Towards Best-Effort Autonomy

Rüdiger Ehlers
University of Bremen

Dagstuhl Seminar 17071, February 2017

Based on Joint work with Salar Moarref & Ufuk Topcu (CDC 2016)

Motivation

Highly autonomous systems...

- ... degrade in performance over time
- ... need to work correctly in off-nominal conditions
- ... need to adapt without the need of a human operator

Motivation

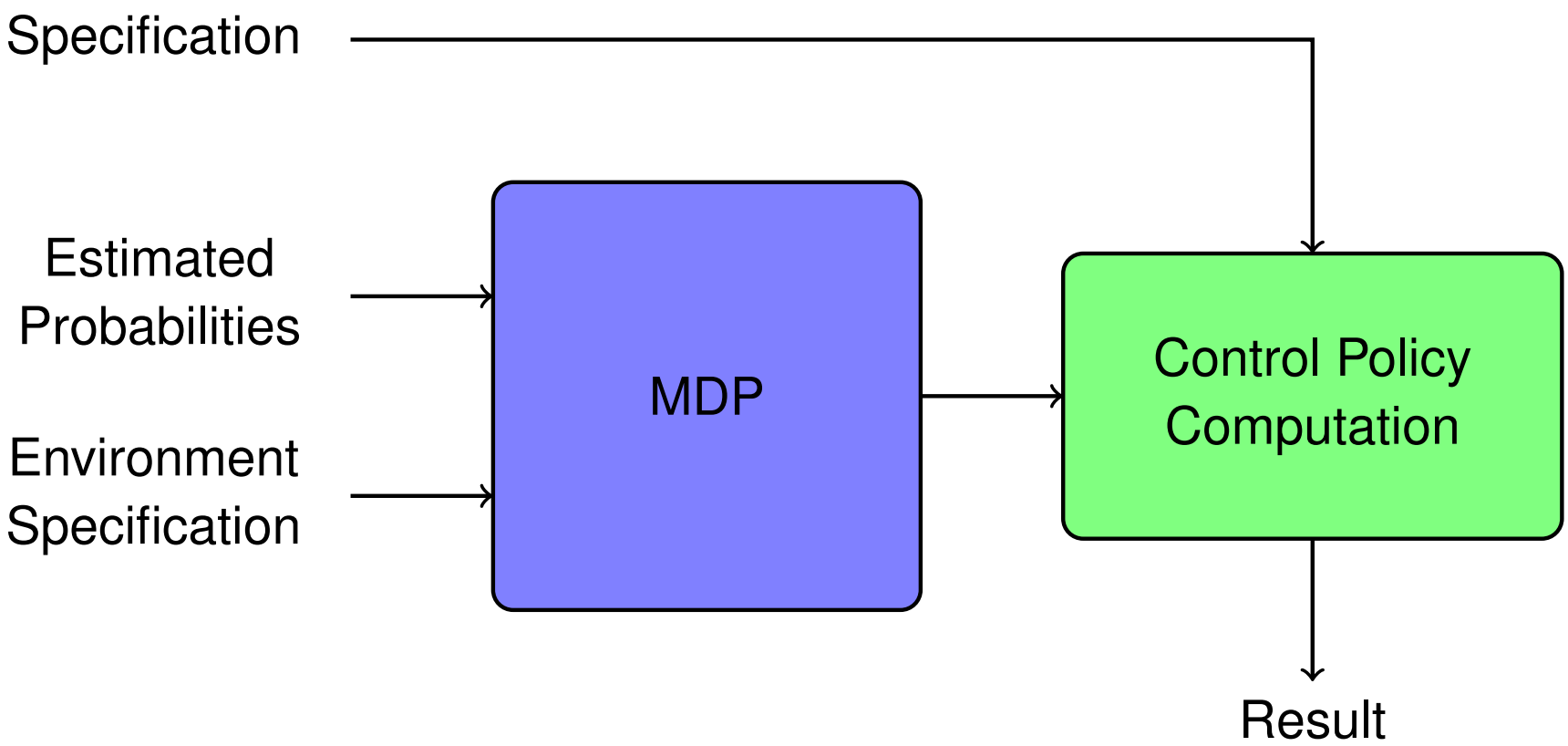
Highly autonomous systems...

- ... degrade in performance over time
- ... need to work correctly in off-nominal conditions
- ... need to adapt without the need of a human operator

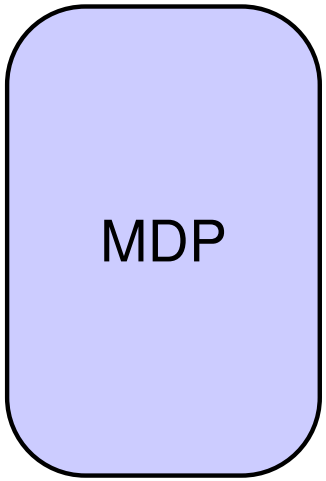
Problem:

- We do not always know in advance how they are degrading...
- ...so we should be able to *synthesize an adapted strategy* in the field

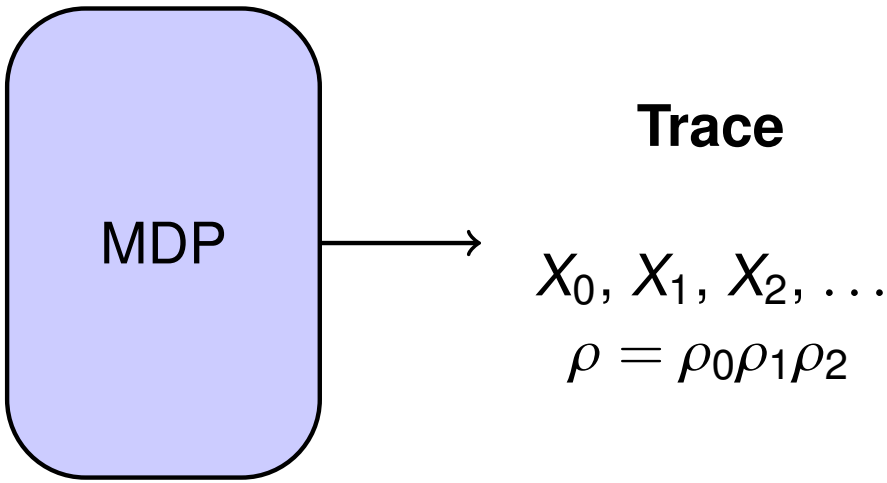
Connecting Theory and Practice....



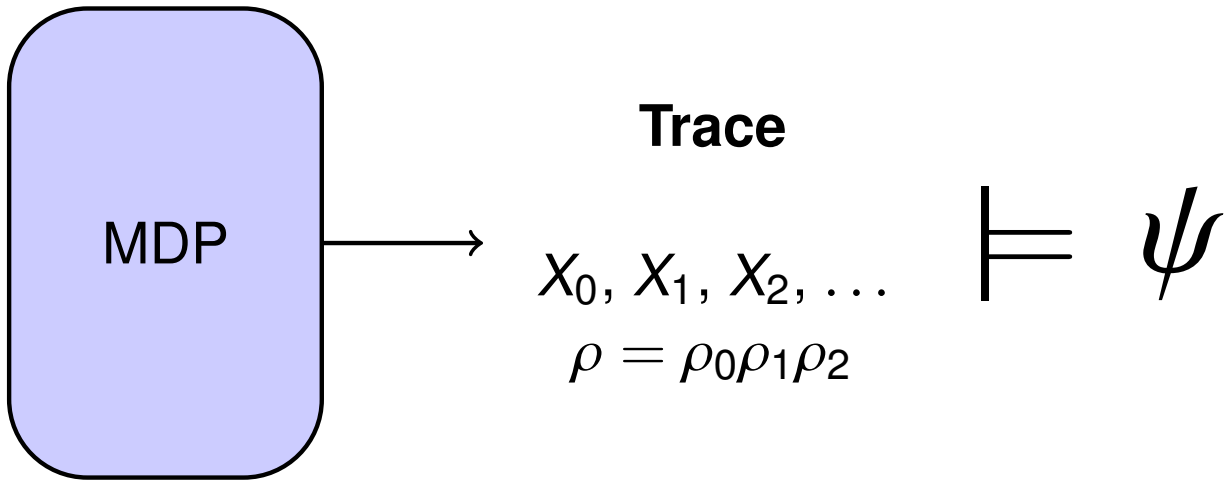
ω-regular control of MDPs – basic setting



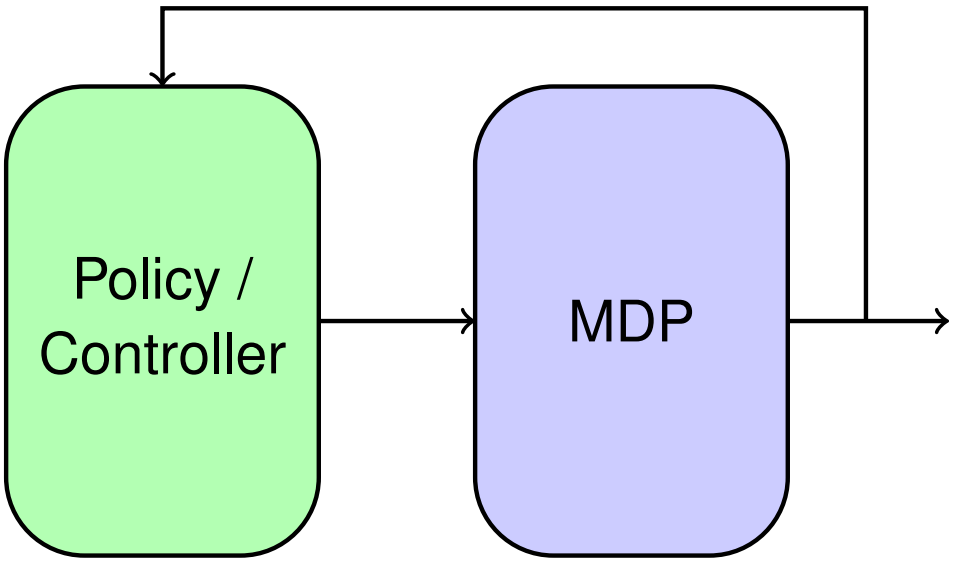
ω-regular control of MDPs – basic setting



