

Satisfiability Bounds for ω -Regular Properties in Bounded-Parameter Markov Decision Processes

M. Weininger T. Meggendorfer J. Kretinsky

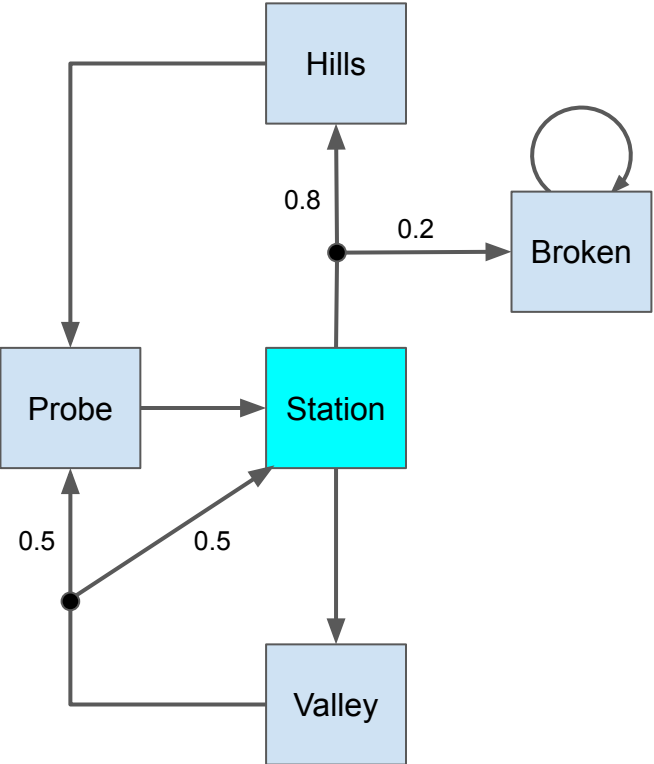


Satisfiability Bounds for ω -Regular Properties in Bounded-Parameter **Markov Decision Processes**

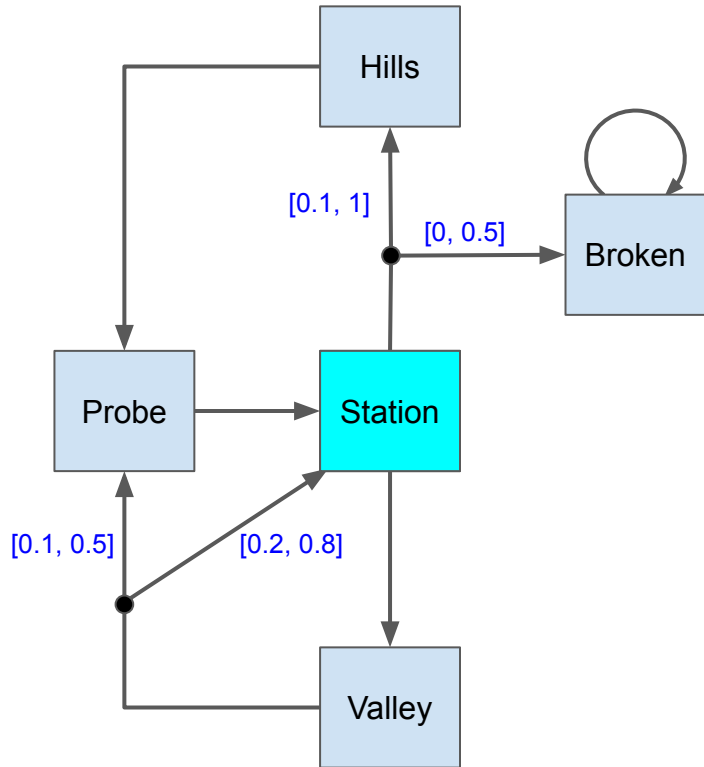
M. Weininger T. Meggendorfer J. Kretinsky



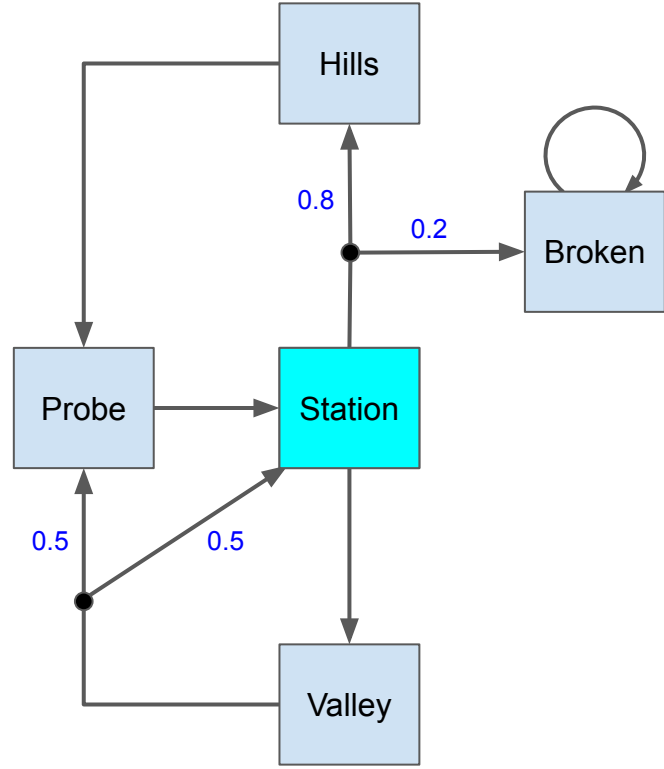
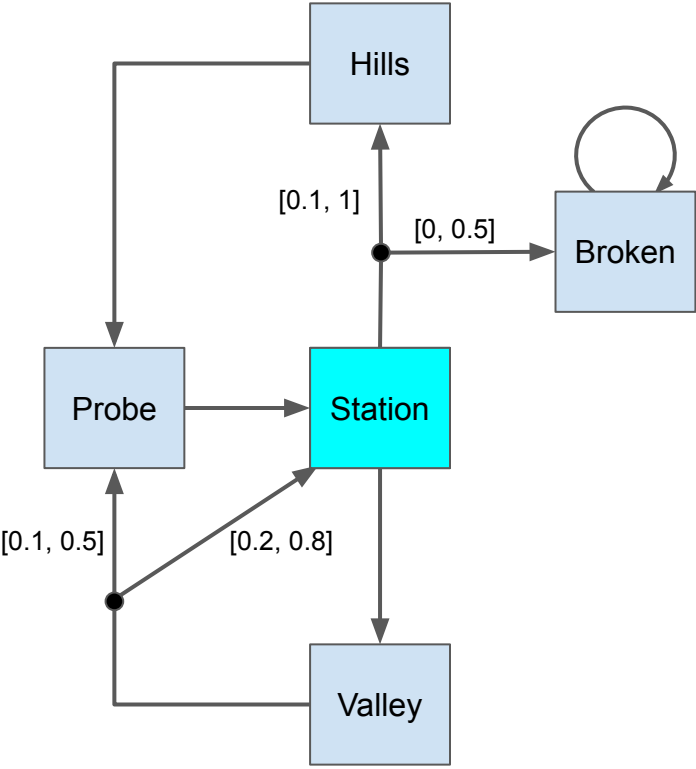
Markov Decision Process



