Probabilistic Temporal Logic for Reasoning about Bounded Policies

Nima Motamed
Utrecht University

Work with Natasha Alechina, Mehdi Dastani,
Dragan Doder, and Brian Logan

IJCAI
Macau, China
August 2023

Yet another temporal logic. . .

Our intention is to study **intentions**: to build a theory of intention revision under uncertainty

Yet another temporal logic...

Our intention is to study **intentions**: to build a theory of intention revision under uncertainty

Reasoning about such intentions requires an appropriate probabilistic temporal logic, allowing us to explicitly reason about the execution and executability of the agent's **actions**

But well-known (**infinite-trace**) ones like PCTL, pCTL*, PATL/PATL* or Probabilistic Strategy Logic do not have this! Plus: they have high complexity (sometimes undecidable) **model checking**, and the decidability of **satisfiability** is still open

The idea

Consider **finite traces** and **bounded policies** - a policy/strategy that holds for a certain number of steps.

The idea

Consider **finite traces** and **bounded policies** - a policy/strategy that holds for a certain number of steps.

We introduce the **Probabilistic Logic of Bounded Policies (PLBP)**

PLBP is expressive enough for our needs (and can in fact express properties important for other AI applications), and it has good computational properties.

Markov Decision Processes and bounded policies

Fix a finite set $\mathcal{A}$ of **actions**, and for every $a \in \mathcal{A}$ a **precondition** pre_a and finite **set of postconditions** Post_a - these are conjunctions of literals.

Definition

An **MDP** is a tuple $\mathbb{M} = \langle S, P, V \rangle$, where S is a set of *states*, $P : S \times \mathcal{A} \rightsquigarrow \Delta(S)$ is the *partial probabilistic transition function*, and $V : S \to 2^{\text{Prop}}$ is the *valuation*.

(+ coherence conditions ensuring that pre- and postconditions are meaningful: pre_a *holds iff a is executable,* Post_a *corresponds precisely to outcomes of a at each state)*

Markov Decision Processes and bounded policies

Fix a finite set $\mathcal{A}$ of **actions**, and for every $a \in \mathcal{A}$ a **precondition** pre_a and finite **set of postconditions** Post_a - these are conjunctions of literals.

Definition

An **MDP** is a tuple $\mathbb{M} = \langle S, P, V \rangle$, where S is a set of *states*, $P : S \times \mathcal{A} \rightsquigarrow \Delta(S)$ is the *partial probabilistic transition function*, and $V : S \to 2^{\mathrm{Prop}}$ is the *valuation*.

(+ coherence conditions ensuring that pre- and postconditions are meaningful: pre_a *holds iff a is executable,* Post_a *corresponds precisely to outcomes of a at each state)*

Definition

For $n \geqslant 1$, an ***n*-step policy from a state *s*** is a function $\pi : S_s^{\leqslant n} \to \mathcal{A}$ (where $S_s^{\leqslant n}$ is the set of all length $\leqslant n$ sequences from s) such that $s_k \models \mathrm{pre}_{\pi(s_1 \cdots s_k)}$.

Syntax & semantics

State formulas:

$$\varphi ::= x(\in \text{Prop}) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Diamond^n_{\bowtie r}\Phi^n$$

here, $n \geqslant 1$, $r \in [0,1]$ *and* $\bowtie \in \{<,=,>\}$

For states: $s \models \Diamond^n_{\bowtie r}\Phi$ iff there is an n-step policy π from s such that under the policy, the probability that the next n steps of states and actions satisfies Φ is $\bowtie r$

Syntax & semantics

State formulas:

$$\varphi ::= x (\in \mathsf{Prop}) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Diamond^n_{\bowtie r} \Phi^n$$

here, $n \geqslant 1$, $r \in [0,1]$ *and* $\bowtie \in \{<, =, >\}$

For states: $s \models \Diamond^n_{\bowtie r}\Phi$ iff there is an n-step policy π from s such that under the policy, the probability that the next n steps of states and actions satisfies Φ is $\bowtie r$

n-step path formulas (defined inductively w.r.t. n):

$$\Phi^0 ::= \varphi \quad \Phi^{n+1} ::= \varphi \mid \mathsf{do}_a (a \in A) \mid \Phi^{n+1} \wedge \Phi^{n+1} \mid \neg\Phi^{n+1} \mid \mathsf{X}\Phi^n$$

Interpreted over **state-action paths** $\mathbf{w} = s_1 a_1 \cdots s_n a_n s_{n+1}$, with $\mathbf{w} \models \mathsf{do}_a$ iff $a_1 = a$, and $\mathbf{w} \models \mathsf{X}\Phi$ iff $s_2 a_2 \cdots s_{n+1} \models \Phi$

What can we express

- $$\Box^{n}_{=1} \mathsf{X}^{n} \Diamond^{m}_{>0.5} \bigvee_{0 \leqslant i \leqslant m} \mathsf{X}^{i} \neg \mathsf{safe}$$

  "No matter what the agent does in the next *n* steps, it is certain that they can afterwards act in such a way that they will most likely be in danger"

What can we express

- $$\Box^{n}_{=1} X^{n} \Diamond^{m}_{>0.5} \bigvee_{0 \leqslant i \leqslant m} X^{i} \neg \mathrm{safe}$$

