Relaxing the Relaxed Exist-Step Parallel Planning Semantics

Tomáš Balyo (biotomas@gmail.com)
Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic
What is Planning?

- World states are described as values of state variables.
- Actions change the state of the world by changing the values of state variables by their effects.
- Actions also have preconditions and are applicable only when their preconditions hold in the given state.

**Objective:** given a set $a$ of actions, an initial world state and the description of a goal state find a valid sequence of actions that transforms the world from the initial state to a goal state.
State Variables and their domains:
- Truck location $T$, $\text{dom}(T) = \{\text{LA, SF, LV}\}$
- Package locations $P$ and $Q$
  $\text{dom}(P) = \text{dom}(Q) = \{\text{LA, SF, LV, Tr}\}$

Initial State: $T=\text{LA}$, $P=\text{LA}$, $Q=\text{SF}$
Goal State: $P=\text{LV}$, $Q=\text{LV}$

Actions:
- $\text{move}(x,y) = \{\text{prec: } T=x, \text{ eff: } T=y\}$
- $\text{loadP}(x) = \{\text{prec: } T=x, P=x, \text{ eff: } P=\text{Tr}\}$
- $\text{loadQ}(x) = \{\text{prec: } T=x, Q=x, \text{ eff: } Q=\text{Tr}\}$
- $\text{dropP}(x) = \{\text{prec: } T=x, P=\text{Tr}, \text{ eff: } P=x\}$
- $\text{dropQ}(x) = \{\text{prec: } T=x, Q=\text{Tr}, \text{ eff: } Q=x\}$

Where $x, y$ are LA, SF, and LV

Plan: $\text{loadP(LA)}$, $\text{move(LA, SF)}$, $\text{loadQ(SF)}$, $\text{move(SF, LV)}$, $\text{dropP(LV)}$, $\text{dropQ(LV)}$
Enabling Graph

\[ V = \{ \text{Actions} \} \]
\[ E = \{ (A_1 \rightarrow A_2), \text{eff}(A_1) \cap \text{prec}(A_2) \neq 0 \} \]
Planning as SATisfiability

- Construct a formula $F_k$ such that it is satisfiable (if and) only if there is a plan of at most $k$ steps.
- Solve $F_1$, $F_2$, … using a SAT solver until you reach a satisfiable formula $F_n$.
- Extract a plan from the satisfying assignment of $F_n$.
- $n$ is the called the makespan of the plan.

- **What actions can go inside a step together?**
  - If more action could be in a step then we would need fewer steps to find a plan.
What actions can go inside a step together?

1. foreach step semantics

- The preconditions of all actions in a step must already hold in the beginning of the step
- The effects of all actions must hold at the end of this step
- The actions in a step do not interfere – they cannot destroy each others preconditions by their effects
- The actions in a step can be turned into a valid subplan sequence

Plan: \{\text{loadP(LA)}\} \;\&\; \{\text{move(LA, SF)}\} \;\&\; \{\text{loadQ(SF)}\} \;\&\; \{\text{move(SF, LV)}\} \;\&\; \{\text{dropP(LV), dropQ(LV)}\} – 5 \text{ steps}
What actions can go inside a step together?

2. exist step semantics

- The preconditions of all actions in a step must already hold in the beginning of the step
- The effects of all actions must hold at the end of this step
- The actions in a step do not interfere—they cannot destroy each others preconditions by their effects
- The actions in a step can be turned into a valid subplan sequence

Plan: {loadP(LA), move(LA, SF)} ♦ {loadQ(SF), move(SF, LV)} ♦ {dropP(LV), dropQ(LV)} – 3 steps
What actions can go inside a step together?

3. relaxed exist step semantics

- The preconditions of all actions in a step must already hold in the beginning of the step
- The effects of all actions must hold at the end of this step
- The actions in a step do not interfere—they cannot destroy each others preconditions by their effects
- The actions in a step can be turned into a valid subplan sequence

Plan: \{loadP(LA), move(LA, SF), loadQ(SF)\} ⊗ \{move(SF, LV), dropP(LV), dropQ(LV)\} – 2 steps
What actions can go inside a step together?

4. relaxed relaxed exist step semantics

- The preconditions of all actions in a step must already hold in the beginning of the step.
- The effects of all actions must hold at the end of this step.
- The actions in a step do not interfere—they cannot destroy each others preconditions by their effects.
- The actions in a step can be turned into a valid subplan sequence.

Plan: \{loadP(LA), move(LA, SF), loadQ(SF), move(SF, LV), dropP(LV), dropQ(LV)\} – 1 step
Basic ideas of the relaxed relaxed exist step SAT encoding

- The SAT encoding only approximates the semantics, i.e., the satisfiability of the constructed formula $F_k$ implies the existence of a $k$-step plan (not vice versa)
- The actions are ranked using cycle–ignoring topological sorting on the enabling graph (action ranking can be arbitrary as long as it is injective)
- The encoding allows only lower ranking actions before higher ranking ones in a step
- The encoding uses implication chains similar to those used in the exist step and relaxed exist step step encoding
Experimental Results

- IPC 2012 domains (20 problems each), time limit 30 minutes

<table>
<thead>
<tr>
<th>Domain</th>
<th>Foreach Step</th>
<th>Exist Step</th>
<th>Relaxed Relaxed E.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Solved</td>
<td>Avg. Steps</td>
<td>Solved</td>
</tr>
<tr>
<td>Barman</td>
<td>8</td>
<td>46.3</td>
<td>8</td>
</tr>
<tr>
<td>Elevators</td>
<td>20</td>
<td>9.5</td>
<td>20</td>
</tr>
<tr>
<td>Parcprinter</td>
<td>20</td>
<td>13.5</td>
<td>20</td>
</tr>
<tr>
<td>Pegsol</td>
<td>7</td>
<td>22.8</td>
<td>13</td>
</tr>
<tr>
<td>Storage</td>
<td>15</td>
<td>9.2</td>
<td>19</td>
</tr>
<tr>
<td>Visitall</td>
<td>9</td>
<td>27.0</td>
<td>11</td>
</tr>
<tr>
<td>Woodwork</td>
<td>20</td>
<td>3.4</td>
<td>20</td>
</tr>
<tr>
<td>Zenotravel</td>
<td>16</td>
<td>5.9</td>
<td>16</td>
</tr>
</tbody>
</table>
Conclusion

- We have defined a novel parallel planning semantics and a SAT encoding which approximates it.
- The results of the experiments show that the new encoding is successful in solving IPC benchmark problems.
- For the domains Pegsol, Barman, and Visitall, we achieved a significant improvement in the number of solved instances.
- The average number of required steps decreased for all domains, most significantly for the Visitall domain.
- The encoding can be further improved to produce smaller formulas and to better approximate the defined semantics.