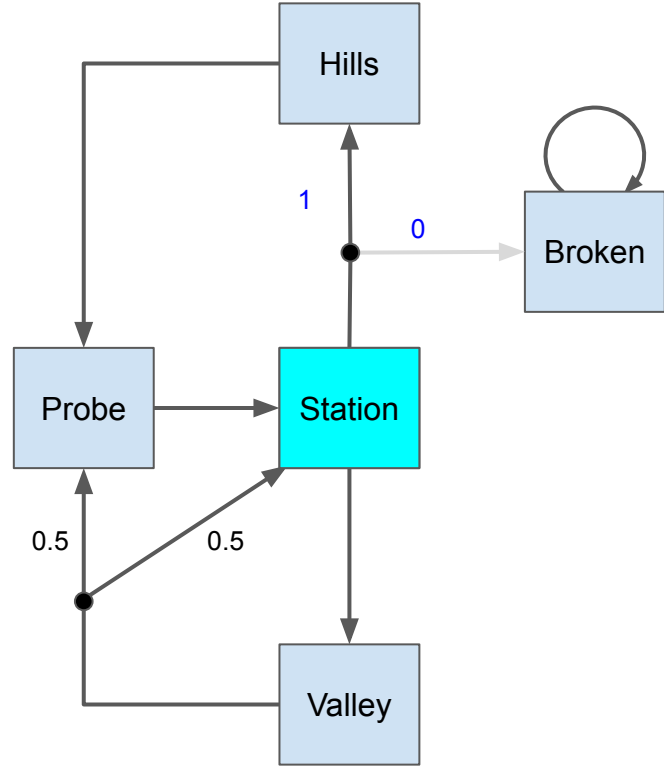
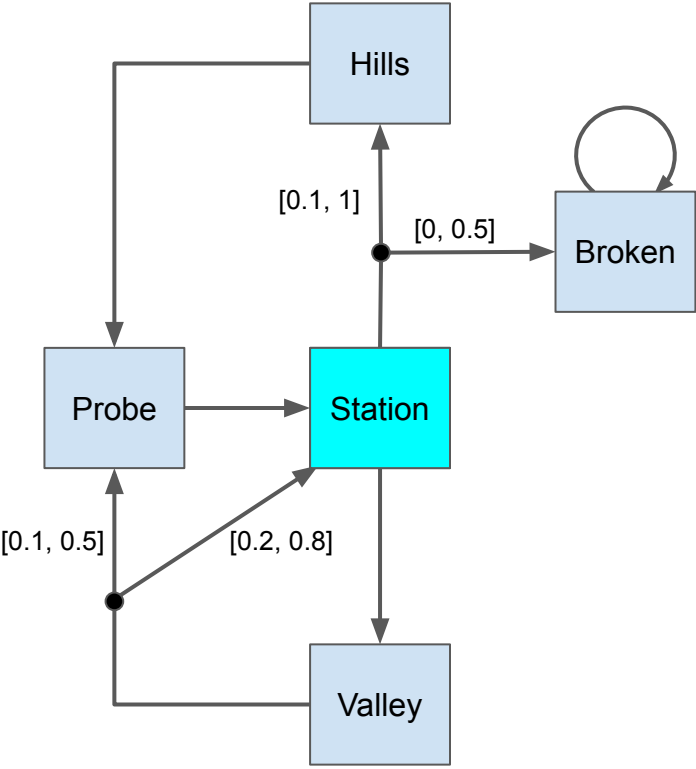
Bounded-Parameter Markov Decision Process



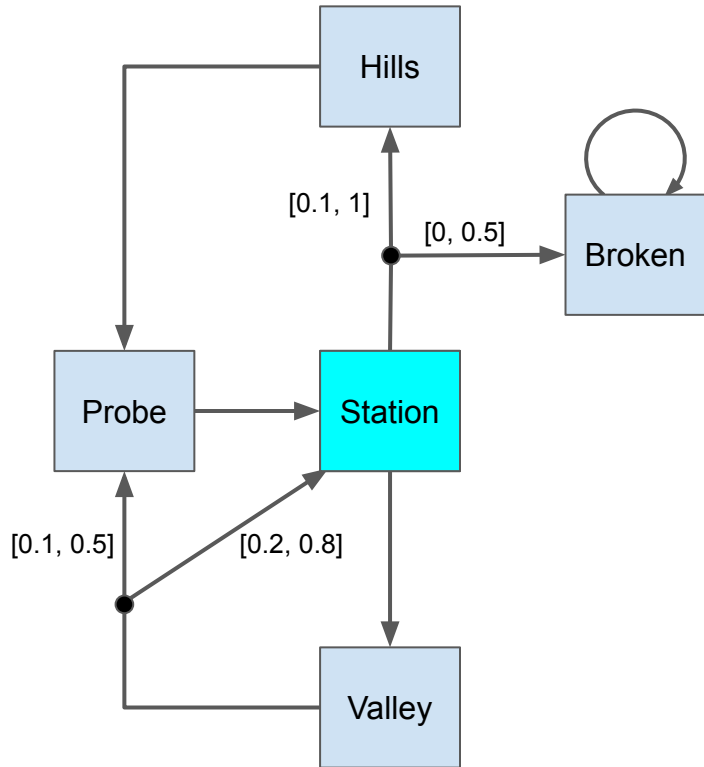
Bounded-Parameter Markov Decision Process



