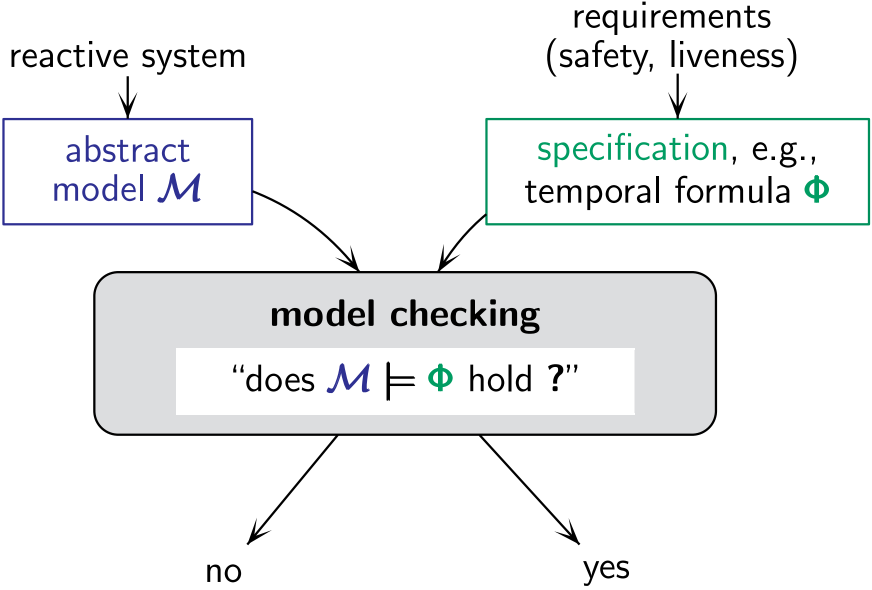
  ⋮

Probability elsewhere

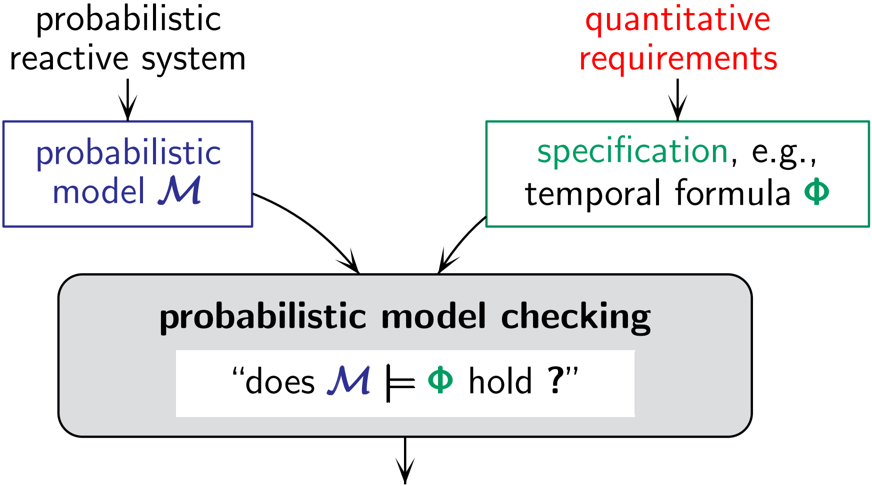
- randomized algorithms [Rabin 1960]
  breaking symmetry, fingerprints, input sampling, ...
  models: discrete-time Markov chains
  Markov decision processes
- stochastic control theory [Bellman 1957]
  operations research
  models: Markov decision processes
- performance modeling [Markov, Erlang, Kolm., $\sim$ 1900]
  emphasis on steady-state and transient measures
  models: continuous-time Markov chains
- biological systems, resilient systems, security protocols

  ⋮

Model checking [Clarke/Emerson, Queille/Sifakis]

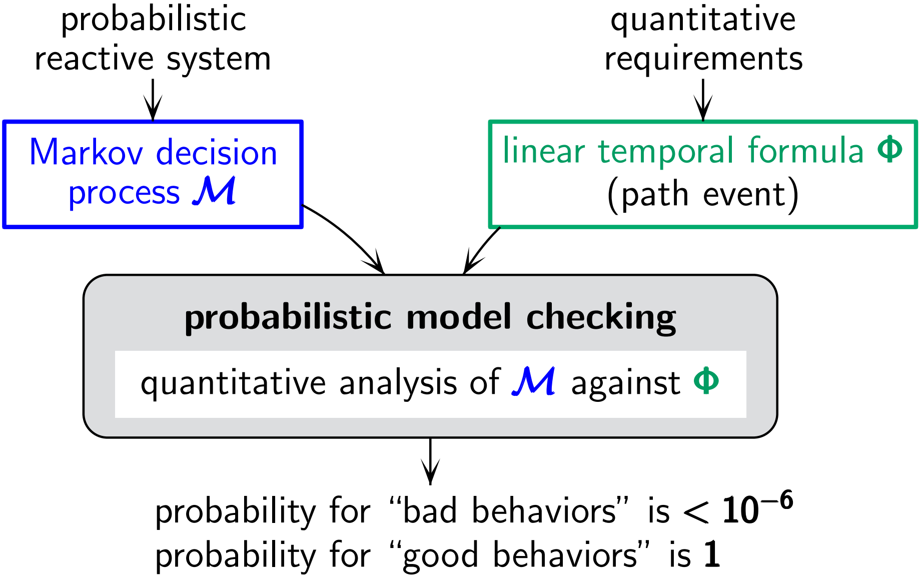


Probabilistic model checking



probability for “bad behaviors” is $< 10^{-6}$
probability for “good behaviors” is **1**
expected costs for ....

Probabilistic model checking



Outline

- Markov decision processes (MDP) and quantitative analysis against path events
- partial order reduction for MDP
- partially-oberservable MDP
- conclusions

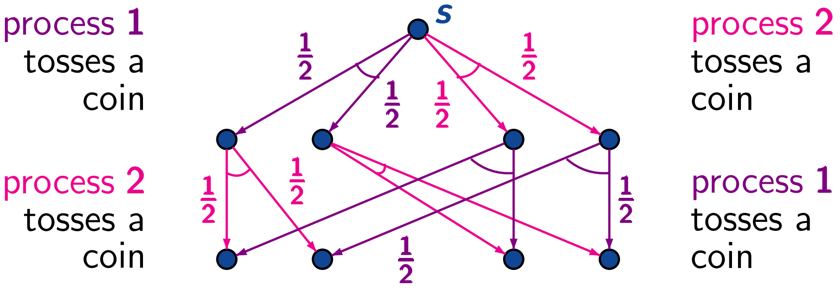
Markov decision process (MDP)

operational model with nondeterminism and probabilism

Markov decision process (MDP)

operational model with nondeterminism and probabilism

- modeling randomized distributed systems
  by interleaving



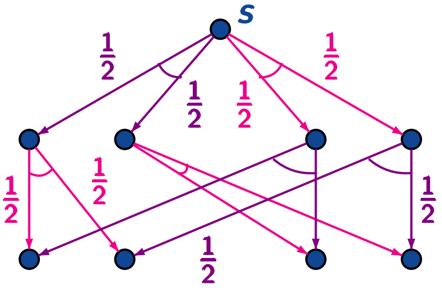
Markov decision process (MDP)

operational model with nondeterminism and probabilism

- modeling randomized distributed systems by interleaving
- nondeterminism useful for abstraction, underspec., modeling interactions with an unkown environment

process 1 tosses a coin

process 2 tosses a coin



process 2 tosses a coin

process 1 tosses a coin

Markov decision process (MDP)

$$\mathcal{M} = (S, Act, P, \ldots)$$

Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, \ldots)$

- finite state space S

Markov decision process (MDP)

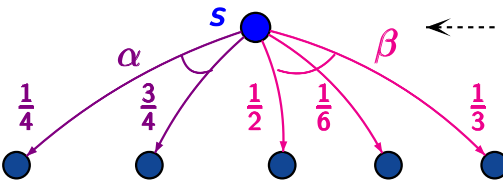
$\mathcal{M} = (S, Act, P, \ldots)$

- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0, 1]$

Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, \ldots)$

- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0, 1]$



nondeterministic choice

probabilistic choice

Markov decision process (MDP)

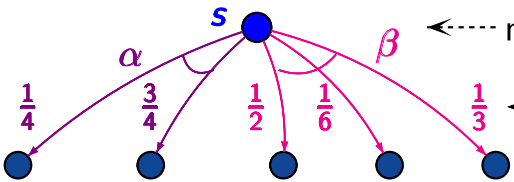
$\mathcal{M} = (S, Act, P, \ldots)$

- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0,1]$ s.t.

$Act(s) =$ set of actions that are enabled in state s

$$\forall s \in S \;\; \forall \alpha \in Act. \;\; \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$$

$\alpha \notin Act(s)$ → 0 $\quad$ $\alpha \in Act(s)$ → 1



← nondeterministic choice

← probabilistic choice

Markov decision process (MDP)

$$\mathcal{M} = (S, Act, P, s_0, AP, L, rew, \ldots)$$

- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0,1]$ s.t.

$$\forall s \in S \ \ \forall \alpha \in Act. \ \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$$

(0 if $\alpha \notin Act(s)$; 1 if $\alpha \in Act(s)$)

- s_0 initial state
- AP set of atomic propositions
- labeling $L : S \to 2^{AP}$
- reward function $rew : S \times Act \to \mathbb{R}$

Randomized mutual exclusion protocol

- 2 concurrent processes $\mathcal{P}_1$, $\mathcal{P}_2$ with 3 phases:

  n_i noncritical actions of process $\mathcal{P}_i$
  w_i waiting phase of process $\mathcal{P}_i$
  c_i critical section of process $\mathcal{P}_i$

- competition of both processes are waiting

Randomized mutual exclusion protocol

- 2 concurrent processes $\mathcal{P}_1$, $\mathcal{P}_2$ with 3 phases:

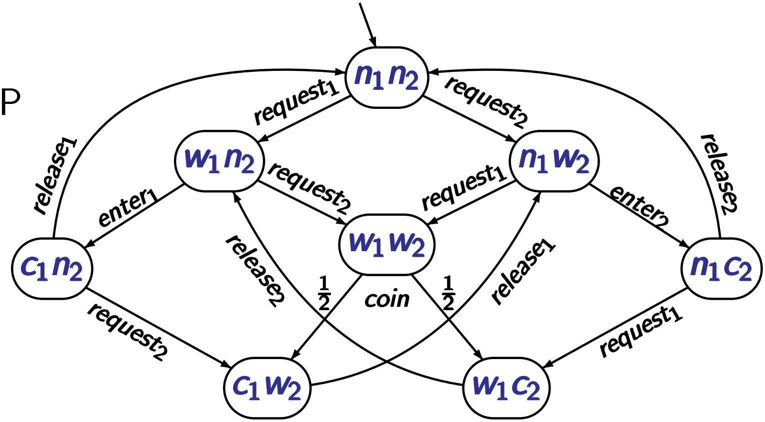
  n_i noncritical actions of process $\mathcal{P}_i$
  w_i waiting phase of process $\mathcal{P}_i$
  c_i critical section of process $\mathcal{P}_i$

- competition of both processes are waiting
- resolved by a randomized arbiter who tosses a coin

Randomized mutual exclusion protocol

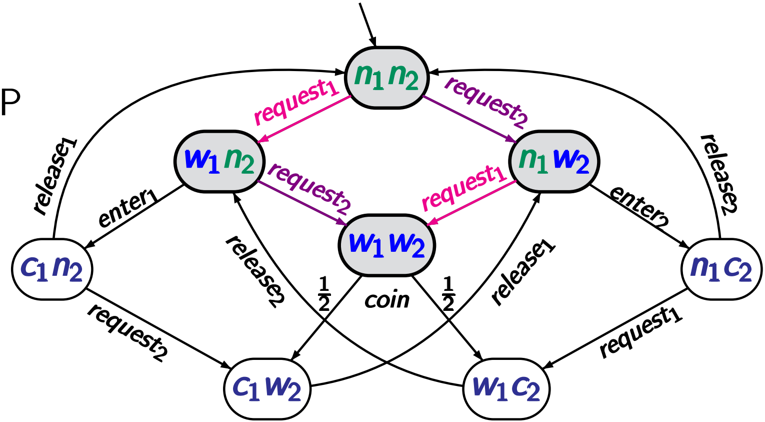
- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

MDP



- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

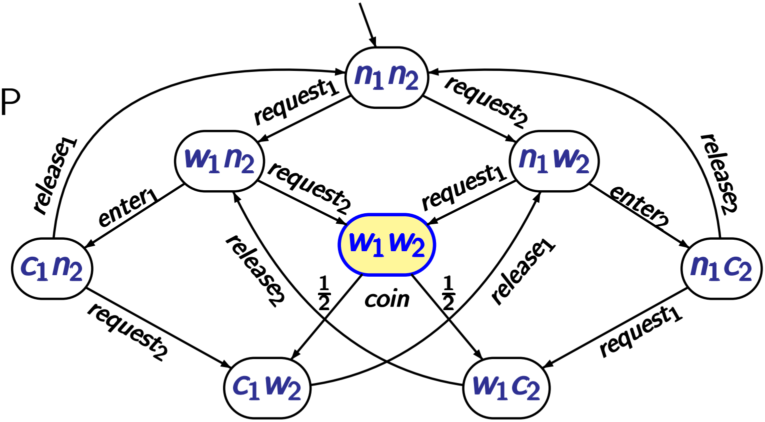
MDP



Randomized mutual exclusion protocol

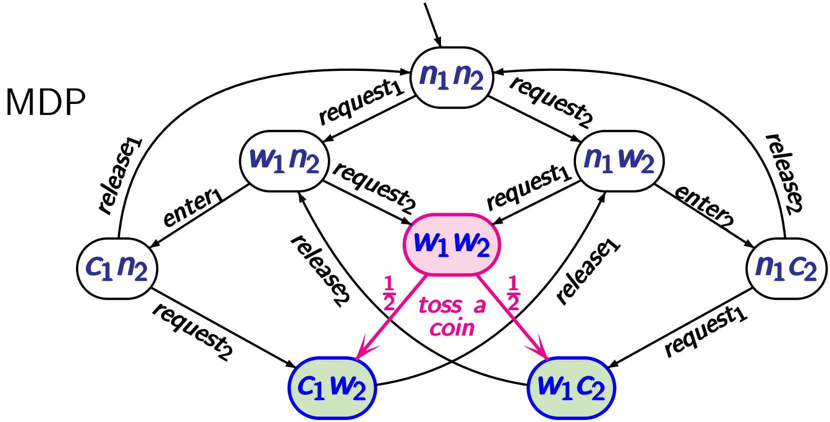
- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

