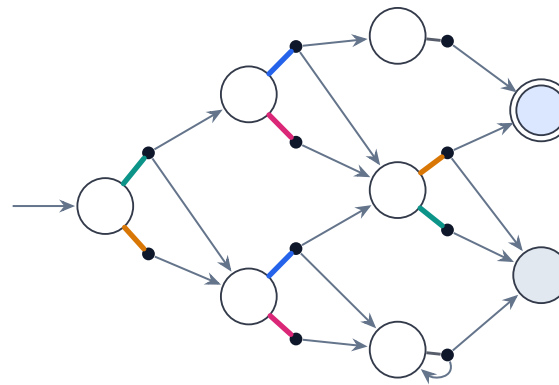
# PAYNT

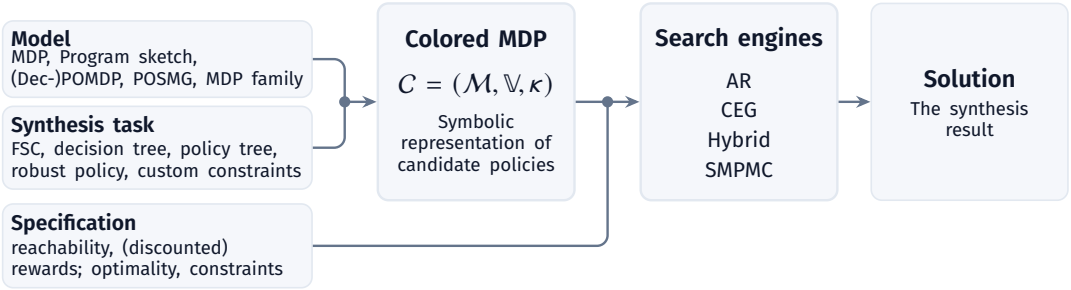
Tool (not only) for
Constrained Policy Synthesis in MDPs

**Filip Macák**
Brno University of Technology



BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

# PAYNT Overview




**Key idea:** Colored MDP compactly represents a set of feasible MDP policies. Dedicated search engines find a policy satisfying the specification.

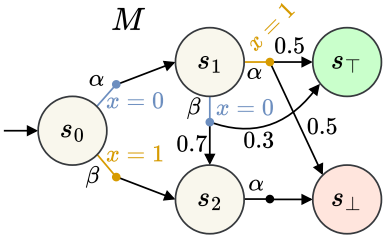
pip install paynt

Built on top of **Storm**

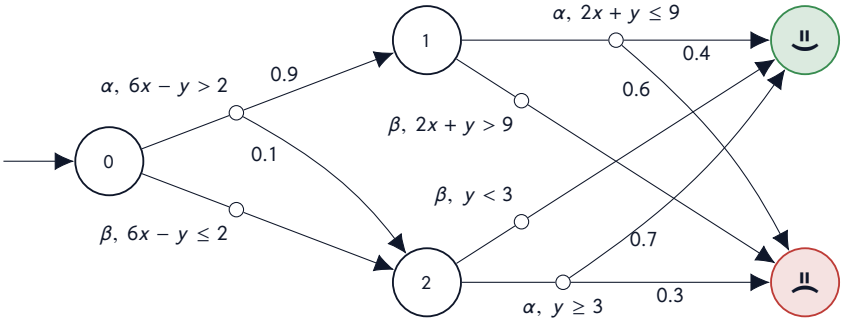
# Colored MDP

Colored MDP $\mathcal{C} = (\mathcal{M}, \mathbb{V}, \kappa)$

- $\mathcal{M}$ is an MDP
- $\mathbb{V}$ is the (restricted) parameter space
- $\kappa(s, \alpha) \subseteq \mathbb{V}$ is the coloring function

