Basics of AI and Machine Learning

The STRIPS Planning Formalism

David Speck

Linköping University

Slides modified from Basel AI group, with permission

Automated Planning: Overview

Chapter overview: automated planning

- Introduction
- The STRIPS Planning Formalism
- Other Planning Formalisms
- Planning Heuristics
- Alternatives to Heuristic Search

Four Formalisms

Four Planning Formalisms

- A description language for state spaces (planning tasks) is called a planning formalism.
- We introduce four planning formalisms:
  1. STRIPS (Stanford Research Institute Problem Solver)
  2. ADL (Action Description Language)
  3. SAS^+ (Simplified Action Structures)
  4. PDDL (Planning Domain Definition Language)

STRIPS

STRIPS: Basic Concepts

basic concepts of STRIPS:

- STRIPS is the most simple common planning formalism.
- state variables are binary (true or false)

STRIPS: Basic Concepts

basic concepts of STRIPS:

- STRIPS is the most simple common planning formalism.
- state variables are binary (true or false)
- states s (based on a given set of state variables V) can be represented in two equivalent ways:
  - as assignments $s : V \to \{\mathbf{F}, \mathbf{T}\}$
  - as sets $s \subseteq V$, where s encodes the set of state variables that are true in s

We will use the set representation.

STRIPS: Basic Concepts

basic concepts of STRIPS:

- STRIPS is the most simple common planning formalism.
- state variables are binary (true or false)
- states s (based on a given set of state variables V) can be represented in two equivalent ways:
  - as assignments $s : V \to \{\mathbf{F}, \mathbf{T}\}$
  - as sets $s \subseteq V$, where s encodes the set of state variables that are true in s

  We will use the set representation.
- goals and preconditions of actions are given as sets of variables that must be true (values of other variables do not matter)
- effects of actions are given as sets of variables that are set to true and set to false, respectively

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
- $I \subseteq V$: the initial state
- $G \subseteq V$: the set of goals
- A: finite set of actions, where for all actions $a \in A$, the following is defined:
  - $pre(a) \subseteq V$: the preconditions of a
  - $add(a) \subseteq V$: the add effects of a
  - $del(a) \subseteq V$: the delete effects of a
  - $cost(a) \in \mathbb{N}_0$: the costs of a

remark: action costs are an extension of "traditional" STRIPS

State Space for STRIPS Planning Task

Definition (state space induced by STRIPS planning task)

Let $\Pi = \langle V, I, G, A \rangle$ be a STRIPS planning task.

Then Π induces the state space $\mathcal{S}(\Pi) = \langle S, A, cost, T, s_0, S_\star \rangle$:

- set of states: $S = 2^V$ (= power set of V)
- actions: actions A as defined in Π
- action costs: $cost$ as defined in Π
- transitions: $s \xrightarrow{a} s'$ for states s, s' and action a iff
  - $pre(a) \subseteq s$ (preconditions satisfied)
  - $s' = (s \setminus del(a)) \cup add(a)$ (effects are applied)
- initial state: $s_0 = I$
- goal states: $s \in S_\star$ for state s iff $G \subseteq s$ (goals reached)

Example: Blocks World in STRIPS

Example (A Blocks World Planning Task in STRIPS)

$\Pi = \langle V, I, G, A \rangle$ with:

- $V = \{on_{R,B}, on_{R,G}, on_{B,R}, on_{B,G}, on_{G,R}, on_{G,B},$
  $on\text{-}table_R, on\text{-}table_B, on\text{-}table_G,$
  $clear_R, clear_B, clear_G\}$
- $I = \{on_{G,R}, on\text{-}table_R, on\text{-}table_B, clear_G, clear_B\}$
- $G = \{on_{R,B}, on_{B,G}\}$
- $A = \{move_{R,B,G}, move_{R,G,B}, move_{B,R,G},$
  $move_{B,G,R}, move_{G,R,B}, move_{G,B,R},$
  $to\text{-}table_{R,B}, to\text{-}table_{R,G}, to\text{-}table_{B,R},$
  $to\text{-}table_{B,G}, to\text{-}table_{G,R}, to\text{-}table_{G,B},$
  $from\text{-}table_{R,B}, from\text{-}table_{R,G}, from\text{-}table_{B,R},$
  $from\text{-}table_{B,G}, from\text{-}table_{G,R}, from\text{-}table_{G,B}\}$

...

Example: Blocks World in STRIPS

Example (A Blocks World Planning Task in STRIPS)

move actions encode moving a block
from one block to another

example:

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

Example: Blocks World in STRIPS

Example (A Blocks World Planning Task in STRIPS)

to-table actions encode moving a block
from a block to the table

example:

- $pre(\mathit{to\text{-}table}_{R,B}) = \{on_{R,B}, clear_R\}$
- $add(\mathit{to\text{-}table}_{R,B}) = \{\mathit{on\text{-}table}_R, clear_B\}$
- $del(\mathit{to\text{-}table}_{R,B}) = \{on_{R,B}\}$
- $cost(\mathit{to\text{-}table}_{R,B}) = 1$

Example: Blocks World in STRIPS

Example (A Blocks World Planning Task in STRIPS)

from-table actions encode moving a block
from the table to a block

example:

- $\mathit{pre}(\mathit{from\text{-}table}_{R,B}) = \{\mathit{on\text{-}table}_R, \mathit{clear}_R, \mathit{clear}_B\}$
- $\mathit{add}(\mathit{from\text{-}table}_{R,B}) = \{\mathit{on}_{R,B}\}$
- $\mathit{del}(\mathit{from\text{-}table}_{R,B}) = \{\mathit{on\text{-}table}_R, \mathit{clear}_B\}$
- $\mathit{cost}(\mathit{from\text{-}table}_{R,B}) = 1$

Why STRIPS?

- STRIPS is particularly simple.

$\rightsquigarrow$ simplifies the design and implementation of planning algorithms

- often cumbersome for the "user" to model tasks directly in STRIPS
- but: STRIPS is equally "powerful" to much more complex planning formalisms

$\rightsquigarrow$ automatic "compilers" exist that translate more complex formalisms (like ADL and SAS^+) to STRIPS

Summary

Summary

- STRIPS planning formalism: Stanford Research Institute Problem Solver
  - particularly simple, easy to handle for algorithms
  - binary state variables
  - preconditions, add and delete effects, goals: sets of variables
- ADL: Action Description Language
- SAS^+: Simplified Action Structures
- PDDL: Planning Domain Definition Language