MDP



Randomized mutual exclusion protocol

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting



Reasoning about probabilities in MDP

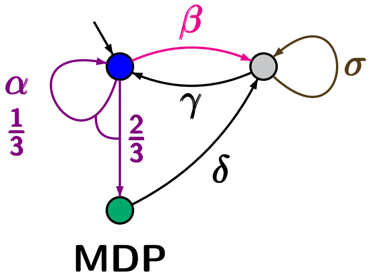
- requires resolving the nondeterminism by schedulers

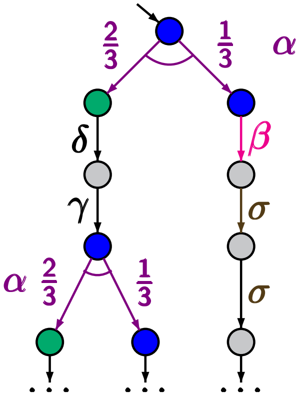
Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n

Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain





Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \ldots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain

↑

yields a notion of probability measure $\mathbf{Pr}^D$ on measurable sets of infinite paths

Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \ldots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain

$\uparrow$

yields a notion of probability measure Pr^D on measurable sets of infinite paths

typical task: given a measurable path event E,

∗ check whether E holds almost surely, i.e., $\mathrm{Pr}^D(E) = 1$ for all schedulers D

Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain

↑

yields a notion of probability measure $\Pr^D$ on measurable sets of infinite paths

typical task: given a measurable path event E,

- ∗ check whether E holds almost surely
- ∗ compute the worst-case probability for E, i.e.,

$$\sup_D \Pr^D(E) \quad \text{or} \quad \inf_D \Pr^D(E)$$

Quantitative analysis of MDP

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ with initial state s_0

ω-regular path event E,

e.g., given by an LTL formula

task: compute $\Pr^{\mathcal{M}}_{\max}(s_0, E) = \sup_{D} \Pr^{D}(s_0, E)$

Quantitative analysis of MDP

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ with initial state s_0

ω-regular path event E,

e.g., given by an LTL formula

task: compute $\Pr^{\mathcal{M}}_{\max}(s_0, E) = \sup_{D} \Pr^{D}(s_0, E)$

method: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, E)$ for all $s \in S$

via graph analysis and linear program

[Vardi/Wolper'86]
[Courcoubetis/Yannakakis'88]
[Bianco/de Alfaro'95]
[Baier/Kwiatkowska'98]

probabilistic system

“bad behaviors”

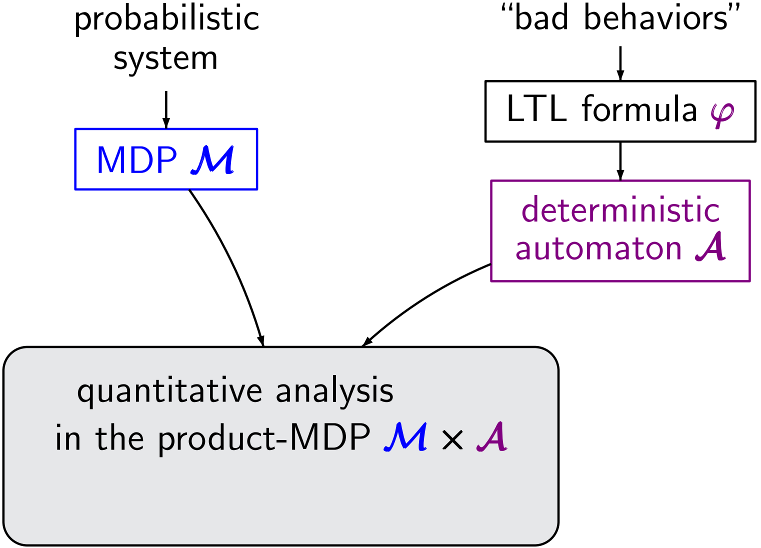
probabilistic system

↓

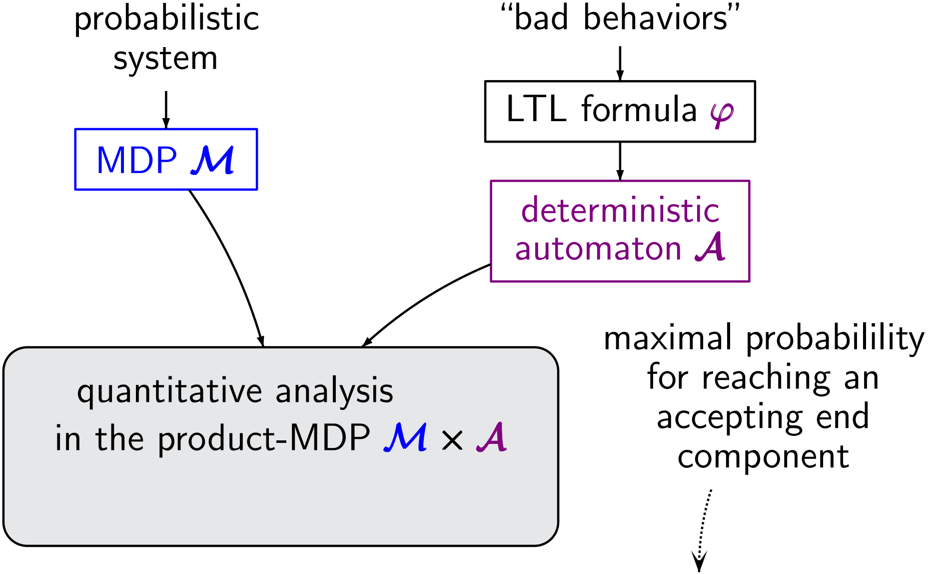
MDP $\mathcal{M}$

"bad behaviors"

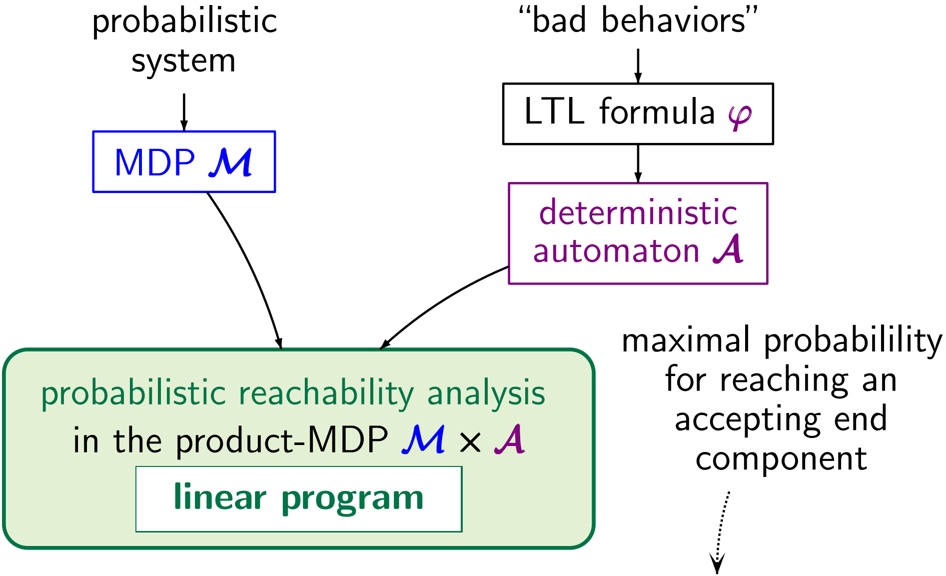

probabilistic system
MDP $\mathcal{M}$
"bad behaviors"
LTL formula φ
deterministic automaton $\mathcal{A}$



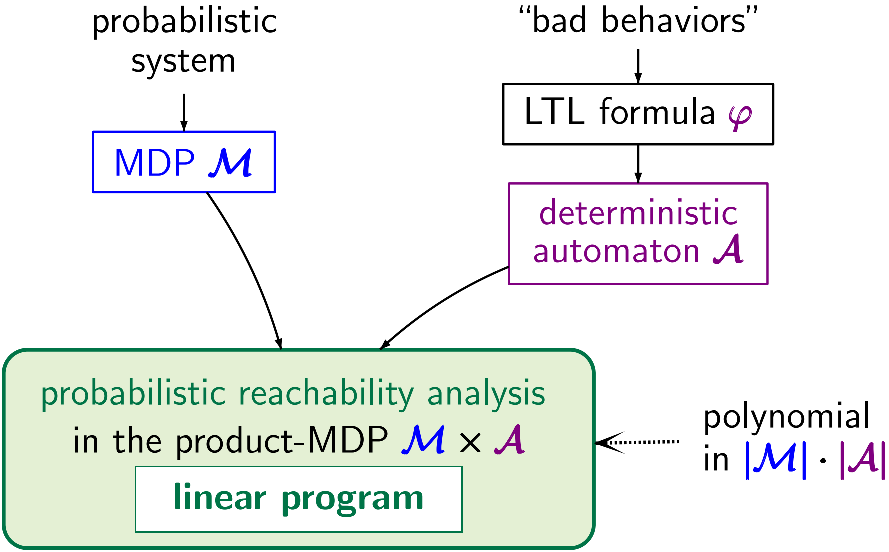
$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\Big(\langle s, init_s\rangle, \text{ acceptance cond. of } \mathcal{A}\Big)$$



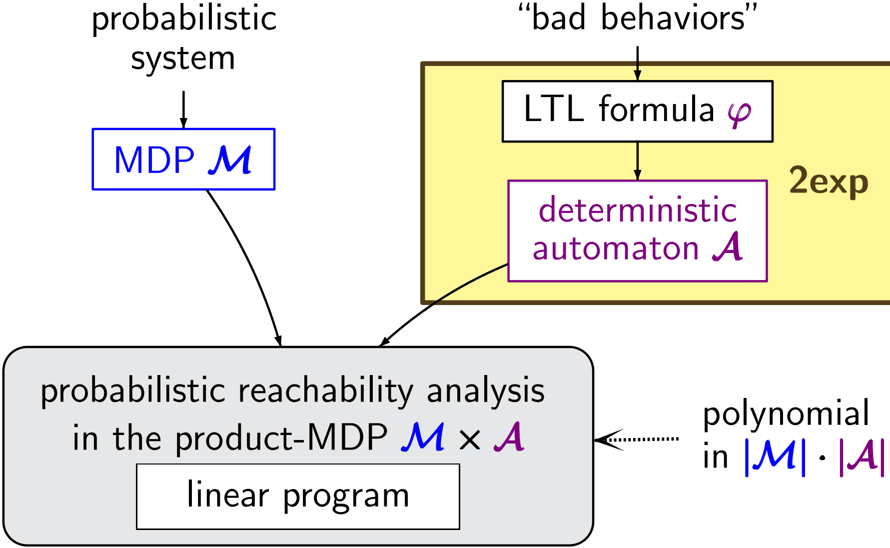
$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}(\langle s, init_s \rangle, \Diamond accEC)$$



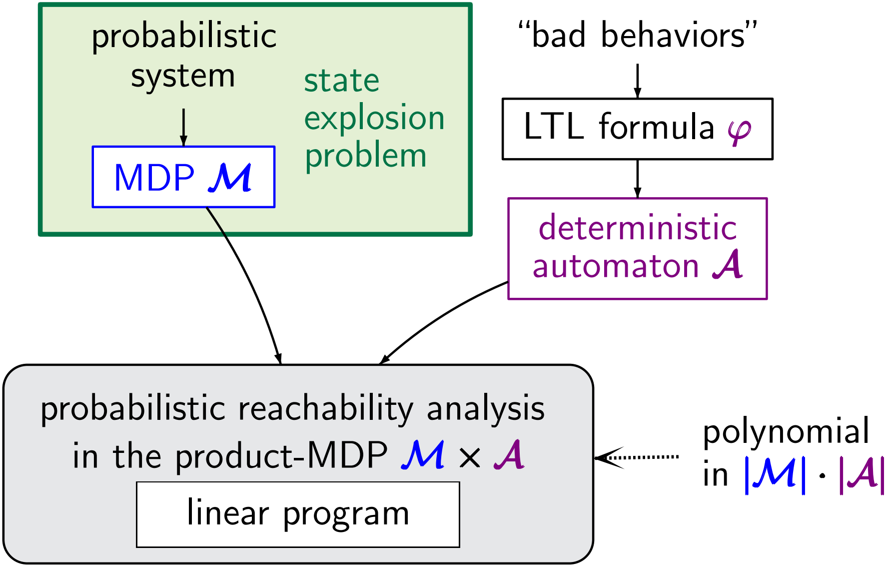
$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}(\langle s, init_s \rangle, \Diamond accEC\ )$$



$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}(\langle s, init_s \rangle, \Diamond accEC\ )$$



$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}(\langle s, init_s \rangle, \Diamond accEC \;)$$



$$\Pr_{\max}^{\mathcal{M}}(s, \varphi) \;=\; \Pr_{\max}^{\mathcal{M}\times\mathcal{A}}(\langle s, init_s \rangle, \Diamond accEC)$$

Advanced techniques for PMC

- symbolic model checking with variants of BDDs
  e.g., in PRISM [Kwiatkowska/Norman/Parker]
  ProbVerus [Hartonas-Garmhausen, Campos, Clarke]
- state aggregation with bisimulation
  e.g., in MRMC [Katoen et al]
- abstraction-refinement
  e.g., in RAPTURE [d'Argenio/Jeannet/Jensen/Larsen]
  PASS [Hermanns/Wachter/Zhang]
- partial order reduction
  e.g., in LiQuor [Baier/Ciesinski/Größer]

Advanced techniques for PMC

- symbolic model checking with variants of BDDs
  e.g., in PRISM [Kwiatkowska/Norman/Parker]
  ProbVerus [Hartonas-Garmhausen, Campos, Clarke]

  randomized distributed algorithms,
  communication and multimedia protocols,
  power management, security, ...

