Planning and Optimization

M. Helmert, G. Röger | University of Basel
P. Ferber, T. Keller, S. Sievers | Fall Semester 2020

Classroom Exercise 3

In this classroom exercise, we denote states as sets of variables that contain exactly the variables that are true in that state. Now consider the following propositional planning task:

$$
\begin{aligned}
V &= \{\textit{A-on-B}, \textit{A-on-C}, \textit{A-on-table},\\
&\quad \textit{B-on-A}, \textit{B-on-C}, \textit{B-on-table},\\
&\quad \textit{C-on-A}, \textit{C-on-B}, \textit{C-on-table}\}\\
I &= \{\textit{A-on-table}, \textit{B-on-A}, \textit{C-on-table}\}\\
O &= \{\textit{move-A-to-table}, \textit{move-A-onto-B}, \textit{move-A-onto-C},\\
&\quad \textit{move-B-to-table}, \textit{move-B-onto-A}, \textit{move-B-onto-C},\\
&\quad \textit{move-C-to-table}, \textit{move-C-onto-A}, \textit{move-C-onto-B}\},\ \text{where}\\
\textit{move-A-to-table} &= \langle \neg \textit{B-on-A} \land \neg \textit{C-on-A}, \neg \textit{A-on-B} \land \neg \textit{A-on-C} \land \textit{A-on-table} \rangle,\\
\textit{move-A-onto-B} &= \langle \neg \textit{B-on-A} \land \neg \textit{C-on-A} \land \neg \textit{C-on-B}, \textit{A-on-B} \land \neg \textit{A-on-C} \land \neg \textit{A-on-table} \rangle,\\
\textit{move-A-onto-C} &= \langle \neg \textit{B-on-A} \land \neg \textit{C-on-A} \land \neg \textit{B-on-C}, \textit{A-on-C} \land \neg \textit{A-on-B} \land \neg \textit{A-on-table} \rangle,\\
\textit{move-B-to-table} &= \langle \neg \textit{A-on-B} \land \neg \textit{C-on-B}, \neg \textit{B-on-A} \land \neg \textit{B-on-C} \land \textit{B-on-table} \rangle,\\
\textit{move-B-onto-A} &= \langle \neg \textit{A-on-B} \land \neg \textit{C-on-B} \land \neg \textit{C-on-A}, \textit{B-on-A} \land \neg \textit{B-on-C} \land \neg \textit{B-on-table} \rangle,\\
\textit{move-B-onto-C} &= \langle \neg \textit{A-on-B} \land \neg \textit{C-on-B} \land \neg \textit{A-on-C}, \textit{B-on-C} \land \neg \textit{B-on-A} \land \neg \textit{B-on-table} \rangle,\\
\textit{move-C-to-table} &= \langle \neg \textit{A-on-C} \land \neg \textit{B-on-C}, \neg \textit{C-on-A} \land \neg \textit{C-on-B} \land \textit{C-on-table} \rangle,\\
\textit{move-C-onto-A} &= \langle \neg \textit{A-on-C} \land \neg \textit{B-on-C} \land \neg \textit{B-on-A}, \textit{C-on-A} \land \neg \textit{C-on-B} \land \neg \textit{C-on-table} \rangle,\\
\textit{move-C-onto-B} &= \langle \neg \textit{A-on-C} \land \neg \textit{B-on-C} \land \neg \textit{A-on-B}, \textit{C-on-B} \land \neg \textit{C-on-A} \land \neg \textit{C-on-table} \rangle,\ \text{and}\\
\gamma &= (\textit{A-on-C} \land \textit{B-on-table} \land \textit{C-on-table})
\end{aligned}
$$

Let furthermore $\phi = \bigwedge_{(v=\mathbf{T}) \in I} v \land \bigwedge_{(v=\mathbf{F}) \in I} \neg v$ be the logical formula that describes the initial state I.

Exercise 1

(a) Provide all successors of I that are created by progression search.

