Planning and Optimization

B2. Introduction to Planning Tasks

Malte Helmert and Gabriele Röger

Universität Basel

September 25, 2024

Planning and Optimization

September 25, 2024 — B2. Introduction to Planning Tasks

B2.1 Introduction

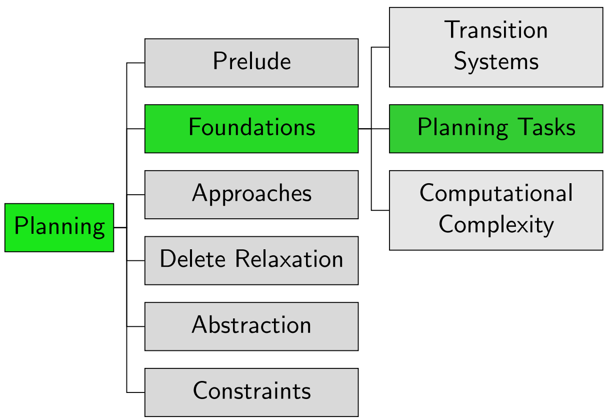
B2.2 State Variables

B2.3 State Formulas

B2.4 Operators and Effects

B2.5 Summary

Content of the Course



B2.1 Introduction

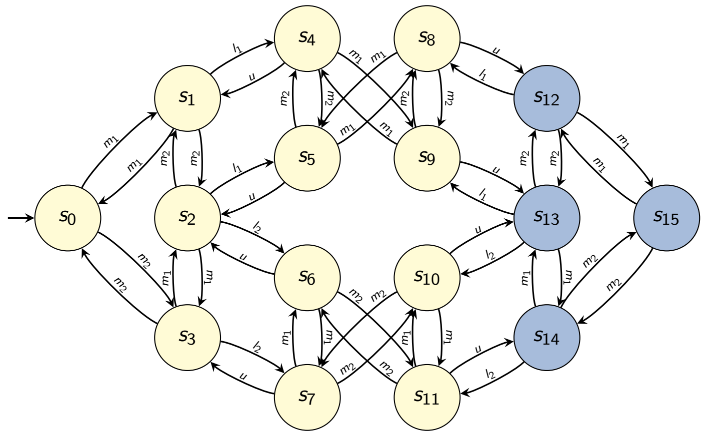
The State Explosion Problem

- We saw in blocks world:
  n blocks $\rightsquigarrow$ number of states exponential in n
- same is true everywhere we look
- known as the state explosion problem

To represent transitions systems compactly,
need to tame these exponentially growing aspects:

- states
- goal states
- transitions

Running Example: Transition System



$c(m_1) = 5,\ c(m_2) = 5,\ c(l_1) = 1,\ c(l_2) = 1,\ c(u) = 1$

B2.2 State Variables

Compact Descriptions of Transition Systems

How to specify huge transition systems
without enumerating the states?

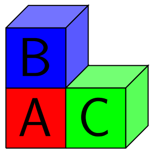
- represent different aspects of the world
  in terms of different (propositional) state variables
- individual state variables are atomic propositions
  $\rightsquigarrow$ a state is an interpretation of state variables
- n state variables induce 2^n states
  $\rightsquigarrow$ exponentially more compact than "flat" representations

Example: n^2 variables suffice for blocks world with n blocks

Blocks World State with Propositional Variables

Example

$s(A\text{-}on\text{-}B) = \mathbf{F}$
$s(A\text{-}on\text{-}C) = \mathbf{F}$
$s(A\text{-}on\text{-}table) = \mathbf{T}$
$s(B\text{-}on\text{-}A) = \mathbf{T}$
$s(B\text{-}on\text{-}C) = \mathbf{F}$
$s(B\text{-}on\text{-}table) = \mathbf{F}$
$s(C\text{-}on\text{-}A) = \mathbf{F}$
$s(C\text{-}on\text{-}B) = \mathbf{F}$
$s(C\text{-}on\text{-}table) = \mathbf{T}$



$\rightsquigarrow$ 9 variables for 3 blocks

Propositional State Variables

Definition (Propositional State Variable)

A propositional state variable is a symbol X.

