Automated Planning

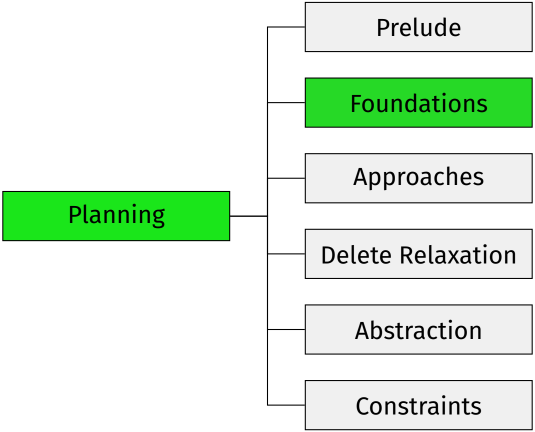
B3. Formal Definition of Planning

Jendrik Seipp

Linköping University

based on slides from the AI group at the University of Basel

Content of this Course



Semantics of Effects and Operators

Semantics of Effects: Effect Conditions

Definition (Effect Condition for an Effect)

Let ℓ be an atomic effect, and let e be an effect.

The effect condition $\mathit{effcond}(\ell, e)$ under which ℓ triggers given the effect e is a propositional formula defined as follows:

- $\mathit{effcond}(\ell, \top) = \bot$
- $\mathit{effcond}(\ell, e) = \top$ for the atomic effect $e = \ell$
- $\mathit{effcond}(\ell, e) = \bot$ for all atomic effects $e = \ell' \neq \ell$
- $\mathit{effcond}(\ell, (e \wedge e')) = (\mathit{effcond}(\ell, e) \vee \mathit{effcond}(\ell, e'))$
- $\mathit{effcond}(\ell, (\chi \rhd e)) = (\chi \wedge \mathit{effcond}(\ell, e))$

Intuition: $\mathit{effcond}(\ell, e)$ represents the condition that must be true in the current state for the effect e to lead to the atomic effect ℓ

Effect Condition: Example (1)

Example

Consider the move operator m_1 from the running example:
$\mathit{eff}(m_1) = ((t_1 \rhd \neg t_1) \wedge (\neg t_1 \rhd t_1))$.

Under which conditions does it set t_1 to false?

$$
\begin{aligned}
\mathit{effcond}(\neg t_1, \mathit{eff}(m_1)) &= \mathit{effcond}(\neg t_1, ((t_1 \rhd \neg t_1) \wedge (\neg t_1 \rhd t_1))) \\
&= \mathit{effcond}(\neg t_1, (t_1 \rhd \neg t_1)) \vee \\
&\quad\; \mathit{effcond}(\neg t_1, (\neg t_1 \rhd t_1)) \\
&= (t_1 \wedge \mathit{effcond}(\neg t_1, \neg t_1)) \vee \\
&\quad\; (\neg t_1 \wedge \mathit{effcond}(\neg t_1, t_1)) \\
&= (t_1 \wedge \top) \vee (\neg t_1 \wedge \bot) \\
&\equiv t_1 \vee \bot \\
&\equiv t_1
\end{aligned}
$$

Effect Condition: Example (2)

Example

Consider the move operator m_1 from the running example:
$\mathit{eff}(m_1) = ((t_1 \rhd \neg t_1) \land (\neg t_1 \rhd t_1))$.

Under which conditions does it set i to true?

$$
\begin{aligned}
\mathit{effcond}(i, \mathit{eff}(m_1)) &= \mathit{effcond}(i, ((t_1 \rhd \neg t_1) \land (\neg t_1 \rhd t_1))) \\
&= \mathit{effcond}(i, (t_1 \rhd \neg t_1)) \lor \\
&\ \mathit{effcond}(i, (\neg t_1 \rhd t_1)) \\
&= (t_1 \land \mathit{effcond}(i, \neg t_1)) \lor \\
&\ (\neg t_1 \land \mathit{effcond}(i, t_1)) \\
&= (t_1 \land \bot) \lor (\neg t_1 \land \bot) \\
&\equiv \bot \lor \bot \\
&\equiv \bot
\end{aligned}
$$

Semantics of Effects: Applying an Effect

first attempt:

Definition (Applying Effects)

Let V be a set of propositional state variables.
Let s be a state over V, and let e be an effect over V.

