Statistical Model Checking for Markov Decision Processes

David Henriques

Joint work with
João Martins, Paolo Zuliani, André Platzer and Edmund M. Clarke

QEST, September 18^{th}, 2012

Outline of this Presentation

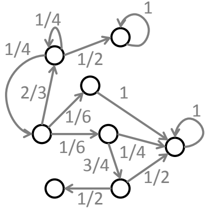
1. Markov Decision Processes
2. Probabilisitic MC and Statistical MC
3. SMC for MDPs
4. Why does it work?
5. Experimental Validation

Summary

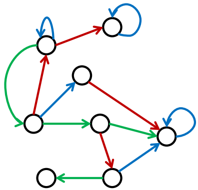
1. Markov Decision Processes
2. Probabilisitic MC and Statistical MC
3. SMC for MDPs
4. Why does it work?
5. Experimental Validation

Common Settings in MC

Fully probabilistic systems



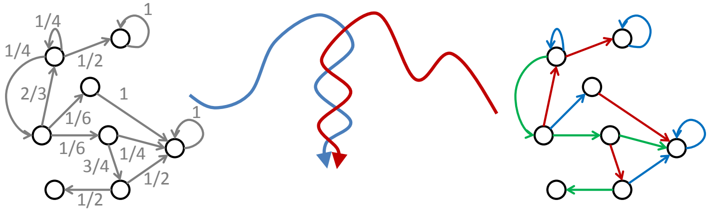
Non-deterministic systems



Common Settings in MC

Fully probabilistic systems

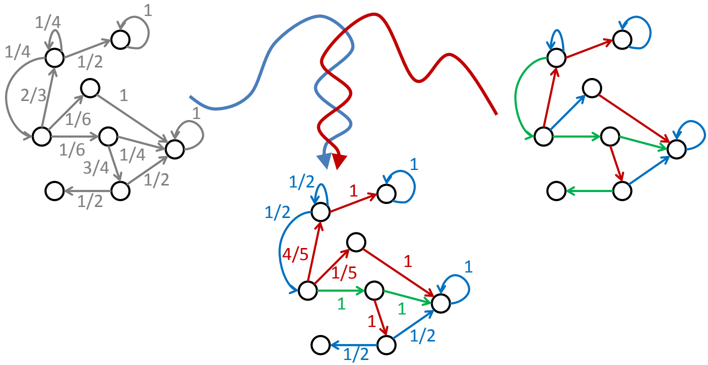
Non-deterministic systems



Common Settings in MC

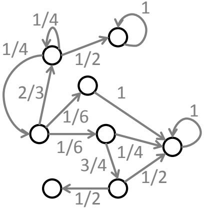
Fully probabilistic systems

Non-deterministic systems

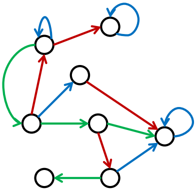


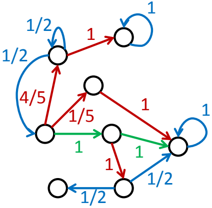
Common Settings in MC

Fully probabilistic systems



Non-deterministic systems





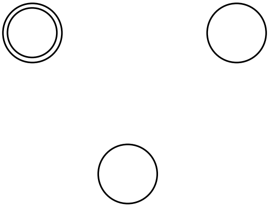
Non-deterministic Probabilistic Systems

Markov Decision Processes

Definition [Markov Decision Process]

A (finite, state labeled) MDP, $\mathcal{M}$, is a tuple $\langle S; s_i; \mathcal{A}; \tau; \Lambda; \mathcal{L} \rangle$ where:

- S is a finite set of states with initial state s_i;
- $\mathcal{A}$ is a finite set of action names;
- $\tau : S \times \mathcal{A} \rightarrow Dist(S)$ is a probabilistic transition function;
- Λ is a set of propositions and $\mathcal{L} : S \rightarrow 2^{\Lambda}$ is a labeling function.

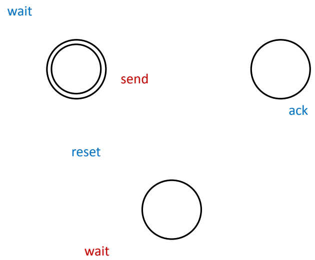


Markov Decision Processes

Definition [Markov Decision Process]

A (finite, state labeled) MDP, $\mathcal{M}$, is a tuple $\langle S; s_i; \mathcal{A}; \tau; \Lambda; \mathcal{L} \rangle$ where:

- S is a finite set of states with initial state s_i;
- $\mathcal{A}$ is a finite set of action names;
- $\tau : S \times \mathcal{A} \to Dist(S)$ is a probabilistic transition function;
- Λ is a set of propositions and $\mathcal{L} : S \to 2^{\Lambda}$ is a labeling function.

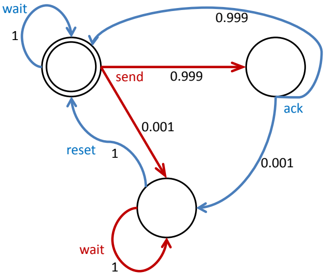


Markov Decision Processes

Definition [Markov Decision Process]

A (finite, state labeled) MDP, $\mathcal{M}$, is a tuple $\langle S; s_i; \mathcal{A}; \tau; \Lambda; \mathcal{L} \rangle$ where:

- S is a finite set of states with initial state s_i;
- $\mathcal{A}$ is a finite set of action names;
- $\tau : S \times \mathcal{A} \rightarrow Dist(S)$ is a probabilistic transition function;
- Λ is a set of propositions and $\mathcal{L} : S \rightarrow 2^{\Lambda}$ is a labeling function.

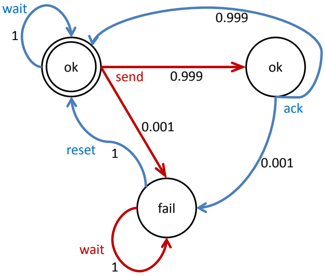


Markov Decision Processes

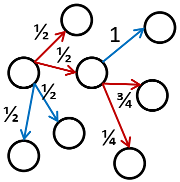
Definition [Markov Decision Process]

A (finite, state labeled) MDP, $\mathcal{M}$, is a tuple $\langle S; s_i; \mathcal{A}; \tau; \Lambda; \mathcal{L} \rangle$ where:

- S is a finite set of states with initial state s_i;
- $\mathcal{A}$ is a finite set of action names;
- $\tau : S \times \mathcal{A} \to Dist(S)$ is a probabilistic transition function;
- Λ is a set of propositions and $\mathcal{L} : S \to 2^{\Lambda}$ is a labeling function.



How to choose actions?

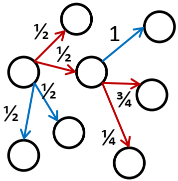


Definition [Scheduler]

A memoryless scheduler for $\mathcal{M}$, σ, is a function $\sigma : S \to Dist(S)$ s.t. for each $s \in S$, $\sigma(s) = \sum_{a \in \mathcal{A}} p_{s,a} \tau(s, a)$ with $\sum_{a \in \mathcal{A}} p_{s,a} = 1$.