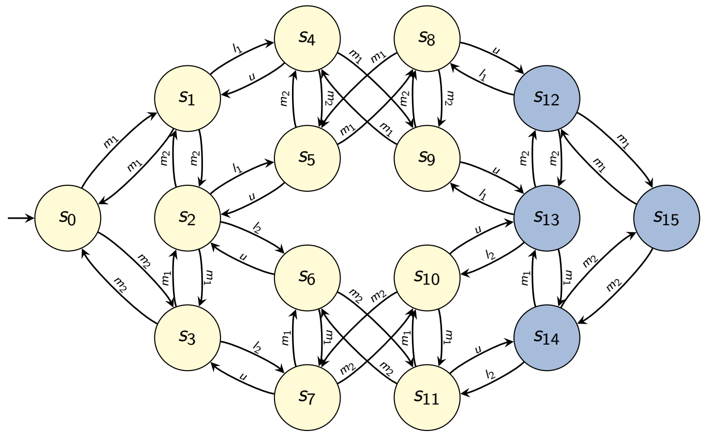
Let V be a finite set of propositional state variables.

A state s over V is an interpretation of V, i.e., a truth assignment $s : V \to \{\mathbf{T}, \mathbf{F}\}$.

Running Example: Compact State Descriptions

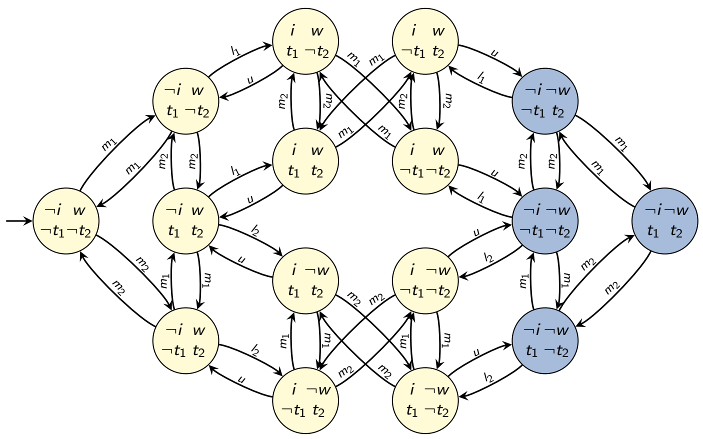
- In the running example, we describe 16 states with 4 propositional state variables ($2^4 = 16$).

Running Example: Opaque States



Running Example: Using State Variables

state variables $V = \{i, w, t_1, t_2\}$



states shown by true literals

example: $\{i \mapsto \mathbf{T}, w \mapsto \mathbf{F}, t_1 \mapsto \mathbf{T}, t_2 \mapsto \mathbf{F}\} \rightsquigarrow i\ \neg w\ t_1\ \neg t_2$

Running Example: Intuition

Intuition: delivery task with 2 trucks, 1 package, locations L and R

transition labels:

- m_1/m_2: move first/second truck
- l_1/l_2: load package into first/second truck
- u: unload package from a truck

state variables:

- t_1 true if first truck is at location L (else at R)
- t_2 true if second truck is at location L (else at R)
- i true if package is inside a truck
- w encodes where exactly the package is:
  - if i is true, w true if package in first truck
  - if i is false, w true if package at location L

B2.3 State Formulas

Representing Sets of States

How do we compactly represent sets of states, for example the set of goal states?

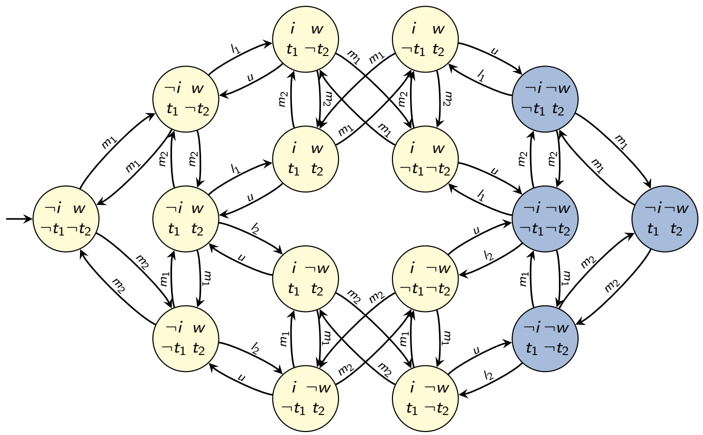
Idea: formula φ over the state variables represents the models of φ.

Definition (State Formula)

Let V be a finite set of propositional state variables.

A formula over V is a propositional logic formula using V as the set of atomic propositions.

Running Example: Representing Goal States



goal formula $\gamma = \neg i \wedge \neg w$ represents goal states $S_\star$

B2.4 Operators and Effects

Operators Representing Transitions

How do we compactly represent transitions?