The resulting state of applying e in s, written $s[\![e]\!]$,
is the state s' defined as follows for all $v \in V$:

$$s'(v) = \begin{cases} \mathbf{T} & \text{if } s \models \mathit{effcond}(v, e) \\ \mathbf{F} & \text{if } s \models \mathit{effcond}(\neg v, e) \\ s(v) & \text{otherwise} \end{cases}$$

What is the problem with this definition?

Semantics of Effects: Applying an Effect

correct definition:

Definition (Applying Effects)

Let V be a set of propositional state variables.
Let s be a state over V, and let e be an effect over V.

The resulting state of applying e in s, written $s[\![e]\!]$,
is the state s' defined as follows for all $v \in V$:

$$s'(v) = \begin{cases} \mathbf{T} & \text{if } s \models \mathit{effcond}(v, e) \\ \mathbf{F} & \text{if } s \models \mathit{effcond}(\neg v, e) \land \neg \mathit{effcond}(v, e) \\ s(v) & \text{otherwise} \end{cases}$$

Add-after-Delete Semantics

Note:

- The definition implies that if a variable is simultaneously "added" (set to **T**) and "deleted" (set to **F**), the value **T** takes precedence.
- This is called add-after-delete semantics.
- This detail of effect semantics is somewhat arbitrary, but has proven useful in applications.

Semantics of Operators

Definition (Applicable, Applying Operators, Resulting State)

Let V be a set of propositional state variables.
Let s be a state over V, and let o be an operator over V.

Operator o is applicable in s if $s \models pre(o)$.

If o is applicable in s, the resulting state of applying o in s, written $s[\![o]\!]$, is the state $s[\![eff(o)]\!]$.

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 an interpretation of V called the initial state,
- O is a finite set of operators over V, and
- γ is a formula over V called the goal.

Running Example: Planning Task

Example

From the previous chapter, we see that the running example can be represented by the task $\Pi = \langle V, I, O, \gamma \rangle$ with

- $V = \{i, w, t_1, t_2\}$
- $I = \{i \mapsto \mathbf{F}, w \mapsto \mathbf{T}, t_1 \mapsto \mathbf{F}, t_2 \mapsto \mathbf{F}\}$
- $O = \{m_1, m_2, l_1, l_2, u\}$ where
  - $m_1 = \langle \top, ((t_1 \rhd \neg t_1) \wedge (\neg t_1 \rhd t_1)), 5 \rangle$
  - $m_2 = \langle \top, ((t_2 \rhd \neg t_2) \wedge (\neg t_2 \rhd t_2)), 5 \rangle$
  - $l_1 = \langle \neg i \wedge (w \leftrightarrow t_1), (i \wedge w), 1 \rangle$
  - $l_2 = \langle \neg i \wedge (w \leftrightarrow t_2), (i \wedge \neg w), 1 \rangle$
  - $u = \langle i, \neg i \wedge (w \rhd ((t_1 \rhd w) \wedge (\neg t_1 \rhd \neg w))) \wedge (\neg w \rhd ((t_2 \rhd w) \wedge (\neg t_2 \rhd \neg w))), 1 \rangle$
- $\gamma = \neg i \wedge \neg w$

Exercise: Modeling a Propositional Planning Task

Example

Model the following task as a propositional planning task:

You are currently at home and have to write an essay. Since your computer is broken, you can only write the essay using a computer at the university library. These computers are always switched off when a user logs off.

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

Positive Normal Form

Flat Effect

Definition (Flat Effect)

An effect is simple if it is either an atomic effect
or of the form $(\chi \rhd e)$, where e is an atomic effect.

An effect e is flat if it is a conjunction of 0 or more simple effects,
and none of these simple effects include the same atomic effect.

An operator o is flat if $\mathit{eff}(o)$ is flat.

