A Verified Compositional Algorithm for AI Planning

Mohammad Abdulaziz[1], Charles Gretton[2], Michael Norrish[3]

[1]Technical University of Munich
[2]Australian National University
[3]Data61

Outline

- Background: what is planning?
- Propositionally factored representations
- Compositional planning algorithms
- A (verified) symmetry based compositional algorithm

AI Planning

- Input: a model of the world in terms of actions, initial state, and a goal

AI Planning

- Lorries: L1 L2, Parcels: P1 P2 P3, Cities: C1 C2
- Initial state: L1@C1, L2@C2, P1@C1, P2@C1, P3@C1
- Driving Actions: L1 drives from C1 to C2, L1 drives from C2 to C1,. . .
- Loading Actions: L1 loads P1@C1, L1 loads P1@C2,. . .
- Unloading: L1 unloads P1@C1, L1 unloads P1@C2,. . .
- Goal: P1@C2 and P2@C2

AI Planning

- Input: a model of the world in terms of actions, initial state, and a goal
- Output: a sequence of actions that, if executed at the initial state, reach the goal

AI Planning

- L2 drives from C2 to C1
- L1 loads P1@C1
- L1 drives from C1 to C2
- L1 unloads P1@C2
- L2 loads P2@C1
- L2 drives from C1 to C2
- L2 unloads P2@C2
- L1 drives from C2 to C1
- L1 loads P3@C1
- L1 drives from C1 to C2
- L1 unloads P3@C2

Propositionally Factored Representations

- A planning problem is a reachability problem in a state space, i.e. digraph
- STRIPS [Fikes and Nilsson 1971], SMV [McMillan 1993]

Propositionally Factored Representations

- A planning problem is a reachability problem in a state space, i.e. digraph
- State variables
  - e.g. L1@C1

Propositionally Factored Representations

- A planning problem is a reachability problem in a state space, i.e. digraph
- Actions representing the dynamics
  - e.g. Load(P1,L1,C1) $\equiv$ ({P1@C1,L1@C1},{$\neg$P1@C1,P1@L1})
- Executing an action at a state results in a new state
  - $\text{ex}(x, (p, e)) = \text{if } p \subseteq x \text{ then } e \uplus x \text{ else } x$

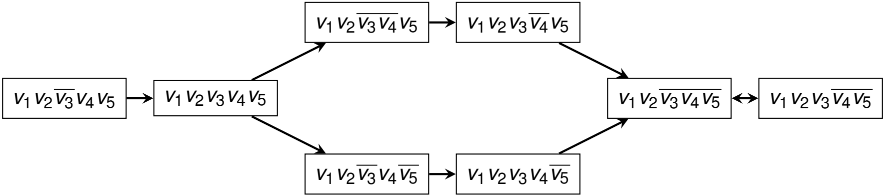
Propositionally Factored Representations

- A planning problem is a reachability problem in a state space, i.e. digraph
- Initial state: an assignment of all variables
  - e.g. {P1@C1,¬P1@C2,P2@C1,¬P2@C2,L1@C1...}
- Goal: an assignment of a subset of the variables
  - e.g. {P1@C2,P2@C2}

Propositionally Factored Representations

- A planning problem is a reachability problem in a state space, i.e. digraph
- Factored representations are more natural and succinct

$$\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},$$
$$\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}$$



Propositionally Factored Representations

- A planning problem is a reachability problem in a state space, i.e. digraph
- In HOL:
  - A state is of type $\alpha \mapsto \texttt{bool}$
  - An action is of type $(\alpha \mapsto \texttt{bool}) \times (\alpha \mapsto \texttt{bool})$
  - A planning problem is a tuple of
    - I: $\alpha \mapsto \texttt{bool}$
    - δ: $(\alpha \mapsto \texttt{bool}) \times (\alpha \mapsto \texttt{bool}) \rightarrow \texttt{bool}$
    - G: $\alpha \mapsto \texttt{bool}$
  - Action execution is

$$\texttt{state-succ}\ x\ (p, e) \stackrel{\text{def}}{=} \textbf{if}\ p \sqsubseteq x\ \textbf{then}\ e \uplus x\ \textbf{else}\ x$$