- state aggregation with bisimulation
  e.g., in MRMC [Katoen et al]
- abstraction-refinement
  e.g., in RAPTURE [d'Argenio/Jeannet/Jensen/Larsen]
  PASS [Hermanns/Wachter/Zhang]
- partial order reduction
  e.g., in LiQuor [Baier/Ciesinski/Größer]

Advanced techniques for PMC

- symbolic model checking with variants of BDDs

  e.g., in PRISM [Kwiatkowska/Norman/Parker]

  ProbVerus [Hartonas-Garmhausen, Campos, Clarke]

  randomized distributed algorithms,
  communication and multimedia protocols,
  power management, security, ...

- state aggregation with bisimulation

  e.g., in MRMC [Katoen et al]

- abstraction-refinement

  e.g., in RAPTURE [d'Argenio/Jeannet/Jensen/Larsen]

  PASS [Hermanns/Wachter/Zhang]

- partial order reduction

  e.g., in LiQuor [Baier/Ciesinski/Größer]

Partial order reduction

technique for reducing the state space of concurrent systems [Godefroid,Peled,Valmari, ca. 1990]

- attempts to analyze a sub-system by identifying "redundant interleavings"
- explores representatives of paths that agree up to the order of independent actions

Partial order reduction

technique for reducing the state space of concurrent systems [Godefroid,Peled,Valmari, ca. 1990]

- attempts to analyze a sub-system by identifying "redundant interleavings"
- explores representatives of paths that agree up to the order of independent actions

e.g., $\underbrace{x := x + y}_{\text{action } \alpha} \parallel \underbrace{z := z + 3}_{\text{action } \beta}$

has the same effect as $\alpha;\beta$ or $\beta;\alpha$

Partial order reduction

technique for reducing the state space of concurrent systems [Godefroid,Peled,Valmari, ca. 1990]

- attempts to analyze a sub-system by identifying "redundant interleavings"
- explores representatives of paths that agree up to the order of independent actions

DFS-based on-the-fly generation of a reduced system

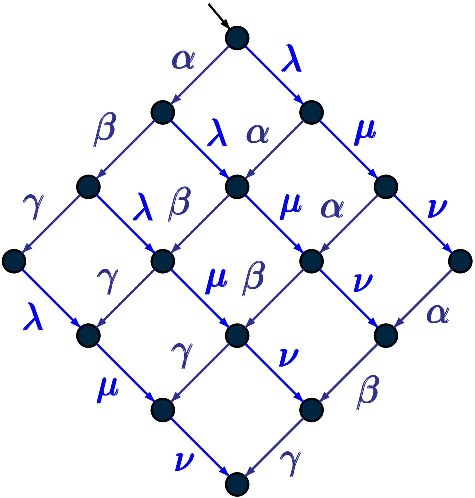
for each expanded state s

- choose an appropriate subset $Ample(s)$ of $Act(s)$
- expand only the α-successors of s for $\alpha \in Ample(s)$ (but ignore the actions in $Act(s) \setminus Ample(s)$)

Partial order reduction

concurrent execution of processes $\mathcal{P}_1$, $\mathcal{P}_2$

- no communication
- no competition



transition system for $\mathcal{P}_1 \| \mathcal{P}_2$ where

$$\mathcal{P}_1 = \alpha; \beta; \gamma$$
$$\mathcal{P}_2 = \lambda; \mu; \nu$$

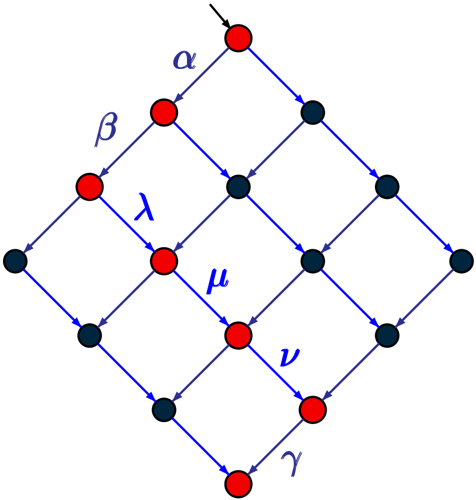
Partial order reduction

concurrent execution of processes $\mathcal{P}_1$, $\mathcal{P}_2$

- no communication
- no competition

transition system for $\mathcal{P}_1 \| \mathcal{P}_2$ where

$$\mathcal{P}_1 = \alpha; \beta; \gamma$$
$$\mathcal{P}_2 = \lambda; \mu; \nu$$



idea: explore just **1** path as representative for all paths

Ample-set method [Peled 1993]

given: processes $\mathcal{P}_i$ of a parallel system $\mathcal{P}_1 \| \ldots \| \mathcal{P}_n$
with transition system $\mathcal{T} = (S, Act, \rightarrow, \ldots)$

task: on-the-fly generation of a sub-system $\mathcal{T}_r$ s.t.

(A1) stutter condition ...
(A2) dependency condition ...
(A3) cycle condition ...

Ample-set method [Peled 1993]

given: processes $\mathcal{P}_i$ of a parallel system $\mathcal{P}_1 \| \ldots \| \mathcal{P}_n$
with transition system $\mathcal{T} = (S, Act, \rightarrow, \ldots)$

task: on-the-fly generation of a sub-system $\mathcal{T}_r$ s.t.

- (A1) stutter condition
- (A2) dependency condition
- (A3) cycle condition

} $\pi \rightsquigarrow \pi_r$ by permutations of independent actions

> Each path π in $\mathcal{T}$ is represented by an "equivalent" path π_r in $\mathcal{T}_r$

Ample-set method [Peled 1993]

given: processes $\mathcal{P}_i$ of a parallel system $\mathcal{P}_1 \| \ldots \| \mathcal{P}_n$
with transition system $\mathcal{T} = (S, Act, \rightarrow, \ldots)$

task: on-the-fly generation of a sub-system $\mathcal{T}_r$ s.t.

(A1) stutter condition
(A2) dependency condition
(A3) cycle condition

} $\pi \rightsquigarrow \pi_r$ by permutations of independent actions

Each path π in $\mathcal{T}$ is represented by an "equivalent" path π_r in $\mathcal{T}_r$

$\Downarrow$

$\mathcal{T}$ and $\mathcal{T}_r$ satisfy the same stutter-invariant events, e.g., next-free LTL formulas

Ample-set method for MDP

given: processes $\mathcal{P}_i$ of a probabilistic system $\mathcal{P}_1 \| \ldots \| \mathcal{P}_n$
with MDP-semantics $\mathcal{M} = (S, Act, P, \ldots)$

task: on-the-fly generation of a sub-MDP $\mathcal{M}_r$ s.t.

$\mathcal{M}_r$ and $\mathcal{M}$ have the same extremal probabilities
for stutter-invariant events

Ample-set method for MDP

given: processes $\mathcal{P}_i$ of a probabilistic system $\mathcal{P}_1 \| \ldots \| \mathcal{P}_n$ with MDP-semantics $\mathcal{M} = (S, Act, P, \ldots)$

task: on-the-fly generation of a sub-MDP $\mathcal{M}_r$ s.t.

For all schedulers D for $\mathcal{M}$ there is a scheduler D_r for $\mathcal{M}_r$ s.t. for all measurable, stutter-invariant events E:

$$\Pr^{D}_{\mathcal{M}}(E) = \Pr^{D_r}_{\mathcal{M}_r}(E)$$

$$\Downarrow$$

$\mathcal{M}_r$ and $\mathcal{M}$ have the same extremal probabilities for stutter-invariant events

Independence of actions

Independence of non-probabilistic actions

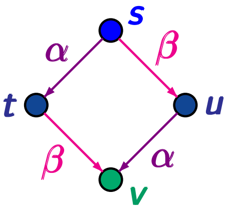
Actions α and β are called independent in a transition system $\mathcal{T}$ iff:

whenever $s \xrightarrow{\alpha} t$ and $s \xrightarrow{\beta} u$ then

(1) α is enabled in u

(2) β is enabled in t

(3) if $u \xrightarrow{\alpha} v$ and $t \xrightarrow{\beta} w$ then $v = w$



Independence of actions in an MDP

Let $\mathcal{M} = (S, Act, P, \ldots)$ be a MDP and $\alpha, \beta \in Act$.

α and β are independent in $\mathcal{M}$ if for each state s s.t. $\alpha, \beta \in Act(s)$:

(1) if $P(s, \alpha, t) > 0$ then $\beta \in Act(t)$

(2) if $P(s, \beta, u) > 0$ then $\alpha \in Act(u)$

(3) $\ldots$

Independence of actions in an MDP

Let $\mathcal{M} = (S, Act, P, \ldots)$ be a MDP and $\alpha, \beta \in Act$.

α and β are independent in $\mathcal{M}$ if for each state s s.t. $\alpha, \beta \in Act(s)$:

(1) if $P(s, \alpha, t) > 0$ then $\beta \in Act(t)$

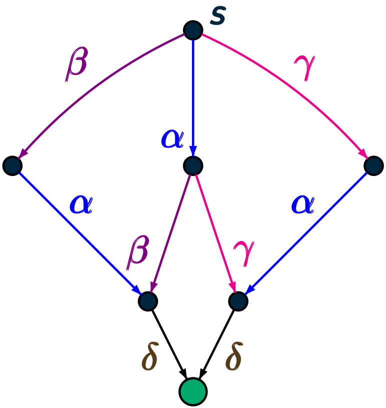
(2) if $P(s, \beta, u) > 0$ then $\alpha \in Act(u)$

(3) for all states w:

$$P(s, \alpha\beta, w) = P(s, \beta\alpha, w)$$

$$\sum_{t \in S} P(s, \alpha, t) \cdot P(t, \beta, w) \qquad \sum_{u \in S} P(s, \beta, u) \cdot P(u, \alpha, w)$$

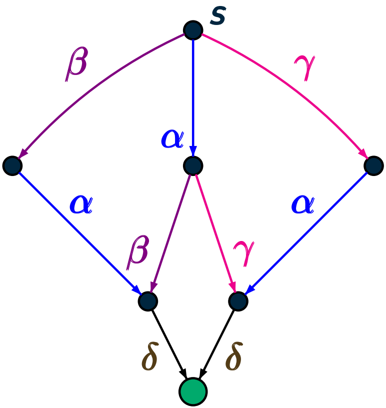
Example: ample set method



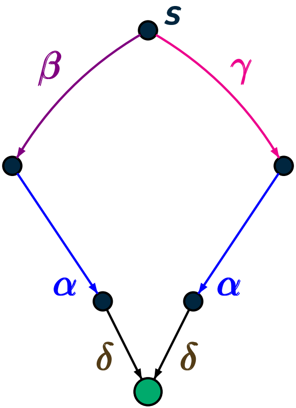
original system $\mathcal{T}$

α independent from β and γ

Example: ample set method



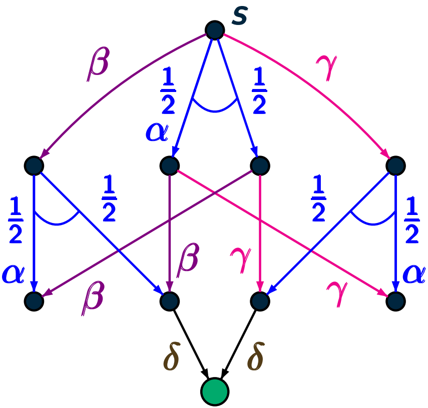
original system $\mathcal{T}$



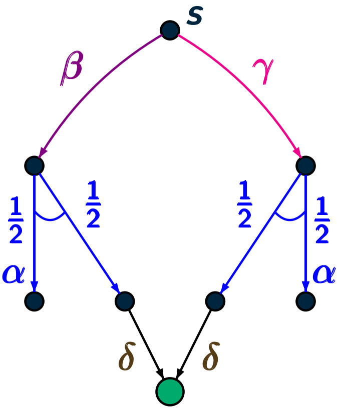
reduced system $\mathcal{T}_r$
(A1)-(A3) are fulfilled

α independent from β and γ

Example: ample set method fails for MDP



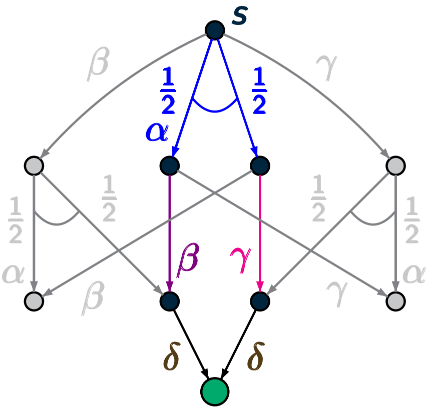
original MDP $\mathcal{M}$



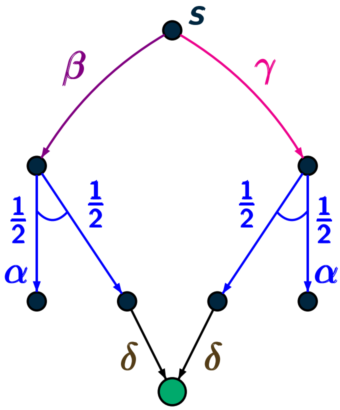
reduced MDP $\mathcal{M}_r$
(A1)-(A3) are fulfilled

