Towards Automated Modeling Assistance: An Efficient Approach for Repairing Flawed Planning Domains

Songtuan Lin Alban Grastien Pascal Bercher
School of Computing, The Australian National University
{songtuan.lin, alban.grastien, pascal.bercher}@anu.edu.au

Introduction

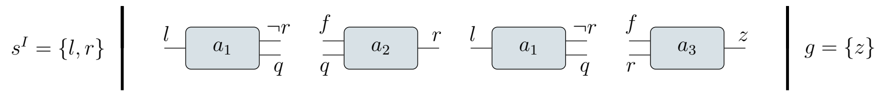
Motivation: The task of modeling planning domains is a major obstacle for deploying AI planning techniques more broadly.

- Tools for modeling assistance are important!
  - E.g., Planning. Domains, itSIMPLE, etc.

Objective: We want to repair a flawed planning domain.

- **Inputs**: A flawed planning domain and a (set of) plan(s) contradicting the flawed domain but demanded to be valid.
- **Output**: A *minimal cardinality* repair set that turns each plan into a solution.

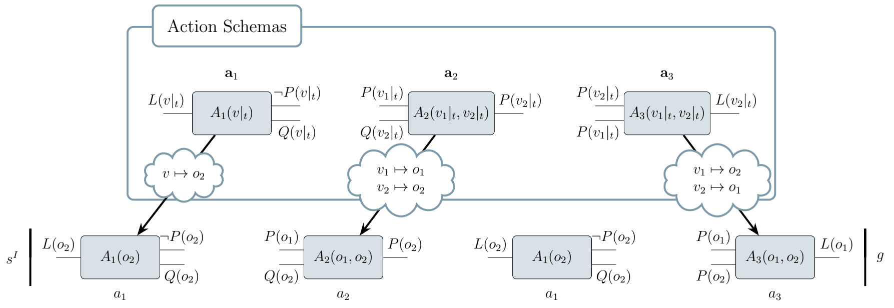
Repairs for Grounded Domains



For each action a, an atomic repair is one of the following:

- $\langle F_a|^p_f\rangle$ with $f \in \texttt{prec}(a)$ – Removing the proposition f from the precondition of a, e.g., $\langle F_{a_3}|^p_r\rangle$ removes r from $\texttt{prec}(a_3)$.
- $\langle F_a|^-_f\rangle$ with $f \in \texttt{eff}^-(a)$ – Removing the proposition f from the negative effects of a, e.g., $\langle F_{a_1}|^-_r\rangle$ removes r from $\texttt{eff}^-(a_1)$.
- $\langle F_a|^+_f\rangle$ – Adding the proposition f to the positive effects of a, e.g., $\langle F_{a_1}|^+_f\rangle$ adds f to $\texttt{eff}^+(a_1)$.

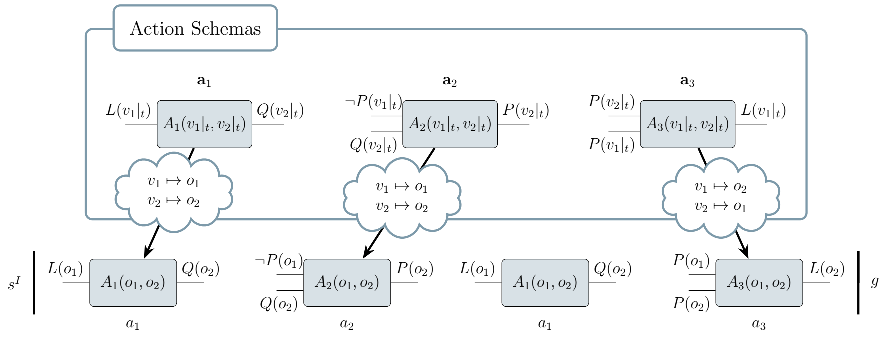
Repairs for Lifted Domains



For each *action schema* $\mathbf{a}$, an atomic repair is one of the following:

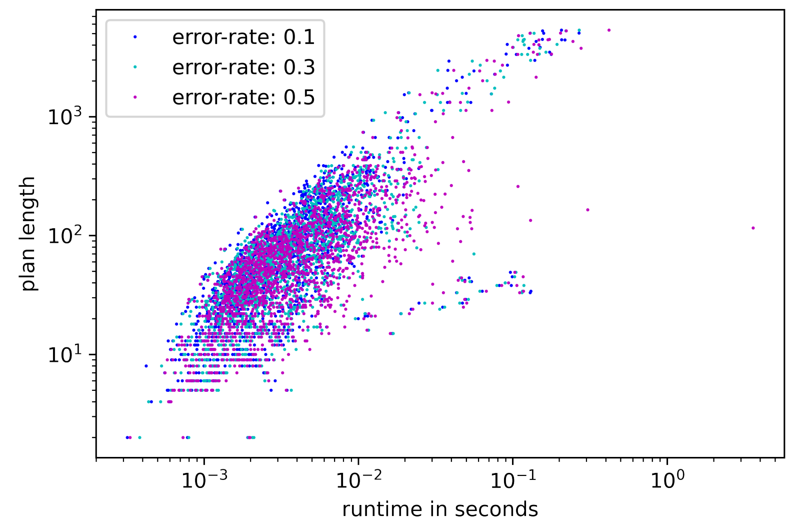
- $\langle \mathbf{F_a}|^p_\mathbf{f}\rangle$ with $\mathbf{f} \in \texttt{prec}(\mathbf{a})$ – Removing the *predicate* $\mathbf{f}$ from the precondition of $\mathbf{a}$.
- $\langle \mathbf{F_a}|^-_\mathbf{f}\rangle$ with $\mathbf{f} \in \texttt{eff}^-(\mathbf{a})$ – Removing the *predicate* $\mathbf{f}$ from the negative effects of $\mathbf{a}$.
- $\langle \mathbf{F_a}|^+_\mathbf{f}\rangle$ – Adding the *predicate* $\mathbf{f}$ to the positive effects of $\mathbf{a}$.
  - Every parameter of $\mathbf{f}$ must also be a parameter of $\mathbf{a}$, e.g., $\langle \mathbf{F_{a_2}}|^+_\mathbf{f}\rangle$ with $\mathbf{f} = Q(v_1|_t)$ or $\mathbf{f} = Q(v_2|_t)$.

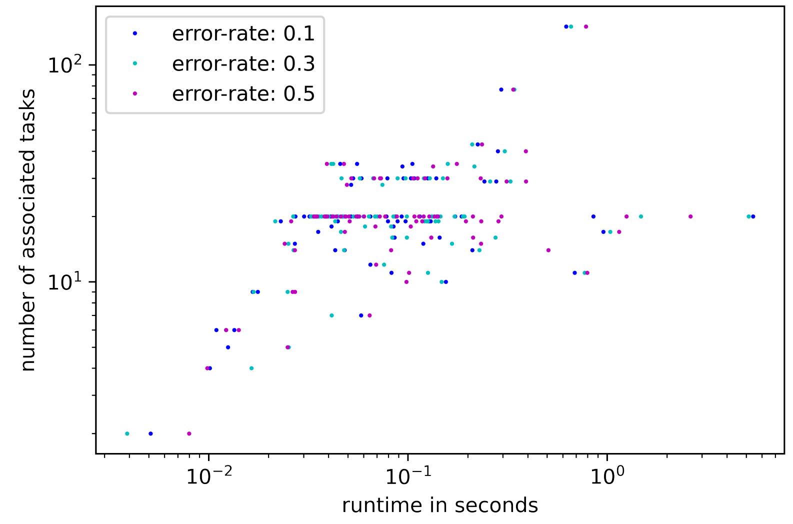
Repairs for Domains with Negative Preconditions



For each action schema $\mathbf{a}$, we define the following *extra* atomic repairs:

- $\langle \mathbf{N_a}|^p_\mathbf{f}\rangle$ with $\mathbf{f} \in \texttt{prec}^-(\mathbf{a})$ – Removing the predicate $\mathbf{f}$ from the *negative* precondition of $\mathbf{a}$.
- $\langle \mathbf{N_a}|^+_\mathbf{f}\rangle$ with $\mathbf{f} \in \texttt{eff}^+(\mathbf{a})$ – *Removing* the predicate $\mathbf{f}$ from the positive effects of $\mathbf{a}$.
- $\langle \mathbf{N_a}|^-_\mathbf{f}\rangle$ – *Adding* the predicate $\mathbf{f}$ to the negative effects of $\mathbf{a}$.





Algorithm

Input: A planning problem Π
A plan $\pi = \langle a_1 \cdots a_n\rangle$
Output: A minimal cardinality repair set δ^* turning π into a solution

$\Theta^* \leftarrow \emptyset$
loop
  $\delta \leftarrow$ a minimal hitting set of Θ^*
  $\Pi^* \leftarrow$ the planning problem obtained by applying δ^* to Π
  if π is a solution to Π^* **then**
    return δ^*
  $\theta \leftarrow$ a conflict with $\theta \cap \delta^* = \emptyset$
  $\Theta^* \leftarrow \Theta^* \cup \{\theta\}$

- A conflict is a set of repairs in which at least one *must* be applied in order to turn π into a solution.

Running Example