ω-regular control of MDPs – basic setting



ω-regular control of MDPs – basic setting



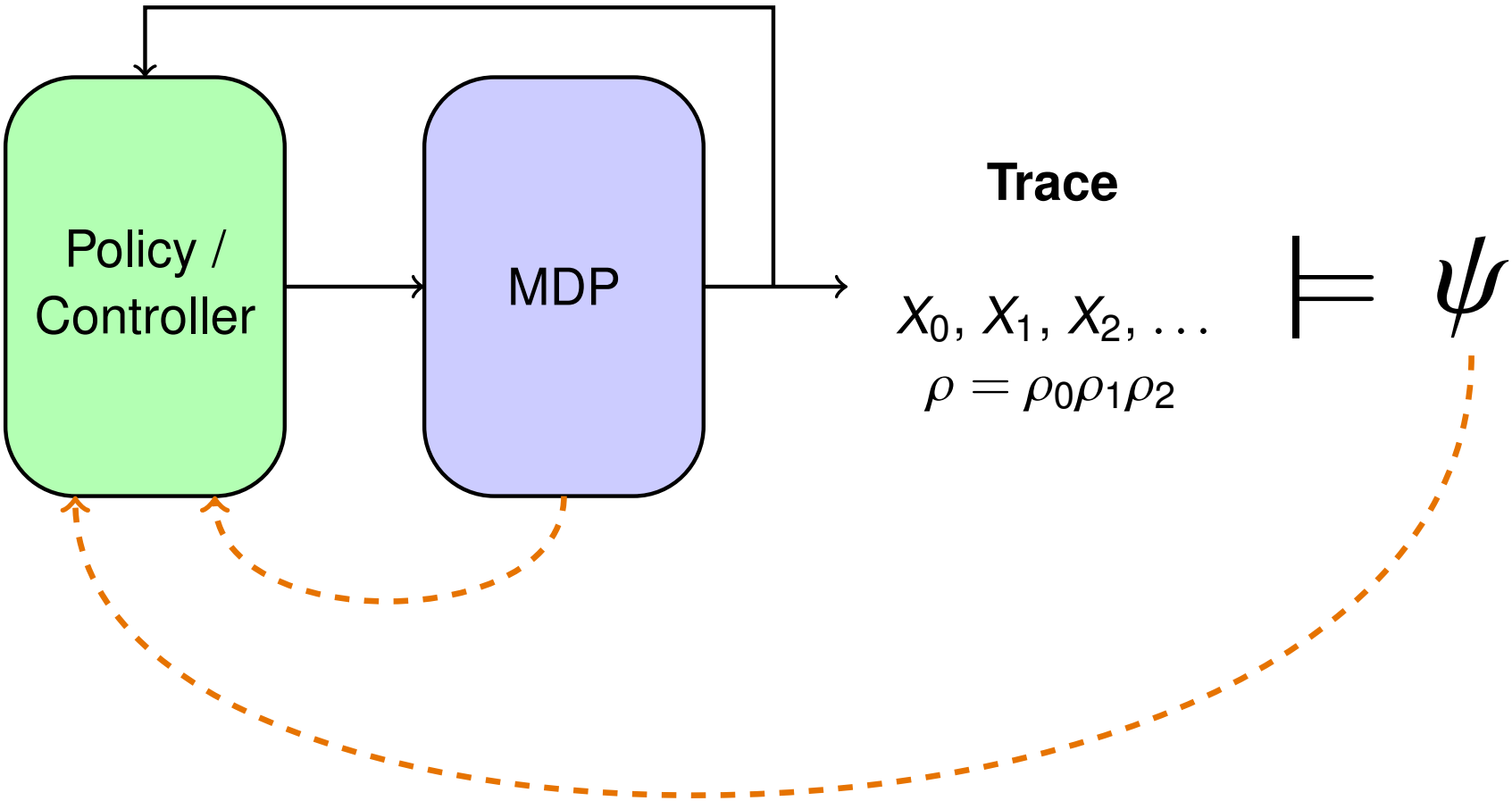
Trace

$X_0, X_1, X_2, \ldots$

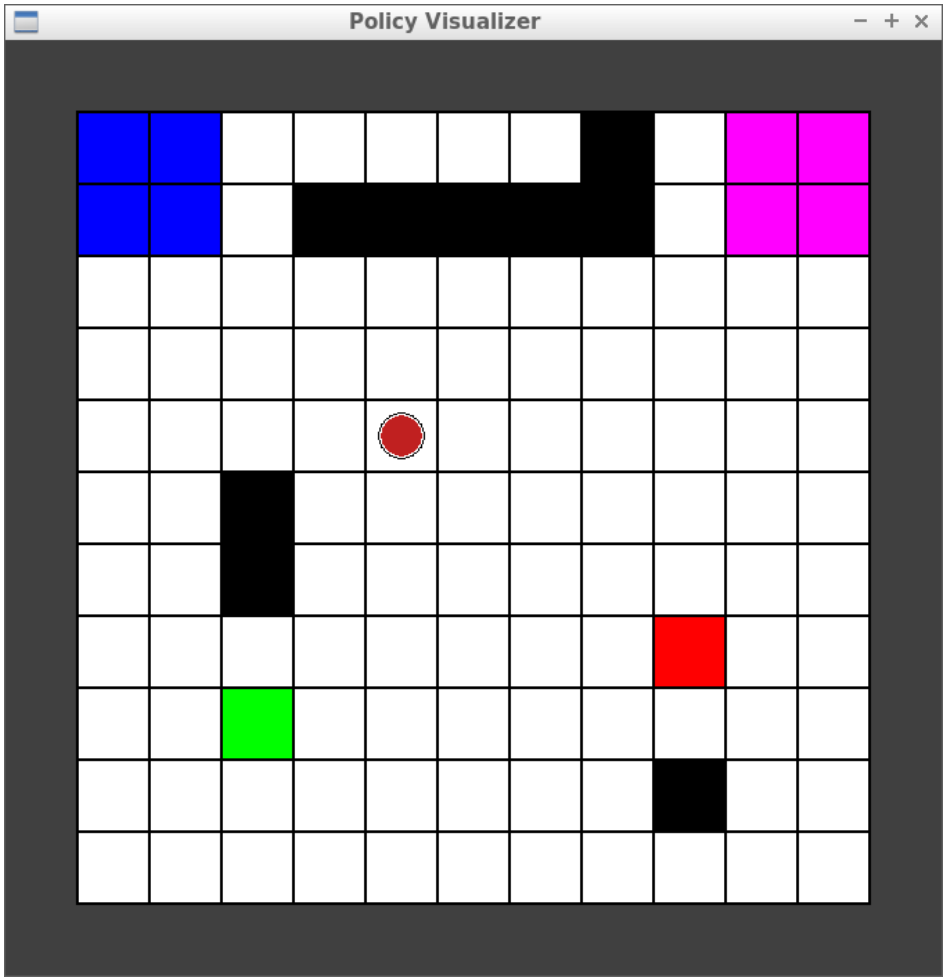
$\rho = \rho_0\rho_1\rho_2$

$\models \psi$

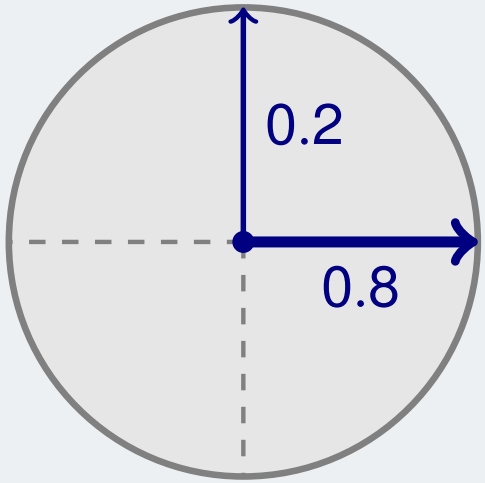
ω-regular control of MDPs – basic setting



Simple example: patrolling



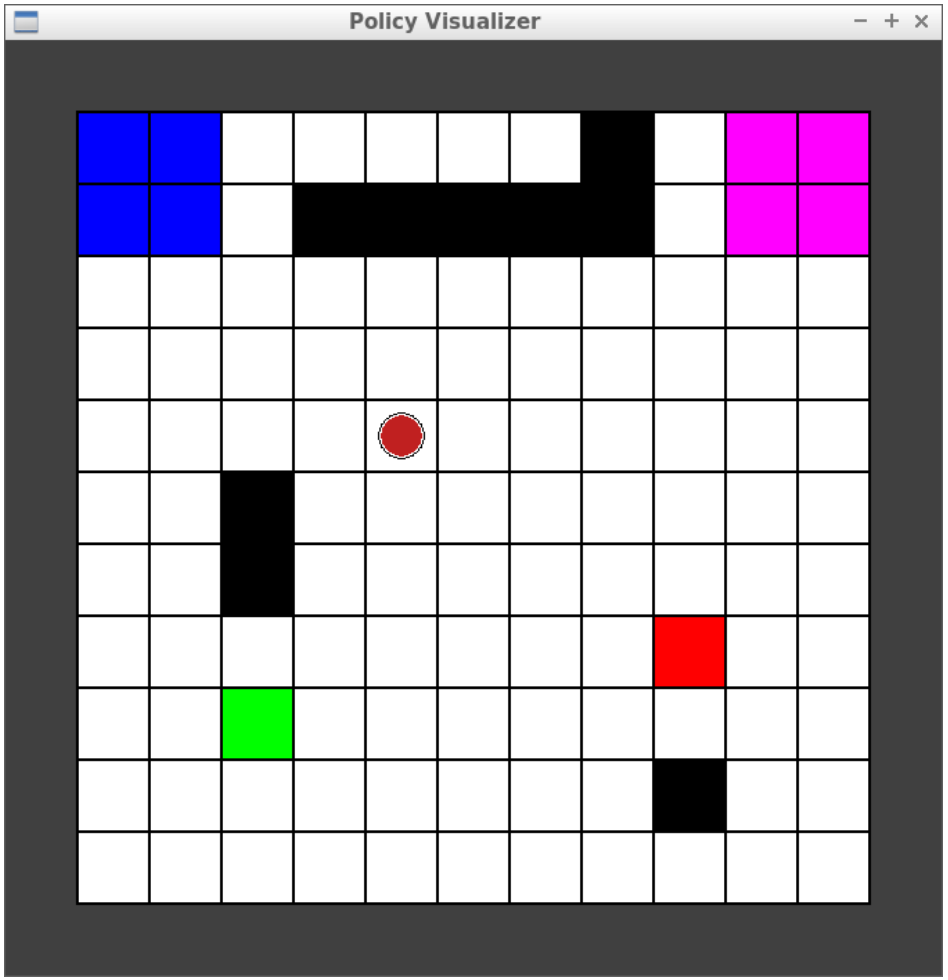
Motion primitives



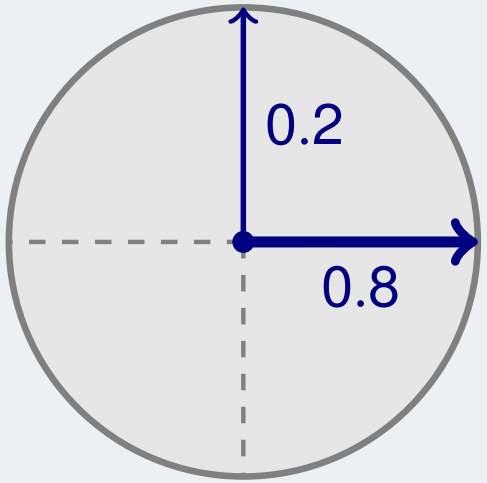
Specification

$$\mathbf{GF}(\mathit{green}) \wedge \mathbf{GF}(\mathit{red}) \wedge \mathbf{GF}(\mathit{purple}) \wedge \mathbf{GF}(\mathit{blue})$$