Compositional Algorithms

- Classical planning is PSPACE-Complete
  - as are reachability problems in other succinct representations
- In practice: incrementally compute the (exponentially) larger explicit state space

- Compositional algorithms:
  - divide the planning problem into sub-problems
  - solve each sub-problem separately
  - compose sub-problem solutions

This talk

- Verifying a compositional algorithm based on symmetries
  - we published it in 2015 in IJCAI
- Why?
  - safety critical applications of planning
  - e.g. Williams and Nayak 1997, check IWPSS 2008-now
- But...
  - planning algorithms are notation/mathematically heavy
  - many easy-to-miss corner cases
  - review process can be sloppy

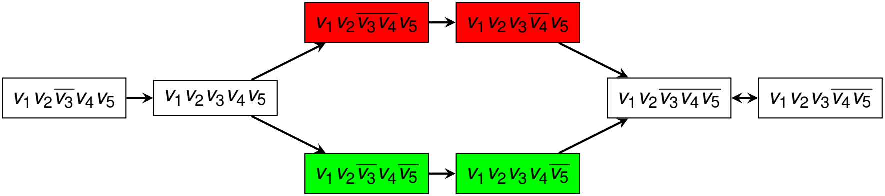
Planning: Symmetries

- Variable symmetry: a problem automorphism
  - a permutation of variables that does not "change" the problem

Planning: Symmetries

- Variable symmetry: a problem automorphism
  - a permutation of variables that does not “change” the problem
- E.g. $\{v_1 \mapsto v_2, v_3 \mapsto v_5\}$

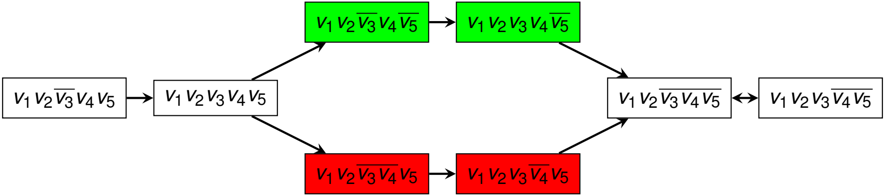
$$\begin{aligned}
&\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},\\
&\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},\\
&\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}
\end{aligned}$$



Planning: Symmetries

- Variable symmetry: a problem automorphism
  - a permutation of variables that does not “change” the problem
- E.g. $\{v_1 \mapsto v_2, v_3 \mapsto v_5\}$

$$\begin{aligned}
&\Pi.I \equiv \{v_2, v_1, v_3, v_5, v_4\},\\
&\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\})\},\\
&\Pi.G \equiv \{\overline{v_5}, \overline{v_4}\}
\end{aligned}$$



Planning: Symmetries

- Variable symmetry: a problem automorphism
  - a permutation of variables that does not “change” the problem
- These automorphisms form a finite group
- Can be computed using graph automorphism tools
  - E.g. NAUTY [McKay 1981]

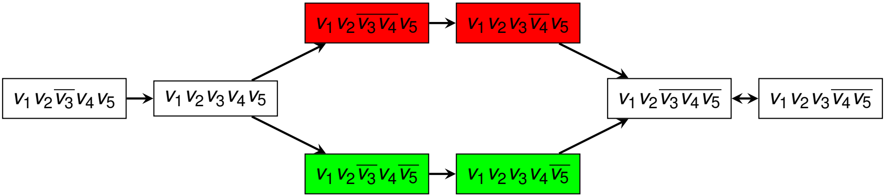
Symmetry-based Compositional Planning

- The automorphism group induces equivalence relations, aka *orbits*
  - on variables, literals, actions, etc.