- most complex aspect of a planning task
- central concept: operators

Idea: one operator o for each transition label ℓ, describing

- in which states s a transition $s \xrightarrow{\ell} s'$ exists (precondition)
- how state s' differs from state s (effect)
- what the cost of ℓ is

Syntax of Operators

Definition (Operator)

An operator o over state variables V is an object with three properties:

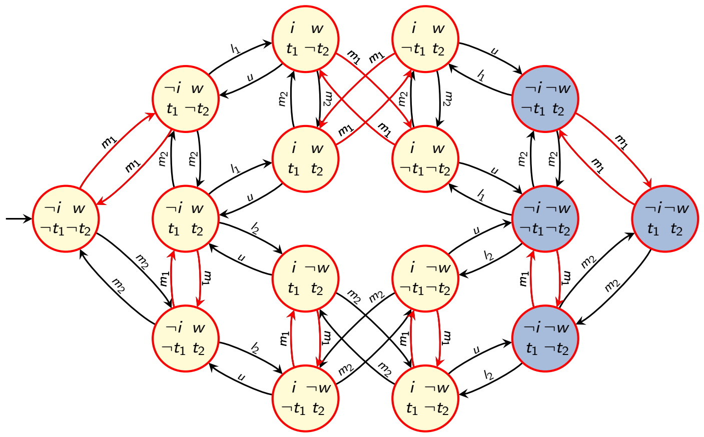
- a precondition $pre(o)$, a formula over V
- an effect $eff(o)$ over V, defined later in this chapter
- a cost $cost(o) \in \mathbb{R}_0^+$

Notes:

- Operators are also called actions.
- Operators are often written as triples $\langle pre(o), eff(o), cost(o)\rangle$.
- This can be abbreviated to pairs $\langle pre(o), eff(o)\rangle$ when the cost of the operator is irrelevant.