Simple example: patrolling



Motion primitives



Specification

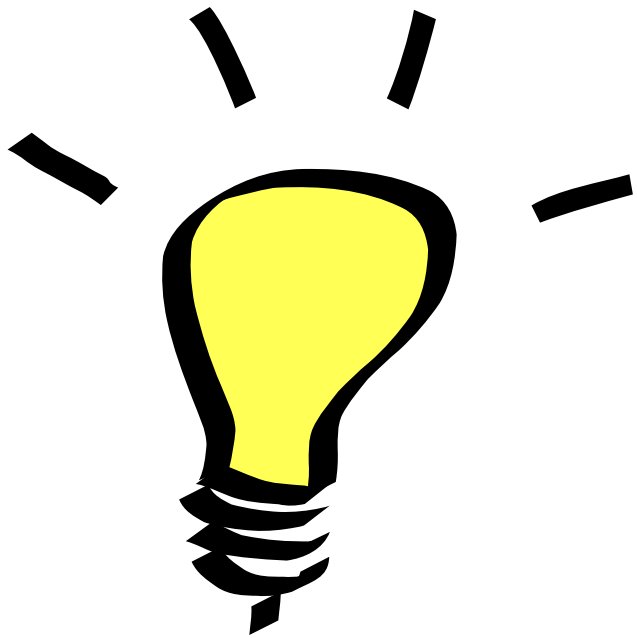
$$\begin{aligned} & \mathbf{GF}(\mathit{green}) \\ \wedge\ & \mathbf{GF}(\mathit{red}) \\ \wedge\ & \mathbf{GF}(\mathit{purple}) \\ \wedge\ & \mathbf{GF}(\mathit{blue}) \end{aligned}$$

$$\mathcal{P}(\rho \models \psi) \geq (0.8)^4$$

Using ω-regular specifications

Ideas

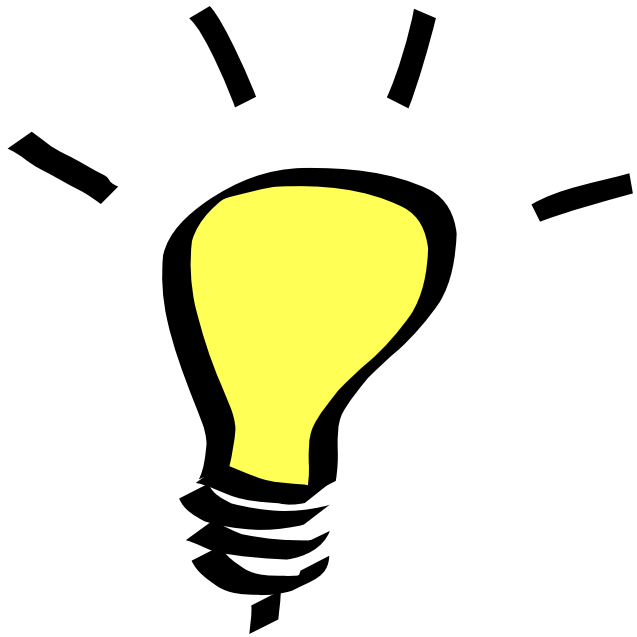
- By assuming that **traces are infinitely long**, we can abstract from an unknown time until the system goes *out of service*.



Using ω-regular specifications

Ideas

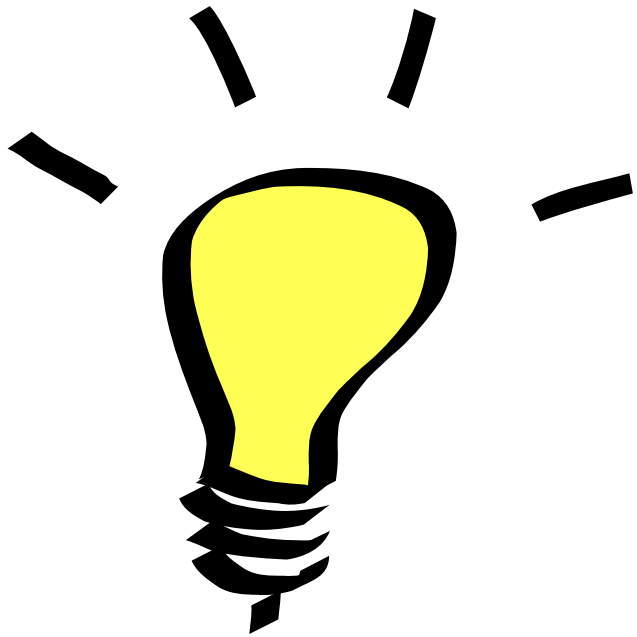
- By assuming that **traces are infinitely long**, we can abstract from an unknown time until the system goes *out of service*.
- Using **temporal logic** for the specification with operators such as “finally” and “globally”, we do not need to set time bounds for reaching the *system goals*, which **helps with maximizing the probability for a trace to satisfy the specification**.



Using ω-regular specifications

Ideas

- By assuming that **traces are infinitely long**, we can abstract from an unknown time until the system goes *out of service*.
- Using **temporal logic** for the specification with operators such as “finally” and “globally”, we do not need to set time bounds for reaching the *system goals*, which **helps with maximizing the probability for a trace to satisfy the specification**.
- ω-regular specifications allow us to specify relatively complex behaviors easily.



But do ω-regular specifications always make sense?

A thought experiment

- Assume that a robot has to patrol between two regions (i.e., it needs to visit both regions infinite often)

But do ω-regular specifications always make sense?

A thought experiment

- Assume that a robot has to patrol between two regions (i.e., it needs to visit both regions infinite often)
- At every second, $\mathcal{P}(\textit{robot breaks}) > 10^{-10}$.

But do ω-regular specifications always make sense?

A thought experiment

- Assume that a robot has to patrol between two regions (i.e., it needs to visit both regions infinite often)
- At every second, $\mathcal{P}(\textit{robot breaks}) > 10^{-10}$.
- What is the maximum probability of satisfying the specification that some control policy can achieve?

But do ω-regular specifications always make sense?

A thought experiment

- Assume that a robot has to patrol between two regions (i.e., it needs to visit both regions infinite often)
- At every second, $\mathcal{P}(\textit{robot breaks}) > 10^{-10}$.
- What is the maximum probability of satisfying the specification that some control policy can achieve? It's

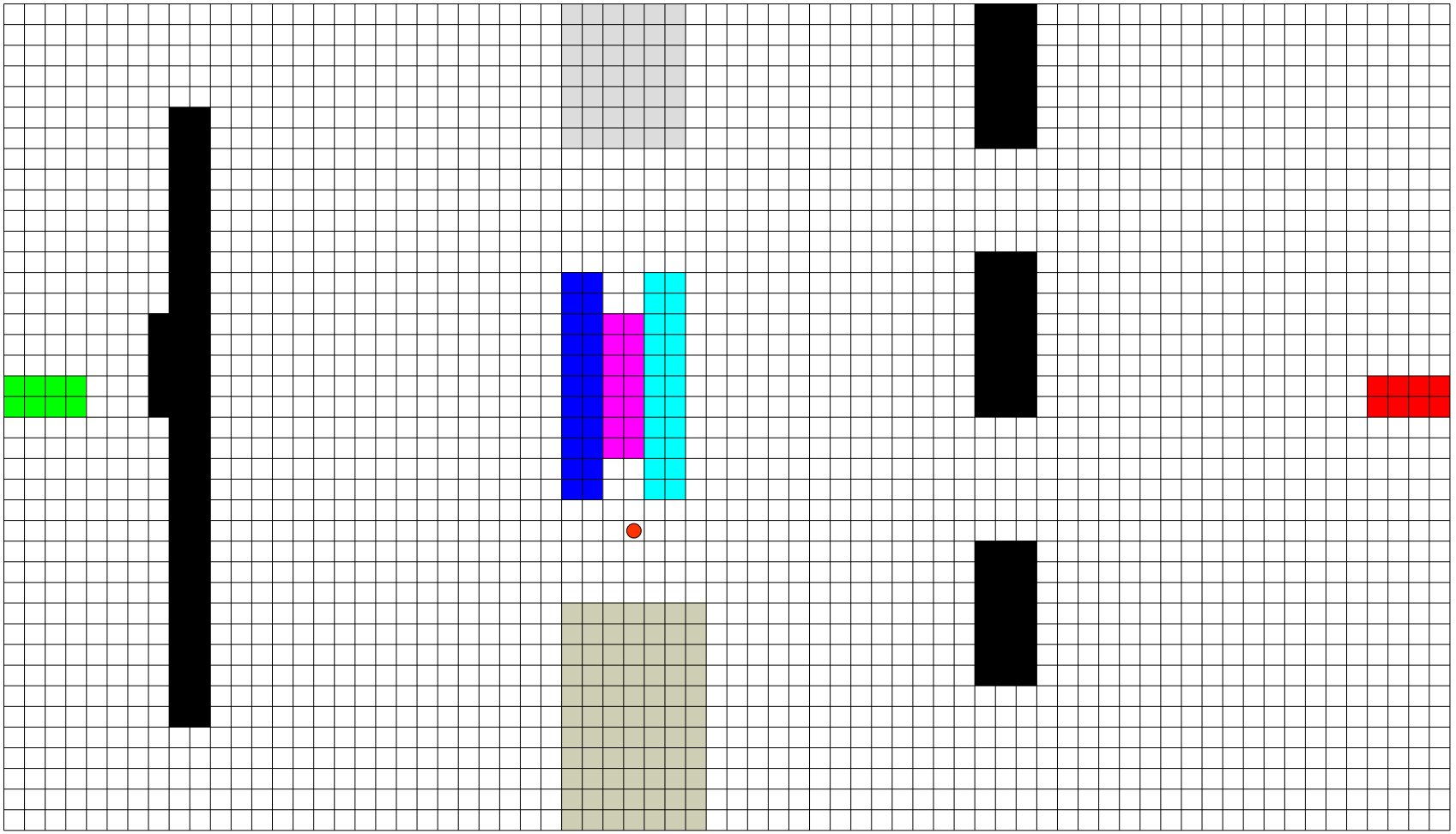
$$\mathbf{0}$$

as the robot will almost surely eventually break down.