Symmetry-based Compositional Planning

- The automorphism group induces equivalence relations, aka *orbits*
  - on variables, literals, actions, etc.
- E.g. partition $P \equiv \{p_1 \equiv \{v_1, v_2\}, p_2 \equiv \{v_3\}, p_3 \equiv \{v_4, v_5\}\}$

$$\begin{aligned}
&\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},\\
&\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},\\
&\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}
\end{aligned}$$



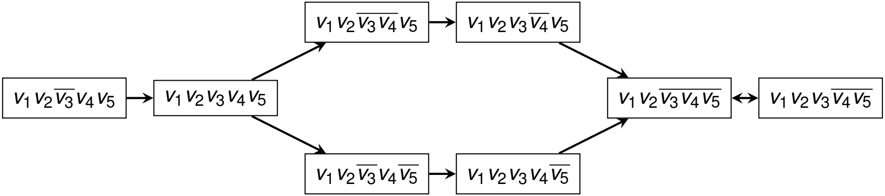
Symmetry-based Compositional Planning

- The automorphism group induces equivalence relations, aka *orbits*
  - on variables, literals, actions, etc.
- Quotient problem: replace every proposition with its orbit

Symmetry-based Compositional Planning

- The automorphism group induces equivalence relations, aka *orbits*
  - on variables, literals, actions, etc.
- Quotient problem: replace every proposition with its orbit
- E.g. $P \equiv \{p_1 \equiv \{v_1, v_2\}, p_2 \equiv \{v_3\}, p_3 \equiv \{v_4, v_5\}\}$

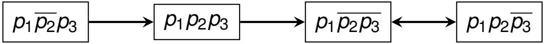
$$
\begin{aligned}
&\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},\\
&\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},\\
&\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}
\end{aligned}
$$



Symmetry-based Compositional Planning

- The automorphism group induces equivalence relations, aka *orbits*
  - on variables, literals, actions, etc.
- Quotient problem: replace every proposition with its orbit
- E.g. $P \equiv \{p_1 \equiv \{v_1, v_2\}, p_2 \equiv \{v_3\}, p_3 \equiv \{v_4, v_5\}\}$

$$
\begin{aligned}
&(\Pi/P).I \equiv \{p_1, p_2, p_3\},\\
&(\Pi/P).\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\},\\
&(\Pi/P).G \equiv \{\overline{p_3}\}
\end{aligned}
$$



Symmetry-based Compositional Planning

- The automorphism group induces equivalence relations, aka *orbits*
  - on variables, literals, actions, etc.
- Quotient problem: replace every proposition with its orbit
- Solving the quotient problem instead of the concrete problem would be great
- We can, if we satisfy two conditions

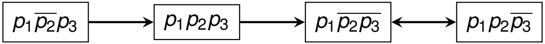
Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- E.g. for $\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}$, $\pitchfork\llparenthesis\Pi/P\rrparenthesis$ is

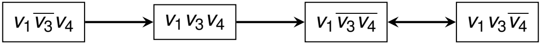
$$
\begin{array}{l}
(\Pi/P).I \equiv \{p_1, p_2, p_3\}, \\
(\Pi/P).\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\}, \\
(\Pi/P).G \equiv \{\overline{p_3}\}
\end{array}
$$



Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- E.g. for $\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}$, $\pitchfork\llparenthesis \Pi / P \rrparenthesis$ is

$$
\begin{aligned}
&(\pitchfork_1 \llparenthesis(\Pi/P)\rrparenthesis).I = \{v_1, v_3, v_4\},\\
&(\pitchfork_1 \llparenthesis(\Pi/P)\rrparenthesis).\delta = \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\})\},\\
&(\pitchfork_1 \llparenthesis(\Pi/P)\rrparenthesis).G = \{\overline{v_4}\}
\end{aligned}
$$



Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{I}$ of the quotient Π/P that *cover* the problem Π

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - for any $\pitchfork \in \mathfrak{T}$, $\pitchfork(\!|\Pi/P|\!) \subseteq \Pi$
    - where $\Pi_1 \subseteq \Pi_2$ iff $\Pi_1.I \subseteq \Pi_2.I$ and $\Pi_1.\delta \subseteq \Pi_2.\delta$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - E.g.