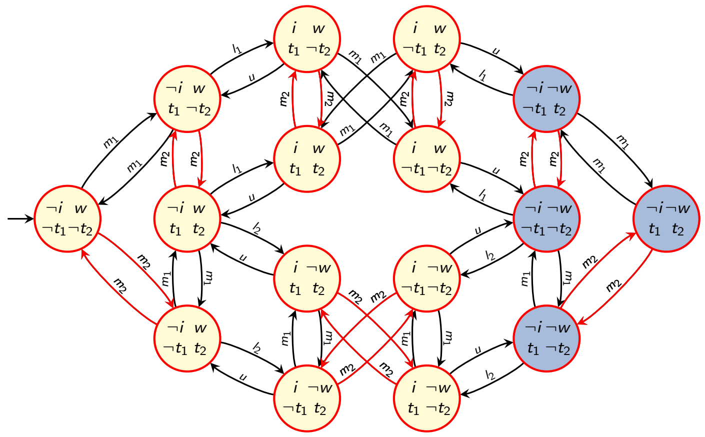
Running Example: Operator Preconditions

$pre(m_1) = \top$



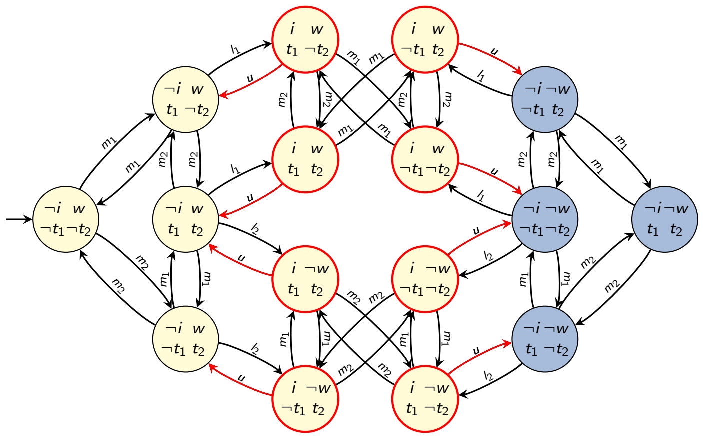
Running Example: Operator Preconditions

$pre(m_2) = \top$



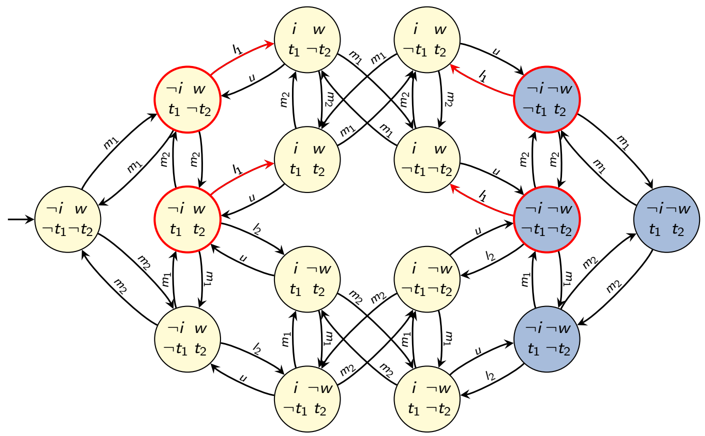
Running Example: Operator Preconditions

$pre(u) = i$



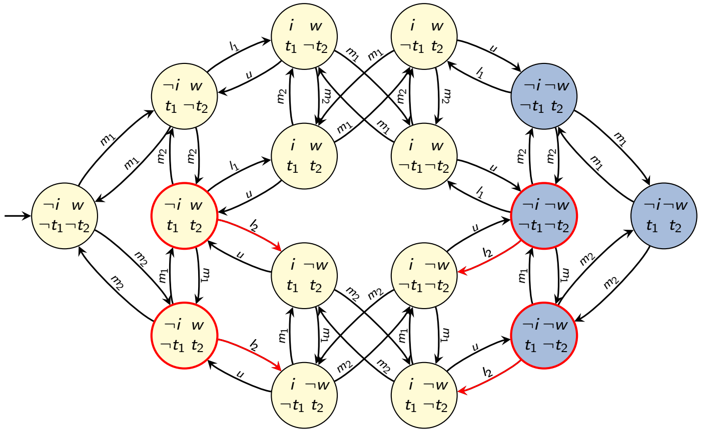
Running Example: Operator Preconditions

$pre(l_1) = \neg i \land (w \leftrightarrow t_1)$



Running Example: Operator Preconditions

$pre(l_2) = \neg i \wedge (w \leftrightarrow t_2)$



Syntax of Effects

Definition (Effect)

Effects over propositional state variables V are inductively defined as follows:

- $\top$ is an effect (empty effect).
- If $v \in V$ is a propositional state variable, then v and $\neg v$ are effects (atomic effect).
- If e and e' are effects, then $(e \wedge e')$ is an effect (conjunctive effect).
- If χ is a formula over V and e is an effect, then $(\chi \rhd e)$ is an effect (conditional effect).

We may omit parentheses when this does not cause ambiguity.

Example: we will later see that $((e \wedge e') \wedge e'')$ behaves identically to $(e \wedge (e' \wedge e''))$ and will write this as $e \wedge e' \wedge e''$.

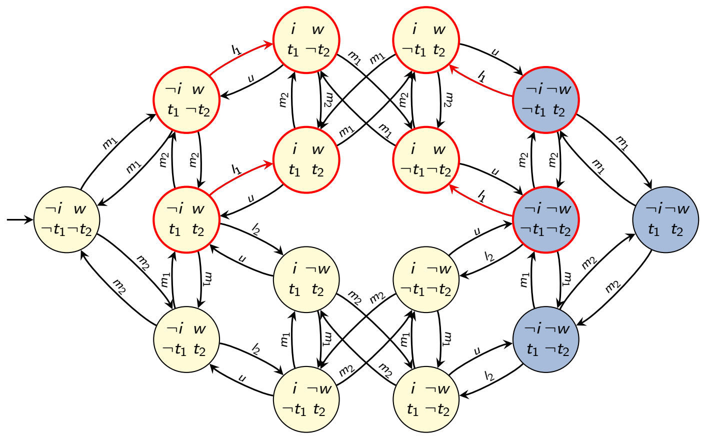
Effects: Intuition

Intuition for effects:

- The empty effect $\top$ changes nothing.
- Atomic effects can be understood as assignments that update the value of a state variable.
  - v means "$v :=$ **T**"
  - $\neg v$ means "$v :=$ **F**"
- A conjunctive effect $e = (e' \wedge e'')$ means that both subeffects e and e' take place simultaneously.
- A conditional effect $e = (\chi \rhd e')$ means that subeffect e' takes place iff χ is true in the state where e takes place.