α independent from β and γ

Example: ample set method fails for MDP



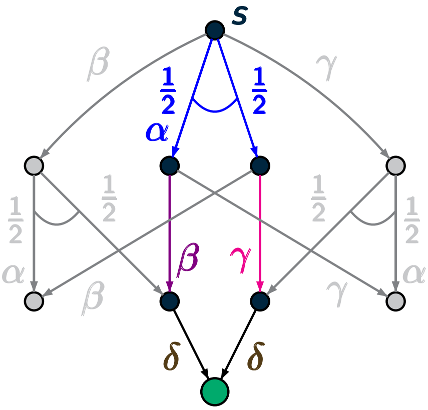
original MDP $\mathcal{M}$



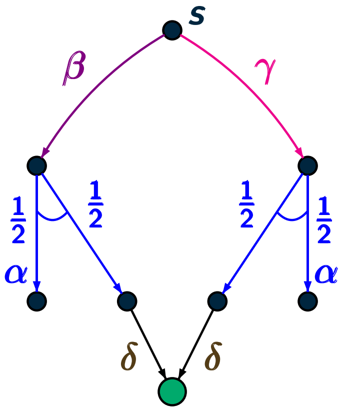
reduced MDP $\mathcal{M}_r$

$\Pr^{\mathcal{M}}_{\max}(s, \Diamond green) = 1$ $\qquad$ $\Diamond$ "eventually"

Example: ample set method fails for MDP



original MDP $\mathcal{M}$



reduced MDP $\mathcal{M}_r$

$$\Pr^{\mathcal{M}}_{\max}(s, \Diamond green) = 1 \; > \; \frac{1}{2} = \Pr^{\mathcal{M}_r}_{\max}(s, \Diamond green)$$

Partial order reduction for MDP

extend Peled's conditions (A1)-(A3) for the ample-sets

(A1) stutter condition ...
(A2) dependency condition ...
(A3) cycle condition ...
(A4) probabilistic condition

> If there is a path $s \xrightarrow{\beta_1} \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_n} \xrightarrow{\alpha}$ in $\mathcal{M}$ s.t. $\beta_1, \ldots, \beta_n, \alpha \notin Ample(s)$ and α is probabilistic then $|Ample(s)| = 1$.

Partial order reduction for MDP

extend Peled's conditions (A1)-(A3) for the ample-sets

(A1) stutter condition ...

(A2) dependency condition ...

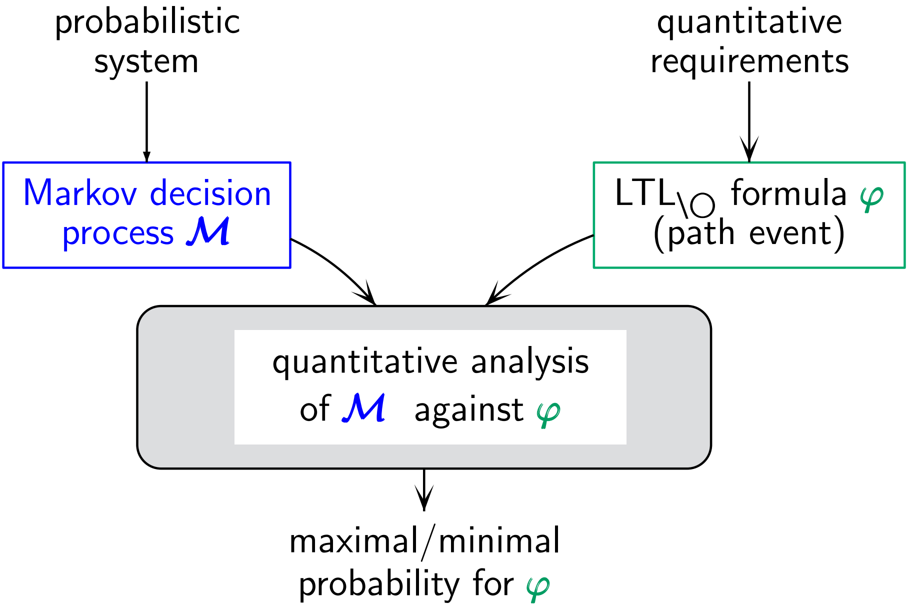
(A3) cycle condition ...

(A4) probabilistic condition

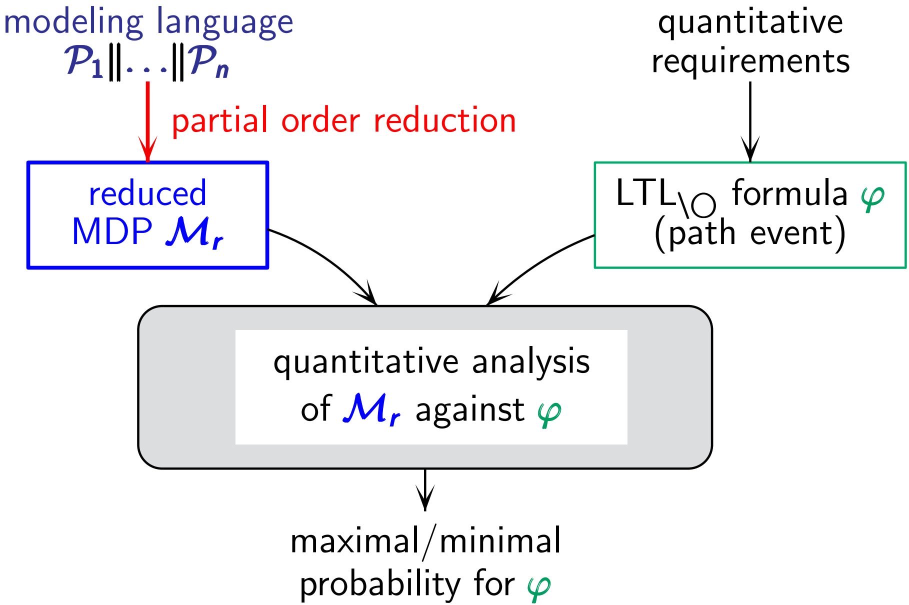
If there is a path $s \xrightarrow{\beta_1}\xrightarrow{\beta_2} \ldots \xrightarrow{\beta_n}\xrightarrow{\alpha}$ in $\mathcal{M}$ s.t.
$\beta_1, \ldots, \beta_n, \alpha \notin Ample(s)$ and α is probabilistic
then $|Ample(s)| = 1$.

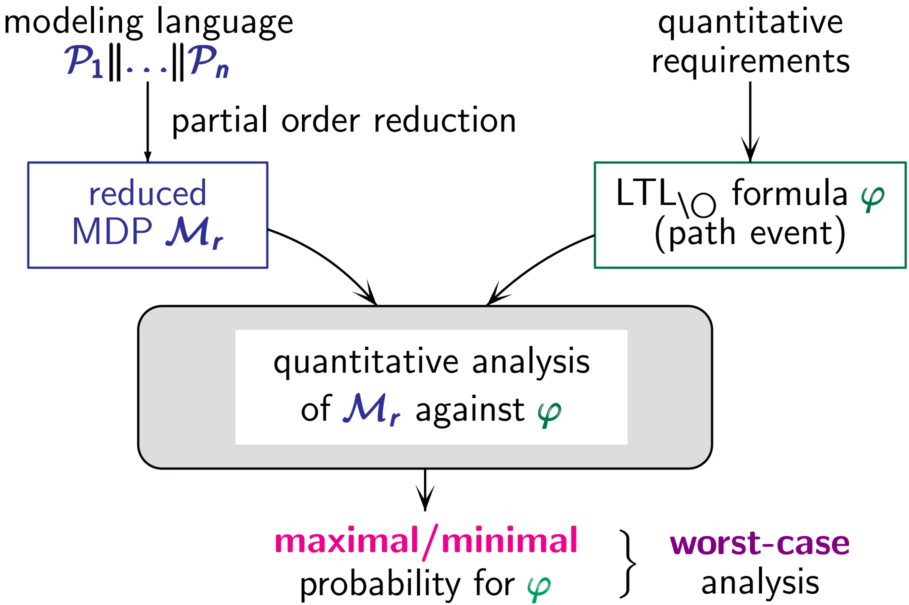
If (A1)-(A4) hold then $\mathcal{M}$ and $\mathcal{M}_r$ have the same extremal probabilities for all stutter-invariant properties.

Probabilistic model checking



Probabilistic model checking, e.g., LiQuor



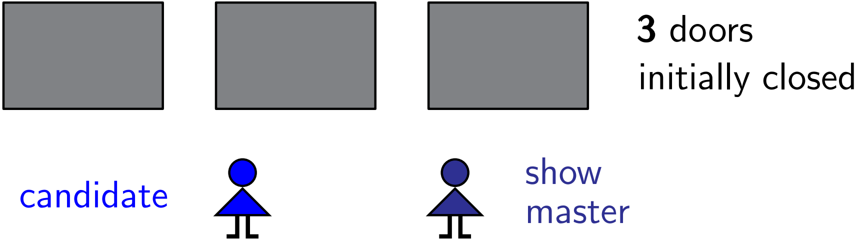


modeling language
$\mathcal{P}_1 \| \ldots \| \mathcal{P}_n$
partial order reduction
reduced MDP $\mathcal{M}_r$
quantitative requirements
$LTL_{\setminus\bigcirc}$ formula φ (path event)
quantitative analysis of $\mathcal{M}_r$ against φ
maximal/minimal probability for φ
worst-case analysis

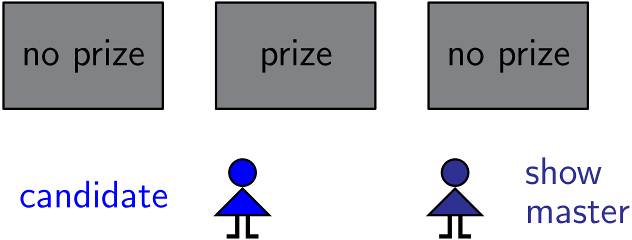
Outline

- Markov decision processes (MDP) and quantitative analysis against path events
- partial order reduction for MDP
- partially-oberservable MDP ⟵
- conclusions

Monty-Hall problem

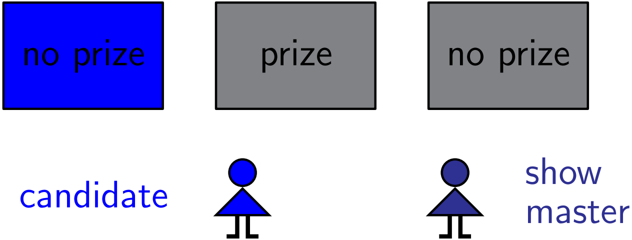


Monty-Hall problem



3 doors
initially closed

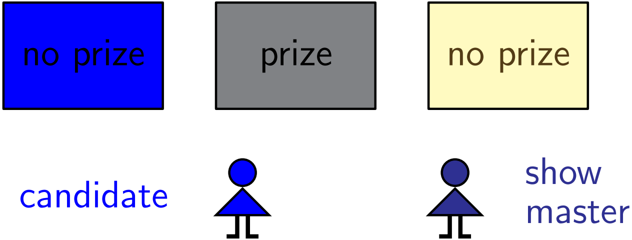
Monty-Hall problem



3 doors
initially closed

1. candidate chooses one of the doors

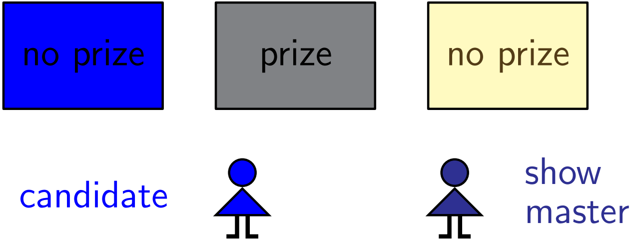
Monty-Hall problem



3 doors
initially closed

1. candidate chooses one of the doors
2. show master opens a non-chosen, non-winning door

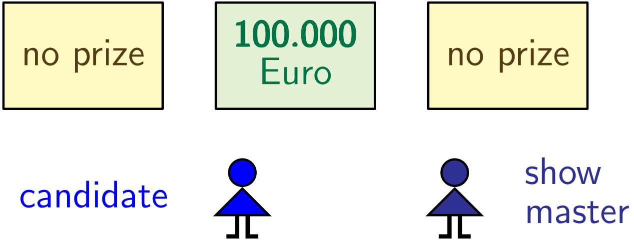
Monty-Hall problem



3 doors
initially closed

1. candidate chooses one of the doors
2. show master opens a non-chosen, non-winning door
3. candidate has the choice:
   - keep the choice or
   - switch to the other (still closed) door

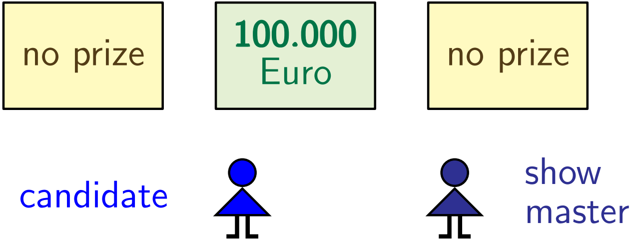
Monty-Hall problem



3 doors
initially closed

1. candidate chooses one of the doors
2. show master opens a non-chosen, non-winning door
3. candidate has the choice:
   - keep the choice   or
   - switch to the other (still closed) door
4. show master opens all doors