  "No matter what the agent does in the next n steps, it is certain that they can afterwards act in such a way that they will most likely be in danger"
- We can reason about (abstractions of) **specific policies / conditional plans**: e.g. consider a 2-step policy saying to do a now, and afterwards b_1 if we got the first postcondition of a, otherwise b_2. The formula

  $$\mathrm{exec} = \Diamond^{2}_{=1}(\mathrm{do}_a \wedge \bigwedge_{i=1,2} X(\mathrm{post}_{a,i} \rightarrow \mathrm{do}_{b_i}))$$

  states that this policy is executable. The formula

  $$\mathrm{exec} \wedge \Diamond^{2}_{>0.5}(\mathrm{do}_a \wedge (\bigwedge_{i=1,2} X(\mathrm{post}_{a,i} \rightarrow \mathrm{do}_{b_i})) \wedge (\mathrm{safe} \wedge X\mathrm{safe}))$$

  states the policy is executable, and most likely leads to a safe state

What can we express

- Following Shoham[1] in considering basic intentions to be pairs (a, t) denoting 'the agent intends to do a at time t', **coherence** of a set I of such intentions with respect to a set Γ of formulas representing the agent's beliefs is stating that

$$\Gamma \cup \{\Diamond_{>0}^{t_{\max}} \bigwedge_{(a,t)\in I} \mathrm{X}^t \mathrm{do}_a\} \quad (\text{where } t_{\max} = \max_{(a,t)\in I} t)$$

is **satisfiable** - the agent does not believe that their intentions are not realizable

[1] Shoham, Yoav (2009), "Logical Theories of Intention and the Database Perspective"

Model checking

Model checking asks given MDP M, state s and φ, if $M, s \models \varphi$

Model checking

Model checking asks given MDP M, state s and φ, if $M, s \models \varphi$

Membership in **PSPACE** is shown using an **nondeterministic** polyspace algorithm that labels states by the subformulas they satisfy: for $\Diamond^n_{\bowtie r}\Phi$, traverse the MDP by **guessing** actions to take. PSPACE follows from PSPACE = NPSPACE by Savitch's theorem[2]

[2] Savitch, Walter J. (1970), "Relationships between nondeterministic and deterministic tape complexities"

Model checking

Model checking asks given MDP M, state s and φ, if $M, s \models \varphi$

Membership in **PSPACE** is shown using an **nondeterministic** polyspace algorithm that labels states by the subformulas they satisfy: for $\Diamond^{n}_{\bowtie r}\Phi$, traverse the MDP by **guessing** actions to take. PSPACE follows from PSPACE = NPSPACE by Savitch's theorem[2]

It is also **PSPACE-hard** by a reduction from QSAT inspired by Bulling & Jamroga (2010)[3]

[2]Savitch, Walter J. (1970), "Relationships between nondeterministic and deterministic tape complexities"

[3]Bulling, Nils and Jamroga, Wojciech (2010), "Verifying Agents with Memory is Harder Than It Seemed"

Satisfiability

Satisfiability asks given φ, whether there exist M and s with $M, s \models \varphi$

Can be used to check **coherence of intentions**, compute **logical consequences**, or model check **partially specified MDPs**

[4] Decidable in PSPACE by Canny, John (1988), "Some algebraic and geometric computations in PSPACE."

Satisfiability

Satisfiability asks given φ, whether there exist M and s with $M, s \models \varphi$

Can be used to check **coherence of intentions**, compute **logical consequences**, or model check **partially specified MDPs**

Satisfiability is decidable in **2-EXPSPACE**:

- PLBP has the **finite model property**: satisfiable φ is satisfied in MDP of size $\leqslant b_\varphi$
- Algorithm iterates over S of size $\leqslant b_\varphi$ and V
- Determines whether there is P and s with $(S, P, V), s \models \varphi$ by checking whether a certain **existential first-order logic** sentence is valid in the theory of **real closed fields**[4]

[4] Decidable in PSPACE by Canny, John (1988), "Some algebraic and geometric computations in PSPACE."

Wrapping up

Summing up:

- We have a logic allowing us to reason about **finite traces** and **bounded policies**
- The logic can express important properties for AI, while still having **strong computational properties**

Wrapping up

Summing up:

- We have a logic allowing us to reason about **finite traces** and **bounded policies**
- The logic can express important properties for AI, while still having **strong computational properties**

Work being done:

- We have a definition of **bisimilarity** with soundness & completeness: what are applications in e.g. MDP minimization
- Defining and axiomatizing **belief and intention revision operators** using PLBP
- Developing a **quantitative extension** incorporating reward signals, for applications in Reinforcement Learning

Wrapping up

Summing up:

- We have a logic allowing us to reason about **finite traces** and **bounded policies**
- The logic can express important properties for AI, while still having **strong computational properties**

Work being done:

- We have a definition of **bisimilarity** with soundness & completeness: what are applications in e.g. MDP minimization
- Defining and axiomatizing **belief and intention revision operators** using PLBP
- Developing a **quantitative extension** incorporating reward signals, for applications in Reinforcement Learning

Thank you!

n.motamed@uu.nl — https://motamed.nl