Main question of the this paper

How can we compute policies
that work towards the satisfaction
of ω-regular specifications
even in the case of
inevitable non-satisfication?

Motivational example problem



Solving the problem by intuition

A fact

Solving the problem by intuition

A fact

We will all die, and it can happen any moment!

Solving the problem by intuition

A fact

We will all die, and it can happen any moment!

Human behavior

But that does not keep us from planning for the long term (e.g., getting a PhD)!

Solving the problem by intuition

A fact

We will all die, and it can happen any moment!

Human behavior

But that does not keep us from planning for the long term (e.g., getting a PhD)!

Rationale

We normally ignore the risk of catastrophic but very sparse events in decision making

Solving the problem by intuition

A fact

We will all die, and it can happen any moment!

Human behavior

But that does not keep us from planning for the long term (e.g., getting a PhD)!

Rationale

We normally ignore the risk of catastrophic but very sparse events in decision making

However...

... while planning for the long term, humans minimize the risk of catastrophic events.

Solving the problem by intuition

A fact

We will all die, and it can happen any moment!

Human behavior

But that does not keep us from planning for the long term (e.g., getting a PhD)!

Rationale

We normally ignore the risk of catastrophic but very sparse events in decision making

However...

... while planning for the long term, humans minimize the risk of catastrophic events.

Example

Not doing risky driving

Solving the problem by intuition

A fact

We will all die, and it can happen any moment!

Human behavior

But that does not keep us from planning for the long term (e.g., getting a PhD)!

Rationale

We normally ignore the risk of catastrophic but very sparse events in decision making

However...

... while planning for the long term, humans minimize the risk of catastrophic events.

Example

Not doing risky driving

So what we want is...

...a method to compute *risk-averse* policies that are at the same time *optimistic* that the catastrophic event does not happen.

Towards optimistic, but risk-averse policies (1)

Try 1

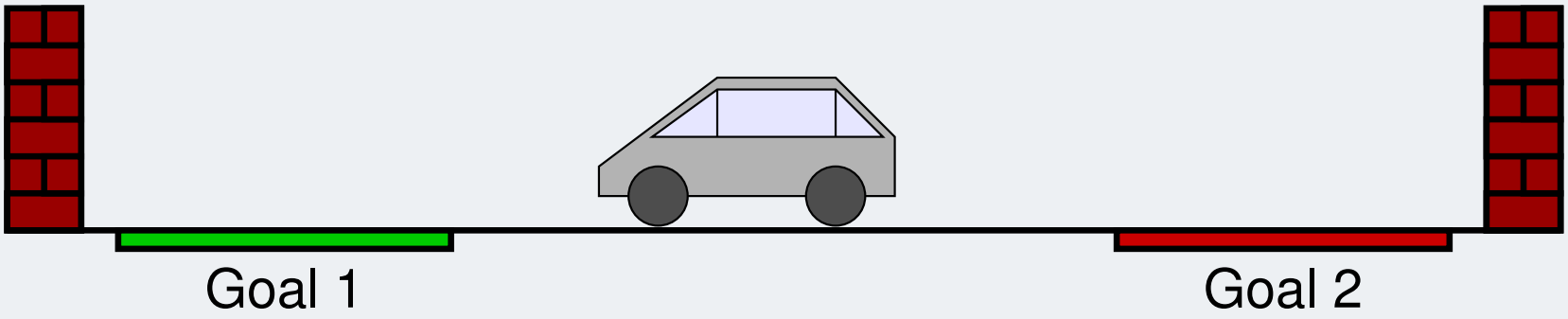
Compute policies that after reaching a *goal* maximize the probability of reaching the respective next goal.

Towards optimistic, but risk-averse policies (1)

Try 1

Compute policies that after reaching a *goal* maximize the probability of reaching the respective next goal.

Example



Specification: $\mathbf{GF}(goal_1) \wedge \mathbf{GF}(goal_2) \wedge \mathbf{G}(\neg crash)$

Prob. car breaks: 10^{-10} (every second)

Towards optimistic, but risk-averse policies (2)

Try 2 (similar to the work by Svorenova et al., 2013)

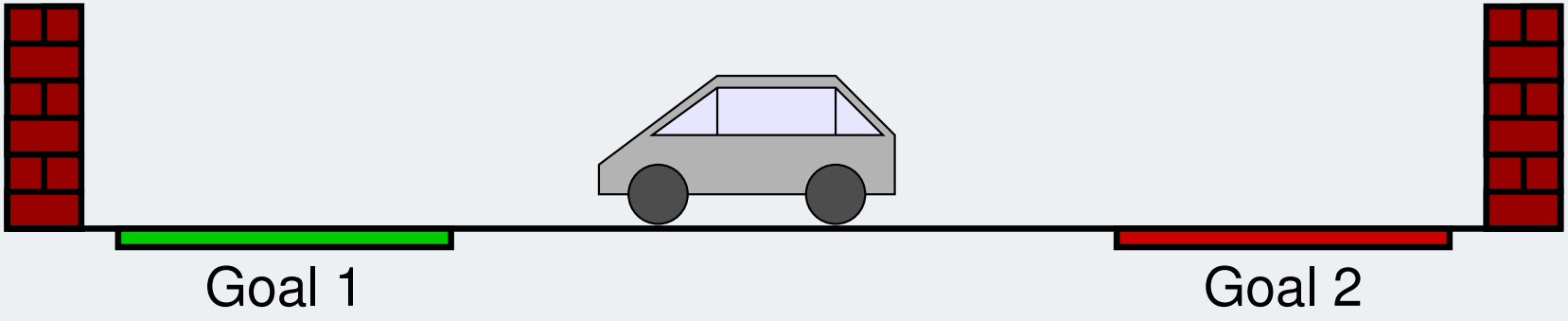
Compute policies that maximize some value p such that whenever a *goal* is reached, the probability of reaching the respective next goal is at least p.

Towards optimistic, but risk-averse policies (2)

Try 2 (similar to the work by Svorenova et al., 2013)

Compute policies that maximize some value p such that whenever a *goal* is reached, the probability of reaching the respective next goal is at least p.

The same example as before



Specification: $\mathbf{GF}(goal_1) \wedge \mathbf{GF}(goal_2) \wedge \mathbf{G}(\neg crash)$

Prob. car breaks: 10^{-10} (every second)

Towards optimistic, but risk-averse policies (3)

But what about general ω-regular specifications?

Example:

$$(\mathbf{GF}(\mathit{red}) \wedge (\neg \mathbf{blue}\, \mathcal{U}\, \mathit{green})) \vee (\mathbf{FG}(\neg \mathit{blue}) \wedge \mathbf{GF}(\mathit{yellow}))$$

What are the goals here and how can we compute risk-averse policies?

Towards optimistic, but risk-averse policies (3)

But what about general ω-regular specifications?

Example:

$$(\mathbf{GF}(\mathit{red}) \wedge (\neg \mathbf{blue}\, \mathcal{U}\, \mathit{green})) \vee (\mathbf{FG}(\neg \mathit{blue}) \wedge \mathbf{GF}(\mathit{yellow}))$$

What are the goals here and how can we compute risk-averse policies?

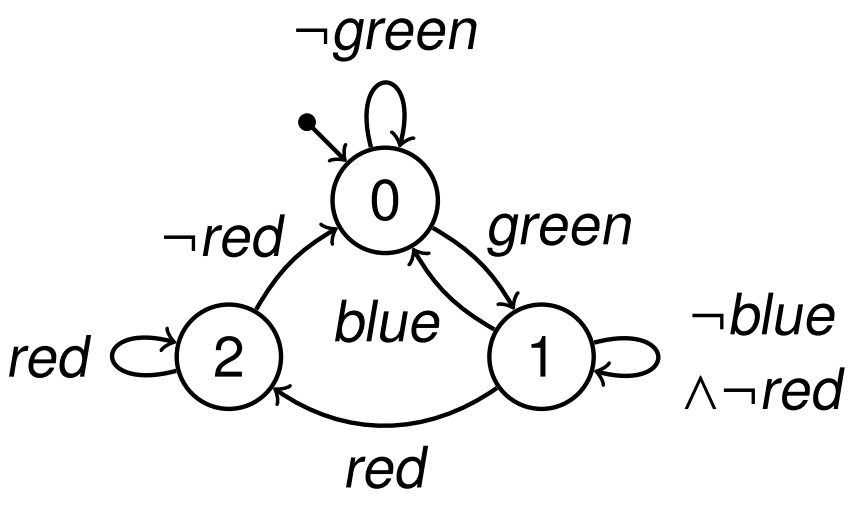
Idea

Let the policy **declare** the goals. Then we can compute a policy together with its declaration.

Declaring goals

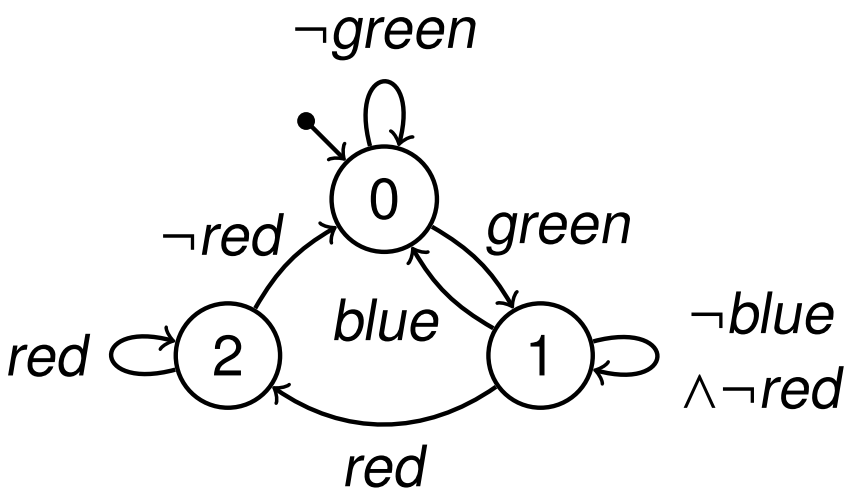
$$
\begin{aligned}
& (\mathbf{FG}(\mathit{red}) \\
\vee\ & \mathbf{F}(\mathit{blue} \wedge \mathbf{XG}\neg\mathit{green}) \\
\vee\ & \mathbf{G}\neg\mathit{green} \\
\vee\ & \mathbf{GF}((\mathit{green} \wedge (\neg\mathit{blue}\,\mathcal{U}\,\mathit{red}))
\end{aligned}
$$