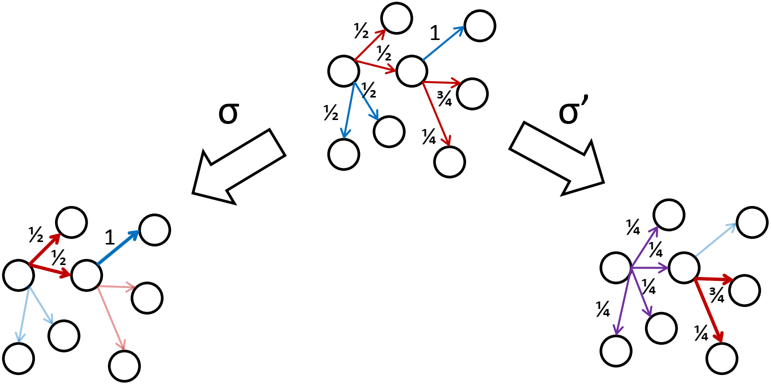
Schedulers "solve" the nondeterminism.

How to choose actions?



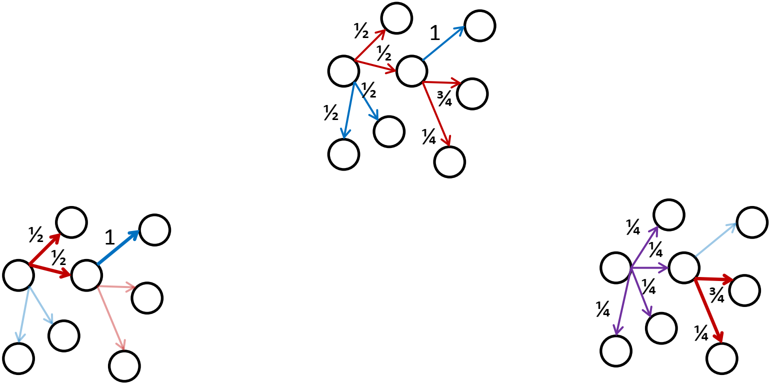
Schedulers “solve” the nondeterminism.

How to choose actions?



Schedulers "solve" the nondeterminism.

How to choose actions?

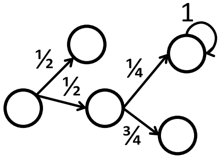


Schedulers "solve" the nondeterminism.

Paths and Probabilities (Paths)

Definition [Path]

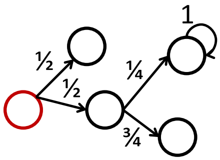
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



Paths and Probabilities (Paths)

Definition

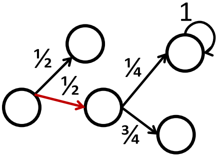
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



Paths and Probabilities (Paths)

Definition

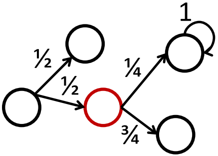
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



Paths and Probabilities (Paths)

Definition

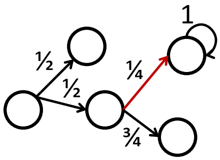
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



Paths and Probabilities (Paths)

Definition

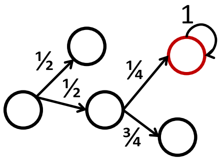
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



Paths and Probabilities (Paths)

Definition

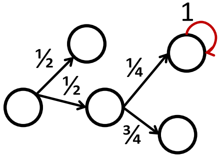
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



Paths and Probabilities (Paths)

Definition

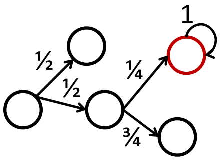
For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



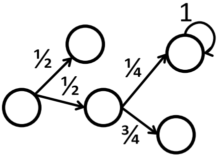
Paths and Probabilities (Paths)

Definition

For $\mathcal{M}, \sigma$, a path π is a sequence of states $\pi_1 \cdot \pi_2 ...$ s.t. $\forall i, \sigma(\pi_i)(\pi_{i+1}) > 0$.



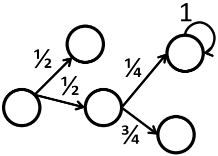
Paths and Probabilities



Paths and Probabilities (Probabilities)

Proposition

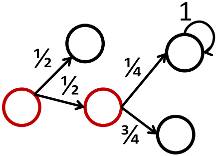
Each σ induces a probability measure P^σ over the set of paths given by
$P^\sigma(\{\pi_0 \cdot \pi_1 \cdot \ldots \cdot \pi_n \cdot * \mid * \text{ is a path, } \pi_0 = s_i\}) = \prod_{0 \leq i < n} \sigma(\pi_i)(\pi_{i+1})$



Paths and Probabilities (Probabilities)

Proposition

Each σ induces a probability measure P^σ over the set of paths given by
$P^\sigma(\{\pi_0 \cdot \pi_1 \cdot \ldots \cdot \pi_n \cdot * \mid * \text{ is a path}, \pi_0 = s_i\}) = \prod_{0 \le i < n} \sigma(\pi_i)(\pi_{i+1})$

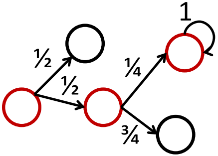


$$\frac{1}{2}$$

Paths and Probabilities (Probabilities)

Proposition

Each σ induces a probability measure P^σ over the set of paths given by
$P^\sigma(\{\pi_0 \cdot \pi_1 \cdot \ldots \cdot \pi_n \cdot * \mid * \text{ is a path}, \pi_0 = s_i\}) = \prod_{0 \leq i < n} \sigma(\pi_i)(\pi_{i+1})$

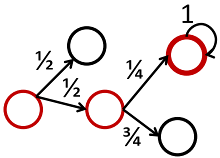


$$\frac{1}{2} \times \frac{1}{4}$$

Paths and Probabilities (Probabilities)

Proposition

Each σ induces a probability measure P^σ over the set of paths given by
$P^\sigma(\{\pi_0 \cdot \pi_1 \cdot \ldots \cdot \pi_n \cdot * \mid * \text{ is a path}, \pi_0 = s_i\}) = \prod_{0 \le i < n} \sigma(\pi_i)(\pi_{i+1})$



$$\frac{1}{2} \times \frac{1}{4} \times 1$$

Summary

Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$$\begin{array}{ll}
\pi \models \lambda & \text{if } \lambda \in \mathcal{L}(\pi_0) \\
\pi \models \neg\varphi & \text{if } \pi \not\models \varphi \\
\pi \models \varphi_1 \vee \varphi_2 & \text{if } \pi \models \varphi_1 \text{ or } \pi \models \varphi_2 \\
\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi & \text{if } \exists_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \mathbf{G}^{\leq n}\varphi & \forall_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2 & \exists_{i \leq n} \forall_{k \leq i} : \pi|^k \models \varphi_1 \text{ and } \pi|^i \models \varphi_2
\end{array}$$