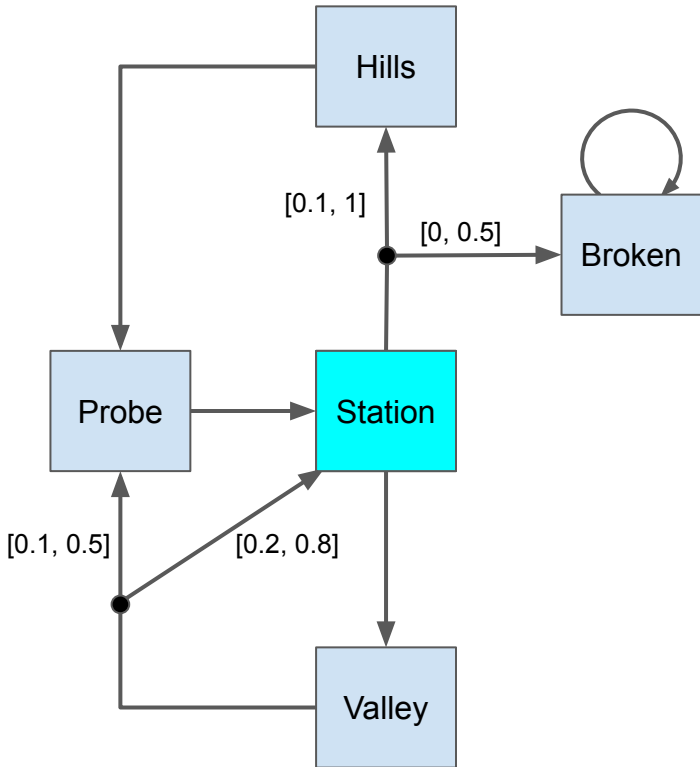
Bounded-Parameter Markov Decision Process



ω -Regular Properties



ω -Regular Properties



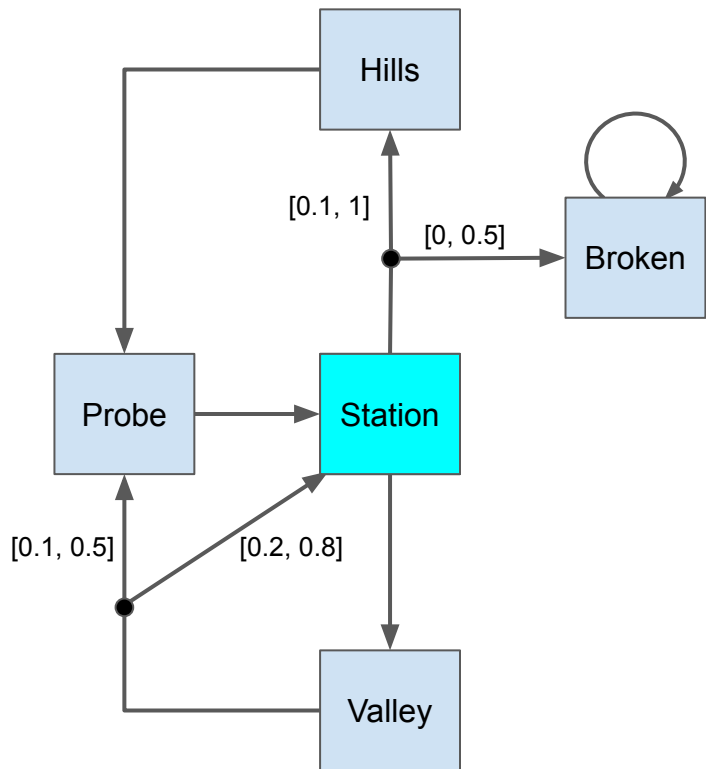
“Eventually take a probe”

$F(\text{Probe})$

“Always take a probe in the future and
bring it to the station”

$G(F(\text{Probe}) \wedge \text{Probe} \Rightarrow X(\text{Station}))$

Satisfiability bounds for ω -Regular Properties



“Eventually take a probe”

$F(\text{Probe})$

“Always take a probe in the future and bring it to the station”

$G(F(\text{Probe}) \wedge \text{Probe} \Rightarrow X(\text{Station}))$

Find optimal controller

$\mathcal{L} \leq \mathbb{P}(\text{System} \models \text{Property}) \leq \mathcal{U}$