Specification



Deterministic Parity Automaton

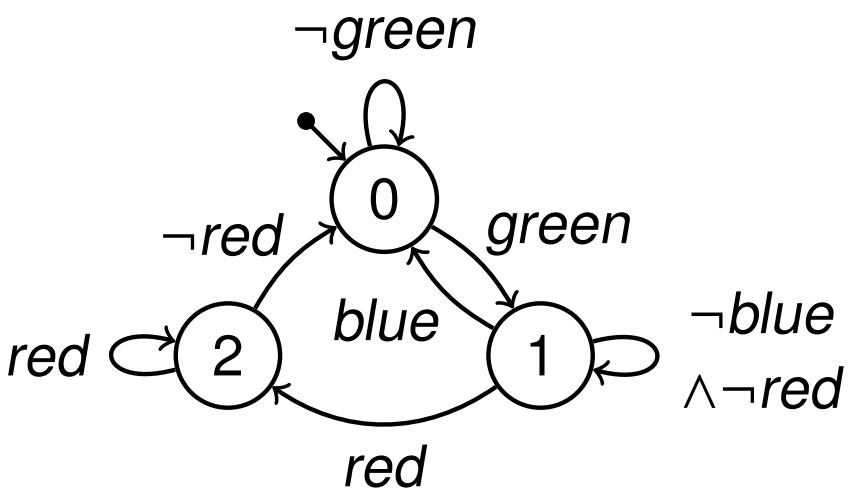
Declaring goals (2)



Definition of parity acceptance

A parity automaton accepts a trace if the highest color that occurs infinitely often along the automaton's run for the trace is **even**.

Declaring goals (2)



Definition of parity acceptance

A parity automaton accepts a trace if the highest color that occurs infinitely often along the automaton's run for the trace is **even**.

So what are possible goals to be reached?

Colors 0 and 2.

Declaring goals (3)

Main idea

We require the system to decrease goal colors at most k times (for some $k \in \mathbb{N}$), and whenever an odd-colored state is visited, the goal color must be higher than the odd color.

Declaring goals (3)

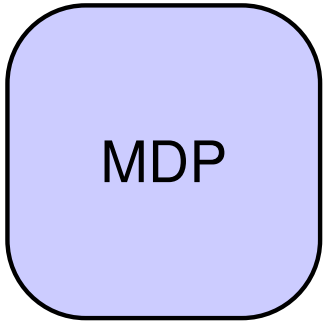
Main idea

We require the system to decrease goal colors at most k times (for some $k \in \mathbb{N}$), and whenever an odd-colored state is visited, the goal color must be higher than the odd color.

Effect

All infinite traces satisfying this new condition satisfy the original parity objective as well.

Overall workflow

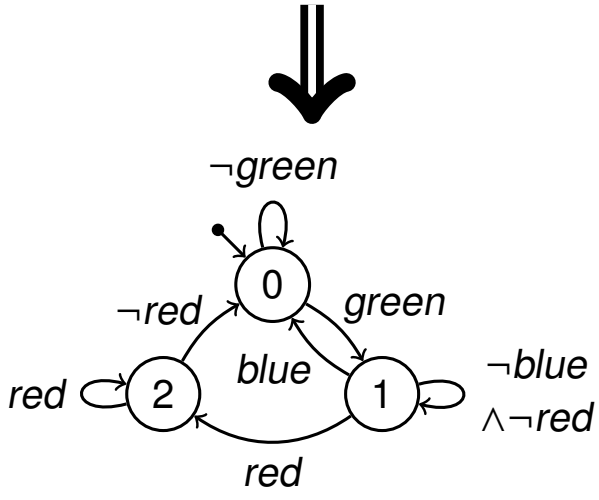


$$
\begin{aligned}
& (\mathbf{FG}(\mathit{red}) \\
\vee\ & \mathbf{F}(\mathit{blue} \wedge \mathbf{XG}\neg \mathit{green}) \\
\vee\ & \mathbf{G}\neg \mathit{green} \\
\vee\ & \mathbf{GF}((\mathit{green} \wedge (\neg \mathit{blue}\, \mathcal{U}\, \mathit{red}))
\end{aligned}
$$

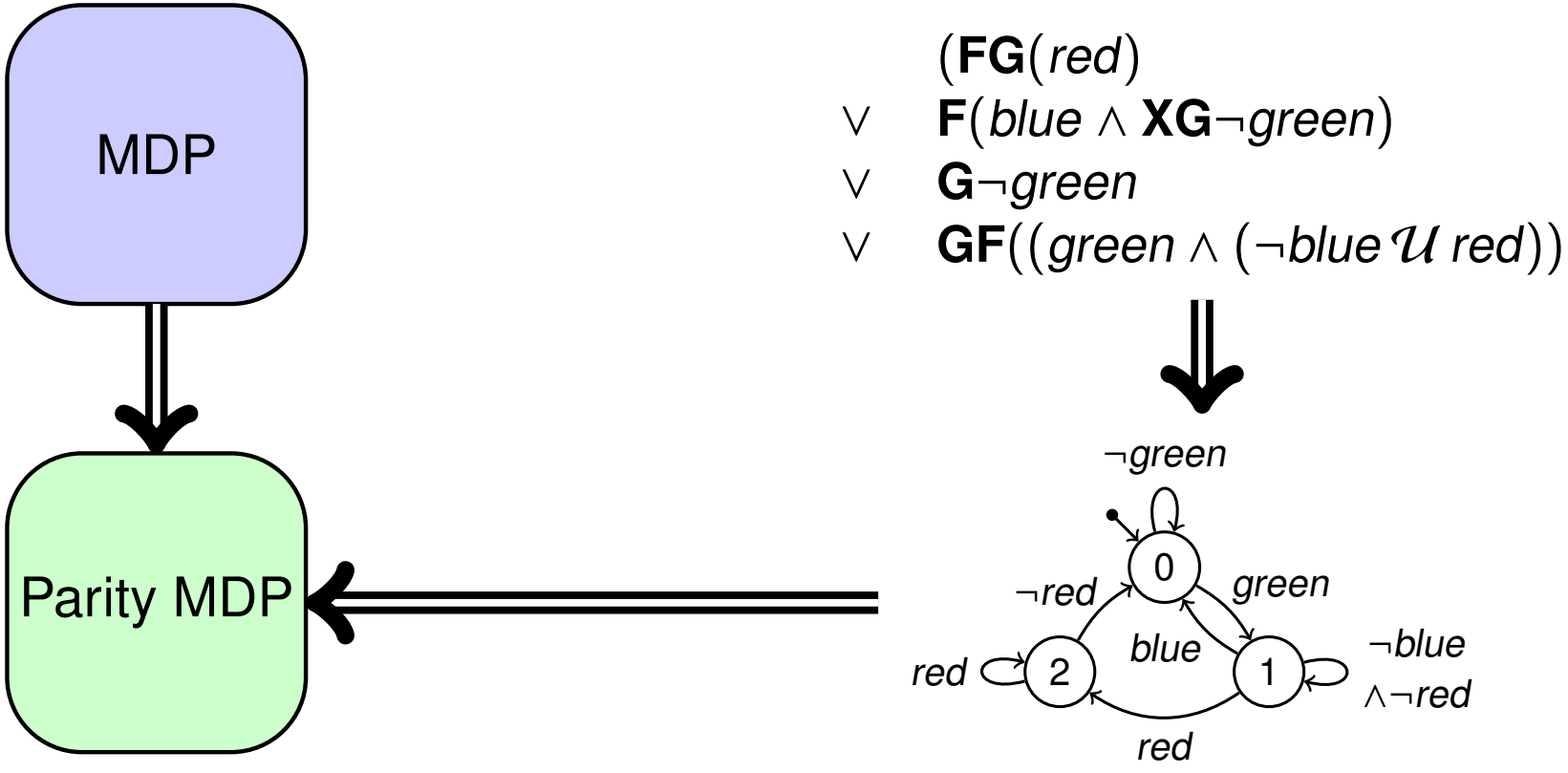
Overall workflow

MDP

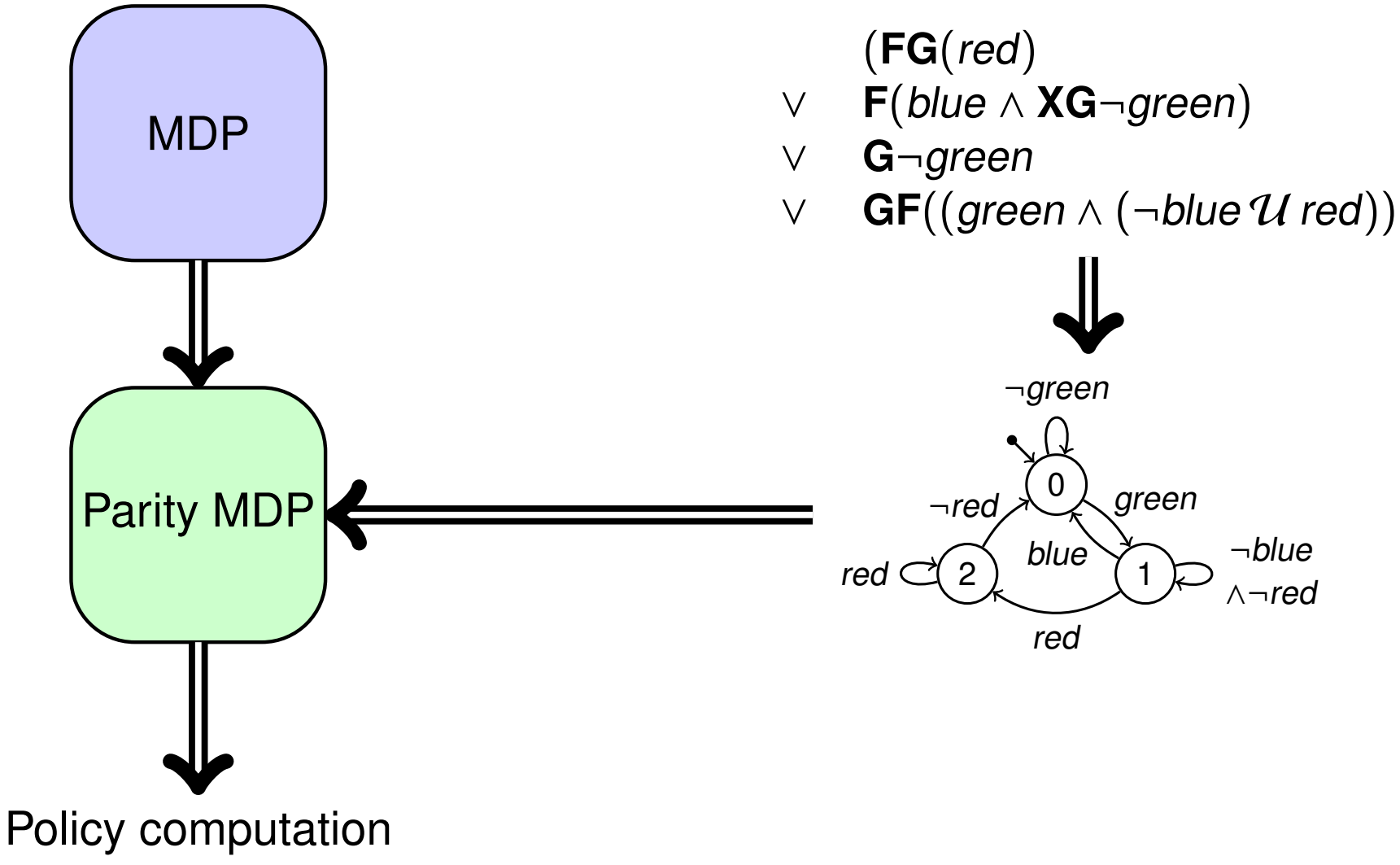
$$
\begin{aligned}
& (\mathbf{FG}(\mathit{red}) \\
\vee\ & \mathbf{F}(\mathit{blue} \wedge \mathbf{XG}\neg \mathit{green}) \\
\vee\ & \mathbf{G}\neg \mathit{green} \\
\vee\ & \mathbf{GF}((\mathit{green} \wedge (\neg \mathit{blue}\, \mathcal{U}\, \mathit{red}))
\end{aligned}
$$