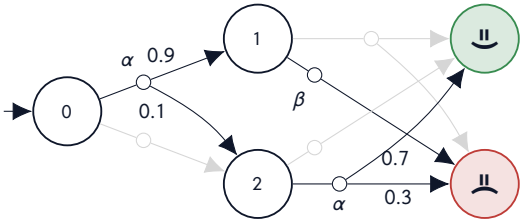


# Colored MDP - Advanced example



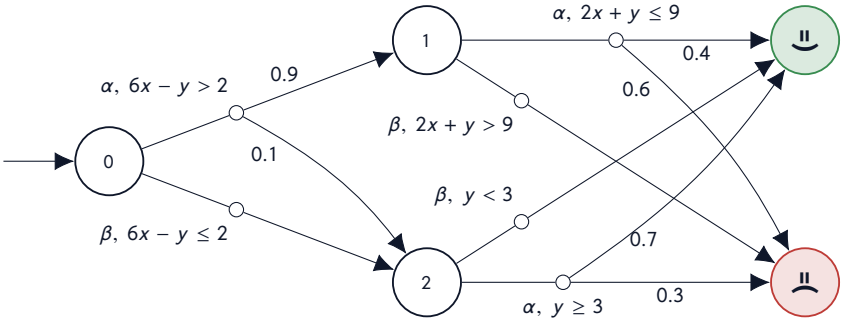

$$\mathbb{V} = (1 \leq x \leq 10) \wedge (1 \leq y \leq 5)$$

# Colored MDP - Advanced example




$x = 4, y = 5$

# Colored MDP - Advanced example




$$\mathbb{V} = (1 \leq x \leq 10) \wedge (1 \leq y \leq 5)$$

# Search Engines

## Abstraction refinement

- uses the underlying MDP $M$ as an abstraction
- $V^{\mathcal{M}}_{\text{min}} \leq V^{\mathcal{M}^\theta} \leq V^{\mathcal{M}}_{\text{max}}$ for all $\theta \in \mathbb{V}$

## Hybrid

- dynamically switches between AR and CEG

## CEG

- full assignment $\theta$ induces MC $\mathcal{M}^\theta$
- if $\mathcal{M}^\theta$ violates specification, we can construct a counter-example $G(\theta)$

## SMPMC

- Satisfiability Modulo Probabilistic Model Checking
- search space handled by an SMT solver and solving the MDP dynamics handled by PMC

# Out-of-the-box Features

## 1. Program sketches

PRISM programs with holes

TACAS'21 CAV'21 JAIR'25

## 2. FSC synthesis

FSCs for (Dec-)POMDPs, POSGs

UAI'22 CAV'23 AAMAS'25

## 3. Policy trees

set of policies for family of MDPs

ATVA'24

## 4. Decision trees

DT policies for MDPs

CAV'25 UAI'25

## 5. Robust policies

policies that work in a set of environments

AAAI'26

## 6. Custom constraints

support for custom constraint formulation

AAAI'26 WIP

# Roadmap

- updated installation ✓
- getting all of the published features into PAYNT releases:
  - dtNEST ✓
  - multi-constraint specifications ✓
  - SMPMC and robust policy synthesis ✓ (not merged yet)
  - SAYNT WIP
  - Conditional reachability properties WIP
- python code refactor ✓
- support for general coloring functions WIP (mostly done)
- examples of how to use paynt (Oct 2026)
- CLI ease-of-use update (Nov 2026)
- unified export formats for results (Nov 2026)
- C++ code refactor (Dec 2026)
- even more user-friendly API (Dec 2026)

# WORLD PREMIERE

# PAYNT 1.0

Q1 2027

imagine if we had a cool logo :(

# Conclusions

1. **One framework**: many interesting features available out-of-the-box
2. **Active development**: Feature requests are welcome!
3. **Coming soon**: define your own colored MDPs directly in Python
4. **Developers and users** of PMC tools, let's stay in touch

# Thank you!

**Try PAYNT**



github.com/randriu/paynt

**Join the PMC Tools Zulip**



pmc-tools.zulipchat.com

# References


TACAS'21 Andriushchenko, Češka, Junges, Katoen. *Inductive synthesis for probabilistic programs reaches new horizons.*

CAV'21 Andriushchenko, Češka, Junges, Katoen, Stupinský. *PAYNT: a tool for inductive synthesis of probabilistic programs.*

UAI'22 Andriushchenko, Češka, Junges, Katoen. *Inductive synthesis of finite-state controllers for POMDPs.*

CAV'23 Andriushchenko, Bork, Češka, Junges, Katoen, Macák. *Search and explore: symbiotic policy synthesis in POMDPs.*

ATVA'24 Andriushchenko, Češka, Junges, Macák. *Policies grow on trees: model checking families of MDPs.*

JAIR'25 Andriushchenko, Češka, Macák, Junges, Katoen. *An oracle-guided approach to constrained policy synthesis under uncertainty.*

AAMAS'25 Pontiggia, Macák, Andriushchenko, Chiari, Češka. *Decentralized planning using probabilistic hyperproperties.*

CAV'25 Andriushchenko, Češka, Junges, Macák. *Small decision trees for MDPs with deductive synthesis.*

UAI'25 Andriushchenko, Češka, Chakraborty, Junges, Křetínský, Macák. *Symbiotic local search for small decision tree policies in MDPs.*

AAAI'26 Heck, Macák, Češka, Junges. *Constrained and robust policy synthesis with satisfiability-modulo-probabilistic-model-checking.*