Notes: analogously to CNF, we consider

- a single simple effect as a conjunction of 1 simple effect
- the empty effect as a conjunction of 0 simple effects

Flat Effect: Example

Example

Consider the effect

$$c \wedge (a \rhd (\neg b \wedge (c \rhd (b \wedge \neg d \wedge \neg a)))) \wedge (\neg b \rhd \neg a)$$

An equivalent flat (and conflict-free) effect is

$$\begin{aligned} & c \wedge \\ & ((a \wedge \neg c) \rhd \neg b) \wedge \\ & ((a \wedge c) \rhd b) \wedge \\ & ((a \wedge c) \rhd \neg d) \wedge \\ & ((\neg b \vee (a \wedge c)) \rhd \neg a) \end{aligned}$$

Note: if we want, we can write c as $(\top \rhd c)$ to make the structure even more uniform, with each simple effect having a condition.

Positive Formulas, Operators and Tasks

Definition (Positive Formula)

A logical formula φ is positive if no negation symbols appear in φ.

Note: This includes the negation symbols implied by $\rightarrow$ and $\leftrightarrow$.

Definition (Positive Operator)

An operator o is positive if $pre(o)$ and all effect conditions in $\mathit{eff}(o)$ are positive.

Definition (Positive Propositional Planning Task)

A propositional planning task $\langle V, I, O, \gamma \rangle$ is positive if all operators in O and the goal γ are positive.

Positive Normal Form

Definition (Positive Normal Form)

A propositional planning task is in positive normal form
if it is positive and all operator effects are flat.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$
\begin{aligned}
V &= \{\mathit{home}, \mathit{uni}, \mathit{lecture}, \mathit{bike}, \mathit{bike\text{-}locked}\} \\
I &= \{\mathit{home} \mapsto \mathbf{T}, \mathit{bike} \mapsto \mathbf{T}, \mathit{bike\text{-}locked} \mapsto \mathbf{T}, \\
&\qquad \mathit{uni} \mapsto \mathbf{F}, \mathit{lecture} \mapsto \mathbf{F}\} \\
O &= \{\langle \mathit{home} \land \mathit{bike} \land \neg \mathit{bike\text{-}locked}, \neg \mathit{home} \land \mathit{uni}\rangle, \\
&\qquad \langle \mathit{bike} \land \mathit{bike\text{-}locked}, \neg \mathit{bike\text{-}locked}\rangle, \\
&\qquad \langle \mathit{bike} \land \neg \mathit{bike\text{-}locked}, \mathit{bike\text{-}locked}\rangle, \\
&\qquad \langle \mathit{uni}, \mathit{lecture} \land ((\mathit{bike} \land \neg \mathit{bike\text{-}locked}) \rhd \neg \mathit{bike})\rangle\} \\
\gamma &= \mathit{lecture} \land \mathit{bike}
\end{aligned}
$$

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$
\begin{aligned}
V &= \{home, uni, lecture, bike, bike\text{-}locked\} \\
I &= \{home \mapsto \mathbf{T}, bike \mapsto \mathbf{T}, bike\text{-}locked \mapsto \mathbf{T}, \\
&\qquad uni \mapsto \mathbf{F}, lecture \mapsto \mathbf{F}\} \\
O &= \{\langle home \land bike \land \neg bike\text{-}locked, \neg home \land uni\rangle, \\
&\qquad \langle bike \land bike\text{-}locked, \neg bike\text{-}locked\rangle, \\
&\qquad \langle bike \land \neg bike\text{-}locked, bike\text{-}locked\rangle, \\
&\qquad \langle uni, lecture \land ((bike \land \neg bike\text{-}locked) \rhd \neg bike)\rangle\} \\
\gamma &= lecture \land bike
\end{aligned}
$$

Identify state variable v occurring negatively in conditions.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$V = \{home, uni, lecture, bike, bike\text{-}locked, bike\text{-}unlocked\}$$

$$I = \{home \mapsto \mathbf{T}, bike \mapsto \mathbf{T}, bike\text{-}locked \mapsto \mathbf{T}, \quad uni \mapsto \mathbf{F}, lecture \mapsto \mathbf{F}, bike\text{-}unlocked \mapsto \mathbf{F}\}$$

$$O = \{\langle home \land bike \land \neg bike\text{-}locked, \neg home \land uni\rangle, \quad \langle bike \land bike\text{-}locked, \neg bike\text{-}locked\rangle, \quad \langle bike \land \neg bike\text{-}locked, bike\text{-}locked\rangle, \quad \langle uni, lecture \land ((bike \land \neg bike\text{-}locked) \rhd \neg bike)\rangle\}$$