Overall workflow



Overall workflow



What exactly is now being computed?

We compute for some values of p and k ...

A **control policy** for a **parity MDP** that always declares its respective next goal such that:

- From every goal state, the next goal state is visited with probability at least p.
- Goal states have an even color, and the color of the goal states can only be decreased at most k times along a trace
- The goal color is always greater than or equal to the odd colors of the states visited on along the trace

What exactly is now being computed?

We compute for some values of p and k ...

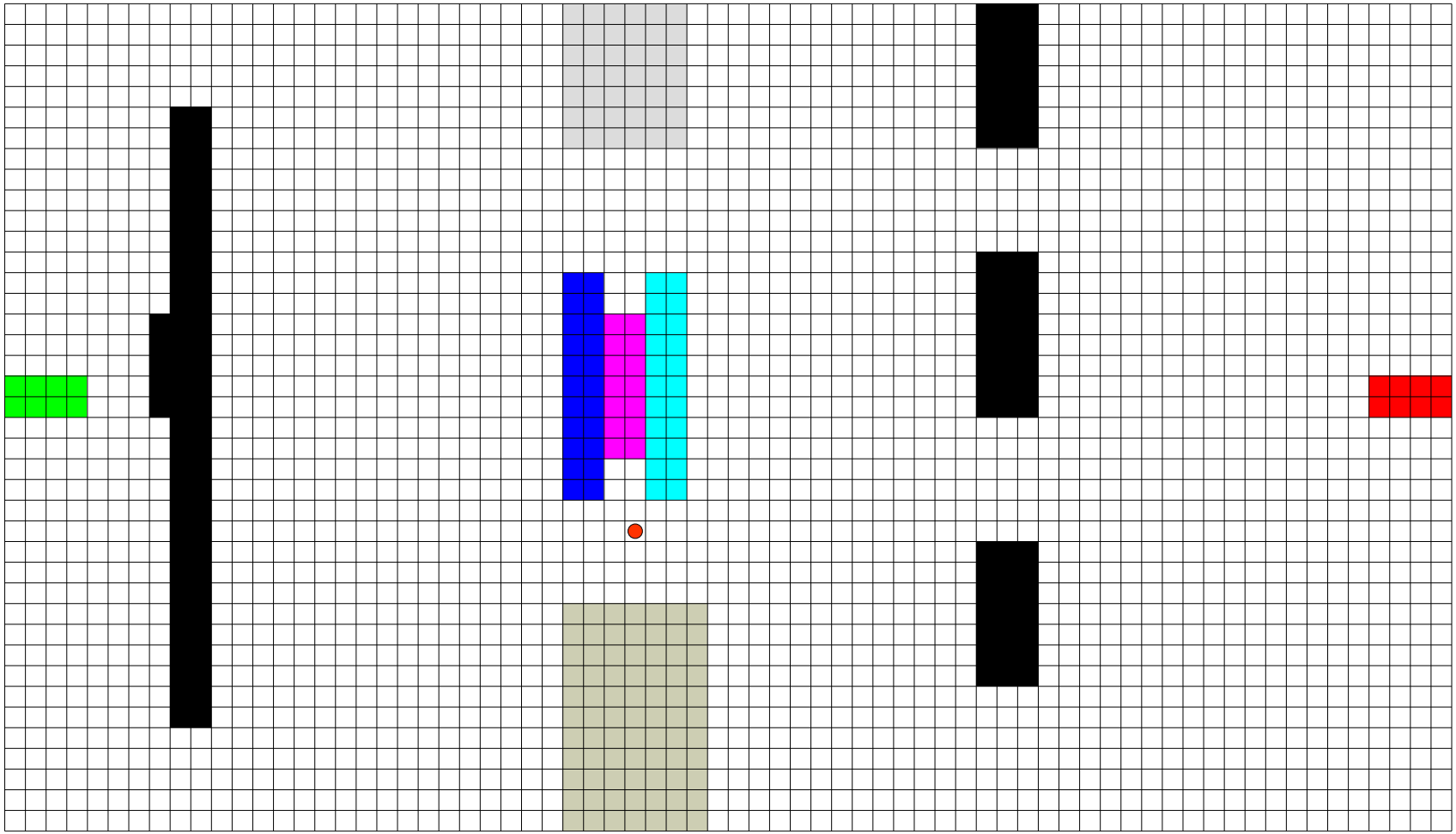
A **control policy** for a **parity MDP** that always declares its respective next goal such that:

- From every goal state, the next goal state is visited with probability at least p.
- Goal states have an even color, and the color of the goal states can only be decreased at most k times along a trace
- The goal color is always greater than or equal to the odd colors of the states visited on along the trace

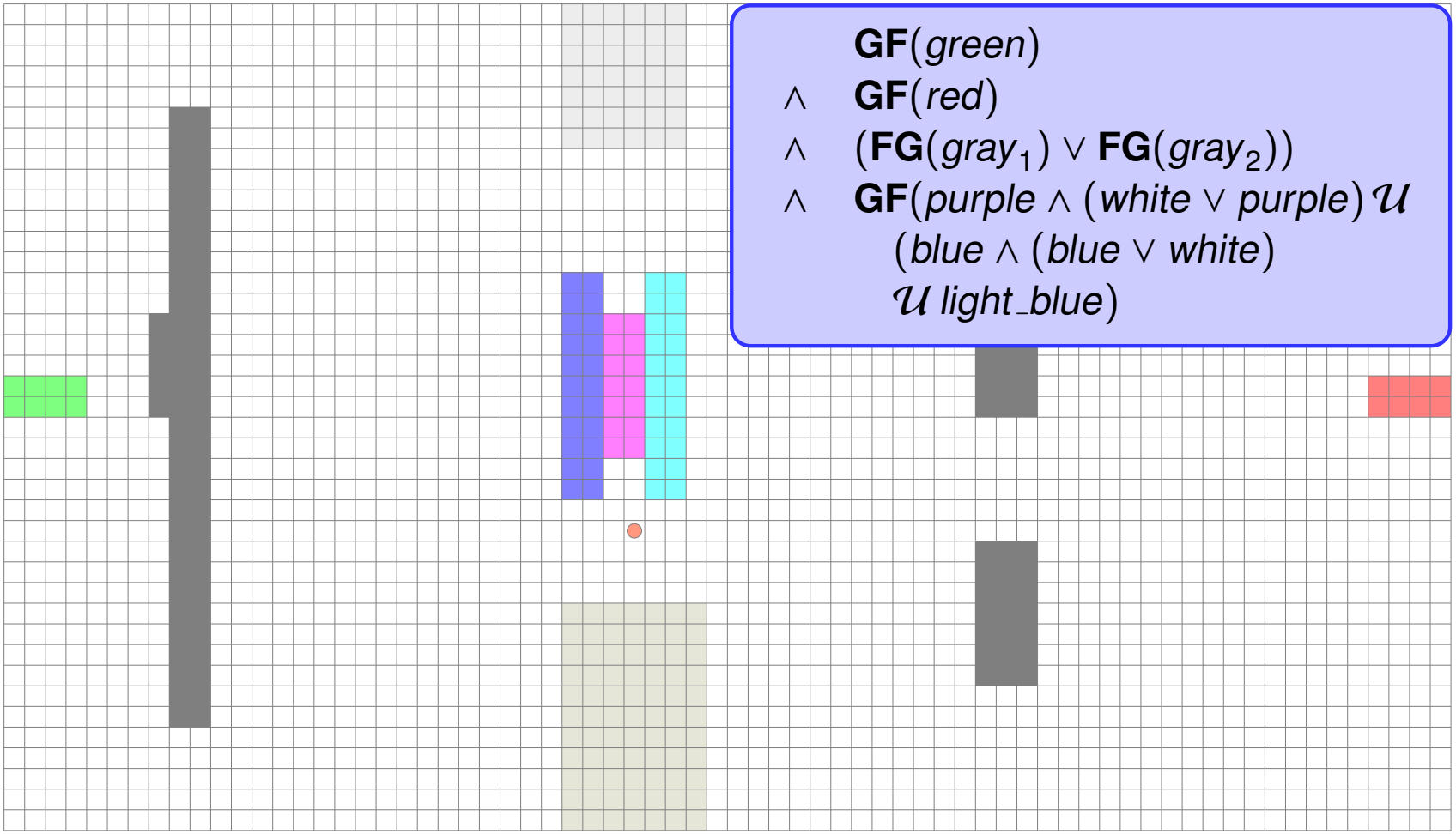
Computing the *best* policies

We perform bisection search over p and compute if there is a k such there exists a *p-risk-averse policy*.