Monty-Hall problem



3 doors
initially closed

optimal strategy for the candidate:

initial choice of the door: arbitrary
revision of the initial choice (switch)

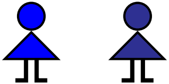
probability for getting the prize: $\frac{2}{3}$

MDP for the Monty-Hall problem

3 doors
initially closed

candidate's actions

1. choose one door
3. keep or switch **?**



show master's actions

2. opens a non-chosen, non-winning door
4. opens all doors

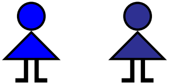
MDP for the Monty-Hall problem

no prize | prize | no prize

3 doors
initially closed

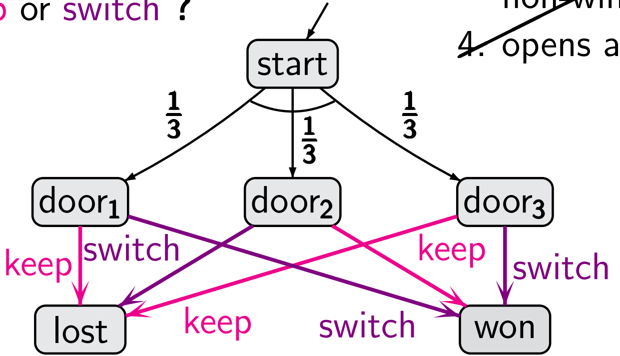
candidate's actions

1. choose one door
3. keep or switch **?**



show master's actions

2. opens a non-chosen, non-winning door
4. opens all doors

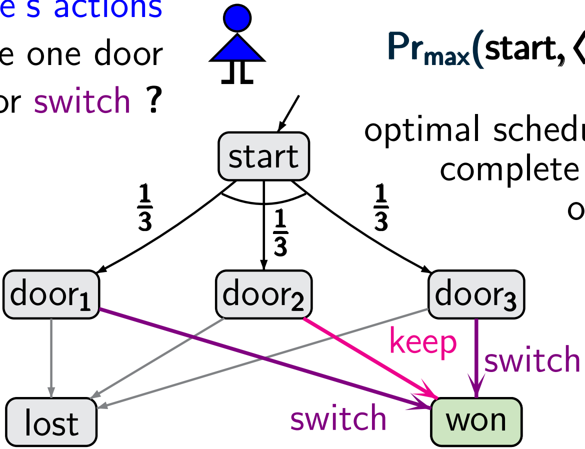


MDP for the Monty-Hall problem

3 doors
initially closed

candidate's actions

1. choose one door
3. keep or switch **?**



$\Pr_{\max}(\text{start}, \Diamond \textit{won}) = 1$

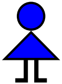
optimal scheduler requires complete information on the states

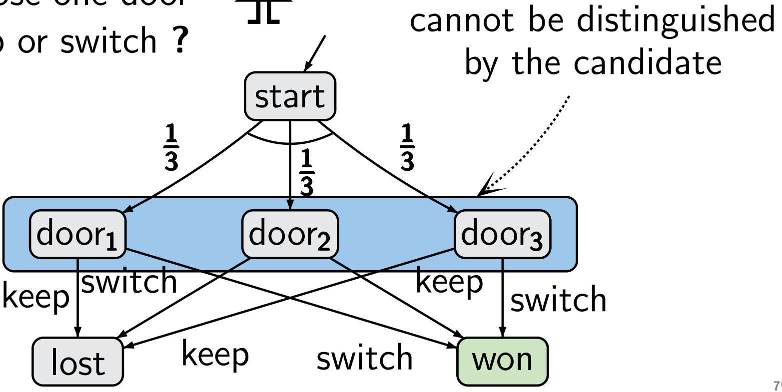
MDP for the Monty-Hall problem

3 doors
initially closed

candidate's actions

1. choose one door
3. keep or switch **?**



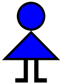


MDP for the Monty-Hall problem

3 doors
initially closed

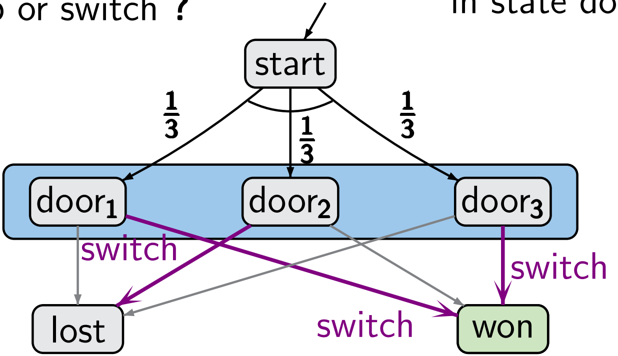
candidate's actions

1. choose one door
3. keep or switch **?**



observation-based strategy:

choose action switch
in state $door_i$

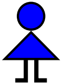


MDP for the Monty-Hall problem

3 doors
initially closed

candidate's actions

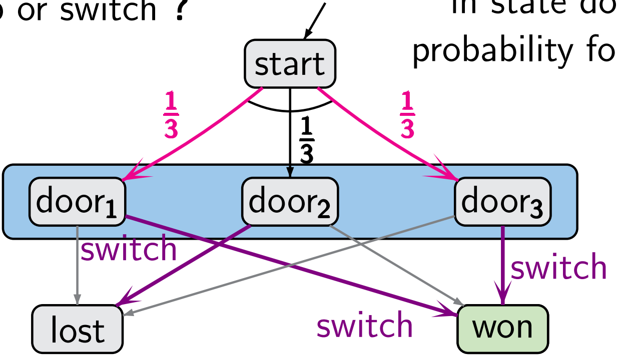
1. choose one door
3. keep or switch **?**



observation-based strategy:

choose action switch
in state door$_i$

probability for $\Diamond$*won*: $\frac{2}{3}$



Partially-observable Markov decision process

A partially-observable MDP (POMDP for short) is an MDP $\mathcal{M} = (S, Act, P, \ldots)$ together with an equivalence relation $\sim$ on S

Partially-observable Markov decision process

A partially-observable MDP (POMDP for short) is an MDP $\mathcal{M} = (S, Act, P, \ldots)$ together with an equivalence relation $\sim$ on S

↑

if $s_1 \sim s_2$ then s_1, s_2 cannot be distinguished from outside (or by the scheduler)

observables: equivalence classes of states

Partially-observable Markov decision process

A partially-observable MDP (POMDP for short) is an MDP $\mathcal{M} = (S, Act, P, \ldots)$ together with an equivalence relation $\sim$ on S

↑

if $s_1 \sim s_2$ then s_1, s_2 cannot be distinguished from outside (or by the scheduler)

observables: equivalence classes of states

observation-based scheduler:

scheduler $D : S^* \to Act$ such that for all $\pi_1, \pi_2 \in S^*$:

$$D(\pi_1) = D(\pi_2) \quad \text{if} \quad obs(\pi_1) = obs(\pi_2)$$

where $obs(s_0\, s_1 \ldots s_n) = [s_0]\,[s_1] \ldots [s_n]$

Extreme cases of POMDP

extreme cases of an POMDP:

- $s_1 \sim s_2$ iff $s_1 = s_2$
- $s_1 \sim s_2$ for all s_1, s_2

Extreme cases of POMDP

extreme cases of an POMDP:

- $s_1 \sim s_2$ iff $s_1 = s_2$ $\longleftarrow$ standard MDP
- $s_1 \sim s_2$ for all s_1, s_2

Probabilistic automata are special POMDP

extreme cases of an POMDP:

- $s_1 \sim s_2$ iff $s_1 = s_2$ $\longleftarrow$ standard MDP
- $s_1 \sim s_2$ for all s_1, s_2 $\longleftarrow$ probabilistic automata

note that for totally non-observable POMDP:

observation-based scheduler	$\widehat{=}$	function $D : \mathbb{N} \to Act$	$\widehat{=}$	infinite word over Act

Undecidability results for POMDP

extreme cases of an POMDP:

- $s_1 \sim s_2$ iff $s_1 = s_2$ ⟵ standard MDP
- $s_1 \sim s_2$ for all s_1, s_2 ⟵ probabilistic automata

note that for totally non-observable POMDP:

observation-based scheduler $\widehat{=}$ function $D : \mathbb{N} \to Act$ $\widehat{=}$ infinite word over Act

undecidability results for PFA carry over to POMDP

maximum probabilistic reachability problem "does $\mathrm{Pr}^{obs}_{\max}(\Diamond F) > p$ hold ?" $\widehat{=}$ non-emptiness problem for **PFA**

Undecidability results for POMDP

- The model checking problem for POMDP and quantitative properties is undecidable, e.g., probabilistic reachability properties.

Undecidability results for POMDP

- The model checking problem for POMDP and quantitative properties is undecidable, e.g., probabilistic reachability properties.
- There is no even no approximation algorithm for reachability objectives.

[Paz'71], [Madani/Hanks/Condon'99], [Giro/d'Argenio'07]

Undecidability results for POMDP

- The model checking problem for POMDP and quantitative properties is undecidable, e.g., probabilistic reachability properties.
- There is no even no approximation algorithm for reachability objectives.
- The model checking problem for POMDP and several qualitative properties is undecidable, e.g.,

> repeated reachability with positive probability
>
> “does $\mathrm{Pr}^{obs}_{\max}(\Box\Diamond F) > \mathbf{0}$ hold **?**”

$\Box\Diamond \mathrel{\widehat{=}}$ “infinitely often”

Undecidability results for POMDP

- The model checking problem for POMDP and quantitative properties is undecidable, e.g., probabilistic reachability properties.
- There is no even no approximation algorithm for reachability objectives.
- The model checking problem for POMDP and several qualitative properties is undecidable, e.g.,

repeated reachability with positive probability

"does $\mathbf{Pr}^{obs}_{\max}(\Box\Diamond F) > \mathbf{0}$ hold **?**"

Many interesting verification problems for distributed probabilistic multi-agent systems are undecidable.

Undecidability results for POMDP

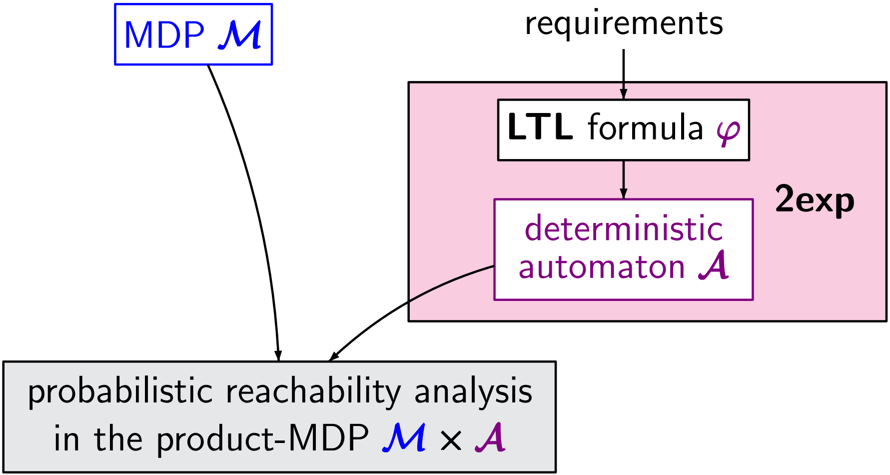
- The model checking problem for POMDP and quantitative properties is undecidable, e.g., probabilistic reachability properties.
- There is no even no approximation algorithm for reachability objectives.
- The model checking problem for POMDP and several qualitative properties is undecidable, e.g.,

repeated reachability with positive probability

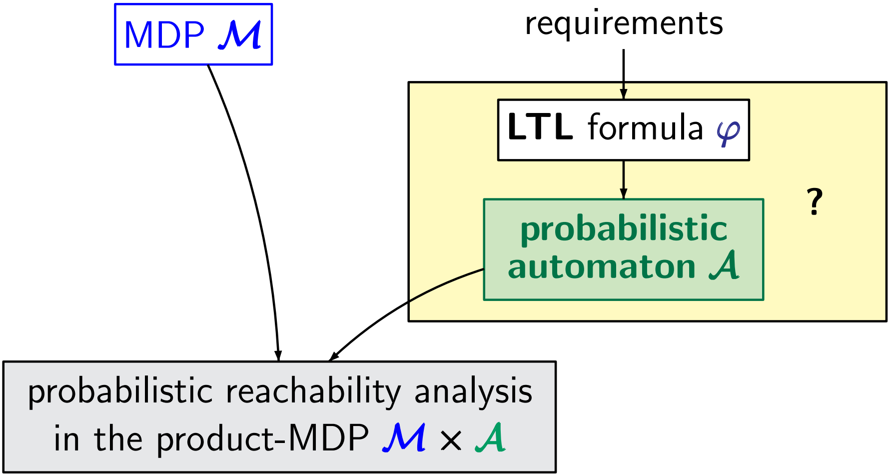
"does $\mathrm{Pr}^{obs}_{\max}(\Box\Diamond F) > 0$ hold ?"