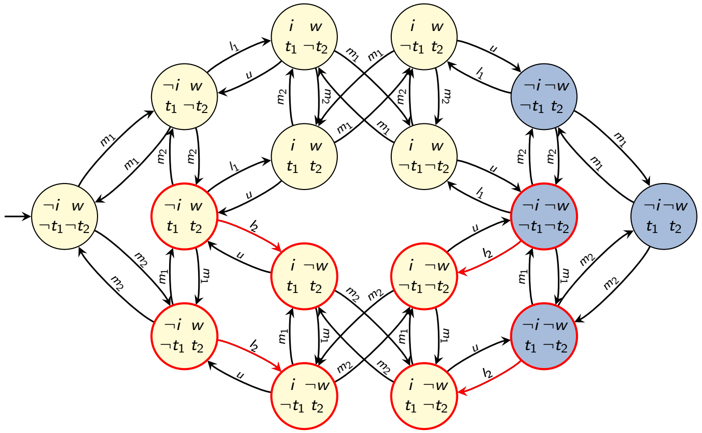
Running Example: Operator Effects

$eff(l_1) = (i \wedge w)$



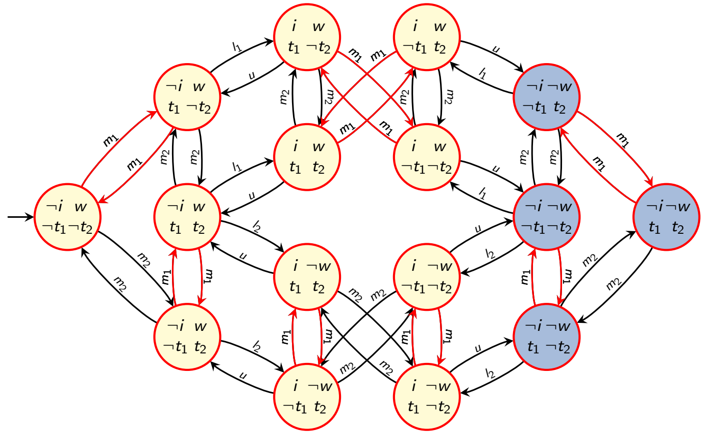
Running Example: Operator Effects

$$\mathit{eff}(l_2) = (i \land \neg w)$$



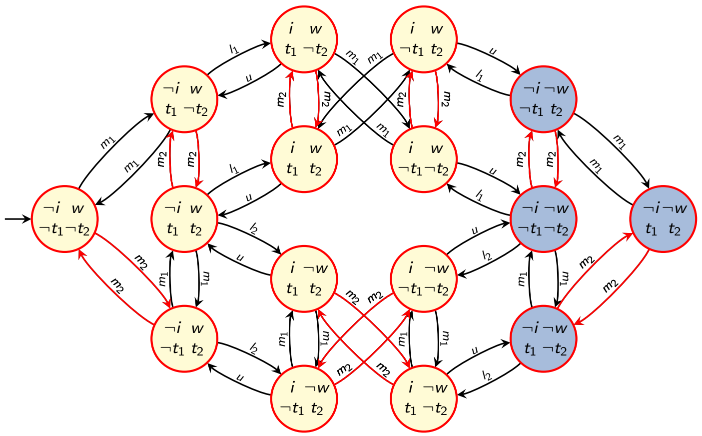
Running Example: Operator Effects

$$\mathit{eff}(m_1) = ((t_1 \rhd \neg t_1) \land (\neg t_1 \rhd t_1))$$



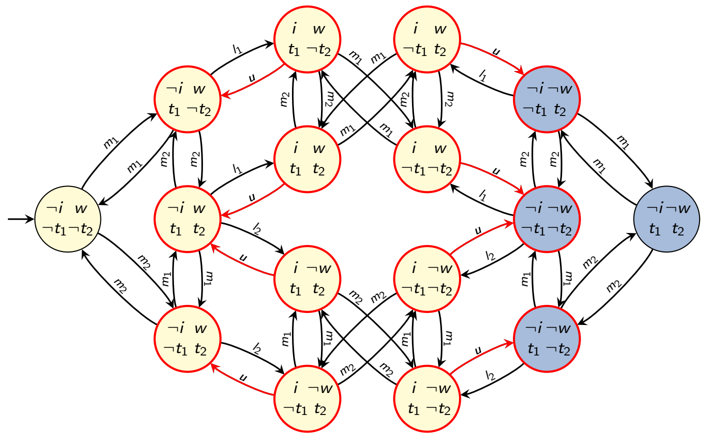
Running Example: Operator Effects

$$\mathit{eff}(m_2) = ((t_2 \rhd \neg t_2) \land (\neg t_2 \rhd t_2))$$



Running Example: Operator Effects

$$\begin{aligned}\mathit{eff}(u) = \neg i &\land (w \rhd ((t_1 \rhd w) \land (\neg t_1 \rhd \neg w))) \\ &\land (\neg w \rhd ((t_2 \rhd w) \land (\neg t_2 \rhd \neg w)))\end{aligned}$$



B2.5 Summary

Summary

- Propositional state variables let us compactly describe properties of large transition systems.
- A state is an assignment to a set of state variables.
- Sets of states are represented as formulas over state variables.
- Operators describe when (precondition), how (effect) and at which cost the state of the world can be changed.
- Effects are structured objects including empty, atomic, conjunctive and conditional effects.