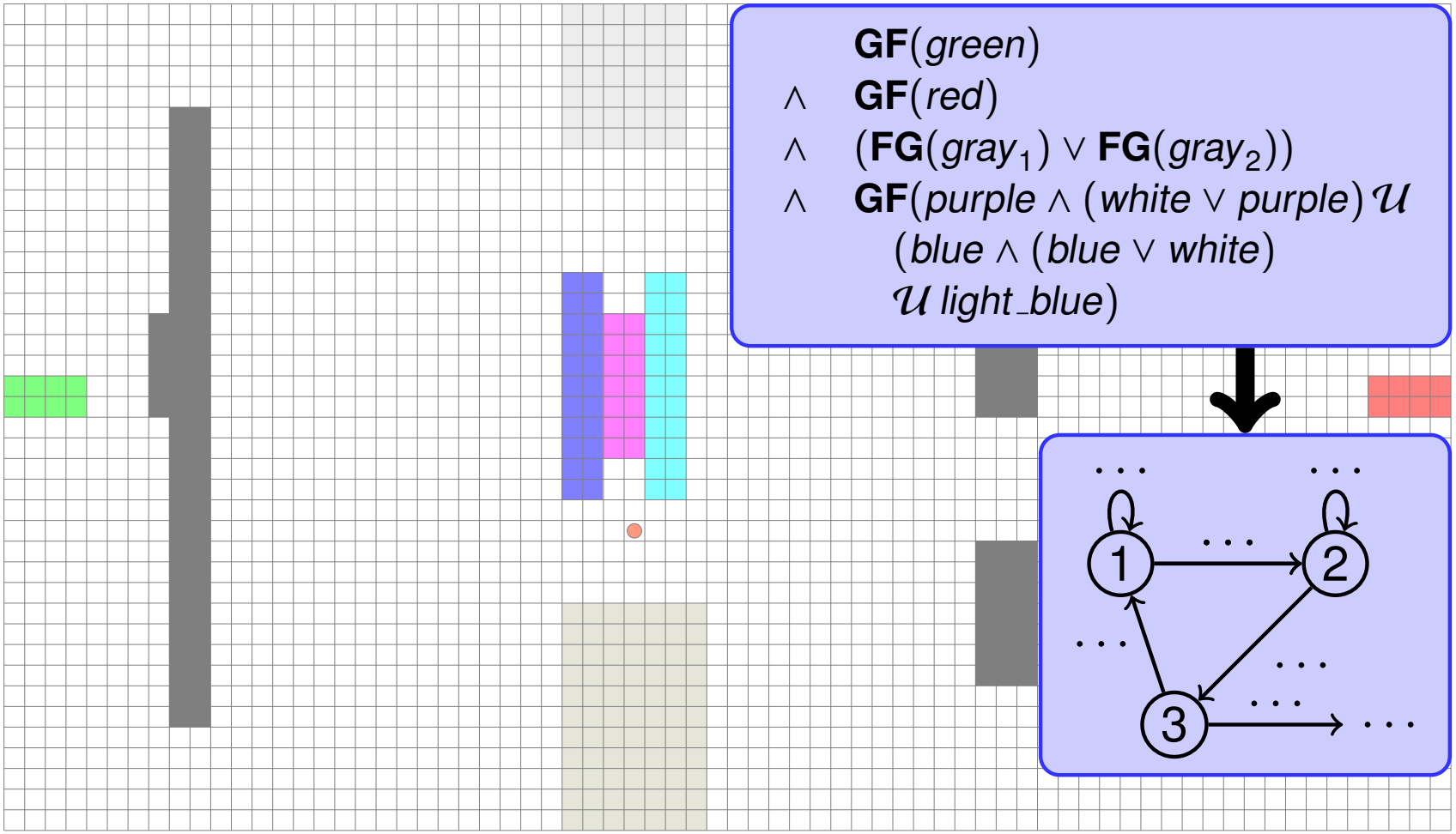
Motivational example problem (revisited)



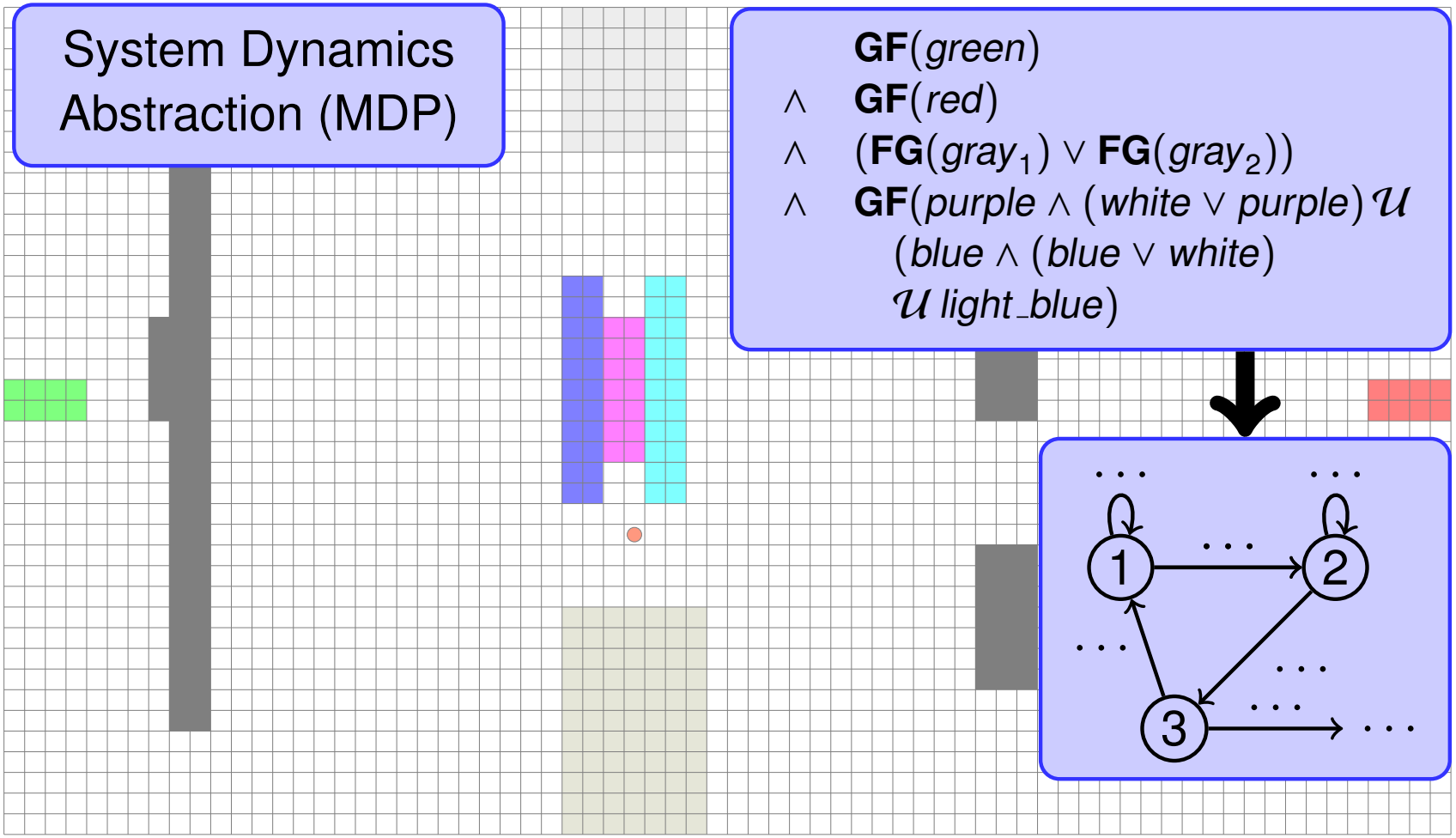
Motivational example problem (revisited)



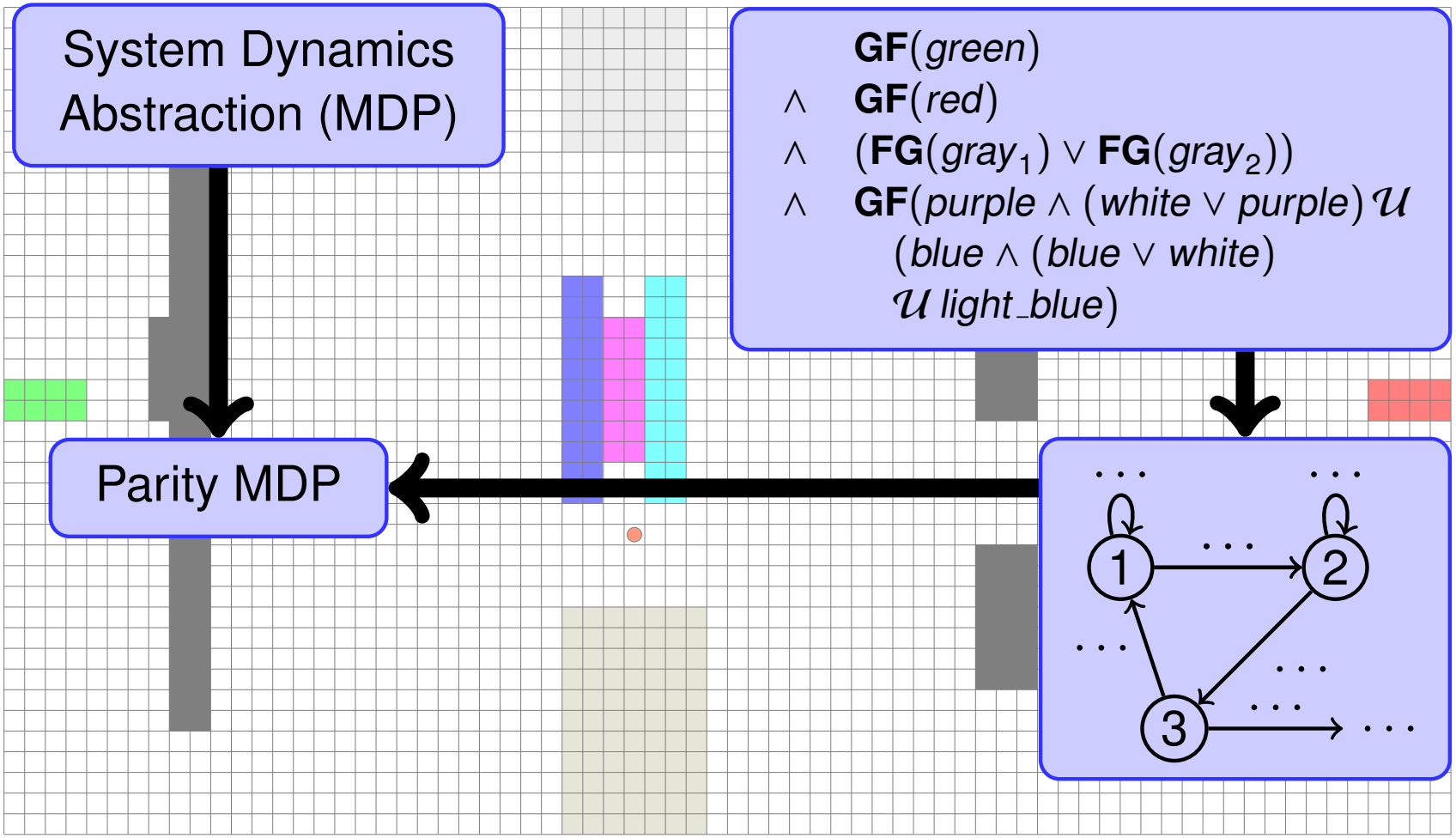
Motivational example problem (revisited)