... already holds for $\underbrace{\text{totally non-observable POMDP}}_{\text{probabilistic Büchi automata}}$

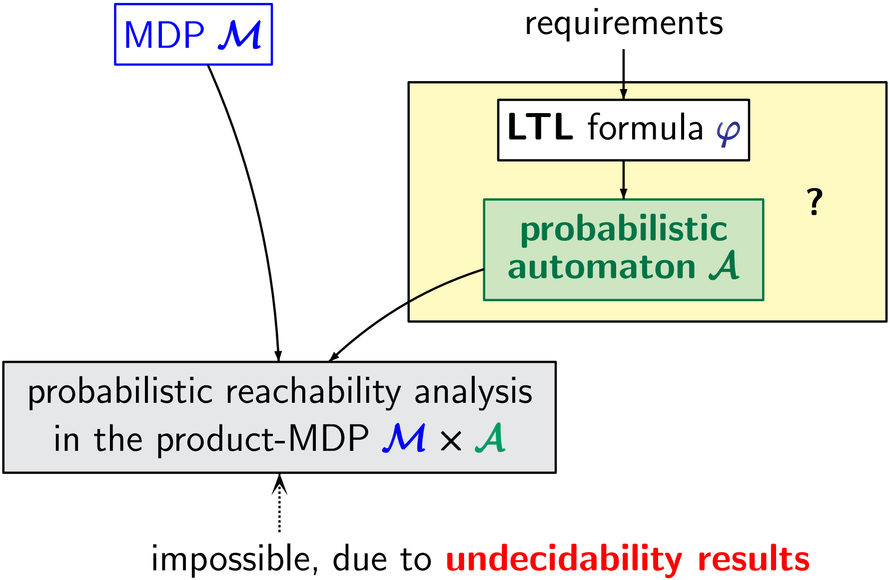
Remind: LTL model checking for MDP



PA rather than DA ?



PA rather than DA ?



Decidability results for POMDP

Decidability results for POMDP

The model checking problem for POMDP and several qualitative properties is decidable, e.g.,

- invariance with positive probability
  "does $\Pr^{obs}_{\max}(\Box F) > 0$ hold ?"
- almost-sure reachability
  "does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"
- almost-sure repeated reachability
  "does $\Pr^{obs}_{\max}(\Box\Diamond F) = 1$ hold ?"
- persistence with positive probability
  "does $\Pr^{obs}_{\max}(\Diamond\Box F) > 0$ hold ?"

The model checking problem for POMDP and several qualitative properties is decidable, e.g.,

- invariance with positive probability

  "does $\Pr^{obs}_{\max}(\Box F) > 0$ hold ?"
- almost-sure reachability

  "does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"
- almost-sure repeated reachability

  "does $\Pr^{obs}_{\max}(\Box\Diamond F) = 1$ hold ?"
- persistence with positive probability

  "does $\Pr^{obs}_{\max}(\Diamond\Box F) > 0$ hold ?"

algorithms use a certain powerset construction

- Markov decision processes (MDP) and quantitative analysis against path events
- partial order reduction for MDP
- partially-oberservable MDP
- conclusions ⟵

Conclusion

- worst/best-case analysis of MDP solvable by
  - numerical methods for solving linear programs
  - known techniques for non-probabilistic systems

↑

graph algorithms, LTL-2-AUT translators, ...
techniques to combat the state explosion problem
(such as partial order reduction)

Conclusion

- worst/best-case analysis of MDP solvable by
  - numerical methods for solving linear programs
  - known techniques for non-probabilistic systems

↑

graph algorithms, LTL-2-AUT translators, ...
techniques to combat the state explosion problem
(such as partial order reduction)

but: strongly simplified definition of schedulers

↑

assumption "full knowledge of the history" is
inadequate, e.g., for agents of distributed systems

Conclusion

- worst/best-case analysis of MDP solvable by
  - numerical methods for solving linear programs
  - known techniques for non-probabilistic systems
- more realistic model: partially-observable MDP
  and multi-agents variants with distributed schedulers

Conclusion

- worst/best-case analysis of MDP solvable by
  - numerical methods for solving linear programs
  - known techniques for non-probabilistic systems
- more realistic model: partially-observable MDP
  and multi-agents variants with distributed schedulers
  - many algorithms for "finite-horizon properties"
  - few decidability results for qualitative properties
  - undecidability for quantitative properties and, e.g.,
    repeated reachability with positive probability

Conclusion

- worst/best-case analysis of MDP solvable by
  - ∗ numerical methods for solving linear programs
  - ∗ known techniques for non-probabilistic systems

- more realistic model: partially-observable MDP and multi-agents variants with distributed schedulers
  - many algorithms for "finite-horizon properties"
  - few decidability results for qualitative properties
  - undecidability for quantitative properties and, e.g., repeated reachability with positive probability

↑ proof via probabilistic language acceptors (PFA/PBA)

Conclusion

- worst/best-case analysis of MDP solvable by
  - ∗ numerical methods for solving linear programs
  - ∗ known techniques for non-probabilistic systems
- more realistic model: partially-observable MDP
  and multi-agents variants with distributed schedulers
  - many algorithms for "finite-horizon properties"
  - few decidability results for qualitative properties
  - undecidability for quantitative properties and, e.g.,
    repeated reachability with positive probability
- probabilistic Büchi automata interesting in their own ...

Probabilistic Büchi automaton (PBA)

$\mathcal{P} = (Q, \Sigma, \delta, \mu, F)$

- Q finite state space
- Σ alphabet
- $\delta : Q \times \Sigma \times Q \to [0,1]$ s.t. for all $q \in Q$, $a \in \Sigma$:

$$\sum_{p \in Q} \delta(q, a, p) \in \{0, 1\}$$

- initial distribution μ
- set of final states $F \subseteq Q$

Probabilistic Büchi automaton (PBA)

$\mathcal{P} = (Q, \Sigma, \delta, \mu, F)$ $\longleftarrow$ POMDP where $\Sigma = Act$ and $\sim\ \widehat{=}\ Q \times Q$

- Q finite state space
- Σ alphabet
- $\delta : Q \times \Sigma \times Q \to [0,1]$ s.t. for all $q \in Q$, $a \in \Sigma$:

$$\sum_{p \in Q} \delta(q, a, p) \in \{0, 1\}$$

- initial distribution μ
- set of final states $F \subseteq Q$

$\mathcal{P} = (Q, \Sigma, \delta, \mu, F)$ ⟵ POMDP where $\Sigma = Act$ and $\sim \,\widehat{=}\, Q \times Q$

- Q finite state space
- Σ alphabet
- $\delta : Q \times \Sigma \times Q \to [0,1]$ s.t. for all $q \in Q$, $a \in \Sigma$:
  $$\sum_{p \in Q} \delta(q, a, p) \in \{0, 1\}$$
- initial distribution μ
- set of final states $F \subseteq Q$

For each infinite word $x \in \Sigma^\omega$:

$\Pr(x) =$ probability for the accepting runs for x

↑ accepting run: visits F infinitely often

Probabilistic Büchi automaton (PBA)

$\mathcal{P} = (Q, \Sigma, \delta, \mu, F)$ ⟵ POMDP where $\Sigma = Act$ and $\sim \widehat{=} Q \times Q$

- Q finite state space
- Σ alphabet
- $\delta : Q \times \Sigma \times Q \to [0,1]$ s.t. for all $q \in Q$, $a \in \Sigma$:
  $$\sum_{p \in Q} \delta(q, a, p) \in \{0, 1\}$$
- initial distribution μ
- set of final states $F \subseteq Q$

For each infinite word $x \in \Sigma^\omega$:

$\Pr(x) =$ probability for the accepting runs for x

↑

probability measure in the infinite Markov chain induced by x viewed as a scheduler

Accepted language of a PBA

$\mathcal{P} = (Q, \Sigma, \delta, \mu, F)$

- Q finite state space, Σ alphabet
- $\delta : Q \times \Sigma \times Q \to [0,1]$ s.t. ...
- initial distribution μ
- set of final states $F \subseteq Q$

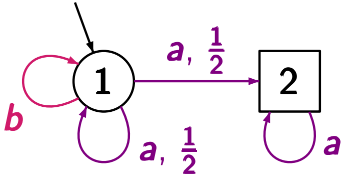
three types of accepted language:

$\mathcal{L}^{>0}(\mathcal{P}) = \{x \in \Sigma^\omega : \Pr(x) > 0\}$ probable semantics

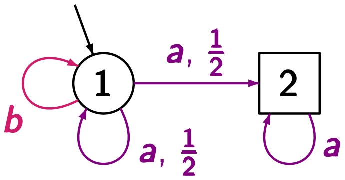
$\mathcal{L}^{=1}(\mathcal{P}) = \{x \in \Sigma^\omega : \Pr(x) = 1\}$ almost-sure sem.

$\mathcal{L}^{>\lambda}(\mathcal{P}) = \{x \in \Sigma^\omega : \Pr(x) > \lambda\}$ threshold semantics where $0 < \lambda < 1$

Example for PBA



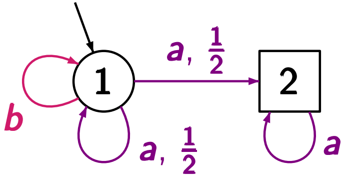
Example for PBA



accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) \;=\; (a+b)^* a^\omega$$

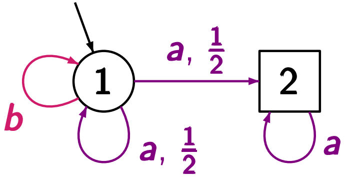
Example for PBA



accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (a + b)^* a^\omega$$
$$\mathcal{L}^{=1}(\mathcal{P}) = b^* a^\omega$$

Example for PBA



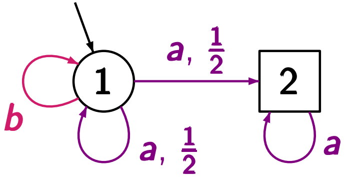
accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (a+b)^* a^\omega$$

$$\mathcal{L}^{=1}(\mathcal{P}) = b^* a^\omega$$

Thus: **PBA$^{>0}$** are strictly more expressive than **DBA**

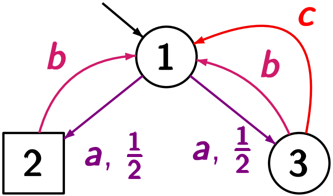
Example for PBA



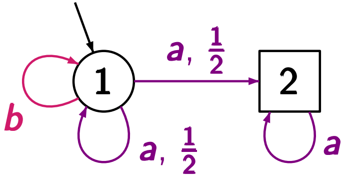
accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (a+b)^* a^\omega$$
$$\mathcal{L}^{=1}(\mathcal{P}) = b^* a^\omega$$

Thus: **PBA$^{>0}$** are strictly more expressive than **DBA**



Example for PBA

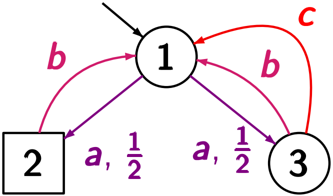


accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (a+b)^* a^\omega$$

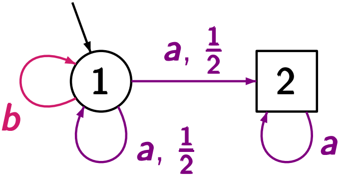
$$\mathcal{L}^{=1}(\mathcal{P}) = b^* a^\omega$$

Thus: **PBA**$^{>0}$ are strictly more expressive than **DBA**



NBA accepts $((ac)^* ab)^\omega$

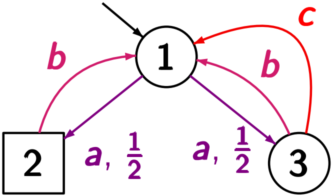
Example for PBA



accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (a+b)^* a^\omega$$
$$\mathcal{L}^{=1}(\mathcal{P}) = b^* a^\omega$$

Thus: **PBA**$^{>0}$ are strictly more expressive than **DBA**

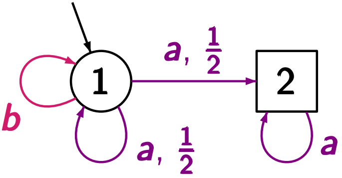


accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (ab+ac)^*(ab)^\omega$$

but **NBA** accepts $((ac)^* ab)^\omega$

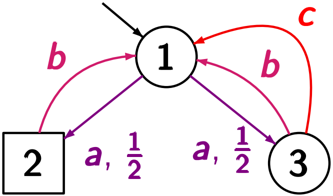
Example for PBA



accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (a+b)^* a^\omega$$
$$\mathcal{L}^{=1}(\mathcal{P}) = b^* a^\omega$$

Thus: **PBA**$^{>0}$ are strictly more expressive than **DBA**



accepted language:

$$\mathcal{L}^{>0}(\mathcal{P}) = (ab+ac)^*(ab)^\omega$$
$$\mathcal{L}^{=1}(\mathcal{P}) = (ab)^\omega$$

but **NBA** accepts $((ac)^* ab)^\omega$

Expressiveness of PBA with probable semantics

Expressiveness of PBA with probable semantics

PBA$^{>0}$ are strictly more expressive than **NBA**

Expressiveness of PBA with probable semantics

$PBA^{>0}$ are strictly more expressive than NBA

from NBA to PBA:

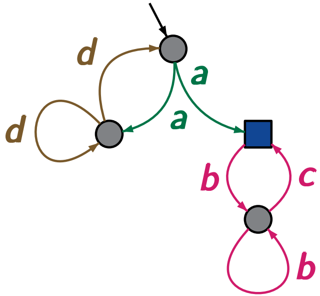
NBA —— Courcoubetis/ Yannakakis ——→ **NBA** deterministic in limit $\widehat{=}$ **$PBA^{>0}$**

Expressiveness of PBA with probable semantics

$\mathbf{PBA}^{>0}$ are strictly more expressive than **NBA**

from NBA to PBA:

NBA ——Courcoubetis/ Yannakakis——→ **NBA** deterministic in limit $\widehat{=}$ **$\mathbf{PBA}^{>0}$**

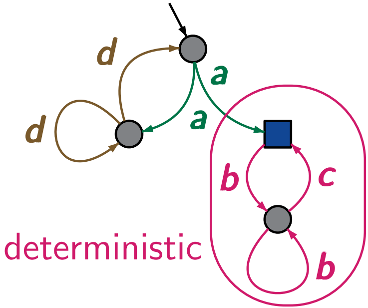


Expressiveness of PBA with probable semantics

$\text{PBA}^{>0}$ are strictly more expressive than NBA

from NBA to PBA:

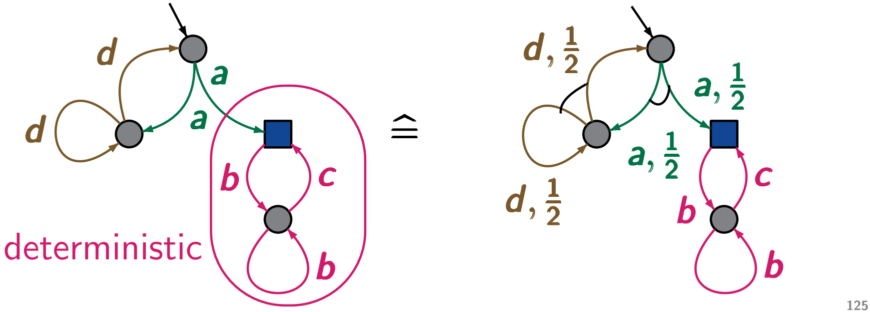
NBA ⟶ (Courcoubetis/Yannakakis) NBA deterministic in limit $\widehat{=}$ $\text{PBA}^{>0}$



$\text{PBA}^{>0}$ are strictly more expressive than NBA

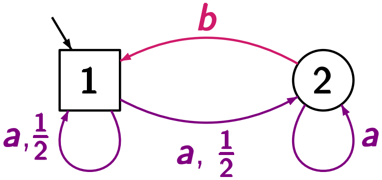
from NBA to PBA:

NBA ⟶ (Courcoubetis/ Yannakakis) NBA deterministic in limit $\hat{=}$ $\text{PBA}^{>0}$



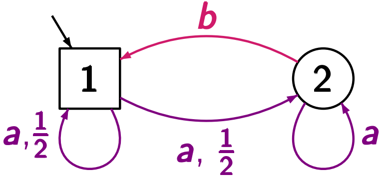
PBA$^{>0}$ are strictly more expressive than NBA

- from NBA to PBA:
  via NBA that are deterministic in limit
- PBA can accept non-ω-regular languages



PBA$^{>0}$ are strictly more expressive than NBA

- from NBA to PBA:
  via NBA that are deterministic in limit
- PBA can accept non-ω-regular languages

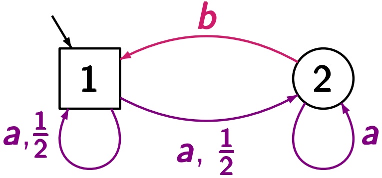


accepted language (probable semantics):

$$\mathcal{L}^{>0}(\mathcal{P}) = \left\{ a^{k_1} b a^{k_2} b a^{k_3} b \ldots \mid \quad \ldots \quad \right\}$$

PBA$^{>0}$ are strictly more expressive than NBA

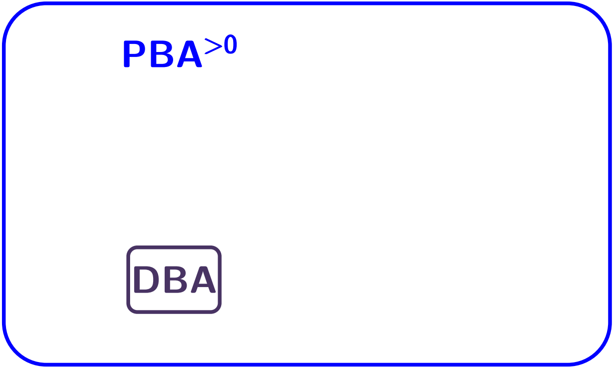
- from NBA to PBA:
  via NBA that are deterministic in limit
- PBA can accept non-ω-regular languages



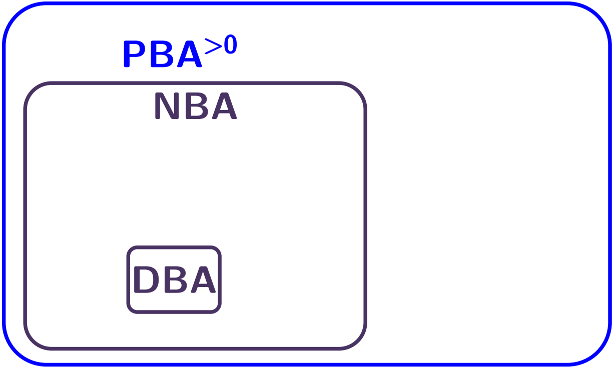
accepted language (probable semantics):

$$\mathcal{L}^{>0}(\mathcal{P}) = \left\{ a^{k_1} b a^{k_2} b a^{k_3} b \ldots \mid \prod_{i=1}^{\infty} \left(1 - \left(\tfrac{1}{2}\right)^{k_i}\right) > 0 \right\}$$

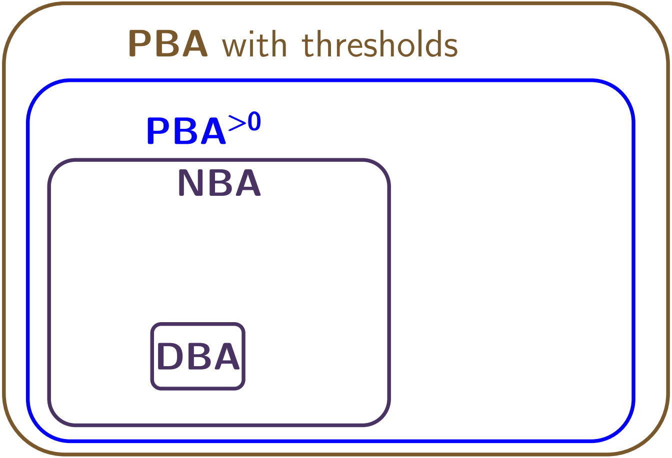
Expressiveness of PBA



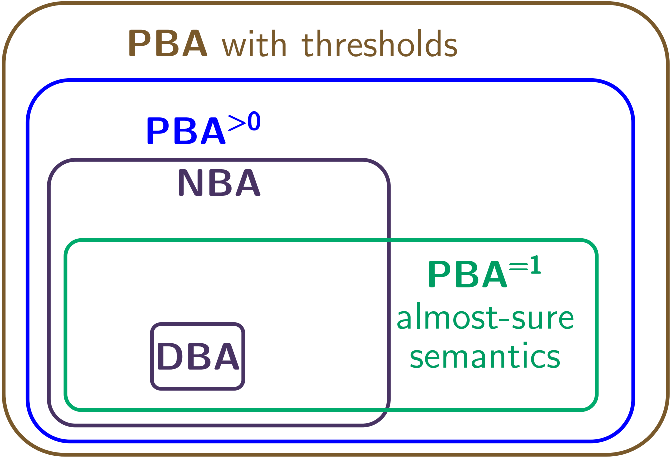
Expressiveness of PBA



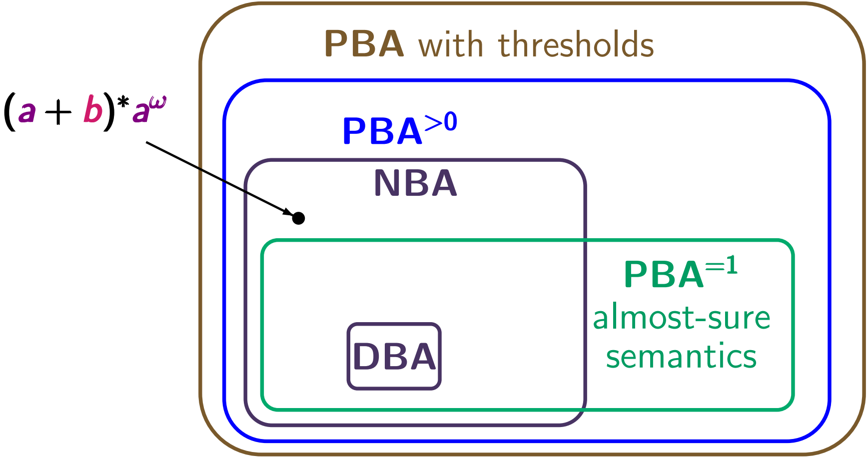
Expressiveness of PBA



Expressiveness of PBA

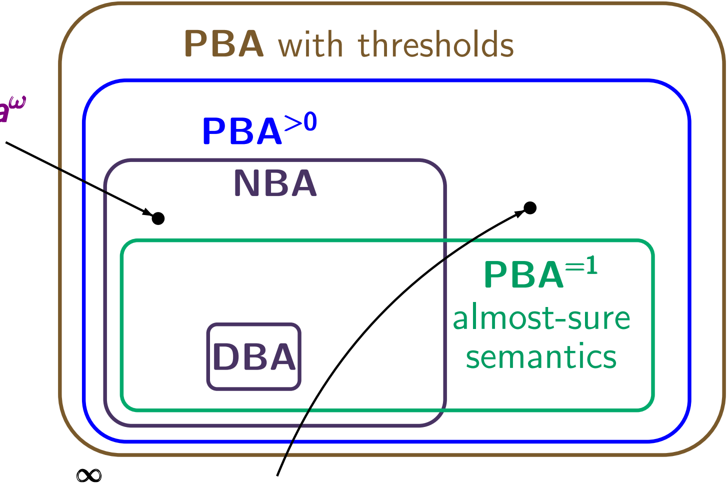


Expressiveness of PBA



Expressiveness of PBA

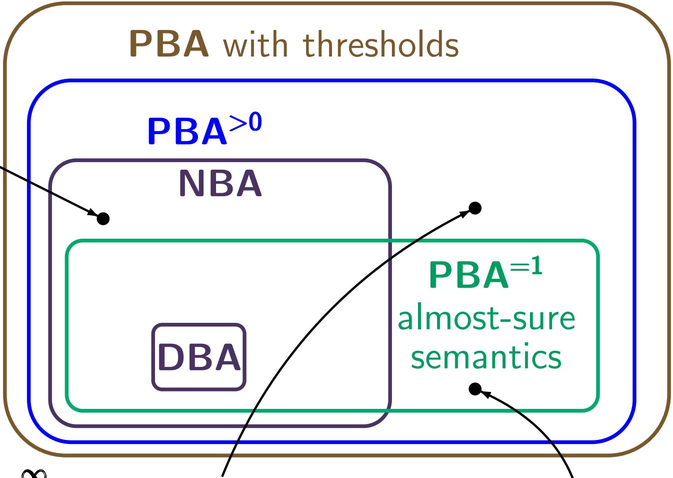
$(a+b)^* a^\omega$



$\{a^{k_1} b a^{k_2} b \ldots : \prod_{i=1}^{\infty} (1 - (\frac{1}{2})^{k_i}) > 0\}$

Expressiveness of PBA

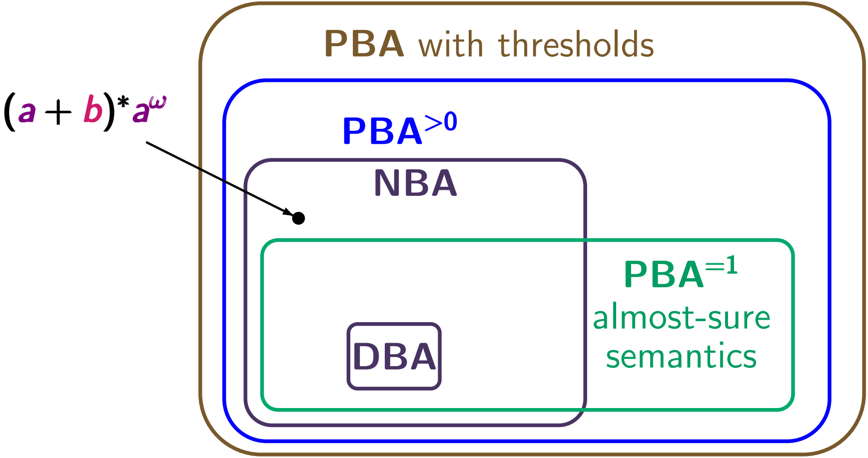
$(a+b)^* a^\omega$



$$\left\{ a^{k_1} b a^{k_2} b \ldots : \prod_{i=1}^{\infty} \left(1 - \left(\tfrac{1}{2}\right)^{k_i}\right) > 0 \right\}$$

$$\left\{ a^{k_1} b a^{k_2} b \ldots : \prod_{i=1}^{\infty} \left(1 - \left(\tfrac{1}{2}\right)^{k_i}\right) = 0 \right\}$$

Expressiveness of PBA



emptiness problem: undecidable for **PBA$^{>0}$**

decidable for **PBA$^{=1}$**

Decidability results for POMDP

The model checking problem for POMDP and several qualitative properties is decidable:

- almost-sure reachability
  "does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"
- invariance with positive probability
  "does $\Pr^{obs}_{\max}(\Box F) > 0$ hold ?"
- almost-sure repeated reachability
  "does $\Pr^{obs}_{\max}(\Box\Diamond F) = 1$ hold ?"
- persistence with positive probability
  "does $\Pr^{obs}_{\max}(\Diamond\Box F) > 0$ hold ?"

algorithms use a certain powerset construction

Decidability results for POMDP

The model checking problem for POMDP and several qualitative properties is decidable:

- almost-sure reachability
  "does $\mathrm{Pr}_{\max}^{obs}(\Diamond F) = 1$ hold ?"
- invariance with positive probability
  "does $\mathrm{Pr}_{\max}^{obs}(\Box F) > 0$ hold ?"
- almost-sure repeated reachability
  "does $\mathrm{Pr}_{\max}^{obs}(\Box\Diamond F) = 1$ hold ?"
- persistence with positive probability
  "does $\mathrm{Pr}_{\max}^{obs}(\Diamond\Box F) > 0$ hold ?"

algorithms use a certain powerset construction

The almost-sure repeated reachability problem

$$\text{“does } \mathsf{Pr}^{obs}_{\max}(\Box\Diamond F) = \mathbf{1} \text{ hold ?”}$$

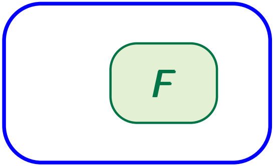
is polynomially reducible to the almost-sure reachability problem “does $\mathsf{Pr}^{obs}_{\max}(\Diamond F) = \mathbf{1}$ hold ?”

