

Foundations of Artificial Intelligence

F5. Automated Planning: Abstraction

Malte Helmert

University of Basel

May 4, 2026

Foundations of Artificial Intelligence

May 4, 2026 — F5. Automated Planning: Abstraction

F5.1 SAS⁺

F5.2 Abstractions

F5.3 Summary

Automated Planning: Overview

Chapter overview: automated planning

- ▶ F1. Introduction
- ▶ F2. Planning Formalisms
- ▶ F3. Delete Relaxation
- ▶ F4. Delete Relaxation Heuristics
- ▶ F5. Abstraction
- ▶ F6. Abstraction Heuristics

Planning Heuristics

We consider **two basic ideas** for general heuristics:

- ▶ Delete Relaxation
- ▶ **Abstraction** \rightsquigarrow this chapter

Abstraction: Idea

Estimate solution costs by considering a **smaller** planning task.

F5.1 SAS⁺

SAS⁺ Encoding

- ▶ in this chapter: SAS⁺ encoding instead of STRIPS (see Chapter F2)
- ▶ difference: state variables v not binary, but with **finite domain** $\text{dom}(v)$
- ▶ accordingly, preconditions, effects, goals specified as **partial assignments**
- ▶ everything else equal to STRIPS

(In practice, planning systems convert automatically between STRIPS and SAS⁺.)

SAS⁺ Planning Task

Definition (SAS⁺ planning task)

A SAS⁺ planning task is a 5-tuple $\Pi = \langle V, \text{dom}, I, G, A \rangle$ with the following components:

- ▶ V : finite set of **state variables**
- ▶ dom : **domain**; $\text{dom}(v)$ finite and non-empty for all $v \in V$
 - ▶ states: **total assignments** for V according to dom
- ▶ I : the **initial state** (state = total assignment)
- ▶ G : **goals** (partial assignment)
- ▶ A : finite set of **actions** a with
 - ▶ $\text{pre}(a)$: its **preconditions** (partial assignment)
 - ▶ $\text{eff}(a)$: its **effects** (partial assignment)
 - ▶ $\text{cost}(a) \in \mathbb{N}_0$: its **cost**

German: SAS⁺-Planungsaufgabe

State Space of SAS⁺ Planning Task

Definition (state space induced by SAS⁺ planning task)

Let $\Pi = \langle V, \text{dom}, I, G, A \rangle$ be a SAS⁺ planning task.

Then Π **induces** the **state space** $\mathcal{S}(\Pi) = \langle S, A, \text{cost}, T, s_1, S_G \rangle$:

- ▶ **set of states**: total assignments of V according to dom
- ▶ **actions**: actions A defined as in Π
- ▶ **action costs**: cost as defined in Π
- ▶ **transitions**: $s \xrightarrow{a} s'$ for states s, s' and action a iff
 - ▶ $\text{pre}(a)$ agrees with s (precondition satisfied)
 - ▶ s' agrees with $\text{eff}(a)$ for all variables mentioned in eff ; agrees with s for all other variables (effects are applied)
- ▶ **initial state**: $s_1 = I$
- ▶ **goal states**: $s \in S_G$ for state s iff G agrees with s

German: durch SAS⁺-Planungsaufgabe induzierter Zustandsraum

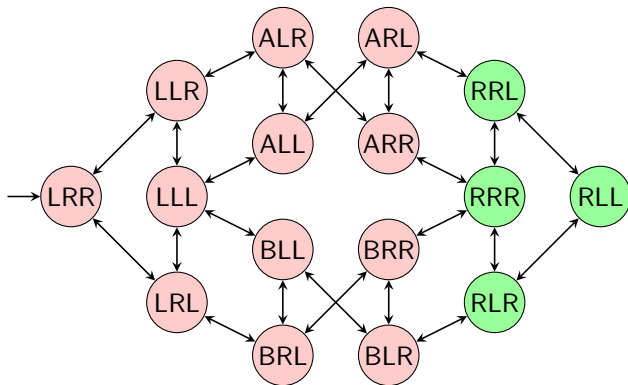
Example: Logistics Task with One Package, Two Trucks

Example (one package, two trucks)

Consider the SAS⁺ planning task $\langle V, \text{dom}, I, G, A \rangle$ with:

- ▶ $V = \{p, t_A, t_B\}$
- ▶ $\text{dom}(p) = \{L, R, A, B\}$ and $\text{dom}(t_A) = \text{dom}(t_B) = \{L, R\}$
- ▶ $I = \{p \mapsto L, t_A \mapsto R, t_B \mapsto R\}$
- ▶ $G = \{p \mapsto R\}$
- ▶ $A = \{load_{i,j} \mid i \in \{A, B\}, j \in \{L, R\}\}$
 $\cup \{unload_{i,j} \mid i \in \{A, B\}, j \in \{L, R\}\}$
 $\cup \{move_{i,j,j'} \mid i \in \{A, B\}, j, j' \in \{L, R\}, j \neq j'\}$ with:
 - ▶ $load_{i,j}$ has preconditions $\{t_i \mapsto j, p \mapsto j\}$, effects $\{p \mapsto i\}$
 - ▶ $unload_{i,j}$ has preconditions $\{t_i \mapsto j, p \mapsto i\}$, effects $\{p \mapsto j\}$
 - ▶ $move_{i,j,j'}$ has preconditions $\{t_i \mapsto j\}$, effects $\{t_i \mapsto j'\}$
 - ▶ All actions have cost 1.

State Space for Example Task



- ▶ state $\{p \mapsto i, t_A \mapsto j, t_B \mapsto k\}$ denoted as ijk
- ▶ annotations of edges not shown for simplicity
- ▶ for example, edge from LLL to ALL has annotation $load_{A,L}$

F5.2 Abstractions

State Space Abstraction

State space abstractions **drop distinctions between certain states**, but preserve the **state space behavior** as well as possible.

- ▶ An abstraction of a state space \mathcal{S} is defined by an **abstraction function** α that determines which states can be distinguished in the abstraction.
- ▶ Based on \mathcal{S} and α , we compute the **abstract state space** \mathcal{S}^α which is “similar” to \mathcal{S} but smaller.
- ▶ main idea: use optimal solution cost in \mathcal{S}^α as heuristic

German: Abstraktionsfunktion, abstrakter Zustandsraum

Induced Abstraction

Definition (induced abstraction)

Let $\mathcal{S} = \langle S, A, cost, T, s_1, S_G \rangle$ be a state space, and let $\alpha : S \rightarrow S'$ be a surjective function.

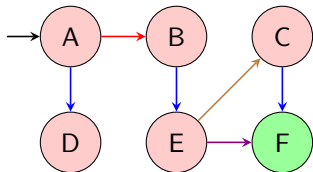
The **abstraction of \mathcal{S} induced by α** , denoted as \mathcal{S}^α , is the state space $\mathcal{S}^\alpha = \langle S', A, cost, T', s'_1, S'_G \rangle$ with:

- ▶ $T' = \{ \langle \alpha(s), a, \alpha(t) \rangle \mid \langle s, a, t \rangle \in T \}$
- ▶ $s'_1 = \alpha(s_1)$
- ▶ $S'_G = \{ \alpha(s) \mid s \in S_G \}$

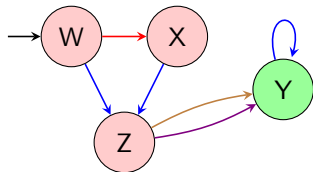
German: induzierte Abstraktion

Abstraction: Example

concrete state space with states $S = \{A, B, C, D, E, F\}$



abstract state space with states $S^\alpha = \{W, X, Y, Z\}$

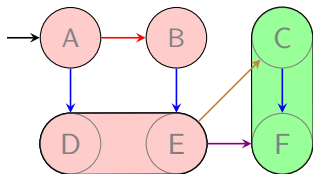


abstraction function $\alpha : S \rightarrow S^\alpha$

$$\alpha(A) = W \quad \alpha(B) = X \quad \alpha(C) = Y$$

$$\alpha(D) = Z \quad \alpha(E) = Z \quad \alpha(F) = Y$$

intuition: grouping states



F5.3 Summary

Summary

- ▶ basic idea of **abstractions**: simplify state space by considering a **smaller** version
- ▶ formally: **abstraction function** α maps states to **abstract states** and thus defines which states can be distinguished by the resulting abstraction
- ▶ induces **abstract state space**