Planning and Optimization

B3. Operator Examples and Planning Tasks

Malte Helmert and Gabriele Röger

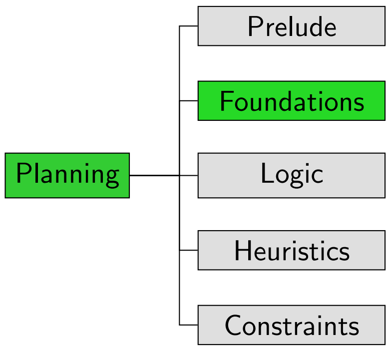
Universität Basel

September 28, 2022

Planning and Optimization

September 28, 2022 — B3. Operator Examples and Planning Tasks

Content of this Course



B3.1 Operator Examples

Applying Operators: Example

We use the state variables $V = \{a, b, c, d\}$.

Example

Consider the operator $o = \langle a, (\neg a \wedge (\neg c \rhd \neg b))\rangle$ and the state $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$.

The operator o is applicable in s because $s \models a$.

Effect conditions of $\mathit{eff}(o)$:

$$\begin{aligned} \mathit{effcond}(a, \mathit{eff}(o)) &= \mathit{effcond}(a, (\neg a \wedge (\neg c \rhd \neg b))) \\ &= (\mathit{effcond}(a, \neg a) \vee \mathit{effcond}(a, (\neg c \rhd \neg b))) \\ &= (\bot \vee (\neg c \wedge \mathit{effcond}(a, \neg b))) \\ &= (\bot \vee (\neg c \wedge \bot)) \\ &\equiv \bot \quad \rightsquigarrow \text{ false in state } s \end{aligned}$$

Applying Operators: Example

We use the state variables $V = \{a, b, c, d\}$.

Example

Consider the operator $o = \langle a, (\neg a \wedge (\neg c \rhd \neg b))\rangle$ and the state $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$.

The operator o is applicable in s because $s \models a$.

Effect conditions of $\mathit{eff}(o)$:

$$\begin{aligned} \mathit{effcond}(\neg a, \mathit{eff}(o)) &= \mathit{effcond}(\neg a, (\neg a \wedge (\neg c \rhd \neg b))) \\ &= (\mathit{effcond}(\neg a, \neg a) \vee \mathit{effcond}(\neg a, (\neg c \rhd \neg b))) \\ &= (\top \vee \mathit{effcond}(\neg a, (\neg c \rhd \neg b))) \\ &\equiv \top \quad \rightsquigarrow \text{ true in state } s \end{aligned}$$

Applying Operators: Example

We use the state variables $V = \{a, b, c, d\}$.

Example

Consider the operator $o = \langle a, (\neg a \wedge (\neg c \rhd \neg b))\rangle$ and the state $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$.

The operator o is applicable in s because $s \models a$.

Effect conditions of $\mathit{eff}(o)$:

$$\begin{aligned} \mathit{effcond}(b, \mathit{eff}(o)) &= \mathit{effcond}(b, (\neg a \wedge (\neg c \rhd \neg b))) \\ &= (\mathit{effcond}(b, \neg a) \vee \mathit{effcond}(b, (\neg c \rhd \neg b))) \\ &= (\bot \vee (\neg c \wedge \mathit{effcond}(b, \neg b))) \\ &= (\bot \vee (\neg c \wedge \bot)) \\ &\equiv \bot \quad \rightsquigarrow \text{ false in state } s \end{aligned}$$

Applying Operators: Example

We use the state variables $V = \{a, b, c, d\}$.

Example

Consider the operator $o = \langle a, (\neg a \wedge (\neg c \rhd \neg b))\rangle$ and the state $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$.

The operator o is applicable in s because $s \models a$.

Effect conditions of $\mathit{eff}(o)$:

$$\begin{aligned} \mathit{effcond}(\neg b, \mathit{eff}(o)) &= \mathit{effcond}(\neg b, (\neg a \wedge (\neg c \rhd \neg b))) \\ &= (\mathit{effcond}(\neg b, \neg a) \vee \mathit{effcond}(\neg b, (\neg c \rhd \neg b))) \\ &= (\bot \vee (\neg c \wedge \mathit{effcond}(\neg b, \neg b))) \\ &= (\bot \vee (\neg c \wedge \top)) \\ &\equiv \neg c \quad \rightsquigarrow \text{ false in state } s \end{aligned}$$

Applying Operators: Example

We use the state variables $V = \{a, b, c, d\}$.

Example

Consider the operator $o = \langle a, (\neg a \land (\neg c \rhd \neg b))\rangle$ and the state $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$.

The operator o is applicable in s because $s \models a$.

Effect conditions of $\mathit{eff}(o)$:

$$\begin{aligned} \mathit{effcond}(c, \mathit{eff}(o)) &\equiv \bot && \leadsto \text{ false in state } s \\ \mathit{effcond}(\neg c, \mathit{eff}(o)) &\equiv \bot && \leadsto \text{ false in state } s \\ \mathit{effcond}(d, \mathit{eff}(o)) &\equiv \bot && \leadsto \text{ false in state } s \\ \mathit{effcond}(\neg d, \mathit{eff}(o)) &\equiv \bot && \leadsto \text{ false in state } s \end{aligned}$$

The resulting state of applying o in s is the state $\{a \mapsto \mathbf{F}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$.