Semantics of the intervals

\mathcal{L} : Adversarial Environment



\mathcal{U} : Design choice



Semantics of the intervals

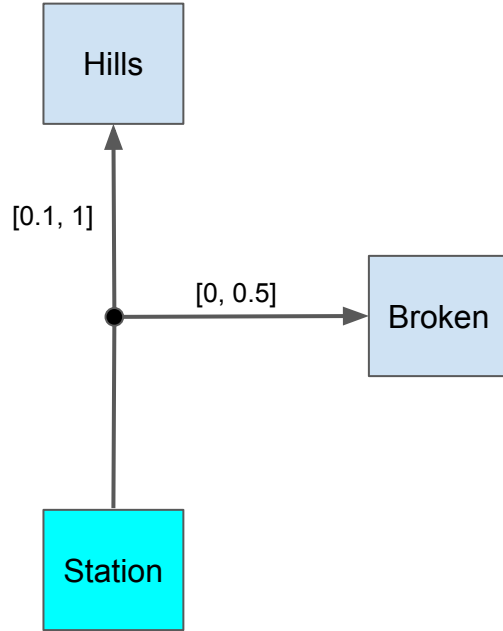
\mathcal{L} : Adversarial Environment



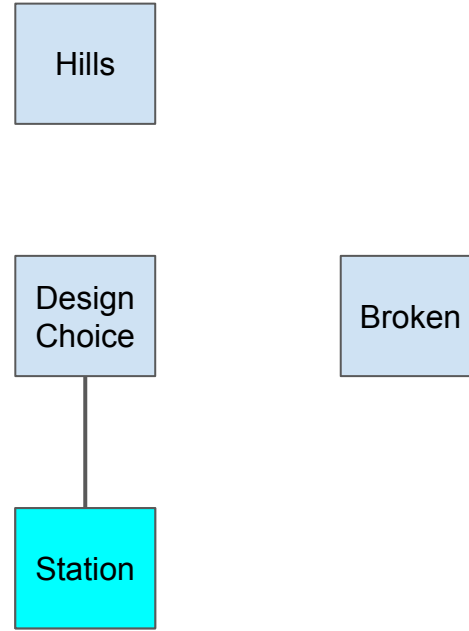
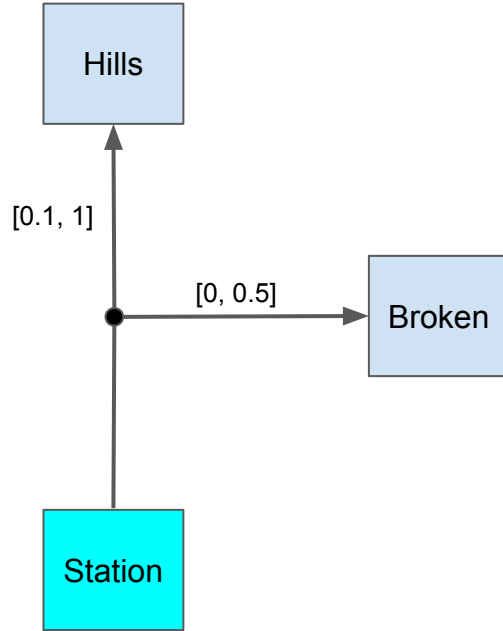
\mathcal{U} : Design choice



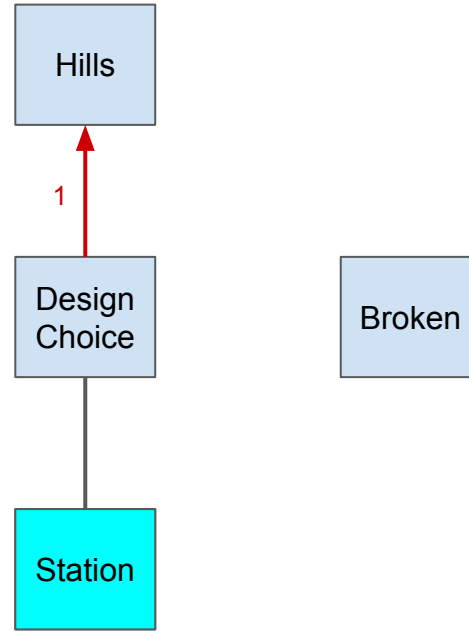
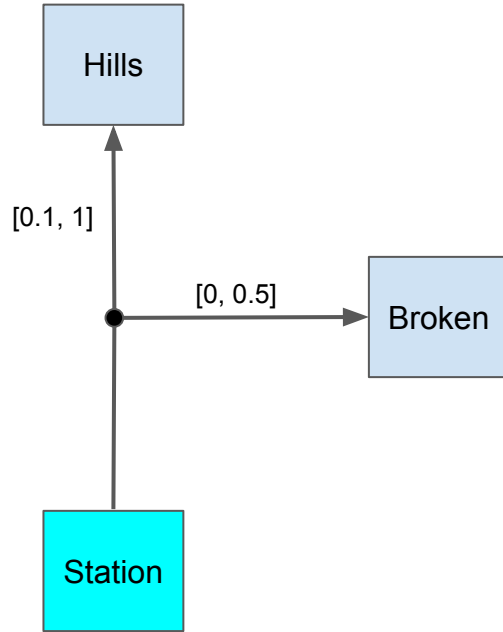
Resolving intervals in "Design choice" setting



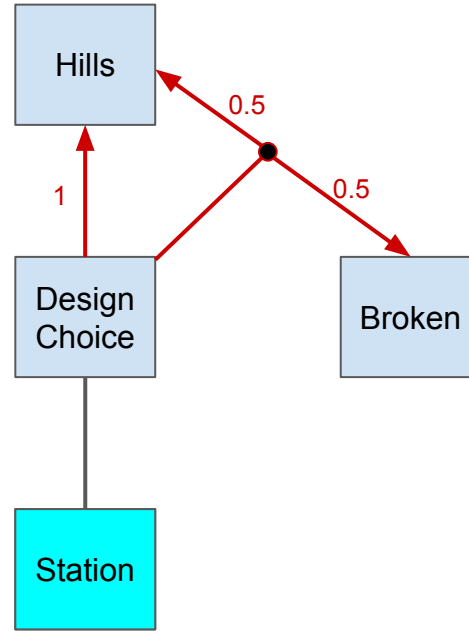
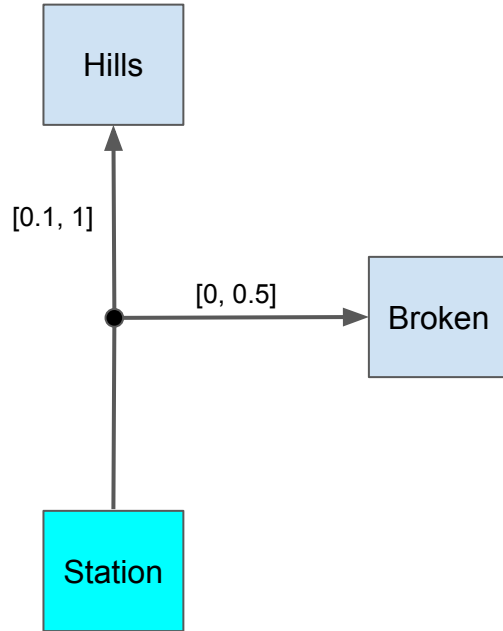
Resolving intervals in "Design choice" setting



