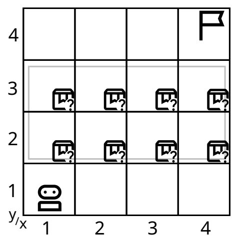
Decision Trees for Multi-Environment Markov Decision Processes

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Author: Ing. Ladislav Dokoupil   Supervisor: doc. RNDr. Milan Češka, Ph.D.

Motivation

Automated decision-making in uncertain environments is crucial for modern systems. Markov Decision Processes (MDPs) provide a mathematical framework for modeling such decisions. Many real-world systems exhibit parametric uncertainty or variable operating conditions, such as robots navigating different environments or autonomous vehicles adapting to changing conditions. **Our goal is to create decision systems that are not only correct across all these variations but also compact and understandable by humans.**



Robot navigating a maze with variable obstacle positions, seeking to exit safely. (*slippery environment*)

Current Approaches and Challenges

Existing methods for creating decision rules for families of related problems often produce controllers that are:

- **Overly Conservative:** They are designed for worst-case scenarios, including many unnecessary decision points.
- **Difficult to Understand:** They are typically represented in tabular formats that humans cannot easily interpret.
- **Redundant:** Similar decision logic is duplicated across multiple controllers for different scenarios.

Experimental Evaluation

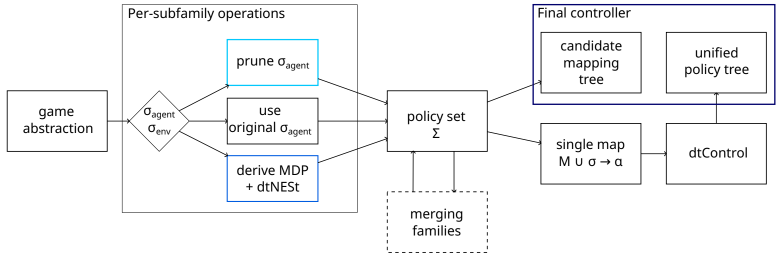
We evaluated our approach on complex models with up to a million variations, each having thousands of states. These include robot navigation, network protocols, and resource allocation scenarios. Our experiments show:

- **Controllers up to 10× smaller** than naive methods while maintaining correctness guarantees.
- Significant reduction in redundant decision logic.

For example, a robot navigating terrain with eight possible obstacle layouts requires just two compact decision trees instead of eight separate controllers, making implementation feasible on resource-constrained devices.

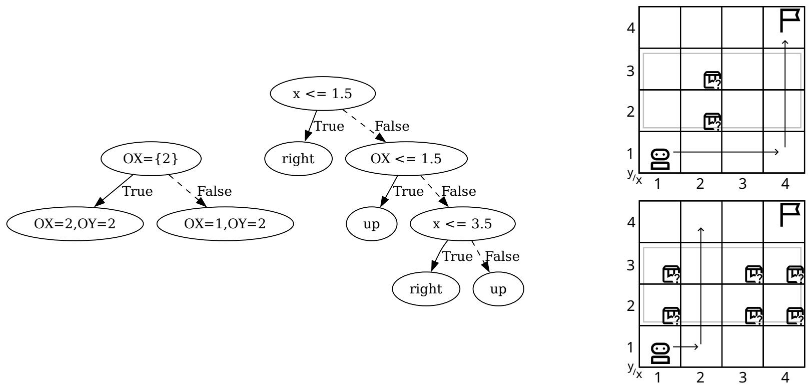
Methodology

We developed a comprehensive framework for synthesizing compact, understandable controllers for families of decision problems:



- **State Pruning Algorithms:** We systematically eliminate irrelevant states and actions from initial controllers while preserving correctness guarantees across all variations.
- **Novel Problem Transformation:** We transform the complex family problem into a derived form that enables the application of advanced synthesis tools, generating more compact controllers that remain robust.
- **Two-Level Decision Structure:** We create a unified representation with two components. One selects strategies based on environmental parameters, the other determines actions based on the selected strategy and current state.

Strategy Visualization



Two-tree solution: left tree selects strategy based on obstacles, right tree determines actions based on robot position. (*slippery environment*)