$$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\}$$

$$\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},$$
$$\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}$$

$$(\pitchfork_1 (\!|(\Pi/P)|\!)).I = \{v_1, v_3, v_4\},$$
$$(\pitchfork_1 (\!|(\Pi/P)|\!)).\delta = \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\})\},$$
$$(\pitchfork_1 (\!|(\Pi/P)|\!)).G = \{\overline{v_4}\}$$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - E.g.

$$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\}$$

$$\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},$$
$$\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}$$

$$(\pitchfork_2 (\!|\Pi/P|\!)).I = \{v_2, v_3, v_5\},$$
$$(\pitchfork_2 (\!|\Pi/P|\!)).\delta = \{(\emptyset, \{v_3\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$(\pitchfork_2 (\!|\Pi/P|\!)).G = \{\overline{v_5}\}$$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - for any $\pitchfork \in \mathfrak{T}$, $\pitchfork(\!|\Pi/P|\!) \subseteq \Pi$
    - where $\Pi_1 \subseteq \Pi_2$ iff $\Pi_1.I \subseteq \Pi_2.I$ and $\Pi_1.\delta \subseteq \Pi_2.\delta$
  - for any $\ell \in \Pi.G$, $\exists \pitchfork \in \mathfrak{T}.\ \ell \in (\pitchfork(\!|\Pi/P|\!)).G$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - E.g.

$$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\}$$

$$\begin{aligned}
&\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},\\
&\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},\\
&\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}
\end{aligned}$$

$$\begin{aligned}
&(\pitchfork_1 (\!|(\Pi/P)|\!)).I = \{v_1, v_3, v_4\},\\
&(\pitchfork_1 (\!|(\Pi/P)|\!)).\delta = \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\})\},\\
&(\pitchfork_1 (\!|(\Pi/P)|\!)).G = \{\overline{v_4}\}
\end{aligned}$$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - E.g.

$$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\}$$

$$\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},$$
$$\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}$$

$$(\pitchfork_2(\!|\Pi/P|\!)).I = \{v_2, v_3, v_5\},$$
$$(\pitchfork_2(\!|\Pi/P|\!)).\delta = \{(\emptyset, \{v_3\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$(\pitchfork_2(\!|\Pi/P|\!)).G = \{\overline{v_5}\}$$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - for any $\pitchfork \in \mathfrak{T}$, $\pitchfork(\!|\Pi/P|\!) \subseteq \Pi$
    - where $\Pi_1 \subseteq \Pi_2$ iff $\Pi_1.I \subseteq \Pi_2.I$ and $\Pi_1.\delta \subseteq \Pi_2.\delta$
  - for any $\ell \in \Pi.G$, $\exists \pitchfork \in \mathfrak{T}.\ \ell \in (\pitchfork(\!|\Pi/P|\!)).G$
  - for any $\pitchfork \in \mathfrak{T}$, $\pitchfork$ is a *transversal* of P
    - i.e. for any $p \in P.\ \pitchfork(p) \in p$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{T}$ of the quotient Π/P that *cover* the problem Π
  - E.g.

$$\begin{aligned}\mathfrak{T} \equiv \{ & \pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \\ & \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\}\end{aligned}$$

and

$$P \equiv \{p_1 \equiv \{v_1, v_2\}, p_2 \equiv \{v_3\}, p_3 \equiv \{v_4, v_5\}\}$$

Symmetry-based Compositional Planning: condition 1

- This is a condition on *instantiations* of the quotient problem
- Instantiation is an analogue of function images on sets
  - applies to states, actions and problems
- There must be a set of instantiations $\mathfrak{I}$ of the quotient Π/P that *cover* the problem Π
  - for any $\pitchfork \in \mathfrak{I}$, $\pitchfork(\!|\Pi/P|\!) \subseteq \Pi$
    - where $\Pi_1 \subseteq \Pi_2$ iff $\Pi_1.I \subseteq \Pi_2.I$ and $\Pi_1.\delta \subseteq \Pi_2.\delta$
  - for any $\ell \in \Pi.G$, $\exists \pitchfork \in \mathfrak{I}.\ \ell \in (\pitchfork(\!|\Pi/P|\!)).G$
  - for any $\pitchfork \in \mathfrak{I}$, $\pitchfork$ is a *transversal* of P
    - i.e. for any $p \in P.\ \pitchfork(p) \in p$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- Needed assignments, $\mathcal{N}(\Pi)$: assignments in the goal or action preconditions that also occur in I
  - i.e. $\mathcal{N}(\Pi) = (\text{pre}(\delta) \cap I) \cup (G \cap I)$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- E.g. $\mathcal{N}(\Pi/P) = \{p_1, p_2\}$

