A Reminder about the Importance of Computing and Exploiting Invariants in Planning

Vidal Alcázar, Álvaro Torralba

PLG @ Universidad Carlos III de Madrid
http://www.plg.inf.uc3m.es/~valcazar
FAI @ Saarland University
http://fai.cs.uni-saarland.de/torralba/

ICAPS – June 9, 2015
Motivation

Invariants are known to be useful:

- FDR representation, regression, partial-order planning, SAT,...
- Several methods proposed: here $h^2$

Some aspects have been overlooked and/or appear scattered in the literature:

- Implementation details of $h^2$
- Direction of the computation of the invariants
- Huge impact in some domains!
State invariants:

- Mutexes: \( \neg((\text{at robot loc}_1) \land (\text{at robot loc}_2)) \)
- “exactly-one” invariant groups:
  \((\text{at robot loc}_1) \lor \cdots \lor (\text{at robot loc}_n)) + \text{pairwise mutexes}\)

A (slightly) more general definition of spurious state:

- State that **cannot belong to a solution path**
  \(\Rightarrow\) instead of state unreachable from \(s_0\)
- Detectable when they are inconsistent with invariants
Spurious State

Floortile domain: robots can only paint up or down

- $s_i$ is a forward dead end, and hence spurious
  - ... but does it violate some invariant?
How does $h^2$ work?

Reachability analysis in $P^2$: with conjunctions of two original atoms

- Unreachable $h^2$ atoms are mutexes
  
  - (at robot loc1) $\land$ (at robot loc2) is an unreachable $h^2$ atom

- Unreachable actions in $P^2$ are spurious!
  
  - Spurious actions are never applicable in progression, but can be (wrongly) used in regression, abstractions, heuristics...
  
  - Kind of obvious, but not highlighted/evaluated yet
Negated atoms in $h^2$

$h^2$ was originally described in STRIPS, atoms are propositions

- **Negated propositions matter**, though. See *Matching-Blocksworld*:

\[
\begin{array}{c}
\text{Mutex } \{ (on \; a \; b), \neg(solid \; b) \} \text{ not found by } h^2! \\
\end{array}
\]

- **Negated atoms must be explicitly represented**, unless they belong to an “exactly-one” invariant group
Encoding extra information in actions

Disambiguate implicit preconditions and effects

→ find the value of some variables
→ Use mutexes in \( h^2 \) propagation

It may allow finding more mutexes and spurious actions!

Example: Throw-paint pre \{\}, eff \{(painted loc4), (low-battery)\} 
If you know that (at-robot loc1) and (low-battery) are mutex then

1. \(\neg\)(at-robot loc1) is a precondition of throw-paint
2. and (painted loc4), (at-robot loc1) may be a mutex now
$h^2$ in regression

$h^2$ is a reachability analysis on $P^2$

- It can be done on a **reversed version** of $P^2$ too!!
  1. Disambiguate $S_*$, assume unknown atoms are true
  2. Perform $h^2$ with reversed and disambiguated actions

- Already implemented by Petterson(2005) and Haslum(2008)
h² in regression

h² is a reachability analysis on P²
- It can be done on a reversed version of P² too!!
  1. Disambiguate $S_\star$, assume unknown atoms are true
  2. Perform h² with reversed and disambiguated actions
- Already implemented by Petterson(2005) and Haslum(2008)

Reason for a more general definition of spurious state
- Doesn’t always depend on $s_0$
- Other invariants are used to enrich h²
h^2 in regression: trucks with fuel

- \( S_\star \) is (at-truck goal)
- The pairs (at-truck goal) \( \land \) (fuel \( n \)) are assumed to be true

\[
(\text{at-truck goal}) \land (\text{fuel } n) \xrightarrow{\text{regression}} (\text{at-truck } \text{loc}_x) \land (\text{fuel } n+1)
\]