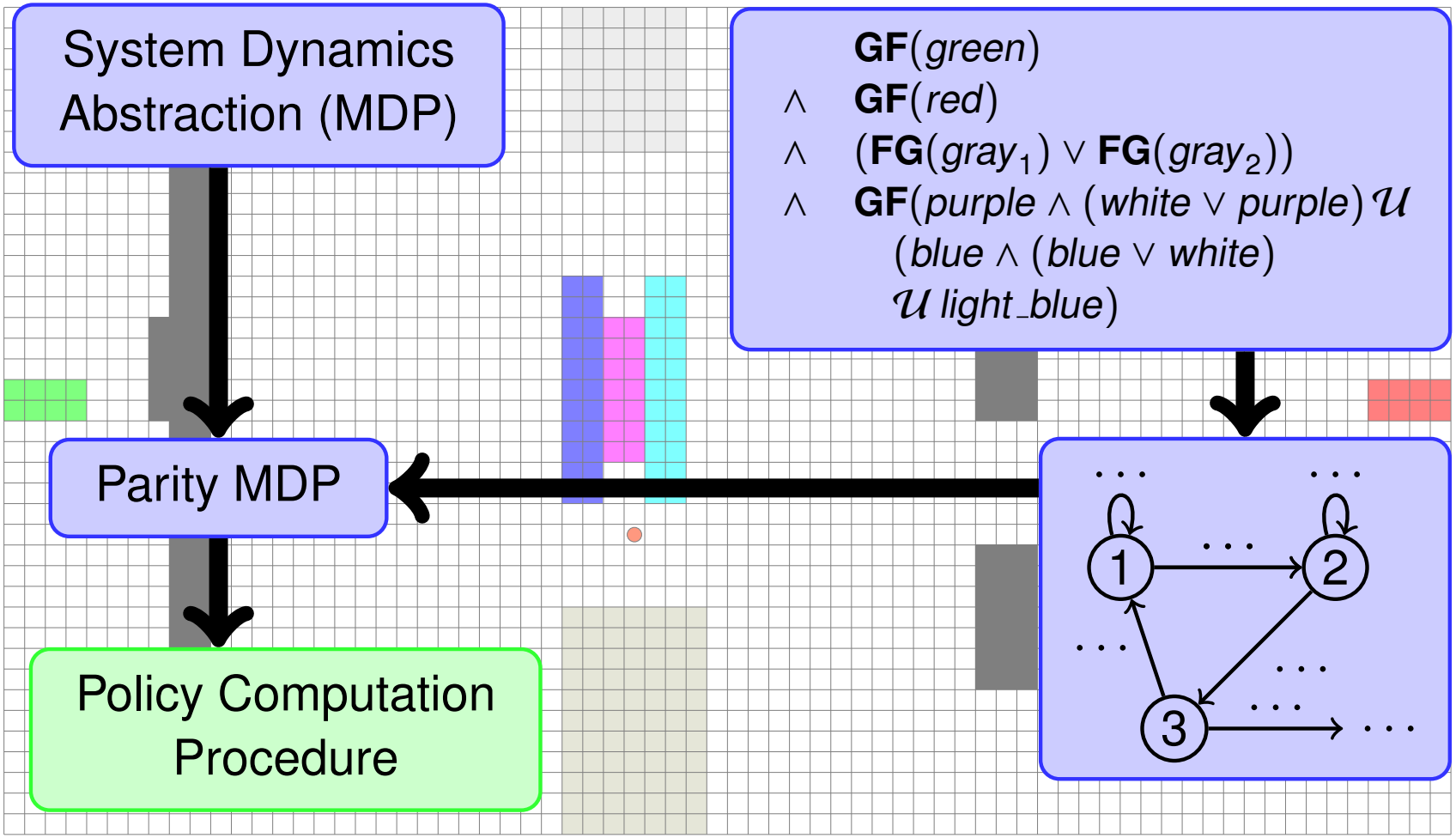
Motivational example problem (revisited)

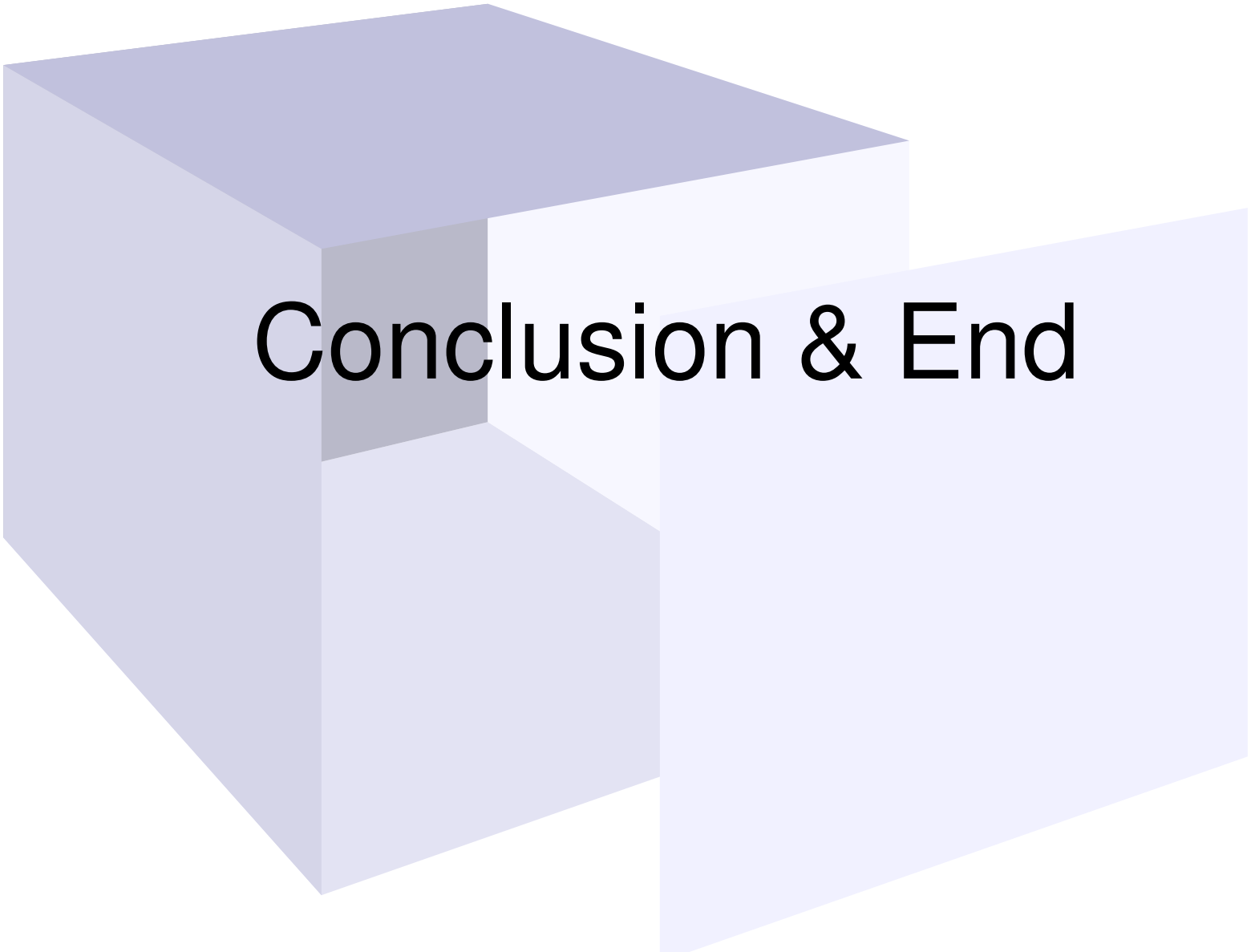


Motivational example problem (revisited)



Motivational example problem (revisited)





Conclusion & End

Conclusion

p-risk-averse policies

- They allow to find reasonable policies even if the specification cannot be fulfilled in the long run.
- Work with all ω-regular specifications
- Computation can be done efficiently
- Tool available: `http://progirep.github.io/ramps/`

Future work

What about 2.5 player games? Combination with costs?

References I

Maria Svorenova, Ivana Cerna, and Calin Belta. Optimal control of MDPs with temporal logic constraints. In *52nd IEEE Conference on Decision and Control (CDC)*, pages 3938–3943, 2013.

Computing a p-risk-averse policy

Approach in the paper

- For every $p \in [0, \ldots, 1]$, a p-risk-averse control policy has a finite number of states.
- Optimal strategies can be computed by solving a series of optimal reachability policy computations in MDPs.

Formal definition

Definition: p-risk-averse strategy

Let $\mathcal{M} = (S, A, \Sigma, P, C, s_0)$ be a parity MDP. We say that some control policy $f : S^* \to A$ has a *risk-averseness probability* $p \in [0, 1]$ if there exist labelings $l : S^* \to \mathbb{N}$ and $l' : S^* \to \mathbb{B}$ and a Markov chain $\mathcal{C}'$ induced by $\mathcal{M}$ and f with the following properties:

- There exists some number $k \in \mathbb{N}$ such that for all $t_0 t_1 t_2 \ldots \in S^\omega$, there are at most k many indices $i \in \mathbb{N}$ for which we have $l(t_0 \ldots t_i) > l(t_0 \ldots t_i t_{i+1})$.
- For all $t_0 t_1 \ldots t_n \in S^*$, we have that $l(t_0 \ldots t_n)$ is even, and $l'(t_0 \ldots t_n) = \mathbf{tt}$ implies that $C(t_n) \geq l(t_0 \ldots t_n)$ and that $C(t_n)$ is even.
- For all $t_0 t_1 \ldots t_n \in S^*$, if $C(t_n)$ is odd, then $l(t_0 \ldots t_n) > C(t_n)$.
- For all $t = t_0 t_1 \ldots t_n \in S^*$ with either (a) $l'(t) = \mathbf{tt}$ or (b) $t = s_0$, the probability measure in $\mathcal{C}'$ to reach some state $t\, t'_0 \ldots t'_m \in S^*$ with $l'(t\, t'_0 \ldots t'_m) = \mathbf{tt}$ from state t is at least p.