Example Operators: Blocks World

Example (Blocks World Operators)

To model blocks world operators conveniently, we use auxiliary state variables *A-clear*, *B-clear*, and *C-clear* to express that there is nothing on top of a given block.

Then blocks world operators can be modeled as:

- $\langle \mathit{A\text{-}clear} \land \mathit{A\text{-}on\text{-}table} \land \mathit{B\text{-}clear},\ \mathit{A\text{-}on\text{-}B} \land \neg \mathit{A\text{-}on\text{-}table} \land \neg \mathit{B\text{-}clear}\rangle$
- $\langle \mathit{A\text{-}clear} \land \mathit{A\text{-}on\text{-}table} \land \mathit{C\text{-}clear},\ \mathit{A\text{-}on\text{-}C} \land \neg \mathit{A\text{-}on\text{-}table} \land \neg \mathit{C\text{-}clear}\rangle$
- $\langle \mathit{A\text{-}clear} \land \mathit{A\text{-}on\text{-}B},\ \mathit{A\text{-}on\text{-}table} \land \neg \mathit{A\text{-}on\text{-}B} \land \mathit{B\text{-}clear}\rangle$
- $\langle \mathit{A\text{-}clear} \land \mathit{A\text{-}on\text{-}C},\ \mathit{A\text{-}on\text{-}table} \land \neg \mathit{A\text{-}on\text{-}C} \land \mathit{C\text{-}clear}\rangle$
- $\langle \mathit{A\text{-}clear} \land \mathit{A\text{-}on\text{-}B} \land \mathit{C\text{-}clear},\ \mathit{A\text{-}on\text{-}C} \land \neg \mathit{A\text{-}on\text{-}B} \land \mathit{B\text{-}clear} \land \neg \mathit{C\text{-}clear}\rangle$
- $\langle \mathit{A\text{-}clear} \land \mathit{A\text{-}on\text{-}C} \land \mathit{B\text{-}clear},\ \mathit{A\text{-}on\text{-}B} \land \neg \mathit{A\text{-}on\text{-}C} \land \mathit{C\text{-}clear} \land \neg \mathit{B\text{-}clear}\rangle$
- ...

Example Operator: 4-Bit Counter

Example (Incrementing a 4-Bit Counter)

Operator to increment a 4-bit number $b_3 b_2 b_1 b_0$ represented by 4 state variables $b_0, \ldots, b_3$:

precondition:

$$\neg b_0 \lor \neg b_1 \lor \neg b_2 \lor \neg b_3$$

effect:

$$\begin{gathered} (\neg b_0 \rhd b_0) \land \\ ((\neg b_1 \land b_0) \rhd (b_1 \land \neg b_0)) \land \\ ((\neg b_2 \land b_1 \land b_0) \rhd (b_2 \land \neg b_1 \land \neg b_0)) \land \\ ((\neg b_3 \land b_2 \land b_1 \land b_0) \rhd (b_3 \land \neg b_2 \land \neg b_1 \land \neg b_0)) \end{gathered}$$

B3.2 Planning Tasks

Planning Tasks

Definition (Planning Task)

A (propositional) planning task is a 4-tuple $\Pi = \langle V, I, O, \gamma \rangle$ where

- V is a finite set of propositional state variables,
- I is a valuation over V called the initial state,
- O is a finite set of operators over V, and
- γ is a formula over V called the goal.

Mapping Planning Tasks to Transition Systems

Definition (Transition System Induced by a Planning Task)

The planning task $\Pi = \langle V, I, O, \gamma \rangle$ induces the transition system $\mathcal{T}(\Pi) = \langle S, L, c, T, s_0, S_\star \rangle$, where

- S is the set of all states over V,
- L is the set of operators O,
- $c(o) = cost(o)$ for all operators $o \in O$,
- $T = \{\langle s, o, s' \rangle \mid s \in S,\ o \text{ applicable in } s,\ s' = s[\![o]\!]\}$,
- $s_0 = I$, and
- $S_\star = \{s \in S \mid s \models \gamma\}$.

Planning Tasks: Terminology

- Terminology for transitions systems is also applied to the planning tasks Π that induce them.
- For example, when we speak of the states of Π, we mean the states of $\mathcal{T}(\Pi)$.
- A sequence of operators that forms a solution of $\mathcal{T}(\Pi)$ is called a plan of Π.

Satisficing and Optimal Planning

By planning, we mean the following two algorithmic problems:

Definition (Satisficing Planning)

Given: a planning task Π

Output: a plan for Π, or **unsolvable** if no plan for Π exists

Definition (Optimal Planning)

Given: a planning task Π

Output: a plan for Π with minimal cost among all plans for Π, or **unsolvable** if no plan for Π exists

B3.3 Summary

Summary

- Planning tasks compactly represent transition systems and are suitable as inputs for planning algorithms.
- A planning task consists of a set of state variables and an initial state, operators and goal over these state variables.
- In satisficing planning, we must find a solution for a planning task (or show that no solution exists).
- In optimal planning, we must additionally guarantee that generated solutions are of minimal cost.