Certified Unsolvability for SAT Planning with Property Directed Reachability

Salomé Eriksson    Malte Helmert

University of Basel, Switzerland

ICAPS 2020
Certifying Algorithms

Certifying Algorithm

Emit *certificate* alongside answer, verify *independently*.

In planning:
- solvable: plan
- unsolvable: unsolvability certificate, e.g. [E et al. 2018]

Desired Certificate Properties
- sound & complete
- efficient generation $\rightarrow$ polynomial in planner runtime
- efficient verification $\rightarrow$ polynomial in certificate size
- general
Covered So Far

- explicit & symbolic search
- different heuristics
- $h^2$ preprocessing
- Trapper

SAT-based planning?

- traditionally less suited for detecting unsolvability
- verifying properties of CNF formulas \textbf{NP}-complete
reasons about layers $L_i$:

- overapproximates states with distance $\leq i$ to goal
- iterative refinement
- represented as
  - CNF $\rightarrow$ requires SAT solver
  - dual-Horn (for STRIPS tasks)

\[ L_u = L_{u-1} \rightarrow \text{unsolvable} \]
Unsolvability Proof System [E et al. 2018]

collection of knowledge about *sets of states*

- subset relations
- deadness of state sets

\{I\} or \(G\) dead \(\rightarrow\) task unsolvable

gaining & verifying knowledge:

- basic statements \(A \subseteq B\)
  \(\rightarrow\) need to be verified *semantically*

- inference rules \(A \subseteq B\) and \(B\) dead \(\rightarrow\) \(A\) dead
  \(\rightarrow\) need to be verified *syntactically*
PDR Unsolvability Certificate

PDR Argument

$L_u = L_{u-1} \rightarrow \text{unsolvable}$

certificate translation:

<table>
<thead>
<tr>
<th>#</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
</table>
PDR Unsolvability Certificate

PDR Argument

\[ L_u = L_{u-1} \rightarrow \text{unsolvable} \]

certificate translation:

<table>
<thead>
<tr>
<th>#</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>([A]L_u \subseteq L_u)</td>
<td>basic statement</td>
</tr>
</tbody>
</table>
PDR Unsolvability Certificate

PDR Argument

\[ L_u = L_{u-1} \rightarrow \text{unsolvable} \]

certificate translation:

<table>
<thead>
<tr>
<th>#</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>([A]L_u \subseteq L_u)</td>
<td>basic statement</td>
</tr>
<tr>
<td>(2)</td>
<td>({I} \subseteq \overline{L_u})</td>
<td>basic statement</td>
</tr>
</tbody>
</table>
PDR Unsolvability Certificate

PDR Argument

\[ L_u = L_{u-1} \rightarrow \text{unsolvable} \]

certificate translation:

<table>
<thead>
<tr>
<th>#</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>([A]L_u \subseteq L_u)</td>
<td>basic statement</td>
</tr>
<tr>
<td>(2)</td>
<td>({I} \subseteq \overline{L_u})</td>
<td>basic statement</td>
</tr>
<tr>
<td>(3)</td>
<td>(L_u) is dead</td>
<td>from (1) and (2) with rule \textbf{RI}</td>
</tr>
</tbody>
</table>

\[ I \cdot \]

\[ L_u \]
PDR Unsolvability Certificate

PDR Argument

\[ L_u = L_{u-1} \rightarrow \text{unsolvable} \]

certificate translation:

<table>
<thead>
<tr>
<th>#</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>([A]L_u \subseteq L_u)</td>
<td>basic statement</td>
</tr>
<tr>
<td>(2)</td>
<td>({I} \subseteq \overline{L_u})</td>
<td>basic statement</td>
</tr>
<tr>
<td>(3)</td>
<td>(L_u) is dead</td>
<td>from (1) and (2) with rule RI</td>
</tr>
<tr>
<td>(4)</td>
<td>(G \subseteq L_u)</td>
<td>basic statement</td>
</tr>
</tbody>
</table>
PDR Unsolvability Certificate

PDR Argument

\[ L_u = L_{u-1} \rightarrow \text{unsolvable} \]

certificate translation:

<table>
<thead>
<tr>
<th>#</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>([A]L_u \subseteq L_u)</td>
<td>basic statement</td>
</tr>
<tr>
<td>(2)</td>
<td>({I} \subseteq \overline{L_u})</td>
<td>basic statement</td>
</tr>
<tr>
<td>(3)</td>
<td>(L_u) is dead</td>
<td>from (1) and (2) with rule RI</td>
</tr>
<tr>
<td>(4)</td>
<td>(G \subseteq L_u)</td>
<td>basic statement</td>
</tr>
<tr>
<td>(5)</td>
<td>(G) is dead</td>
<td>from (3) and (4) with rule SD</td>
</tr>
</tbody>
</table>
Efficient Verification

bottleneck: basic statements ($A \subseteq B$)

$\rightarrow$ depends on representation of $A$ and $B$

efficient for

- BDDs
- (dual-)Horn formulas
- 2CNF
- explicit enumeration

Not efficient for CNF!
Verifying PDR for positive STRIPS

implemented on top of pdrplan

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>certifying</th>
<th>verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDR</td>
<td>388</td>
<td>-4</td>
<td>-2</td>
</tr>
<tr>
<td>FD-$h^\text{M&amp;S}$</td>
<td>224</td>
<td>-27</td>
<td>-19</td>
</tr>
<tr>
<td>FD-$h^{\text{max}}$</td>
<td>203</td>
<td>-47</td>
<td>-14</td>
</tr>
<tr>
<td>DFS-CL</td>
<td>394</td>
<td>-8</td>
<td>-1</td>
</tr>
</tbody>
</table>

small generation overhead, efficient verification
Integration of SAT Certificates

**Observations**
- PDR must have solved related SAT queries already
- SAT solvers are certifying

→ use SAT certificates from planner’s SAT calls*

**Example**
given: state sets $S_\phi$ and $S_\psi$ described by $\phi$ and $\psi$ (in CNF)
→ $S_\phi \subseteq \overline{S_\psi}$ verified with UNSAT certificate for $\phi \land \psi$

*SAT calls don’t perfectly match basic statements
→ combine knowledge within proof system
## Conclusion & Outlook

### Contributions

- certifying version of PDR
- extension of proof system to CNF formalism

### Outlook:

- traditional SAT solvers with modern upper bound techniques
- problem reformulations (e.g. symmetry, STRIPS duality)
- ...