PAYNT: A Framework for Controller Synthesis Under Uncertainty

Filip Macák Roman Andriuschenko Milan Češka Sebastian Junges

Motivation

Interesting problems arise from considering different kinds of uncertainty

- probabilistic outcomes
- what if the current state is not observable?
- uncontrollable nondeterminism

We want to put additional constraints on the synthesized policies

- cost constrained
- FSC or DT representation

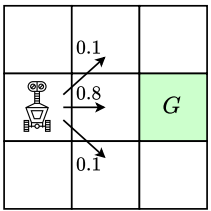
Motivation

Interesting problems arise from considering different kinds of uncertainty

- probabilistic outcomes
- what if the current state is not observable?
- uncontrollable nondeterminism

We want to put additional constraints on the synthesized policies

- cost constrained
- FSC or DT representation



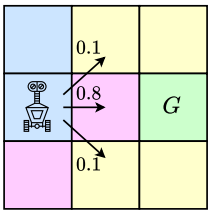
Motivation

Interesting problems arise from considering different kinds of uncertainty

- probabilistic outcomes
- what if the current state is not observable?
- uncontrollable nondeterminism

We want to put additional constraints on the synthesized policies

- cost constrained
- FSC or DT representation



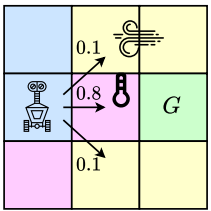
Motivation

Interesting problems arise from considering different kinds of uncertainty

- probabilistic outcomes
- what if the current state is not observable?
- uncontrollable nondeterminism

We want to put additional constraints on the synthesized policies

- cost constrained
- FSC or DT representation



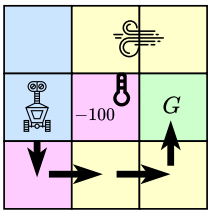
Motivation

Interesting problems arise from considering different kinds of uncertainty

- probabilistic outcomes
- what if the current state is not observable?
- uncontrollable nondeterminism

We want to put additional constraints on the synthesized policies

- cost constrained
- FSC or DT representation



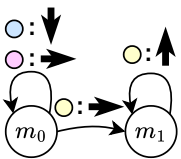
Motivation

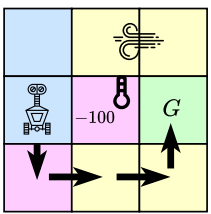
Interesting problems arise from considering different kinds of uncertainty

- probabilistic outcomes
- what if the current state is not observable?
- uncontrollable nondeterminism

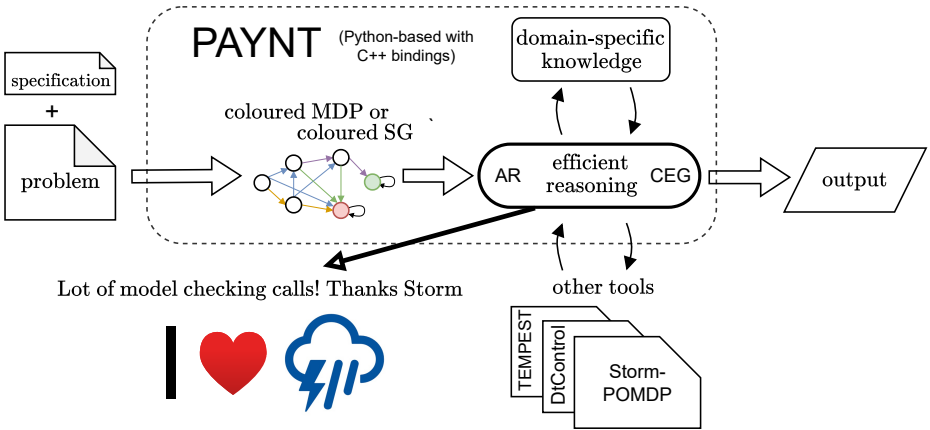
We want to put additional constraints on the synthesized policies

- cost constrained
- FSC or DT representation





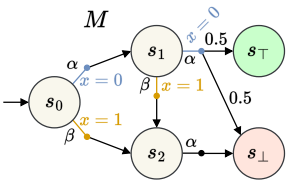
Available via our synthesis framework PAYNT [CAV'21, JAIR'25]



