Certified Unsolvability for SAT Planning with Property Directed Reachability
Salomé Eriksson  Malte Helmert  
University of Basel

Certifying Algorithms

Algorithm emits certificate alongside its output, which is verified independently:

<table>
<thead>
<tr>
<th>solvable</th>
<th>unsolvable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT solvers</td>
<td>unsolvable certificate</td>
</tr>
<tr>
<td>PLANNERS</td>
<td>plan</td>
</tr>
</tbody>
</table>

Desired Properties

- sound & complete
- efficient generation (polynomial in planner runtime)
- efficient verification (polynomial in certificate size)
- generality

Unsolvability Certificates for Planning [E et al. 2018]

The certificate incrementally builds a knowledge base of proven statements:
- objects: state sets $S_i$ (represented by propositional logic formulas $\varphi_i$)
- types of statements:
  - $S_i \subseteq S_j$: dead (no state in $S_i$ can be part of a plan)
  - basic statements:
    - state facts about concrete objects
    - need to be verified semantically
- inference rules:
  - derive new knowledge from existing knowledge
  - universally true $\Rightarrow$ only need to be verified syntactically
- $\neg (S_1 \cup S_2)$ is solvable
- $\neg (S_1 \cup S_2)$ is unsolvable

Basic Statements Examples

- $S_1 \cap S_2 \subseteq S_1 
- (S_1 \cup S_2) \subseteq S_1 
- [A \cap (S_1 \cup S_2) \subseteq S_1 | (A \cap S_1) \subseteq S_1 | \text{for some } a \in A] 

Inference Rules Examples

Rules for showing deadness:

- $S_1$ dead, $S_2 \subseteq S_1$ $\Rightarrow S_2$ dead
- $S_1, S_2$ dead, $S_1 \cap G$ dead $\Rightarrow S_1$ dead
- $\neg (S_1 \cup S_2) \subseteq S_1$ $\Rightarrow \not S_1 \subseteq S_2 \cap S_3, S_2 \subseteq S_3$

ED

Rules from Set Theory:

- $S_1 \subseteq S_2, S_2 \subseteq S_3$ $\Rightarrow S_1 \subseteq S_3$
- $S_1 \subseteq S_2, S_3 \subseteq S_4$ $\Rightarrow S_1 \subseteq S_4$
- $\not U \Rrightarrow S_1 \subseteq S_2 \cup S_3$

Property Directed Reachability [Suda 2014]

Property Directed Reachability (PDR) reasons about layers $L_u$ which:

- overapproximate states with distance $< i$ to goal,
- are iteratively refined, and
- are represented as CNF formulas, or dual-Horn formulas for STRIPS tasks.

for $i = 0$ do ... do
while $i \in L_u$ do
  if exists path of length $i$ from $I$ to $G$ then
    return found plan
  else
    strengthen layers where path cannot be extended
end
if $L_u = L_{u-1}$ for some $u < i$ then
  return unsolvable
end

Certificate Structure

PDR’s unsolvability argument:

- we cannot (backwards) reach new states from $L_u$
- $L_u$ contains all goal states
- $L_u$ does not contain the initial state

Efficient Verification

Basic statements need to be verified semantically. If this can be done efficiently depends on the state set representation:

- $S_i \subseteq S_j$ $\iff \varphi_i \models \varphi_j$
- efficient for BDDs, explicit enumeration, (dual-)Horn and 2CNF formulas
- not efficient for CNF formulas

Basic Statements for CNF

Planner calls SAT solver, which is a certifying algorithm.

- Integrate UNSAT certificates into proof

- required UNSAT certificate(s)

<table>
<thead>
<tr>
<th>statement</th>
<th>C1a</th>
<th>C1b</th>
<th>C2a</th>
<th>C2b</th>
<th>C3a</th>
<th>C3b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_i \subseteq S_j$</td>
<td>$\varphi_i \wedge \gamma$ for each $\gamma$ in $\varphi_j$</td>
<td>$\varphi_i \wedge \varphi_j$</td>
<td>$\varphi_i \wedge \gamma$ for each $\gamma$ in $\varphi_j$</td>
<td>$\varphi_i \wedge \gamma$ for each $\varphi_j$</td>
<td>$\varphi_i \wedge \gamma$ for each $\gamma$ in $\varphi_j$</td>
<td>$\varphi_i \wedge \gamma$ for each $\gamma$ in $\varphi_j$</td>
</tr>
</tbody>
</table>

- $\text{required }$ UNSAT certificate(s)

Modified Certificate for PDR with SAT

The SAT calls performed by PDR don’t match the required certificates.

- $\text{modify basic statements and use additional inference rules:}$

- $\text{required }$ UNSAT certificate(s)

<table>
<thead>
<tr>
<th># statement</th>
<th>SAT certificates provided by planner</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) $[A \notin L_u \subseteq \text{states}(\gamma)]$ for all $\gamma$ in $\varphi_L$</td>
<td>from (1a) with rule $SI$</td>
</tr>
<tr>
<td>(1b) $[A \notin L_u \subseteq \text{L_u}]$</td>
<td>build UNSAT certificate by hand* from (1b) and (2) with rule $RI$</td>
</tr>
<tr>
<td>(2) $[\not \emptyset \subseteq \text{L_u}]$</td>
<td>build UNSAT certificates by hand* from (1a) with rule $SI$</td>
</tr>
<tr>
<td>(3) $L_u$ is dead $\Leftrightarrow \emptyset \subseteq \text{L_u}$</td>
<td>from (1a) with rule $SI$ from (4a) with rule $SI$</td>
</tr>
<tr>
<td>(4a) $G \subseteq \text{states}(\gamma)$ for all $\gamma$ in $\varphi_L$</td>
<td>from (3) and (4b) with rule $SD$</td>
</tr>
<tr>
<td>(4b) $G \subseteq \text{L_u}$</td>
<td>$\text{formula can be proven unsolvable solely by unit propagation}$</td>
</tr>
</tbody>
</table>

Experimental Evaluation (PDR without SAT)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>base</th>
<th>certifying</th>
<th>verifying</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDR</td>
<td>305</td>
<td>304</td>
<td>302</td>
</tr>
<tr>
<td>FD-MAS</td>
<td>224</td>
<td>197</td>
<td>178</td>
</tr>
<tr>
<td>FD-MAS</td>
<td>203</td>
<td>156</td>
<td>140</td>
</tr>
<tr>
<td>DFS-CL</td>
<td>394</td>
<td>386</td>
<td>385</td>
</tr>
</tbody>
</table>

*formula can be proven unsolvable solely by unit propagation