$\mathbf{F}^{\leq \mathbf{n}}$ a

Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n}\forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$

$\mathbf{F}^{\leq \mathbf{n}}$ a

Bounded LTL

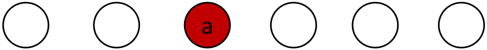
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n} \forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$

$\mathbf{F}^{\leq \mathbf{n}}$ a



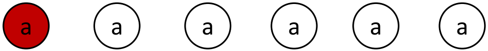
Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$$\begin{array}{ll}
\pi \models \lambda & \text{if } \lambda \in \mathcal{L}(\pi_0) \\
\pi \models \neg\varphi & \text{if } \pi \not\models \varphi \\
\pi \models \varphi_1 \vee \varphi_2 & \text{if } \pi \models \varphi_1 \text{ or } \pi \models \varphi_2 \\
\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi & \text{if } \exists_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \mathbf{G}^{\leq n}\varphi & \forall_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2 & \exists_{i \leq n} \forall_{k \leq i} : \pi|^k \models \varphi_1 \text{ and } \pi|^i \models \varphi_2
\end{array}$$



$\mathbf{G}^{\leq \mathbf{n}}$ a

Bounded LTL

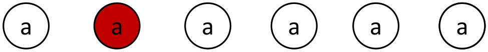
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n}\forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$

$\mathbf{G}^{\leq \mathbf{n}}$ a



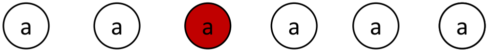
Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n}\forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$



$\mathbf{G}^{\leq \mathbf{n}}$ a

Bounded LTL

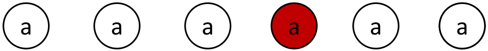
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n}\forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$

$\mathbf{G}^{\leq \mathbf{n}}$ a



Bounded LTL

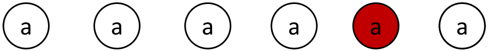
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n}\forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$

$\mathbf{G}^{\leq \mathbf{n}}$ a



Bounded LTL

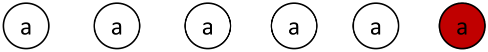
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$\pi \models \lambda$	if $\lambda \in \mathcal{L}(\pi_0)$
$\pi \models \neg\varphi$	if $\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	if $\pi \models \varphi_1$ or $\pi \models \varphi_2$
$\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi$	if $\exists_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \mathbf{G}^{\leq n}\varphi$	$\forall_{i \leq n} : \pi\|^i \models \varphi$
$\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2$	$\exists_{i \leq n}\forall_{k \leq i} : \pi\|^k \models \varphi_1$ and $\pi\|^i \models \varphi_2$

$\mathbf{G}^{\leq \mathbf{n}}$ a



Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$$\begin{array}{ll}
\pi \models \lambda & \text{if } \lambda \in \mathcal{L}(\pi_0) \\
\pi \models \neg\varphi & \text{if } \pi \not\models \varphi \\
\pi \models \varphi_1 \vee \varphi_2 & \text{if } \pi \models \varphi_1 \text{ or } \pi \models \varphi_2 \\
\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi & \text{if } \exists_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \mathbf{G}^{\leq n}\varphi & \forall_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2 & \exists_{i \leq n} \forall_{k \leq i} : \pi|^k \models \varphi_1 \text{ and } \pi|^i \models \varphi_2
\end{array}$$

a $\mathbf{U}^{\leq \mathbf{n}}$ b

Bounded LTL

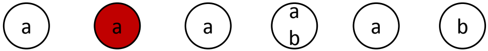
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$$\begin{array}{ll}
\pi \models \lambda & \text{if } \lambda \in \mathcal{L}(\pi_0) \\
\pi \models \neg\varphi & \text{if } \pi \not\models \varphi \\
\pi \models \varphi_1 \vee \varphi_2 & \text{if } \pi \models \varphi_1 \text{ or } \pi \models \varphi_2 \\
\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi & \text{if } \exists_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \mathbf{G}^{\leq n}\varphi & \forall_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2 & \exists_{i \leq n}\forall_{k \leq i} : \pi|^k \models \varphi_1 \text{ and } \pi|^i \models \varphi_2
\end{array}$$

a $\mathbf{U}^{\leq \mathbf{n}}$ b



Bounded LTL

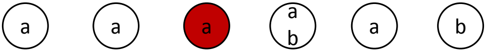
Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$$\begin{array}{ll}
\pi \models \lambda & \text{if } \lambda \in \mathcal{L}(\pi_0) \\
\pi \models \neg\varphi & \text{if } \pi \not\models \varphi \\
\pi \models \varphi_1 \vee \varphi_2 & \text{if } \pi \models \varphi_1 \text{ or } \pi \models \varphi_2 \\
\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi & \text{if } \exists_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \mathbf{G}^{\leq n}\varphi & \forall_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2 & \exists_{i \leq n}\forall_{k \leq i} : \pi|^k \models \varphi_1 \text{ and } \pi|^i \models \varphi_2
\end{array}$$

a $\mathbf{U}^{\leq \mathbf{n}}$ b



Bounded LTL

Syntax of BLTL

$$\varphi := \lambda \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{F}^{\leq \mathbf{n}}\varphi \mid \mathbf{G}^{\leq \mathbf{n}}\varphi \mid \varphi \mathbf{U}^{\leq \mathbf{n}}\varphi \text{ where } \lambda \in \Lambda.$$

Semantics of BLTL

$$\begin{array}{ll}
\pi \models \lambda & \text{if } \lambda \in \mathcal{L}(\pi_0) \\
\pi \models \neg\varphi & \text{if } \pi \not\models \varphi \\
\pi \models \varphi_1 \vee \varphi_2 & \text{if } \pi \models \varphi_1 \text{ or } \pi \models \varphi_2 \\
\pi \models \mathbf{F}^{\leq \mathbf{n}}\varphi & \text{if } \exists_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \mathbf{G}^{\leq n}\varphi & \forall_{i \leq n} : \pi|^i \models \varphi \\
\pi \models \varphi_1 \mathbf{U}^{\leq n}\varphi_2 & \exists_{i \leq n} \forall_{k \leq i} : \pi|^k \models \varphi_1 \text{ and } \pi|^i \models \varphi_2
\end{array}$$

a $\mathbf{U}^{\leq \mathbf{n}}$ b

Probabilistic BLTL

The decision problem of MC in fully probabilistic settings is finding out if, for a given parameter θ,

$$P^{\sigma}(\{\pi : \pi \models \varphi\}) \leq \theta$$

Probabilistic BLTL

The decision problem of MC in fully probabilistic settings is finding out if, for a given parameter θ,

$$P^{\sigma}(\{\pi : \pi \models \varphi\}) \leq \theta$$

Proposition

This is a well posed problem.

We should be so lucky...

We may not have a scheduler, but we still want to guarantee properties...

We should be so lucky...

We may not have a scheduler, but we still want to guarantee properties...