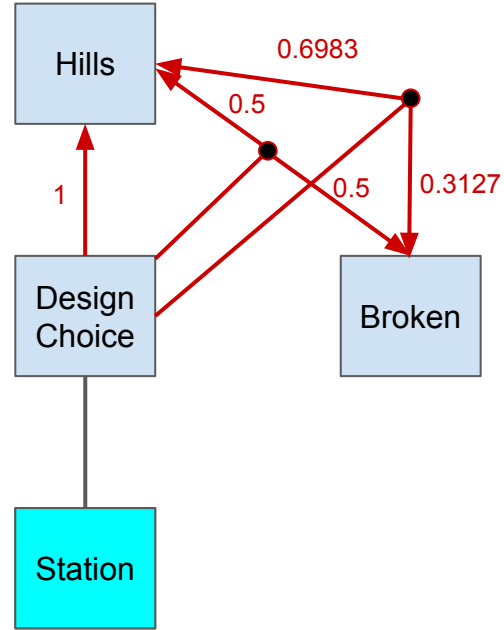
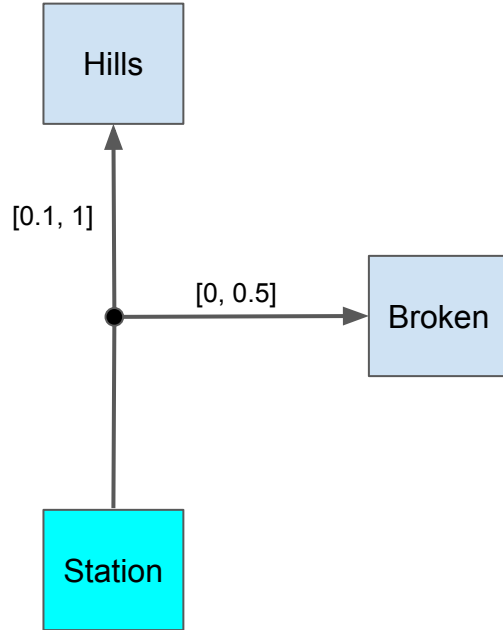
Resolving intervals in "Design choice" setting



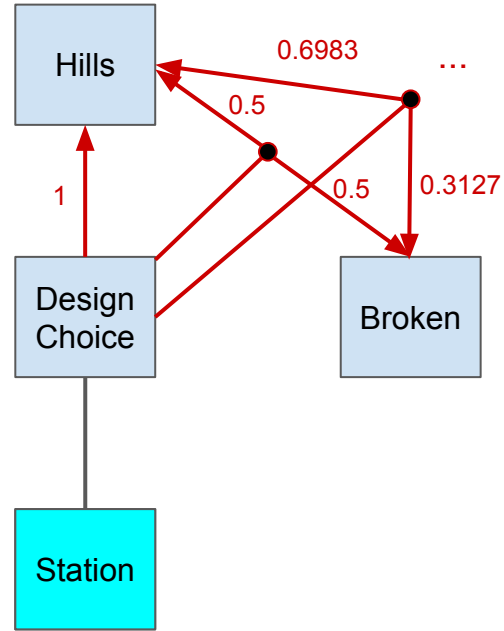
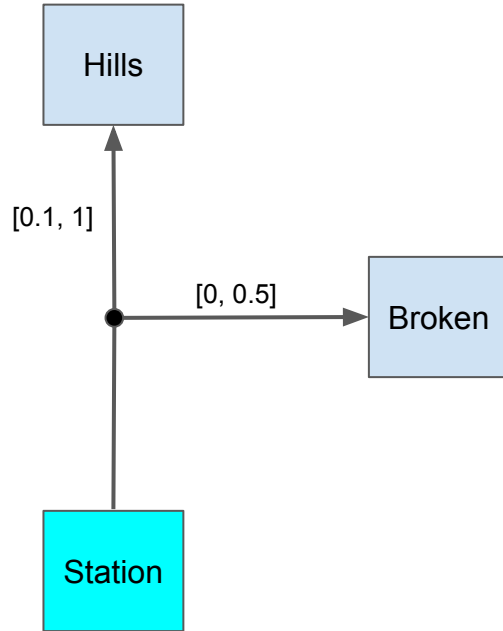
Resolving intervals in "Design choice" setting



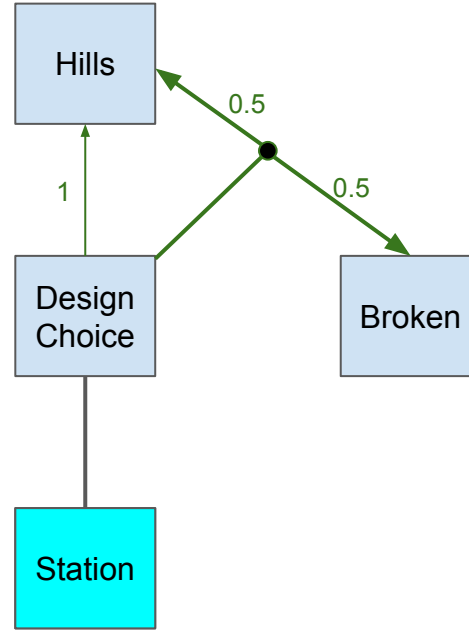
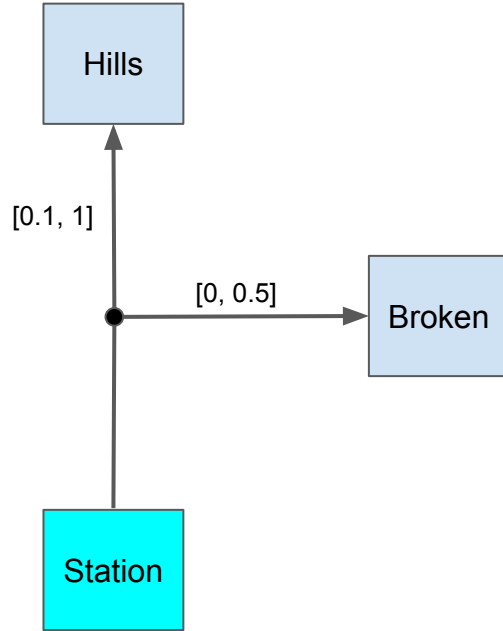
Resolving intervals in "Design choice" setting



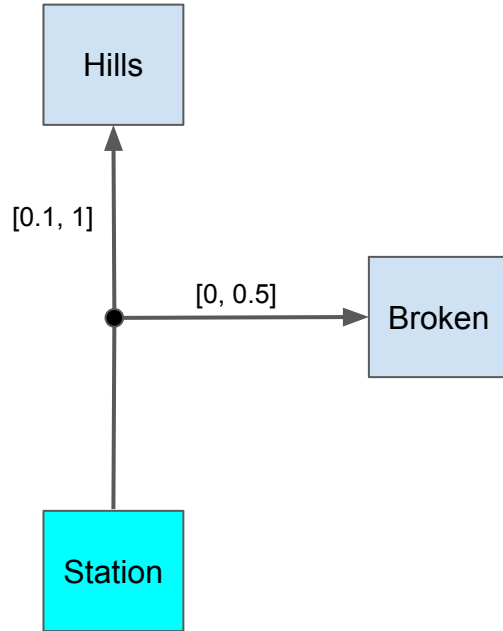
Resolving intervals in "Design choice" setting