(b) Provide a set of states S that contains all states from which I can be reached by applying *move-B-onto-A*. Exploit that $\{x\textit{-on-}y, x\textit{-on-}z, x\textit{-on-table}\}$ are mutex for $x, y, z \in \{A, B, C\}$ and pairwise distinct x, y, z in your solution.

(c) Let $V' = \{\textit{A-on-B}, \textit{A-on-C}, \textit{A-on-table}, \textit{C-on-A}, \textit{C-on-B}, \textit{C-on-table}\}$ and $e = \textit{eff}(\textit{move-B-onto-A})$. Compute $\textit{regr}(v, e)$ for all $v \in V'$.

(d) Let $V'' = \{\textit{B-on-C}, \textit{B-on-table}\}$ and $e = \textit{eff}(\textit{move-B-onto-A})$. Compute $\textit{regr}(v, e)$ for all $v \in V''$.

(e) Let $e = \textit{eff}(\textit{move-B-onto-A})$. Compute $\textit{regr}(\textit{B-on-A}, e)$.

(f) Let $e = \textit{eff}(\textit{move-B-onto-A})$. Compute $\textit{regr}(\phi, e)$.

(g) Compute $\textit{regr}(\phi, \textit{move-B-onto-A})$.

(h) Compare your solutions of Exercises 1.b and 1.g.

Exercise 2

(a) In SAT Planning, we convert the planning task Π with a horizon T into a logical formula ϕ. Then, we try to find a variable assignment that satisfies ϕ. If there is no such assignment, we increase the horizon and repeat the procedure. Provide an intuitive explanation for the horizon T.

(b) Provide the clauses for the initial state.

(c) Provide the clauses for the goal formula.

(d) For every time step $i \in \{1..T\}$, we have an *operator selection clause* $o_1^t \vee \ldots \vee o_n^t$. Explain the purpose of the operator selection clauses. How do the clauses achieve their purpose? *(Hint: The SAT solver receives as input a formula in CNF. Every model of the CNF formula must also be a model of every individual clause.)*

(e) For every time step $i \in \{1..T\}$, we have a set of *operator exclusion clauses* $\{\neg o_j^i \vee \neg o_k^i \mid 1 \leq j < k \leq |O|\}$. Explain the purpose of the operator exclusion clauses. How do the clauses achieve their purpose?

(f) Provide the precondition clauses for the action *move-A-to-table* at time step i.

(g) Provide the positive effect clause for the action *move-A-to-table* and the variable *A-on-table* at time step i.

(h) Provide the positive effect clause for the action *move-A-to-table* and the variable *A-on-C* at time step i.

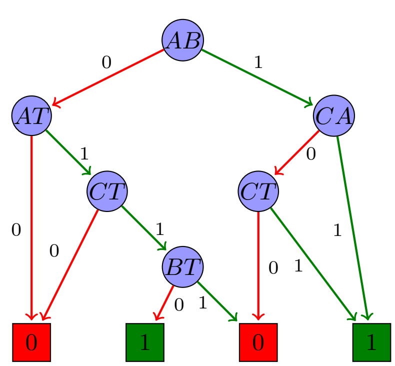
(i) Provide the negative effect clause for the action *move-A-to-table* and the variable *A-on-B* at time step i.

(j) Provide the positive frame clause for the action *move-A-to-table* and the variable *B-on-table* at time step i.

(k) Provide the negative frame clause for the action *move-A-to-table* and the variable *B-on-table* at time step i.

Exercise 3

(a) Provide a logical formula that describes the set of states that is encoded by the following BDD (each variable name *x-on-y* $\in V$ is abbreviated as xy). Give two states that are reachable from I that are in that set.



(b) Provide the BDD that encodes the set of states that is described by the logical formula *A-on-table* $\wedge$ ((*C-on-B* $\wedge$ *B-on-A*) $\vee$ *C-on-table*).

(c) Compare your solutions of Exercises 3.b and 1.a. How do they relate to planning as symbolic search?