We make claims that hold all for *all* schedulers, no matter how adversarial.

We should be so lucky...

We may not have a scheduler, but we still want to guarantee properties...

We make claims that hold all for *all* schedulers, no matter how adversarial.

The (decision) problem for MC for MDPS is finding out if, for a given parameter θ,

$$P^{\sigma}(\{\pi : \pi \models \varphi\}) \leq \theta \text{ for all } \sigma$$

Summary

SMC for MDPS

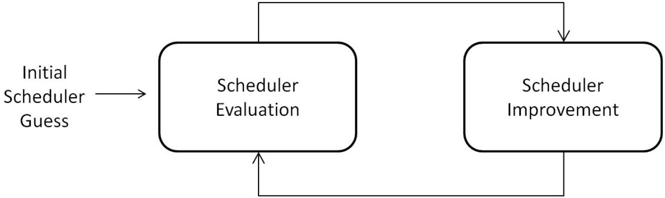
Basic idea

"Learn the most adversarial scheduler (or a good enough approximation) by successively refining an initial guess"

SMC for MDPS

Basic idea

"Learn the most adversarial scheduler (or a good enough approximation) by successively refining an initial guess"

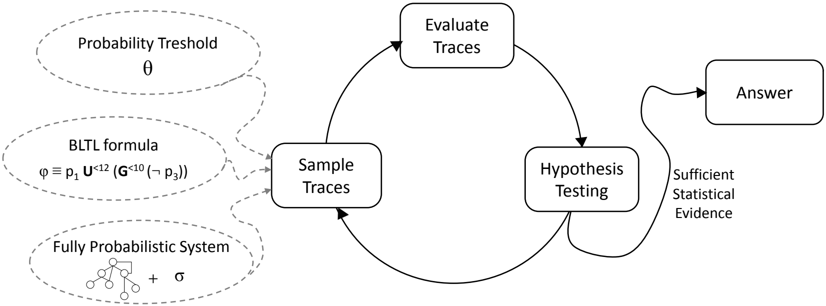


Scheduler Evaluation

Same ideas as classical Statistical Model Checking

Scheduler Evaluation

Same ideas as classical Statistical Model Checking



Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^{\sigma}$ of a visited (s, a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^\sigma$ of a visited (s, a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

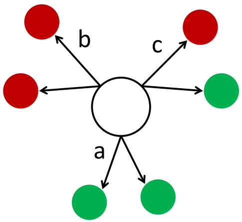
$$\hat{Q}^\sigma(s, a) \overset{\#samples \rightarrow \infty}{\longrightarrow} Q^\sigma(s, a) \equiv P(\pi \models \varphi \mid (s, a) \in \pi)$$

Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^\sigma$ of a visited (s, a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

$$\hat{Q}^\sigma(s, a) \overset{\#samples \to \infty}{\longrightarrow} Q^\sigma(s, a) \equiv P(\pi \models \varphi \mid (s, a) \in \pi)$$

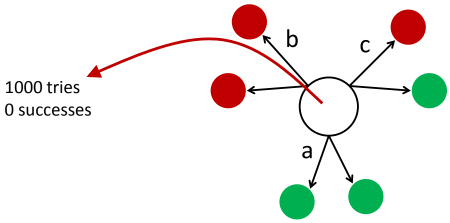


Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^{\sigma}$ of a visited (s, a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

$$\hat{Q}^{\sigma}(s, a) \xrightarrow{\#samples \to \infty} Q^{\sigma}(s, a) \equiv P(\pi \models \varphi \mid (s, a) \in \pi)$$



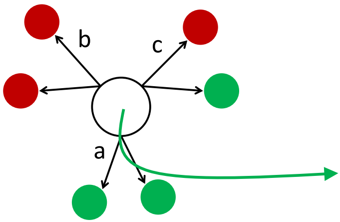
Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^\sigma$ of a visited (s,a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

$$\hat{Q}^\sigma(s,a) \xrightarrow{\#samples \to \infty} Q^\sigma(s,a) \equiv P(\pi \models \varphi \mid (s,a) \in \pi)$$

1000 tries
0 successes



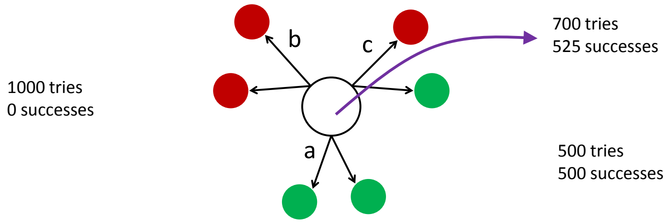
500 tries
500 successes

Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^\sigma$ of a visited (s,a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

$$\hat{Q}^\sigma(s,a) \xrightarrow{\#samples \to \infty} Q^\sigma(s,a) \equiv P(\pi \models \varphi \mid (s,a) \in \pi)$$

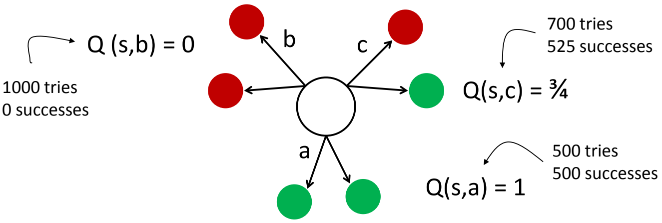


Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^\sigma$ of a visited (s, a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

$$\hat{Q}^\sigma(s, a) \xrightarrow{\#samples \to \infty} Q^\sigma(s, a) \equiv P(\pi \models \varphi \mid (s, a) \in \pi)$$

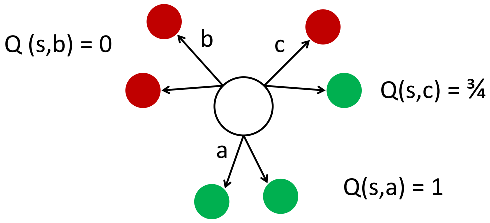


Scheduler Evalutaion

Record whether state action pairs crossed by samples satisfied φ.

Empirical quality $\hat{Q}^\sigma$ of a visited (s, a) is $\frac{\#(s,a) \text{ seen in satisfying traces}}{\# \text{ times } (s,a) \text{ was seen}}$

$$\hat{Q}^\sigma(s, a) \xrightarrow{\#samples \to \infty} Q^\sigma(s, a) \equiv P(\pi \models \varphi \mid (s, a) \in \pi)$$



Scheduler Improvement

New scheduler σ' is obtained from σ by giving higher probability to transitions with higher quality.

Update Rule

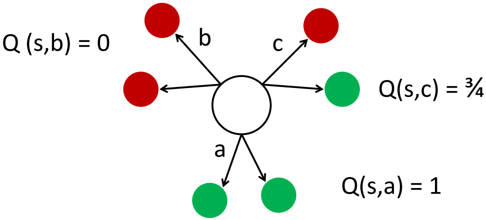
$$\sigma'(s,a) = \frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b \in \mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

Scheduler Improvement

New scheduler σ' is obtained from σ by giving higher probability to transitions with higher quality.