$$\gamma = lecture \land bike$$

Introduce new variable $\hat{v}$ with complementary initial value.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$V = \{home, uni, lecture, bike, bike\text{-}locked, bike\text{-}unlocked\}$$

$$I = \{home \mapsto \mathbf{T}, bike \mapsto \mathbf{T}, bike\text{-}locked \mapsto \mathbf{T},$$
$$uni \mapsto \mathbf{F}, lecture \mapsto \mathbf{F}, bike\text{-}unlocked \mapsto \mathbf{F}\}$$

$$O = \{\langle home \land bike \land \neg bike\text{-}locked, \neg home \land uni\rangle,$$
$$\langle bike \land bike\text{-}locked, \neg bike\text{-}locked\rangle,$$
$$\langle bike \land \neg bike\text{-}locked, bike\text{-}locked\rangle,$$
$$\langle uni, lecture \land ((bike \land \neg bike\text{-}locked) \rhd \neg bike)\rangle\}$$

$$\gamma = lecture \land bike$$

Identify effects on variable v.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$V = \{\mathit{home}, \mathit{uni}, \mathit{lecture}, \mathit{bike}, \mathit{bike\text{-}locked}, \mathit{bike\text{-}unlocked}\}$$

$$I = \{\mathit{home} \mapsto \mathbf{T}, \mathit{bike} \mapsto \mathbf{T}, \mathit{bike\text{-}locked} \mapsto \mathbf{T}, \mathit{uni} \mapsto \mathbf{F}, \mathit{lecture} \mapsto \mathbf{F}, \mathit{bike\text{-}unlocked} \mapsto \mathbf{F}\}$$

$$\begin{aligned} O = \{&\langle \mathit{home} \land \mathit{bike} \land \neg\mathit{bike\text{-}locked}, \neg\mathit{home} \land \mathit{uni}\rangle, \\ &\langle \mathit{bike} \land \mathit{bike\text{-}locked}, \neg\mathit{bike\text{-}locked} \land \mathit{bike\text{-}unlocked}\rangle, \\ &\langle \mathit{bike} \land \neg\mathit{bike\text{-}locked}, \mathit{bike\text{-}locked} \land \neg\mathit{bike\text{-}unlocked}\rangle, \\ &\langle \mathit{uni}, \mathit{lecture} \land ((\mathit{bike} \land \neg\mathit{bike\text{-}locked}) \rhd \neg\mathit{bike})\rangle\} \end{aligned}$$

$$\gamma = \mathit{lecture} \land \mathit{bike}$$

Introduce complementary effects for $\hat{v}$.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$V = \{home, uni, lecture, bike, bike\text{-}locked, bike\text{-}unlocked\}$$

$$I = \{home \mapsto \mathbf{T}, bike \mapsto \mathbf{T}, bike\text{-}locked \mapsto \mathbf{T}, uni \mapsto \mathbf{F}, lecture \mapsto \mathbf{F}, bike\text{-}unlocked \mapsto \mathbf{F}\}$$

$$O = \{\langle home \land bike \land \neg bike\text{-}locked, \neg home \land uni\rangle, \langle bike \land bike\text{-}locked, \neg bike\text{-}locked \land bike\text{-}unlocked\rangle, \langle bike \land \neg bike\text{-}locked, bike\text{-}locked \land \neg bike\text{-}unlocked\rangle, \langle uni, lecture \land ((bike \land \neg bike\text{-}locked) \rhd \neg bike)\rangle\}$$

$$\gamma = lecture \land bike$$

Identify negative conditions for v.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$V = \{home, uni, lecture, bike, \textit{bike-locked}, \textit{bike-unlocked}\}$$

$$I = \{home \mapsto \mathbf{T}, bike \mapsto \mathbf{T}, \textit{bike-locked} \mapsto \mathbf{T},$$
$$uni \mapsto \mathbf{F}, lecture \mapsto \mathbf{F}, \textit{bike-unlocked} \mapsto \mathbf{F}\}$$

$$O = \{\langle home \land bike \land \textit{bike-unlocked}, \neg home \land uni\rangle,$$
$$\langle bike \land \textit{bike-locked}, \neg \textit{bike-locked} \land \textit{bike-unlocked}\rangle,$$
$$\langle bike \land \textit{bike-unlocked}, \textit{bike-locked} \land \neg \textit{bike-unlocked}\rangle,$$
$$\langle uni, lecture \land ((bike \land \textit{bike-unlocked}) \rhd \neg bike)\rangle\}$$

$$\gamma = lecture \land bike$$

Replace by positive condition $\hat{v}$.