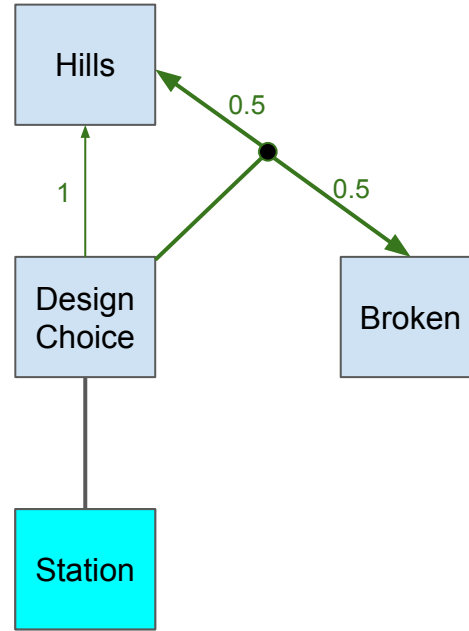
Resolving intervals in "Design choice" setting



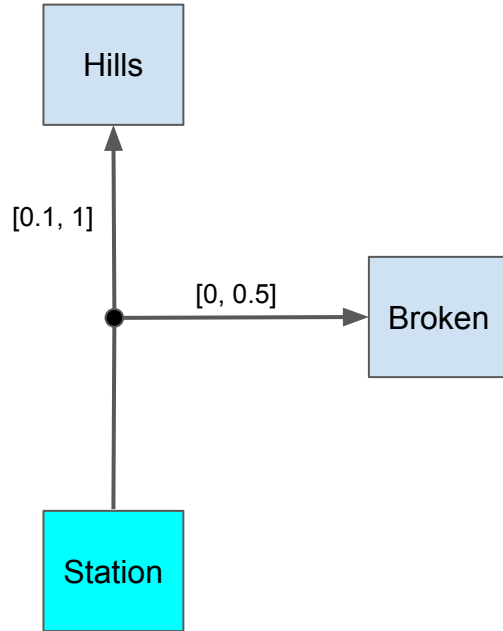
Resolving intervals in "Design choice" setting



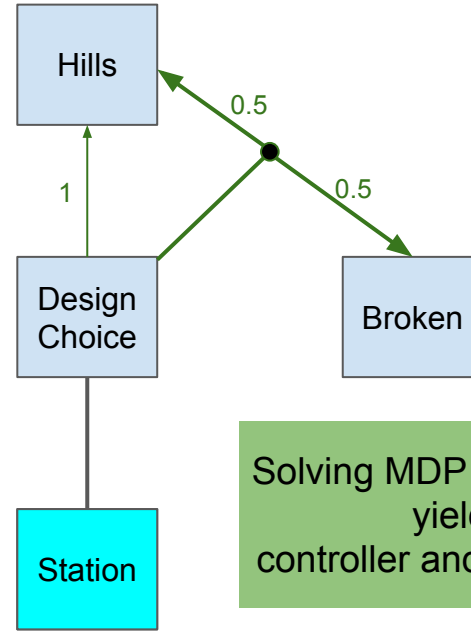
Basic Feasible Solutions
[HM18]



Resolving intervals in "Design choice" setting



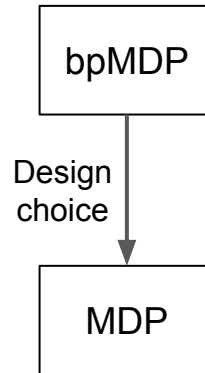
Basic Feasible Solutions
[HM18]



Solving MDP e.g. [Put94]
yields
controller and probability

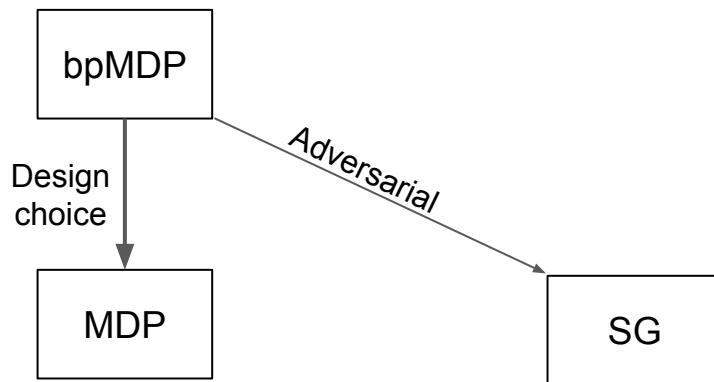
Idea in short

1. New state for every action
2. Basic feasible solutions as its actions
3. Solve MDP



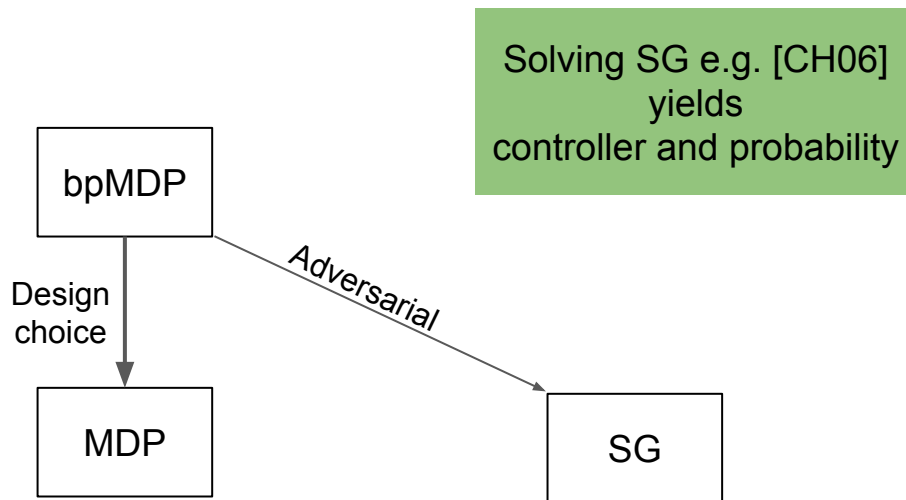
Idea in short

1. New state for every action (other player!)
2. Basic feasible solutions as its actions
3. Solve ~~MDP~~ Stochastic Game