Update Rule

$$\sigma'(s,a) = \frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b \in \mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

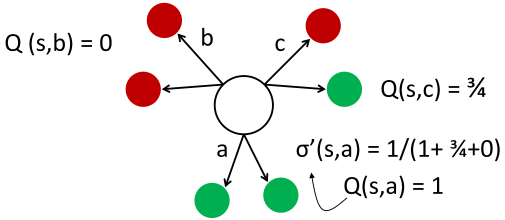


Scheduler Improvement

New scheduler σ' is obtained from σ by giving higher probability to transitions with higher quality.

Update Rule

$$\sigma'(s,a) = \frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b \in \mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

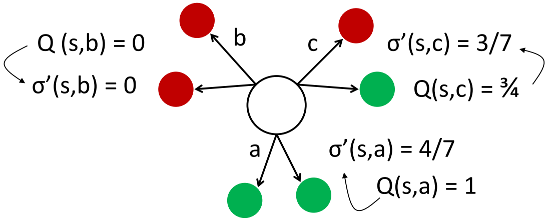


Scheduler Improvement

New scheduler σ' is obtained from σ by giving higher probability to transitions with higher quality.

Update Rule

$$\sigma'(s,a) = \frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b \in \mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

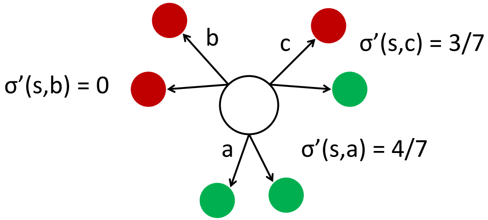


Scheduler Improvement

New scheduler σ' is obtained from σ by giving higher probability to transitions with higher quality.

Update Rule

$$\sigma'(s,a) = \frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b \in \mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$



History and Greediness

What if we explore too little?

In case there are state action pairs such that $\hat{Q}(s,a) = 0$, keep a history parameter h and update instead

$$\sigma'(s,a) = h\sigma(s,a) + (1-h)\frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b\in\mathcal{A}}\hat{Q}^{\sigma}(s,b)}$$

This avoids “blocking” transitions.

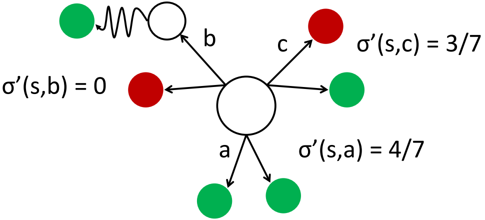
History and Greediness

What if we explore too little?

In case there are state action pairs such that $\hat{Q}(s,a) = 0$, keep a history parameter h and update instead

$$\sigma'(s,a) = h\sigma(s,a) + (1-h)\frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b \in \mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

This avoids “blocking” transitions.



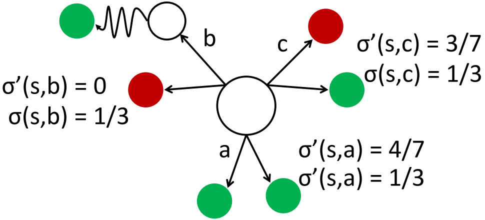
History and Greediness

What if we explore too little?

In case there are state action pairs such that $\hat{Q}(s,a) = 0$, keep a history parameter h and update instead

$$\sigma'(s,a) = h\sigma(s,a) + (1-h)\frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b\in\mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

This avoids "blocking" transitions.



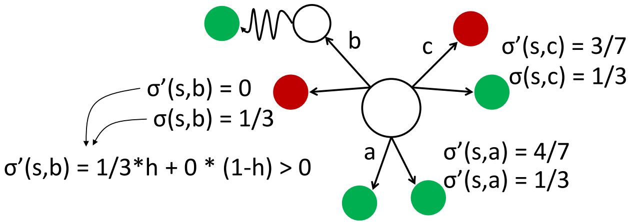
History and Greediness

What if we explore too little?

In case there are state action pairs such that $\hat{Q}(s,a) = 0$, keep a history parameter h and update instead

$$\sigma'(s,a) = h\sigma(s,a) + (1-h)\frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b\in\mathcal{A}}\hat{Q}^{\sigma}(s,b)}$$

This avoids “blocking” transitions.



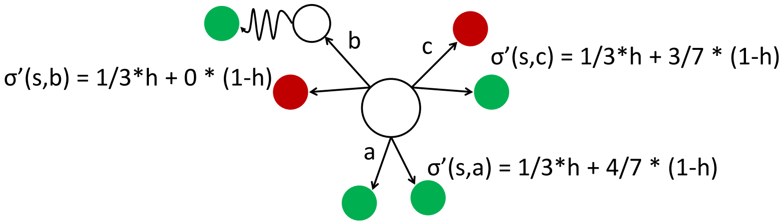
History and Greediness

What if we explore too little?

In case there are state action pairs such that $\hat{Q}(s,a) = 0$, keep a history parameter h and update instead

$$\sigma'(s,a) = h\sigma(s,a) + (1-h)\frac{\hat{Q}^{\sigma}(s,a)}{\sum_{b\in\mathcal{A}} \hat{Q}^{\sigma}(s,b)}$$

This avoids "blocking" transitions.



History and Greediness

What if we explore too much?

Keep a greediness parameter ϵ and give *all* probability to the best action except for ϵ, which is distributed according to the update rule

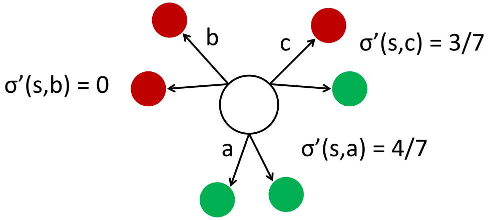
This avoids slow updates.

History and Greediness

What if we explore too much?

Keep a greediness parameter ϵ and give *all* probability to the best action except for ϵ, which is distributed according to the update rule

This avoids slow updates.

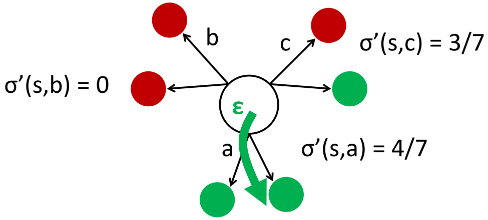


History and Greediness

What if we explore too much?

Keep a greediness parameter ϵ and give *all* probability to the best action except for ϵ, which is distributed according to the update rule

This avoids slow updates.

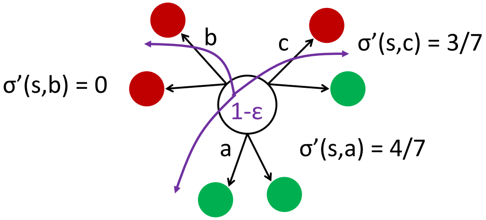


History and Greediness

What if we explore too much?