Almost-sure reachability/repeated reachability

The almost-sure repeated reachability problem

"does $\Pr^{obs}_{\max}(\Box\Diamond F) = 1$ hold ?"

is polynomially reducible to the almost-sure reachability problem "does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"



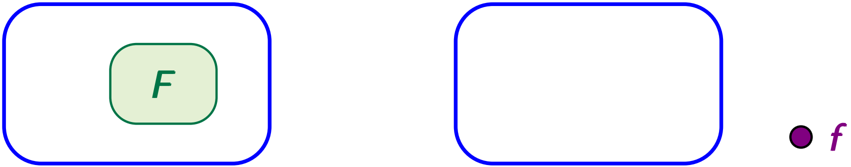
POMDP $\mathcal{M}$
objective:
repeated reachability $\Box\Diamond F$

Almost-sure reachability/repeated reachability

The almost-sure repeated reachability problem

"does $Pr^{obs}_{max}(\Box\Diamond F) = 1$ hold ?"

is polynomially reducible to the almost-sure reachability problem "does $Pr^{obs}_{max}(\Diamond f) = 1$ hold ?"



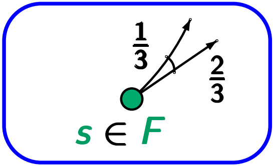
POMDP $\mathcal{M}$
objective:
repeated reachability $\Box\Diamond F$

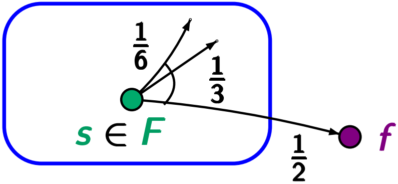
POMDP $\mathcal{M}'$
objective:
reachability $\Diamond f$

The almost-sure repeated reachability problem

$$\text{“does } \Pr^{obs}_{\max}(\Box\Diamond F) = 1 \text{ hold ?”}$$

is polynomially reducible to the almost-sure reachability problem “does $\Pr^{obs}_{\max}(\Diamond f) = 1$ hold ?”





POMDP $\mathcal{M}$
objective:
repeated reachability $\Box\Diamond F$

POMDP $\mathcal{M}'$
objective:
reachability $\Diamond f$

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold **?**"

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

POMDP $\mathcal{M}$ with equivalence $\sim$	$\longrightarrow$	MDP $Pow(\mathcal{M})$ fully observable

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

POMDP $\mathcal{M}$ with equivalence $\sim$	$\longrightarrow$	MDP $Pow(\mathcal{M})$ fully observable

$\Pr^{obs}_{\max}(\Diamond F) = 1$ in $\mathcal{M}$ iff $\Pr_{\max}(\Diamond F') = 1$ in $Pow(\mathcal{M})$

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

POMDP $\mathcal{M}$ with equivalence $\sim$	$\longrightarrow$	MDP $Pow(\mathcal{M})$ fully observable

$\Pr^{obs}_{\max}(\Diamond F) = 1$ in $\mathcal{M}$ iff $\Pr_{\max}(\Diamond F') = 1$ in $Pow(\mathcal{M})$

state s in $\mathcal{M}$ $\mapsto$ states $\langle s, R \rangle$
where $s \in R \subseteq [s]$

$[s]$ = equivalence class of s w.r.t. $\sim$

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

POMDP $\mathcal{M}$ with equivalence $\sim$	$\longrightarrow$	MDP $Pow(\mathcal{M})$ fully observable

$\Pr^{obs}_{\max}(\Diamond F) = 1$ in $\mathcal{M}$ iff $\Pr_{\max}(\Diamond f) = 1$ in $Pow(\mathcal{M})$

state s in $\mathcal{M}$ $\mapsto$ states $\langle s, R \rangle$ where $s \in R \subseteq [s]$

fresh goal state f

$[s]$ = equivalence class of s w.r.t. $\sim$

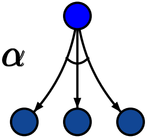
Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

action α



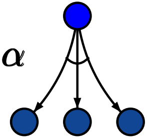
Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

action α



if $Post(s, \alpha) \cap F = \varnothing$

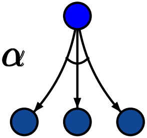
Almost-sure reachability

powerset construction for almost-sure reachability

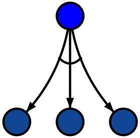
"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

action α



state $\langle s, R \rangle$ in $Pow(\mathcal{M})$



action α

if $Post(s, \alpha) \cap F = \varnothing$

where $s \in R \subseteq [s]$

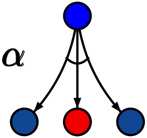
Almost-sure reachability

powerset construction for almost-sure reachability

"does $\mathrm{Pr}^{obs}_{\max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

action α



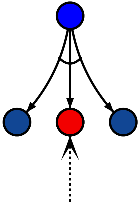
t

if $Post(s, \alpha) \cap F = \varnothing$

where $s \in R \subseteq [s]$

state $\langle s, R \rangle$ in $Pow(\mathcal{M})$

action α



state $\langle t, \ldots \rangle$

$t \in Post(s)$

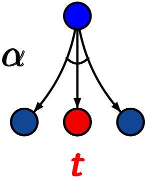
Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

action α



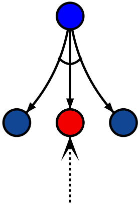
if $Post(s, \alpha) \cap F = \varnothing$

where $s \in R \subseteq [s]$

$U = Post(R, \alpha)$

state $\langle s, R \rangle$ in $Pow(\mathcal{M})$

action α



state $\langle t, U \cap [t] \rangle$

$t \in Post(s)$

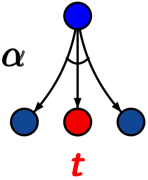
Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

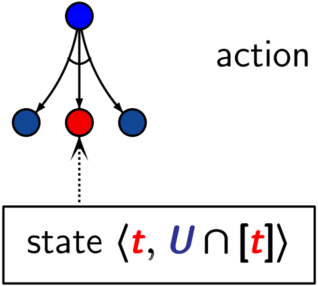
action α



if $Post(s,\alpha) \cap F = \varnothing$

state $\langle s, R\rangle$ in $Pow(\mathcal{M})$

action α



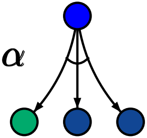
$$P(s,\alpha,t) = P'(\langle s,R\rangle, \alpha, \langle t, U \cap [t]\rangle)$$

powerset construction for almost-sure reachability

"does $\mathrm{Pr}^{obs}_{max}(\Diamond F) = 1$ hold ?"

state s in $\mathcal{M}$

action α



$v \in F$

if $Post(s, \alpha) \cap F \neq \varnothing$

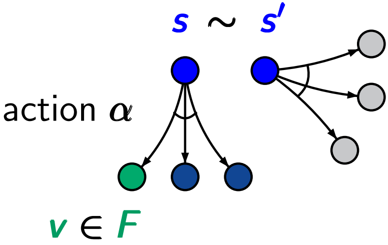
where $s \in R \subseteq [s]$

$U = Post(R, \alpha)$

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"



if $Post(s, \alpha) \cap F \neq \varnothing$

where $s', s \in R \subseteq [s]$

$U = Post(R, \alpha)$

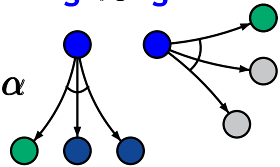
Almost-sure reachability

powerset construction for almost-sure reachability

"does $\mathrm{Pr}^{obs}_{\max}(\Diamond F) = 1$ hold ?"

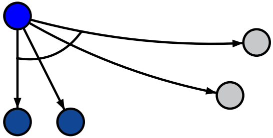
$s \sim s'$ $\quad u \in F$

action α



$v \in F$

state $\langle s, R \rangle$



if $Post(s,\alpha) \cap F \neq \varnothing$

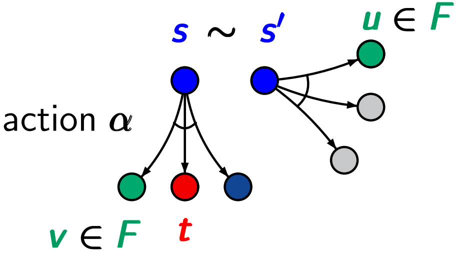
where $s', s \in R \subseteq [s]$

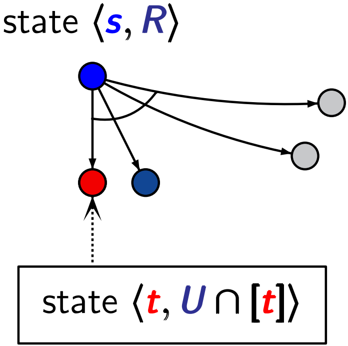
$U = Post(R,\alpha)$

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"





if $Post(s, \alpha) \cap F \neq \varnothing$

where $s', s \in R \subseteq [s]$

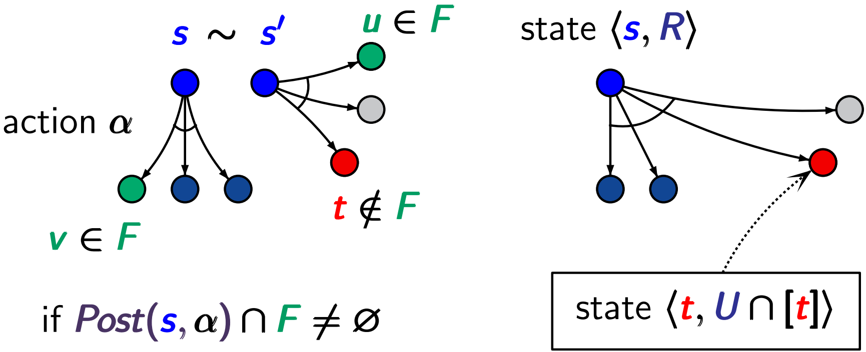
$U = Post(R, \alpha)$

where $t \in U \setminus F$

Almost-sure reachability

powerset construction for almost-sure reachability

“does $\mathrm{Pr}^{obs}_{\max}(\Diamond F) = 1$ hold ?”



if $Post(s, \alpha) \cap F \neq \varnothing$

where $s', s \in R \subseteq [s]$

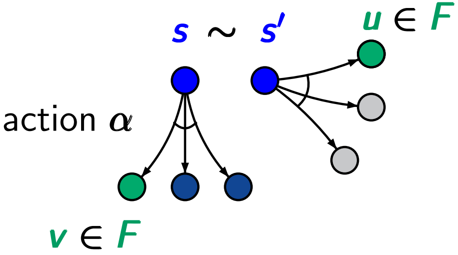
$U = Post(R, \alpha)$

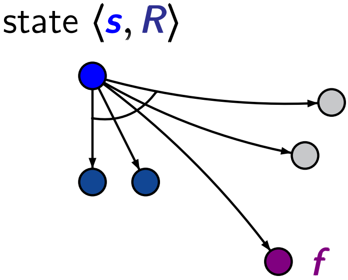
where $t \in U \setminus F$

Almost-sure reachability

powerset construction for almost-sure reachability

"does $\Pr^{obs}_{max}(\Diamond F) = 1$ hold ?"





if $Post(s, \alpha) \cap F \neq \varnothing$

where $s', s \in R \subseteq [s]$

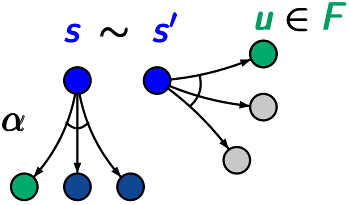
$U = Post(R, \alpha)$

objective: $\Diamond f$

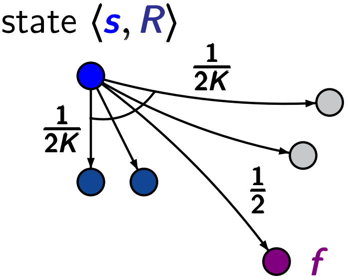
powerset construction for almost-sure reachability

"does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?"

action α



if $Post(s, \alpha) \cap F \neq \varnothing$

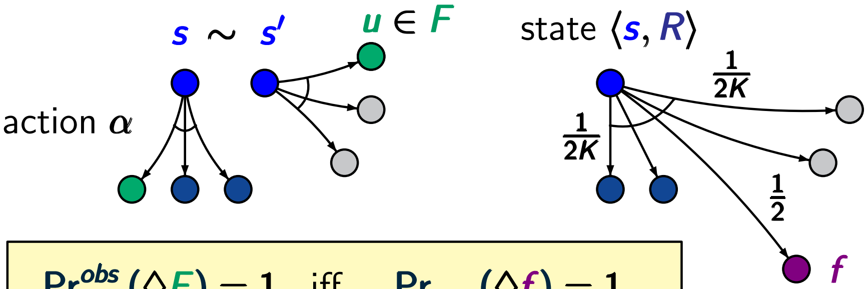


$$P'(\langle s, R\rangle, \alpha, \langle t, U \cap [t]\rangle) = \frac{1}{2K}$$

where $K = |Post(R, \alpha) \setminus F|$

powerset construction for almost-sure reachability

“does $\Pr^{obs}_{\max}(\Diamond F) = 1$ hold ?”



$\Pr^{obs}_{\max}(\Diamond F) = 1$ iff $\Pr_{\max}(\Diamond f) = 1$

$\uparrow$ original POMDP $\mathcal{M}$

$\uparrow$ fully-observable MDP $Pow(\mathcal{M})$