



Foundations of Arti April 30, 2025 — F2. Autom	ficial Intelligence ated Planning: Planning Formalisms		
F2.1 Four Forma	lisms		
F2.2 STRIPS			
F2.3 ADL, SAS ⁺	and PDDL		
F2.4 Summary			
M. Helmert (University of Basel)	Foundations of Artificial Intelligence	April 30, 2025	2 / 20



April 30, 2025

4 / 20



Four Planning Formalisms

 A description language for state spaces (planning tasks) is called a planning formalism.

Four Formalisms

April 30, 2025

5 / 20

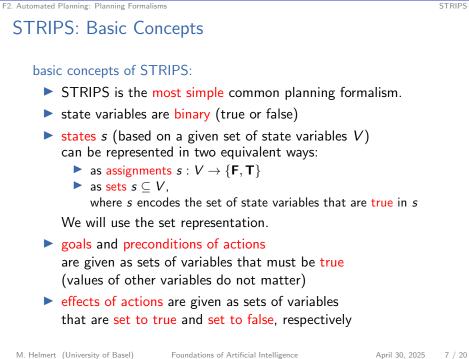
• We introduce four planning formalisms:

STRIPS (Stanford Research Institute Problem Solver)

Foundations of Artificial Intelligence

- ADL (Action Description Language)
- **3** SAS⁺ (Simplified Action Structures)
- PDDL (Planning Domain Definition Language)
- STRIPS and SAS⁺ are the most simple formalisms; in the next chapters, we only consider these.

M. Helmert (University of Basel)

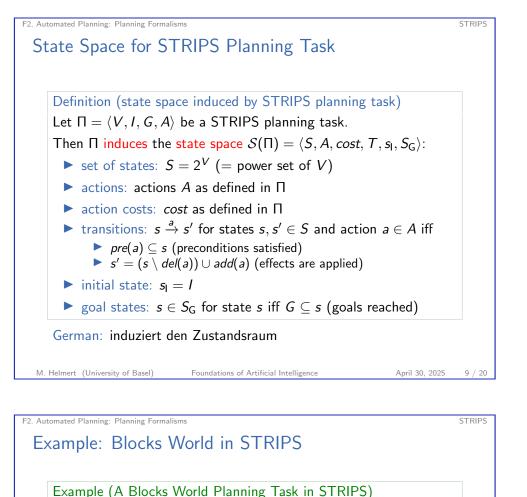


F2.2 STRIPS

F2. Automated Planning: Planning Formalisms STRIPS Planning Task Definition (STRIPS Planning Task) A STRIPS planning task is a 4 tuple $\Pi = \langle V, I, G, A \rangle$ with V: finite set of state variables \blacktriangleright $I \subset V$: the initial state \blacktriangleright *G* \subset *V*: the set of goals ► *A*: finite set of actions. where for all actions $a \in A$, the following is defined: \blacktriangleright pre(a) \subset V: the preconditions of a • $add(a) \subset V$: the add effects of a • $del(a) \subset V$: the delete effects of a ▶ $cost(a) \in \mathbb{N}_0$: the costs of a German: STRIPS-Planungsaufgabe, Zustandsvariablen, Anfangszustand, Ziele, Aktionen, Add-/Delete-Effekte, Kosten remark: action costs are an extension of "traditional" STRIPS

M. Helmert (University of Basel)

F2. Automated Planning: Planning Formalisms



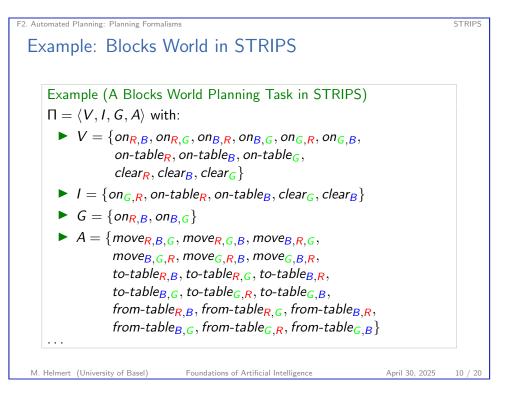
move actions encode moving a block from one block to another

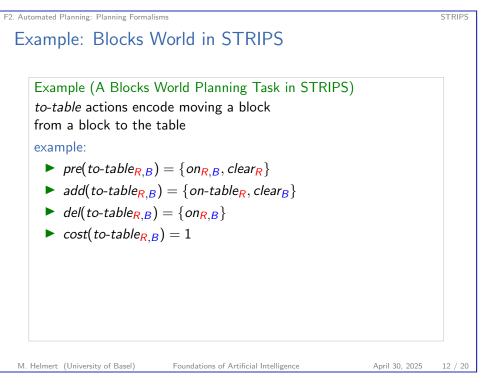
example:

▶ $pre(move_{R,B,G}) = \{on_{R,B}, clear_{R}, clear_{G}\}$

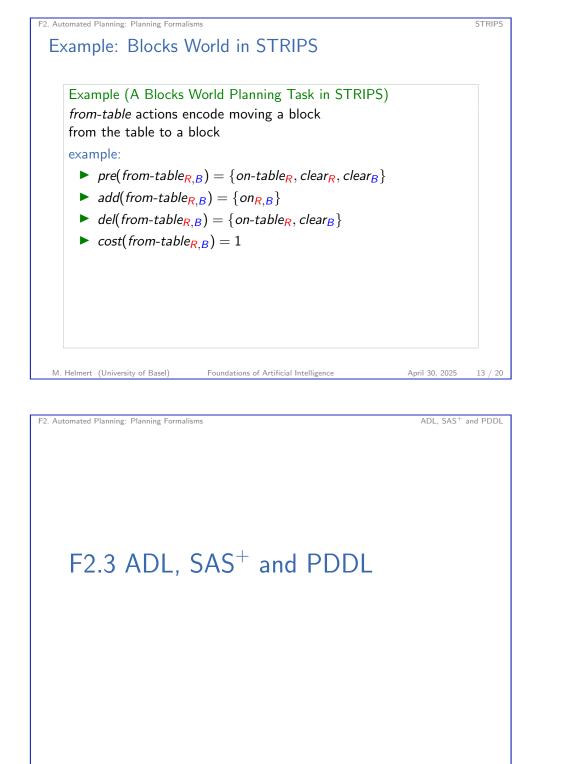
Foundations of Artificial Intelligence

- $add(move_{R,B,G}) = \{on_{R,G}, clear_B\}$
- $del(move_{R,B,G}) = \{on_{R,B}, clear_G\}$
- $cost(move_{R,B,G}) = 1$





11 / 20



F2. Automated Planning: Planning Formal	isms		STRIPS
Why STRIPS?			
STRIPS is part	icularly simple.		
\rightsquigarrow simplifies the d	esign and implementation		
of planning alg	orithms		
often cumberso	ome for the user		
to model tasks	directly in STRIPS		
but: STRIPS is	equally "powerful"		
	complex planning formalisms		
	npilers" exist that translate m	ore complex	
	e ADL and SAS ⁺) to STRIPS		
X	,		
M. Helmert (University of Basel)	Foundations of Artificial Intelligence	April 30, 2025	14 / 20
F2. Automated Planning: Planning Formal	isms	ADL. SAS ⁺	and PDDI

Basic Concepts of ADL

basic concepts of ADL:

- Like STRIPS, ADL uses propositional variables (true/false) as state variables.
- preconditions of actions and goal are arbitrary logic formulas (action applicable/goal reached in states that satisfy the formula)
- in addition to STRIPS effects, there are conditional effects: variable v is only set to true/false if a given logical formula is true in the current state

F2. Automated Planning: Planning Formalisms

ADL, SAS⁺ and PDDL

17 / 20

Summar

Basic Concepts of SAS⁺

basic concepts of SAS⁺:

- very similar to STRIPS: state variables not necessarily binary, but with given finite domain (cf. CSPs)
- states are assignments to these variables (cf. CSPs)
- ▶ preconditions and goals given as partial assignments example: {v₁ → a, v₃ → b} as preconditions (or goals)
 - If s(v₁) = a and s(v₃) = b, then the action is applicable in s (or goal is reached)
 - values of other variables do not matter
- effects are assignments to subset of variables
 - example: effect $\{v_1 \mapsto b, v_2 \mapsto c\}$ means
 - ▶ In the successor state s', $s'(v_1) = b$ and $s'(v_2) = c$.
 - All other variables retain their values.

M. Helmert (University of Basel)

F2. Automated Planning: Planning Formalisms

Foundations of Artificial Intelligence April 30, 2025

F2.4 Summary

Basic Concept of PDDL

- PDDL is the standard language used in practice to describe planning tasks.
- descriptions in (restricted) predicate logic instead of propositional logic (~> even more compact)
- other features like numeric variables and derived variables (axioms) for defining complex logical conditions (formulas that are automatically evaluated in every state and can, e.g., be used in preconditions)
- There exist defined PDDL fragments for STRIPS and ADL; many planners only support the STRIPS fragment. example: blocks world in PDDL

Foundations of Artificial Intelligence

M. Helmert (University of Basel)

April 30, 2025

18 / 20

```
F2. Automated Planning: Planning Formalisms
                                                                             Summar
Summary
    planning formalisms:
     STRIPS: particularly simple, easy to handle for algorithms
           binary state variables
           preconditions, add and delete effects, goals:
              sets of variables
      ADL: extension of STRIPS
           logic formulas for complex preconditions and goals
           conditional effects
      SAS<sup>+</sup>: extension of STRIPS
           state variables with arbitrary finite domains
     ▶ PDDL: input language used in practice
           based on predicate logic
              (more compact than propositional logic)
           only partly supported by most algorithms
              (e.g., STRIPS or ADL fragment)
M. Helmert (University of Basel)
                             Foundations of Artificial Intelligence
                                                                  April 30, 2025
                                                                              20 / 20
```