Keep a greediness parameter ϵ and give *all* probability to the best action except for ϵ, which is distributed according to the update rule

This avoids slow updates.

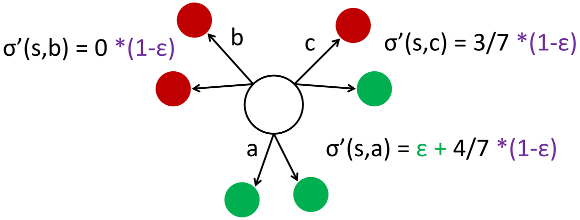


History and Greediness

What if we explore too much?

Keep a greediness parameter ϵ and give *all* probability to the best action except for ϵ, which is distributed according to the update rule

This avoids slow updates.



If at first you don't succeed...

If σ makes $P^{\sigma}(\{\pi : \pi \models \varphi\}) > \theta$, the property is surely false.

If at first you don't succeed...

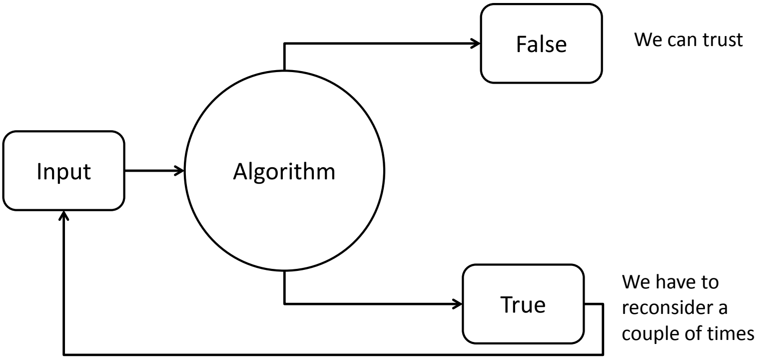
If σ makes $P^{\sigma}(\{\pi : \pi \models \varphi\}) > \theta$, the property is surely false.

If not

- We may be converging towards a local optimum;
- The property may be true;

If at first you don't succeed...

Algorithms like this are called "False-biased Monte Carlo Algorithms"



Confidence increases exponentially with the number of times we restart.

Theorem

Summary

Value

Definition [Value]

The Value of a state s under a scheduler σ is defined as

$$V^{\sigma}(s) = P(\pi \models \varphi \mid (s, a) \in \pi, a \in \mathcal{A}(s))$$

Value

Definition [Value]

The Value of a state s under a scheduler σ is defined as

$$V^{\sigma}(s) = P(\pi \models \varphi \mid (s, a) \in \pi, a \in \mathcal{A}(s))$$

Notice that the MC problem can be reduced to finding $V(^{\sigma} s_i)$

Value

Definition [Value]

The Value of a state s under a scheduler σ is defined as

$$V^{\sigma}(s) = P(\pi \models \varphi \mid (s,a) \in \pi, a \in \mathcal{A}(s))$$

Notice that the MC problem can be reduced to finding $V(^{\sigma} s_i)$

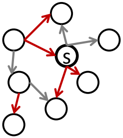
$$V^{\sigma}(s) = \sum_{a \in \mathcal{A}(s)} \sigma(s,a) Q^{\sigma}(s,a)$$

Value

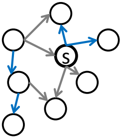
Definition [Local Update]

Let σ and σ' be two schedulers. The local update of σ by σ' in s, $\sigma[\sigma(s) \to \sigma'(s)]$ is the scheduler the behaves like σ everywhere but in s, where it behaves as σ'.

σ



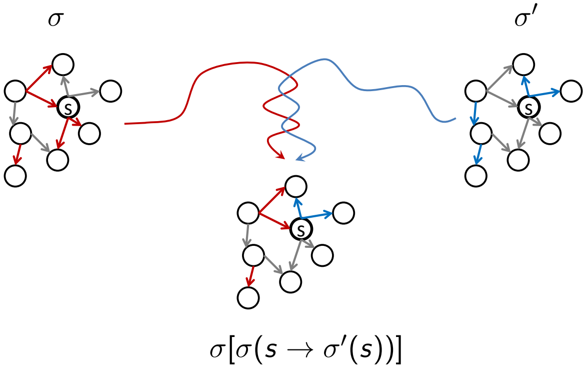
σ'



Value

Definition [Local Update]

Let σ and σ' be two schedulers. The local update of σ by σ' in s, $\sigma[\sigma(s) \to \sigma'(s)]$ is the scheduler the behaves like σ everywhere but in s, where it behaves as σ'.

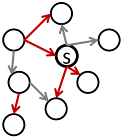


Value

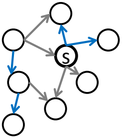
Definition [Local Update]

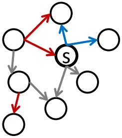
Let σ and σ' be two schedulers. The local update of σ by σ' in s, $\sigma[\sigma(s) \to \sigma'(s)]$ is the scheduler the behaves like σ everywhere but in s, where it behaves as σ'.

σ



σ'





$\sigma[\sigma(s \to \sigma'(s))]$

Value

Theorem [SB]

Let σ and σ' be two schedulers and $\forall s \in S : V^{\sigma[\sigma(s) \to \sigma'(s)]}(s) \geq V^{\sigma}(s)$, then

$$\forall s \in S : V^{\sigma'}(s) \geq V^{\sigma}(s)$$

Corollary

Let σ be the input scheduler and σ' be the output of Scheduler Improvement. Then

$$\forall s \in S : V^{\sigma'}(s) \geq V^{\sigma}(s)$$

and, in particular

$$V^{\sigma'}(s_i) \geq V^{\sigma}(s_i)$$

▸ Proof

Summary

Experimental Validation

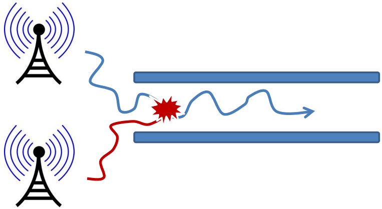
We divided models in three categories

- Heavily structured models
- Structured models
- Unstructured models

Comparisons were made against PRISM, a state-of-the-art probabilistic model checker

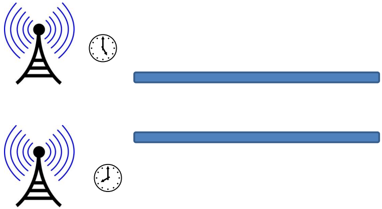
Highly Structured Models

- CSMA - Carrier Sense, Multiple Access protocol
- WLAN - IEEE 802.11 wireless LAN protocol



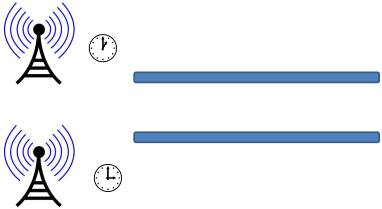
Highly Structured Models

- CSMA - Carrier Sense, Multiple Access protocol
- WLAN - IEEE 802.11 wireless LAN protocol