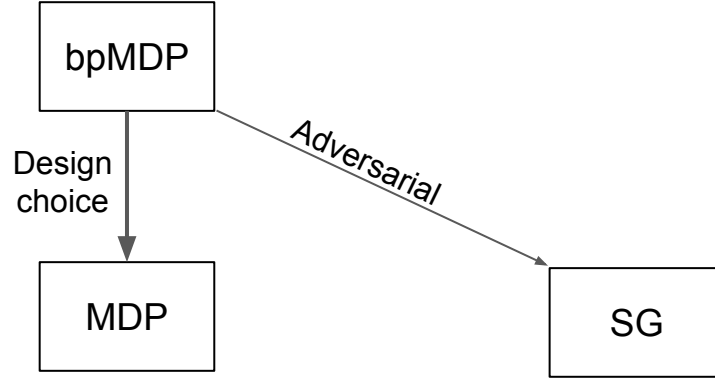


Idea in short

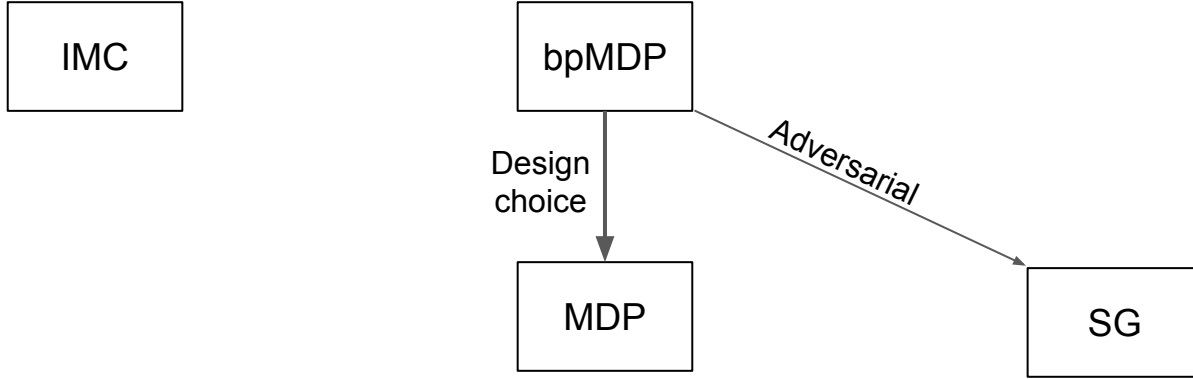
1. New state for every action (other player!)
2. Basic feasible solutions as its actions
3. Solve ~~MDP~~ Stochastic Game



