Basics of AI and Machine Learning

Automated Planning: Introduction

David Speck

Linköping University

Classification

classification:

Automated Planning

environment:

- static vs. dynamic
- deterministic vs. non-deterministic vs. stochastic
- fully vs. partially vs. not observable
- discrete vs. continuous
- single-agent vs. multi-agent

problem solving method:

- problem-specific vs. general vs. learning

Introduction

Automated Planning

What is Automated Planning?

"Planning is the art and practice of thinking before acting."

— P. Haslum

$\rightsquigarrow$ finding plans (sequences of actions) that lead from an initial state to a goal state

Here: classical planning

- general approach to finding solutions for state-space search problems
- classical = static, deterministic, fully observable
- variants: probabilistic planning, planning under partial observability, online planning, ...

Planning: Informally

given:

- state space description in terms of suitable problem description language (planning formalism)

required:

- a plan, i.e., a solution for the described state space (sequence of actions from initial state to goal)
- or a proof that no plan exists

distinguish between

- optimal planning: guarantee that returned plans are optimal, i.e., have minimal overall cost
- suboptimal planning (satisficing): suboptimal plans are allowed

What is New?

Many previously encountered problems are planning tasks:

- blocks world
- route planning in romania
- missionaries and cannibals
- 15-puzzle

New: we are now interested in general algorithms, i.e., the developer of the search algorithm does not know the tasks that the algorithm needs to solve.

⇝ no problem-specific heuristics!

⇝ input language to model the planning task

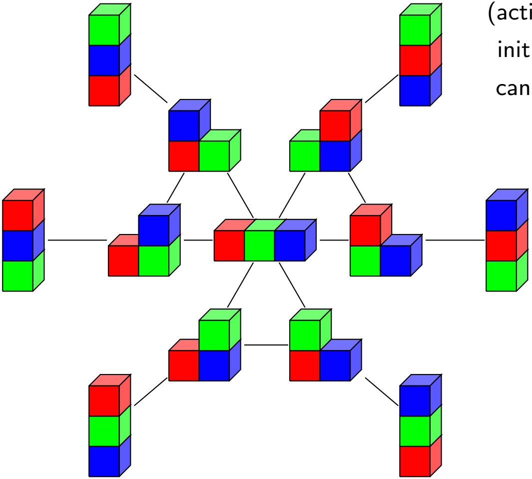
Automated Planning: Overview

Chapter overview: automated planning

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

Compact Descriptions

Reminder: State Space



(action names omitted; initial state and goal can be arbitrary)

- state spaces are (labeled, directed) graphs
- terminology: predecessor, successor, applicable action, path, length, costs, reachable, solution, optimal solution

State Spaces with Declarative Representations

How do we represent state spaces in the computer?

previously: as black box

now: as declarative description

State Spaces with Declarative Representations

represent state spaces declaratively:

- compact description of state space as input to algorithms
  $\rightsquigarrow$ state spaces exponentially larger than the input
- algorithms directly operate on compact description

$\rightsquigarrow$ allows automatic reasoning about problem:
reformulation, simplification, abstraction, etc.

Compact Description of State Spaces

How to describe state spaces compactly?

Compact Description of Several States

- introduce state variables
- states: assignments to state variables

⇝ e.g., n binary state variables can describe 2^n states

- transitions and goal are compactly described with a logic-based formalism

different variants: different planning formalisms

Summary

Summary

- planning: search in general state spaces
- input: compact, declarative description of state space