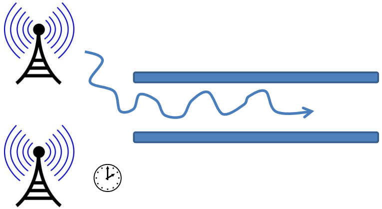
Highly Structured Models

- CSMA - Carrier Sense, Multiple Access protocol
- WLAN - IEEE 802.11 wireless LAN protocol



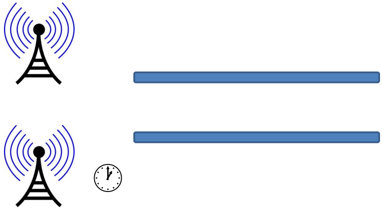
Highly Structured Models

- CSMA - Carrier Sense, Multiple Access protocol
- WLAN - IEEE 802.11 wireless LAN protocol



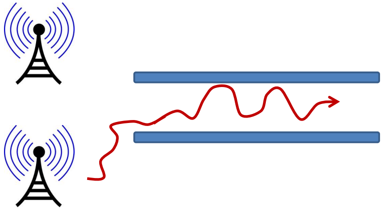
Highly Structured Models

- CSMA - Carrier Sense, Multiple Access protocol
- WLAN - IEEE 802.11 wireless LAN protocol



Highly Structured Models

- CSMA - Carrier Sense, Multiple Access protocol
- WLAN - IEEE 802.11 wireless LAN protocol



Highly Structured Models

CSMA 3 4	θ	0.5	0.8	0.85	0.9	0.95	**PRISM**
	out	F	F	F	T	T	0.86
	t	1.7	11.5	35.9	115.7	111.9	136
CSMA 3 6	θ	0.3	0.4	0.45	0.5	0.8	**PRISM**
	out	F	F	F	T	T	0.48
	t	2.5	9.4	18.8	133.9	119.3	2995
CSMA 4 4	θ	0.5	0.7	0.8	0.9	0.95	**PRISM**
	out	F	F	F	F	T	0.93
	t	3.5	3.7	17.5	69.0	232.8	16244
CSMA 4 6	θ	0.5	0.7	0.8	0.9	0.95	**PRISM**
	out	F	F	F	F	F	timeout
	t	3.7	4.1	4.2	26.2	258.9	timeout
WLAN 5	θ	0.1	0.15	0.2	0.25	0.5	**PRISM**
	out	F	F	T	T	T	0.18
	t	4.9	11.1	124.7	104.7	103.2	1.6
WLAN 6	θ	0.1	0.15	0.2	0.25	0.5	**PRISM**
	out	F	F	T	T	T	0.18
	t	5.0	11.3	127.0	104.9	102.9	1.6

Highly Structured Models

CSMA 3 4	θ	0.5	0.8	**0.85**	**0.9**	0.95	**PRISM**
	out	F	F	F	T	T	**0.86**
	t	1.7	11.5	35.9	115.7	111.9	136

CSMA 3 6	θ	0.3	0.4	**0.45**	**0.5**	0.8	**PRISM**
	out	F	F	F	T	T	**0.48**
	t	2.5	9.4	18.8	133.9	119.3	2995

CSMA 4 4	θ	0.5	0.7	0.8	**0.9**	**0.95**	**PRISM**
	out	F	F	F	F	T	**0.93**
	t	3.5	3.7	17.5	69.0	232.8	16244

CSMA 4 6	θ	0.5	0.7	0.8	0.9	**0.95**	**PRISM**
	out	F	F	F	F	F	timeout
	t	3.7	4.1	4.2	26.2	258.9	timeout

WLAN 5	θ	0.1	**0.15**	**0.2**	0.25	0.5	**PRISM**
	out	F	F	T	T	T	**0.18**
	t	4.9	11.1	124.7	104.7	103.2	1.6

WLAN 6	θ	0.1	**0.15**	**0.2**	0.25	0.5	**PRISM**
	out	F	F	T	T	T	**0.18**
	t	5.0	11.3	127.0	104.9	102.9	1.6

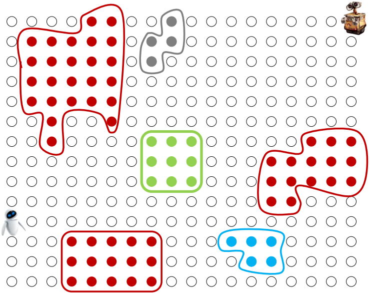
Highly Structured Models

Takeaways

- Symmetry makes the number of "meaningful" actions relatively small;
- SMC works well in highly structured systems;
- Exact methods still work best in most cases;

Structured Models

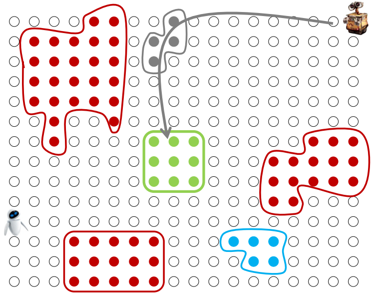
- Motion Planning - Two robots move around an n by n plant



$$P_{\leq\theta}\left(\left[Safe_1 \mathbf{U}^{\leq 30}\left(pickup_1 \wedge \left[Safe'_1 \mathbf{U}^{\leq 30} RendezVous\right]\right)\right]\right.$$
$$\left.\wedge \left[Safe_2 \mathbf{U}^{\leq 30}\left(pickup_2 \wedge \left[Safe'_2 \mathbf{U}^{\leq 30} RendezVous\right]\right)\right]\right)$$

Structured Models

- Motion Planning - Two robots move around an n by n plant

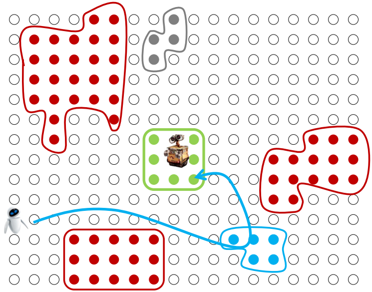


$$P_{\leq\theta}\Big(\Big[Safe_1 \mathbf{U}^{\leq 30}\Big(pickup_1 \wedge \Big[Safe'_1 \mathbf{U}^{\leq 30} RendezVous\Big]\Big)\Big]$$

$$\wedge \Big[Safe_2 \mathbf{U}^{\leq 30}\Big(pickup_2 \wedge \Big[Safe'_2 \mathbf{U}^{\leq 30} RendezVous\Big]\Big)\Big]\Big)$$

Structured Models

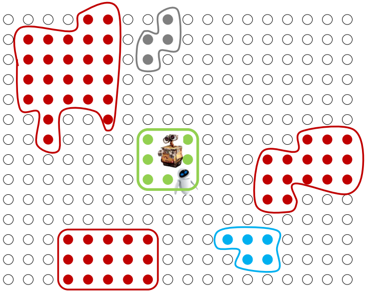
- Motion Planning - Two robots move around an n by n plant



$$P_{\leq\theta}\Big(\Big[Safe_1 \mathbf{U}^{\leq 30}\Big(pickup_1 \wedge \Big[Safe'_1 \mathbf{U}^{\leq 30} RendezVous\Big]\Big)\Big]$$
$$\wedge \Big[Safe_2 \mathbf{U}^{\leq 30}\Big(pickup_2 \wedge \Big[Safe'_2 \mathbf{U}^{\leq 30} RendezVous\Big]\Big)\Big]\Big)$$

