Certified Unsolvability for SAT Planning with Property Directed Reachability

Salomé Eriksen, 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>satisfiable assignment</td>
<td>DRAT proof</td>
</tr>
<tr>
<td>SAT solvers</td>
<td>Planners</td>
</tr>
<tr>
<td>plan</td>
<td>unsatisfiability certificate</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: $S_i \cap \neg \varphi_i$ (true facts about concrete objects)
  - need to be verified semantically
  - derivation rules: $\neg \varphi_i$ (need new knowledge from existing knowledge)
  - universally true $\varphi_i$ (need to be verified syntactically)

A task is proven unsolvable if $\{1\}$ or $G$ have been proven to be dead.

Basic Statements Examples

- $B_1 \cap S_1 \subseteq U \cup S_2$
- $B_2 \cap S_1 | A | \subseteq U \cup S_2$
- $S_i | A | \subseteq S_j | A |$ (for some $a \in A$
- $B_3 \cap S_1 | A | \subseteq U \cup S_2$

Derivation Rules Examples

Rules for showing deadness:

- $S_i \cap \neg \varphi_i$ (true facts about concrete objects)
- $S_i \cap \neg \varphi_i$ (true facts about concrete objects)

Rules from Set Theory:

- $S_i \subseteq S_j \subseteq S_k$ (for all $i \leq j \leq k$)
- $S_i \cap S_j \subseteq S_k$ (for all $i \leq j \leq k$)

Property Directed Reachability [Suda 2014]

Property Directed Reachability (PDR) reasons about layers $L_i$ 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 $i = 0, \ldots, N$ do

while $i \in L_i$ do

- If exists path of length $i$ from $I$ to $G$ then
  - return found plan
- else
  - strengthen layers where path cannot be extended
end

end

if $L_u = L_{u-1}$ for some $u < i$ then

- return unsatisfiable

end

Experimental Evaluation (PDR without SAT)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>PDR</th>
<th>base</th>
<th>certifying</th>
<th>verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD-AHLS</td>
<td>394</td>
<td>224</td>
<td>197</td>
<td>178</td>
</tr>
<tr>
<td>FD-AHLS</td>
<td>203</td>
<td>156</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>DFS-CL</td>
<td>394</td>
<td>386</td>
<td>385</td>
<td></td>
</tr>
</tbody>
</table>

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

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

Efficient Verification

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

- $S_i \subseteq S_j \iff \varphi_i \land \varphi_j$
- 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

Modified Certificate for PDR with SAT

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

- modify basic statements and use additional derivation rules:

<table>
<thead>
<tr>
<th># statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) ${1} \subseteq L_u$</td>
<td>SAT certificates provided by planner from (1a) with rule SI</td>
</tr>
<tr>
<td>(1b) ${1} \subseteq L_u$</td>
<td>build UNSAT certificate by hand* from (1b) and (2) with rule RI</td>
</tr>
<tr>
<td>(2) ${1} \subseteq L_u$</td>
<td>build UNSAT certificates by hand* from (4a) with rule SI</td>
</tr>
<tr>
<td>(3) $L_u$ is dead</td>
<td>from (3) and (4b) with rule SD</td>
</tr>
</tbody>
</table>

*formula can be proven unsolvable solely by unit propagation