Positive Normal Form: Example

Example (Transformation to Positive Normal Form)

$$
\begin{aligned}
V &= \{\mathit{home}, \mathit{uni}, \mathit{lecture}, \mathit{bike}, \mathit{bike\text{-}locked}, \mathit{bike\text{-}unlocked}\} \\
I &= \{\mathit{home} \mapsto \mathbf{T}, \mathit{bike} \mapsto \mathbf{T}, \mathit{bike\text{-}locked} \mapsto \mathbf{T}, \\
&\qquad \mathit{uni} \mapsto \mathbf{F}, \mathit{lecture} \mapsto \mathbf{F}, \mathit{bike\text{-}unlocked} \mapsto \mathbf{F}\} \\
O &= \{\langle \mathit{home} \land \mathit{bike} \land \mathit{bike\text{-}unlocked}, \neg \mathit{home} \land \mathit{uni}\rangle, \\
&\qquad \langle \mathit{bike} \land \mathit{bike\text{-}locked}, \neg \mathit{bike\text{-}locked} \land \mathit{bike\text{-}unlocked}\rangle, \\
&\qquad \langle \mathit{bike} \land \mathit{bike\text{-}unlocked}, \mathit{bike\text{-}locked} \land \neg \mathit{bike\text{-}unlocked}\rangle, \\
&\qquad \langle \mathit{uni}, \mathit{lecture} \land ((\mathit{bike} \land \mathit{bike\text{-}unlocked}) \rhd \neg \mathit{bike})\rangle\} \\
\gamma &= \mathit{lecture} \land \mathit{bike}
\end{aligned}
$$

STRIPS

STRIPS Operators and Planning Tasks

Definition (STRIPS Operator)

An operator o of a propositional planning task is a STRIPS operator if

- $pre(o)$ is a conjunction of state variables, and
- $\mathit{eff}(o)$ is a conflict-free conjunction of atomic effects.

Definition (STRIPS Planning Task)

A propositional planning task $\langle V, I, O, \gamma \rangle$ is a STRIPS planning task if all operators $o \in O$ are STRIPS operators and γ is a conjunction of state variables.

STRIPS Operators: Remarks

- Every STRIPS operator is of the form

$$\langle v_1 \land \cdots \land v_n, \ \ell_1 \land \cdots \land \ell_m \rangle$$

  where v_i are state variables and ℓ_j are atomic effects.
- Often, STRIPS operators o are described via three sets of state variables:
  - the preconditions (state variables occurring in $pre(o)$)
  - the add effects (state variables occurring positively in $\mathit{eff}(o)$)
  - the delete effects (state variables occurring negatively in $\mathit{eff}(o)$)
- Definitions of STRIPS in the literature often do not require conflict-freeness. But it is easy to achieve and makes many things simpler.
- There exists a variant called STRIPS with negation where negative literals are also allowed in conditions.

Why STRIPS is Interesting

- STRIPS is particularly simple, yet expressive enough to capture general planning tasks.
- In particular, STRIPS planning is no easier than planning in general.
- Many algorithms in the planning literature are only presented for STRIPS planning tasks (generalization is often, but not always, obvious).

STRIPS

STanford Research Institute Problem Solver (Fikes & Nilsson, 1971)

Summary

Summary (1/2)

- 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.

Summary (2/2)

- A positive task with flat operators is in positive normal form.
- STRIPS is even more restrictive than positive normal form, forbidding complex preconditions and conditional effects.
- Both forms are expressive enough to capture general propositional planning tasks.
- Transformation to positive normal form is possible with polynomial size increase.
- Isomorphic transformations of propositional planning tasks to STRIPS can increase the number of operators exponentially; non-isomorphic polynomial transformations exist.