$$(\Pi/P).I \equiv \{p_1, p_2, p_3\},$$
$$(\Pi/P).\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\},$$
$$(\Pi/P).G \equiv \{\overline{p_3}\}$$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- Needed assignments, $\mathcal{N}(\Pi)$: assignments in the goal or action preconditions that also occur in I
  - i.e. $\mathcal{N}(\Pi) = (\text{pre}(\delta) \cap I) \cup (G \cap I)$
- Common variables $\bigcap_v \mathfrak{T}$: orbits mapped to the same variable by more than one member of $\mathfrak{T}$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- E.g. for
  $$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\},$$
  the common variables are $\{p_2\}$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- Needed assignments, $\mathcal{N}(\Pi)$: assignments in the goal or action preconditions that also occur in I
  - i.e. $\mathcal{N}(\Pi) = (\text{pre}(\delta) \cap I) \cup (G \cap I)$
- Common variables $\bigcap_v \mathfrak{T}$: orbits mapped to the same variable by more than one member of $\mathfrak{T}$
- A set of variables *vs* is sustainable in a problem Π iff every $v \in vs$ is assigned to the same value by $\Pi.I$ and $\Pi.G$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- E.g. p_3 is *not* sustainable in Π/P

$$
\begin{aligned}
&(\Pi/P).I \equiv \{p_1, p_2, p_3\},\\
&(\Pi/P).\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\},\\
&(\Pi/P).G \equiv \{\overline{p_3}\}
\end{aligned}
$$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- Needed assignments, $\mathcal{N}(\Pi)$: assignments in the goal or action preconditions that also occur in I
  - i.e. $\mathcal{N}(\Pi) = (\text{pre}(\delta) \cap I) \cup (G \cap I)$
- Common variables $\bigcap_v \mathfrak{T}$: orbits mapped to the same variable by more than one member of $\mathfrak{T}$
- A set of variables *vs* is sustainable in a problem Π iff every $v \in vs$ is assigned to the same value by $\Pi.I$ and $\Pi.G$
- $(\bigcap_v \mathfrak{T}) \cap \mathcal{D}(\mathcal{N}(\Pi/P))$ are sustainable in Π/P
  - where $\mathcal{D}(x) \equiv \{v \mid (v \mapsto b) \in x\}$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- Needed assignments, $\mathcal{N}(\Pi)$: assignments in the goal or action preconditions that also occur in I
  - i.e. $\mathcal{N}(\Pi) = (\text{pre}(\delta) \cap I) \cup (G \cap I)$
- Common variables $\bigcap_v \mathfrak{T}$: orbits mapped to the same variable by more than one member of $\mathfrak{T}$
- A set of variables vs is sustainable in a problem Π iff every $v \in vs$ is assigned to the same value by $\Pi.I$ and $\Pi.G$
- $(\bigcap_v \mathfrak{T}) \cap \mathcal{D}(\mathcal{N}(\Pi/P))$ are sustainable in Π/P
  - where $\mathcal{D}(x) \equiv \{v \mid (v \mapsto b) \in x\}$
- We guarantee that if we augment the quotient's goal with the assignments $(\bigcap_v \mathfrak{T}) \cap \mathcal{D}(\mathcal{N}(\Pi))$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- E.g. for

  $$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\},$$
  $$\pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\},$$

  the common variables are $\{p_2\}$
  the variables whose assignments are needed are $\{p_1, p_2\}$

  so we need to add p_2 to the $(\Pi/P).G$

$$(\Pi/P).I \equiv \{p_1, p_2, p_3\},$$
$$(\Pi/P).\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\},$$
$$(\Pi/P).G \equiv \{\overline{p_3}\}$$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- E.g. for

$$\mathfrak{T} \equiv \{\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\}, \pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\},$$

  the common variables are $\{p_2\}$
  the variables whose assignments are needed are $\{p_1, p_2\}$

  so we need to add p_2 to the $(\Pi/P).G$

$$\Pi^q.I \equiv \{p_1, p_2, p_3\},$$
$$\Pi^q.\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\},$$
$$\Pi^q.G \equiv \{p_2, \overline{p_3}\}$$