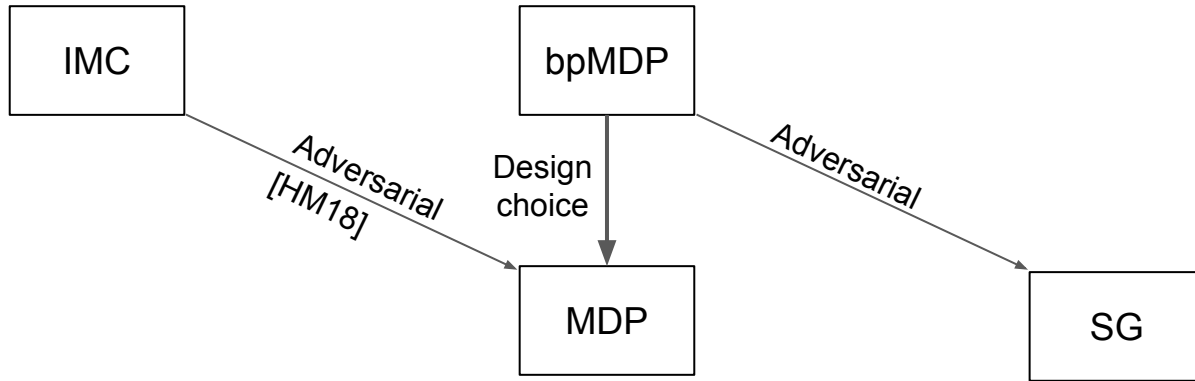
The bigger picture



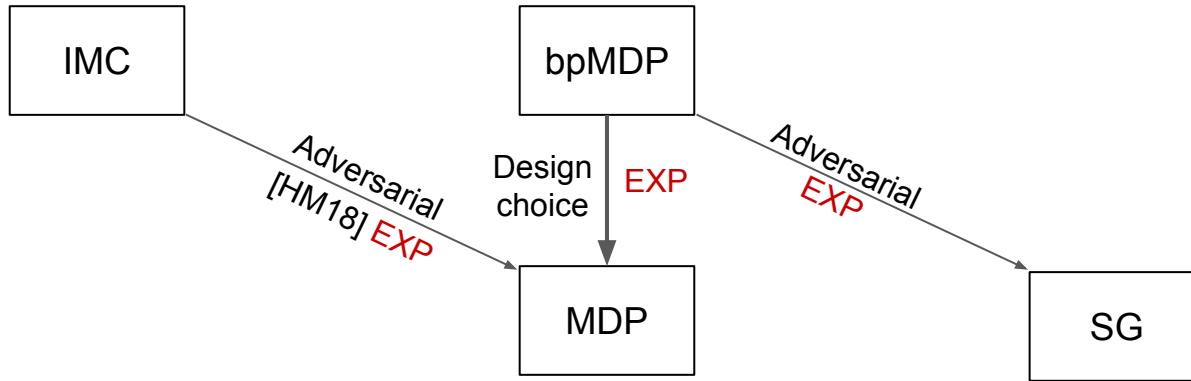
The bigger picture



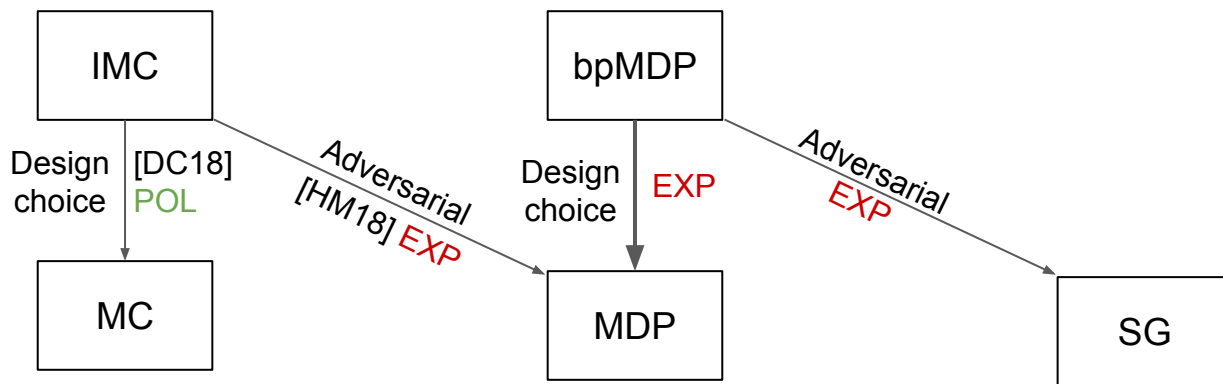
The bigger picture



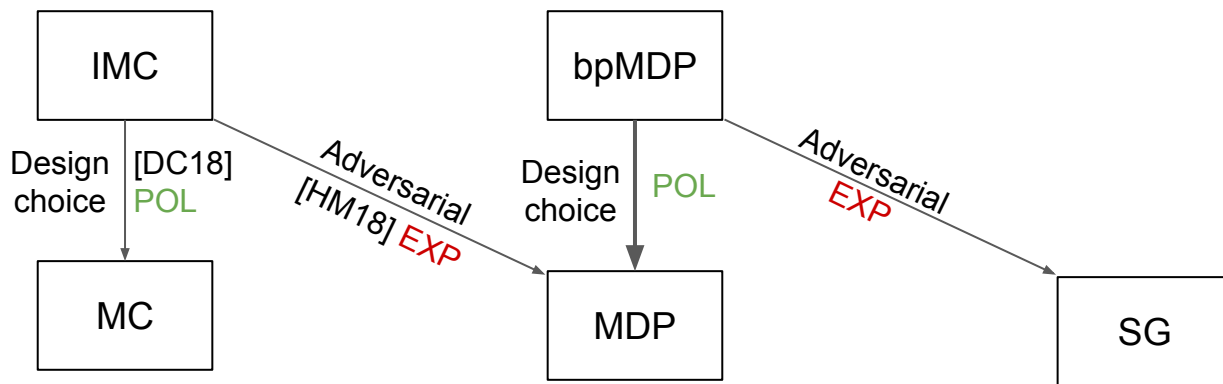
The bigger picture



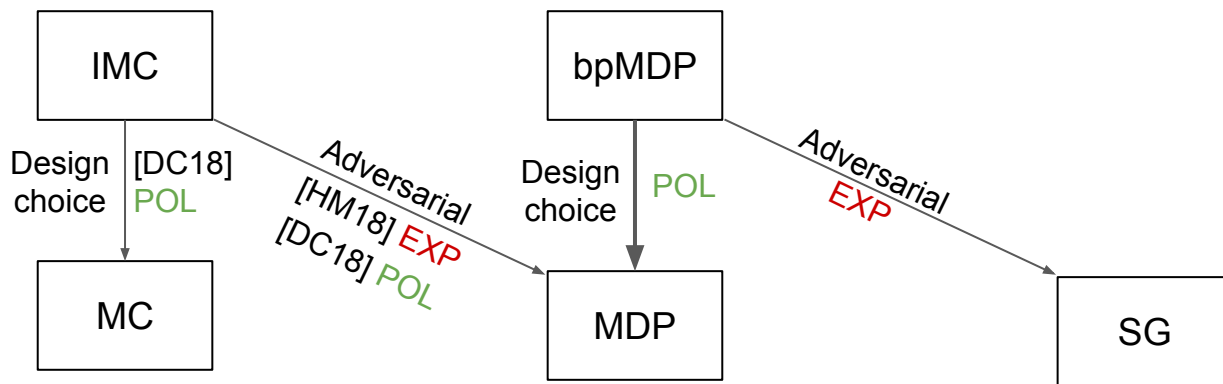
The bigger picture



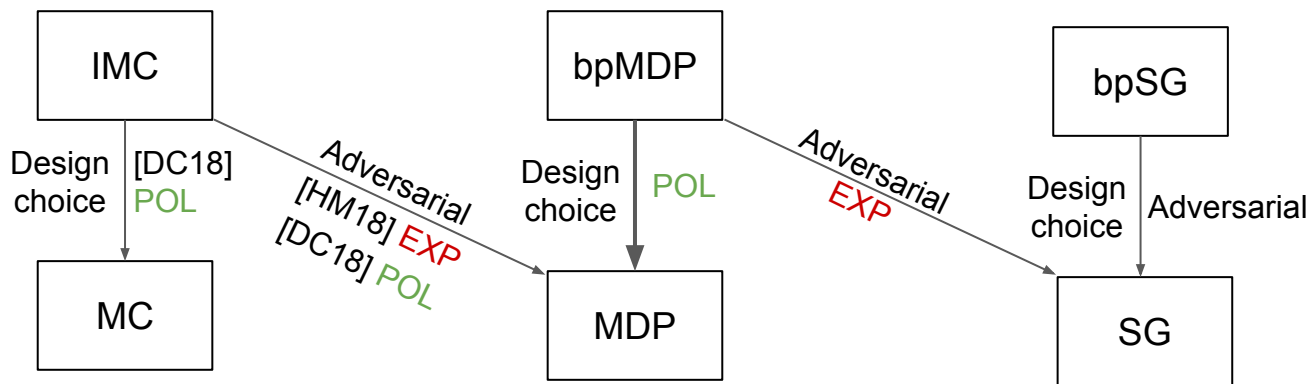
The bigger picture



The bigger picture



The bigger picture



Future work

- Practical implementation (using our previous work)

Future work

- Practical implementation (using our previous work)
- Other imprecisions in system model, e.g. parametrized MDPs
- Multiple objectives