Structured Models

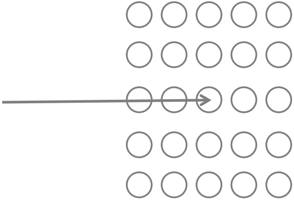
- Motion Planning - Two robots move around an n by n plant



$$P_{\leq\theta}\left(\left[Safe_1 \mathbf{U}^{\leq 30}\left(pickup_1 \wedge \left[Safe'_1 \mathbf{U}^{\leq 30} RendezVous\right]\right)\right]\right.$$
$$\left.\wedge \left[Safe_2 \mathbf{U}^{\leq 30}\left(pickup_2 \wedge \left[Safe'_2 \mathbf{U}^{\leq 30} RendezVous\right]\right)\right]\right)$$

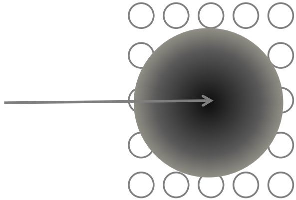
Structured Models

- Motion Planning - Two robots move around an n by n plant



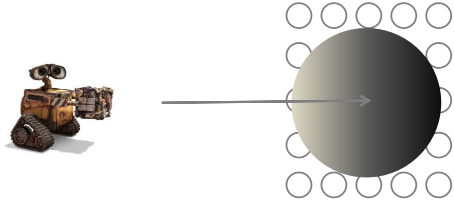
Structured Models

- Motion Planning - Two robots move around an n by n plant



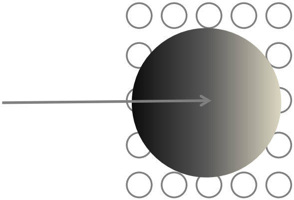
Structured Models

- Motion Planning - Two robots move around an n by n plant



Structured Models

- Motion Planning - Two robots move around an n by n plant



Structured Models

Robot $n = 50$ $r = 1$	θ	0.9	0.95	**0.99**	**PRISM**
	out	F	F	F	**0.999**
	t	23.4	27.5	40.8	1252.7
Robot $n = 50$ $r = 2$	θ	0.9	0.95	**0.99**	**PRISM**
	out	F	F	F	**0.999**
	t	71.7	73.9	250.4	3651.045
Robot $n = 75$ $r = 2$	θ	0.95	0.97	**0.99**	**PRISM**
	out	F	F	F	timeout
	t	382.5	377.1	2676.9	timeout
Robot $n = 200$ $r = 3$	θ	0.85	**0.9**	**0.95**	**PRISM**
	out	F	F	T	timeout
	t	903.1	1129.3	2302.8	timeout

Structured Models

Takeaways

- Exact methods cannot exploit symmetry so much;
- Number of really “meaningful” actions still relatively small;
- SMC works very well in structured systems;

Unstructured Models

- Uniform random model - number of actions enabled follows uniform distribution, number of targets per choice follows uniform distribution, targets picked uniformly, probabilities of transitions uniformely distributed. Objective: as little structure as possible.

Unstructured Models

- Uniform random model - number of actions enabled follows uniform distribution, number of targets per choice follows uniform distribution, targets picked uniformly, probabilities of transitions uniformely distributed. Objective: as little structure as possible.

Results very unpredictable and typically pretty bad.

Unstructured Models

- Uniform random model - number of actions enabled follows uniform distribution, number of targets per choice follows uniform distribution, targets picked uniformly, probabilities of transitions uniformely distributed. Objective: as little structure as possible.

Results very unpredictable and typically pretty bad.

- < 0.3 probability gathered after a few hours with SMC.
- Exact methods fail to produce answers.

Unstructured Models

Takeaways

- Lack of structure makes this problem very hard;
- SMC cannot focus on "good" areas;
- Symbolic methods cannot exploit symmetry when encoding the system.

Conclusions and Future Work

Conclusions

- Statistical method for MC for probabilism + nondeterminism;
- Empirically and theoretically validated;
- Uses bounded memory;
- Efficient for complex but structured models.

Future Work

- Unbounded LTL;
- Distributed systems;
- Schedulers with memory;
- ...

Bibliography

Zuliani P., Platzer A., Clarke E.M.
Bayesian statistical model checking with applications to simulink/stateflow verification.
HSCC, 2010.

Sutton R.S., Barto A.
Reinforcement Learning: An introduction.
MIT Press, 1998.

Brassard G., Bratley P.
Algorithmics - Theory and Practice.
Prentice Hall, 1988.

False Biased Monte Carlo Algorithms

Since our algorithm is false biased (results of "false" are always accurate), we can just run the algorithm again to exponentially increase confidence on a "probably true" result.

Bounding theorem [BB]

If the probability of success of a single trial of a false biased algorithm is greater than

$$p = 1 - 2^{\frac{\log \eta}{T}}$$

where T is the number of iterations of the algorithm, than we can ensure a correcness level of $1 - \eta$, $(0 < \eta < 1)$.

Back

Proof of Improvement Theorem

$$V^{\sigma[\sigma(s)\to\sigma'(s)]}(s)$$

$$= \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a) + (1-\epsilon)\max_{a\in A(s)} Q^\sigma(s,a)$$

$$= \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a) + \left(\sum_{a\in A(s)} \sigma(s,a) - \sum_{a\in A(s)} p_\epsilon(s,a)\right)\max_{a\in A(s)} Q^\sigma(s,a)$$

$$= \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a) + \sum_{a\in A(s)} [\sigma(s,a) - p_\epsilon(s,a)]\max_{a\in A(s)} Q^\sigma(s,a)$$

$$= \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a) + \sum_{a\in A(s)} \left[(\sigma(s,a) - p_\epsilon(s,a))\max_{a\in A(s)} Q^\sigma(s,a)\right]$$

$$\geq \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a) + \sum_{a\in A} [(\sigma(s,a) - p_\epsilon(s,a))Q^\sigma(s,a)]$$

$$= \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a) + \sum_{a\in A(s)} \sigma(s,a)Q^\sigma(s,a) - \sum_{a\in A(s)} p_\epsilon(s,a)Q^\sigma(s,a)$$

$$= \sum_{a\in A(s)} \sigma(s,a)Q^\sigma(s,a) = V^\sigma(s)$$

$$\sigma'(s,a) = (1-h)\left[I\{a = \arg\max_a Q^\sigma(s,a)\}(1-\epsilon) + \epsilon\left(\frac{Q^\sigma(s,a)}{\sum_b Q^\sigma(s,a)}\right)\right] + h\sigma(s,a)$$

Back