Symmetry-based Compositional Planning: condition 2

- This is a condition involving the quotient problem
- Needed assignments, $\mathcal{N}(\Pi)$: assignments in the goal or action preconditions that also occur in I
  - i.e. $\mathcal{N}(\Pi) = (\text{pre}(\delta) \cap I) \cup (G \cap I)$
- Common variables $\bigcap_v \mathfrak{T}$: orbits mapped to the same variable by more than one member of $\mathfrak{T}$
- A set of variables vs is sustainable in a problem Π iff every $v \in vs$ is assigned to the same value by $\Pi.I$ and $\Pi.G$
- $(\bigcap_v \mathfrak{T}) \cap \mathcal{D}(\mathcal{N}(\Pi/P))$ are sustainable in Π/P
  - where $\mathcal{D}(x) \equiv \{v \mid (v \mapsto b) \in x\}$
- We guarantee that if we augment the quotient's goal with the assignments $(\bigcap_v \mathfrak{T}) \cap \mathcal{D}(\mathcal{N}(\Pi))$

Symmetry-based Compositional Planning (Cont.)

- Now, the algorithm is:
  1. compute problem automorphisms
  2. compute quotient problem
  3. compute covering instantiations of the quotient
  4. augment the quotient problem's goal

Symmetry-based Compositional Planning (Cont.)

- Now, the algorithm is:
  1. compute problem automorphisms
  2. compute quotient problem
  3. compute covering instantiations of the quotient
  4. augment the quotient problem's goal
  5. solve the augmented quotient

Symmetry-based Compositional Planning (Cont.)

E.g. $[(\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\}), (\emptyset, \{p_2\})]$

$$\Pi^q.I \equiv \{p_1, p_2, p_3\},$$
$$\Pi^q.\delta \equiv \{(\emptyset, \{p_2\}), (\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\})\},$$
$$\Pi^q.G \equiv \{p_2, \overline{p_3}\}$$

Symmetry-based Compositional Planning (Cont.)

- Now, the algorithm is:
  1. compute problem automorphisms
  2. compute quotient problem
  3. compute covering instantiations of the quotient
  4. augment the quotient problem's goal
  5. solve the augmented quotient
  6. instantiate the quotient sol. with all instantiations

Symmetry-based Compositional Planning (Cont.)

E.g. for $[(\{p_1, p_2\}, \{\overline{p_2}, \overline{p_3}\}), (\emptyset, \{p_2\})]$, and

$$\begin{aligned}\mathfrak{T} \equiv \{&\pitchfork_1 \equiv \{p_1 \mapsto v_1, p_2 \mapsto v_3, p_3 \mapsto v_4\},\\ &\pitchfork_2 \equiv \{p_1 \mapsto v_2, p_2 \mapsto v_3, p_3 \mapsto v_5\}\}\end{aligned}$$

the instantiated plans are
$[(\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\emptyset, \{v_3\})]$, and
$[(\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\}), (\emptyset, \{v_3\})]$

Symmetry-based Compositional Planning (Cont.)

- Now, the algorithm is:
  1. compute problem automorphisms
  2. compute quotient problem
  3. compute covering instantiations of the quotient
  4. augment the quotient problem's goal
  5. solve the augmented quotient
  6. instantiate the quotient sol. with all instantiations
  7. concatenate all instantiations of quotient sol., in any order

Symmetry-based Compositional Planning (Cont.)