- Unreachable pairs in regression are mutex:
  \{ (at-truck distance2toGoal), (fuel level1) \}
- If encountered forward, the state is a dead end
- move (loc\( x \) locDistance2toGoal fuel2 fuel1) is spurious
Disambiguate goal: robot in bottom row

Run bw-h2:

1. All the paint-down actions are discarded by bw-h² in Floortile!
2. $S_i$ contains binary mutexes (painted tile1-2) $\land$ (not-painted tile1-3)
Our algorithm

1. \(Fw-h2 \rightarrow \) find mutexes and spurious actions
2. Disambiguate actions and goal
3. \(Bw-h2 \rightarrow \) find mutexes and spurious actions
4. If \(bw-h2\) found something new: disambiguate and repeat \(fw-h2\)
5. If \(fw-h2\) found something new: disambiguate and repeat \(bw-h2\)

Return set of valid operators, \(fw\)-mutexes and \(bw\)-mutexes
State invariants in benchmark domains

- **Low overhead**: 300s threshold enough except in 3 domains
- **h2 fw-mutexes**: 33 out of 44 domains
- **h2 bw-mutexes**: 16 out of 44 domains
- **Multiple iterations** in 11 out of 44 domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>% Facts</th>
<th>% Ops</th>
<th>Domain</th>
<th>% Facts</th>
<th>% Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tidybot</td>
<td>31</td>
<td>85</td>
<td>Scan-08</td>
<td>0</td>
<td>43</td>
</tr>
<tr>
<td>Airport</td>
<td>38</td>
<td>73</td>
<td>Pegsol-08</td>
<td>14</td>
<td>30</td>
</tr>
<tr>
<td>Parc-11</td>
<td>28</td>
<td>68</td>
<td>Floortile</td>
<td>18</td>
<td>38</td>
</tr>
<tr>
<td>Woodw-11</td>
<td>4</td>
<td>52</td>
<td>Nomystery</td>
<td>6</td>
<td>38</td>
</tr>
<tr>
<td>Trucks</td>
<td>5</td>
<td>46</td>
<td>Sokoban-11</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>TPP</td>
<td>12</td>
<td>45</td>
<td>Mystery</td>
<td>6</td>
<td>23</td>
</tr>
</tbody>
</table>
Empirical Results

Time: (optimal benchmarks)
## Coverage: Highlighted Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>#</th>
<th>Blind h^2</th>
<th>h^2+mut</th>
<th>LM-cut h^2</th>
<th>Satisficing FD h^2</th>
<th>Satisficing LAMA h^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>50</td>
<td>22</td>
<td>+5</td>
<td>28</td>
<td>37</td>
<td>35</td>
</tr>
<tr>
<td>Floortile-11</td>
<td>20</td>
<td>2</td>
<td>+6</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Parcprinter-11</td>
<td>20</td>
<td>6</td>
<td>+10</td>
<td>13</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Pipes-notank</td>
<td>50</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td>44</td>
<td>43</td>
</tr>
<tr>
<td>Sokoban-08</td>
<td>30</td>
<td>22</td>
<td>+5</td>
<td>30</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>Tidybot-11</td>
<td>20</td>
<td>12</td>
<td>0</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Woodwork-11</td>
<td>20</td>
<td>3</td>
<td>+1</td>
<td>12</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td><strong>Σ</strong></td>
<td>1396</td>
<td><strong>533</strong></td>
<td><strong>+41</strong></td>
<td><strong>747</strong></td>
<td><strong>1138</strong></td>
<td><strong>1296</strong></td>
</tr>
</tbody>
</table>

Vidal Alcázar, Álvaro Torralba

A Reminder about Invariants
Conclusions

- Computing $h^2$ invariants is very helpful!
  - Both forward and backward
  - Simply remove operators inconsistent with invariants
  - Increases coverage for different optimal and satisficing planners

- Important implementation details
  - Disambiguation
  - Negated propositions
  - Spurious actions
Thanks for your attention

Questions?