Coloured MDP

Formally coloured MDP $\mathcal{M} = (M, \Delta, \kappa)$

- M - underlying (quotient) MDP
- Δ - design space of parameter assignments
- κ - colouring $\kappa : S \times Act \to \overline{\Delta}$
- $\forall \delta \in \Delta, \forall s \in S, \exists! \alpha \in Act(s) : \kappa(s, \alpha) \sim \delta$



$H = \{x\}$

Design space Δ

$\delta_1 : x = 0$

$\delta_2 : x = 1$

Note this object can be extended to coloured SG

- Informally, the underlying MDP represents a probabilistic model of possibly multiple environments.
- Each action in the MDP corresponds to a partial assignments of parameters.
- We are interested in those policies in the MDP such that the assignment of parameters is consistent.

Analysis of coloured MDP

Abstraction Refinement (AR)

- consistent policies in coloured MDP correspond to total assignments
- uses the underlying MDP M as an abstraction
- $V^M_{\min} \leq \min_{\delta \in \Delta} V^{M^\delta} \leq \max_{\delta \in \Delta} V^{M^\delta} \leq V^M_{\max}$
- refine the abstraction by removing actions from M

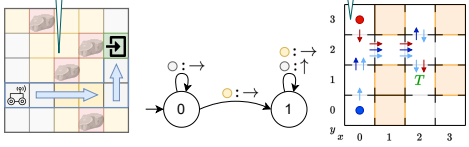
Counterexample Generalisation (CEG)

- candidate assignment $\delta \rightarrow$ induced DTMC M^δ
- if δ violates specification, a counterexample is constructed
- counterexample $G(\delta) \subseteq \Delta$ contains assignments that also violate the specification

Applications & Recent results

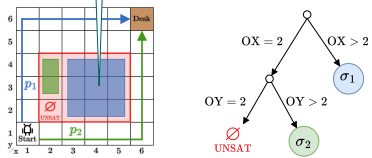
Partial observability

Decentralised agents via a **hyperproperty** spec



Synthesis of finite state controllers for POMDPs (UAI'22, CAV'23) and for decentralised planning (AAMAS'25)

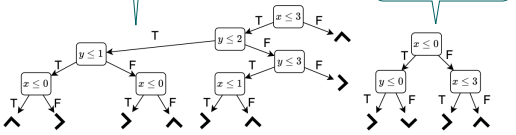
Uncertainty about the model topology



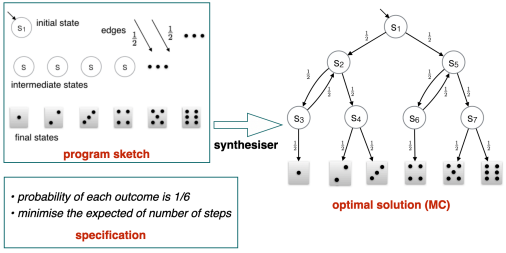
Synthesis of policy trees for multiple-environment MDPs (ATVA'24)

DT found by dtControl for an optimal policy

Our smaller DT achieves 96% of the optimal value



Synthesis of small almost optimal decision trees for MDPs (CAV'25, UAI'25)



Synthesis of finite-state probabilisitic programs from sketches (TACAS'21, JAIR'25)

What's next?

Immediate future:

- Support for OS-POSGs and their use cases
- Combination of policy trees and decision trees

My longer-term goals:

- Improved POMDP specific algorithms
- Synthesis of robust policies
- Focus on explainable policies
- Make PAYNT more user-friendly and able to handle more general problems

RL + PAYNT

Combining the scalability of reinforcement learning and the robustness of PAYNT

Goals:

- RL policy verification through scalable policy extraction
- Robust RL-PAYNT training loop
- Removing the main limitation of model-based methods – knowledge of the user-defined model

Summary

- PAYNT builds on top of the Storm model checker and extends it for synthesis under various uncertainties and structural constraints
- The central object is the so-called coloured MDP
- PAYNT contains many useful features!

Check out PAYNT: `https://github.com/randriu/synthesis`

(In case of any problems please contact: imacak@fit.vut.cz)