E.g. $[(\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\emptyset, \{v_3\})]$, and
$[(\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\}), (\emptyset, \{v_3\})]$

$$\Pi.I \equiv \{v_1, v_2, v_3, v_4, v_5\},$$
$$\Pi.\delta \equiv \{(\emptyset, \{v_3\}), (\{v_1, v_3\}, \{\overline{v_3}, \overline{v_4}\}), (\{v_2, v_3\}, \{\overline{v_3}, \overline{v_5}\})\},$$
$$\Pi.G \equiv \{\overline{v_4}, \overline{v_5}\}$$

Symmetry-based Compositional Planning (Cont.)

- Now, the algorithm is:
    1. compute problem automorphisms
    2. compute quotient problem
    3. compute covering instantiations of the quotient
    4. augment the quotient problem's goal
    5. solve the augmented quotient
    6. instantiate the quotient sol. with all instantiations
    7. concatenate all instantiations of quotient sol., in any order

Fruits of the Project: Bugs

- The previously shown algorithm fails if
  - the quotient plan had an unactivated action due unsatisfied preconditions
  - that action could be activated when instantiations of the quotient sol. are concatenated
  - this can compromise the final state reached by the concatenated instantiations

Fruits of the Project: Bugs

- Now, the algorithm is:
  1. compute problem automorphisms
  2. compute quotient problem
  3. compute covering instantiations of the quotient
  4. augment the quotient problem's goal
  5. solve the augmented quotient
  6. remove unexecutable actions from quotient sol.
  7. instantiate the quotient sol. with all instantiations
  8. concatenate all instantiations of quotient sol., in any order

Fruits of the Project: Bugs

- The previously shown algorithm fails if
  - the quotient plan had an unactivated action due unsatisfied preconditions
  - that action could be activated when instantiations of the quotient sol. are concatenated
  - this can compromise the final state reached by the concatenated instantiations
- In the definition of the "sub-problem" relation
  - $\Pi_1 \subseteq \Pi_2$ iff $\Pi_1.I \subseteq \Pi_2.I$ and $\Pi_1.\delta \subseteq \Pi_2.\delta$

Fruits of the Project: Other

- Generalised the algorithm
  - discovered the algorithm works if state variables are not Boolean
  - i.e. a state is of type $\alpha \mapsto \beta$ and not $\alpha \mapsto \texttt{bool}$

Symmetry-based Compositional Planning (Cont.)

$\vdash$ **let**

Π^q =

Π' with G := $\Pi'.I|_{\bigcap_v \text{(set } \mathfrak{T}) \; \mathcal{D}(\Pi') \cap \mathcal{D}(\mathcal{N}(\Pi'))}$ $\uplus$ $\Pi'.G$;

inst_plans =

MAP $(\lambda \pitchfork.$ rem-cless $(\mathcal{N}(\pitchfork(\!|\Pi^q|\!)), [\,], \pitchfork(\!|\vec{\pi}^q|\!)))$

$\mathfrak{T}$;

concatenated_plans = FLAT *inst_plans*

in

ALL-DISTINCT $\mathfrak{T}$ $\wedge$

$(\forall \pitchfork.$ MEM $\pitchfork$ $\mathfrak{T}$ $\Rightarrow$ valid-inst $\pitchfork)$ $\wedge$ valid-prob Π' $\wedge$

INJ $(\lambda \pitchfork.\ \pitchfork(\!|\Pi^q|\!))$ (set $\mathfrak{T}$)

$\mathcal{U}(:(\alpha,\ \beta)$ problem$)$ $\wedge$

pwise-valid (set $\mathfrak{T}$) $\mathcal{D}(\Pi^q)$ $\wedge$

covers (MAP $(\lambda \pitchfork.\ \pitchfork(\!|\Pi^q|\!))$ $\mathfrak{T}$) Π $\wedge$ Π^q solved-by $\vec{\pi}^q$ $\Rightarrow$

Π solved-by *concatenated_plans*

Conclusion

- We used HOL4 to analyse our own algorithm at an abstract level
- Lead to
  - deeper understanding
  - finding bugs not found via testing nor peer review
  - generalising the algorithm