Basics of AI and Machine Learning

Alternatives to Heuristic Search

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

The Big Three

Planning Approaches: The Big Three

Of the many planning approaches, three techniques stand out:

- explicit search ⇝ previous chapters
  - design choices: search algorithm, heuristic
- SAT planning ⇝ overview in this chapter
- symbolic search ⇝ overview in this chapter

also: many algorithm portfolios

Satisficing or Optimal Planning?

must carefully distinguish:

- satisficing planning: any plan is OK (cheaper ones preferred)
- optimal planning: plans must have minimum cost

solved by similar techniques, but:

- details very different
- almost no overlap between best techniques for satisficing planning and best techniques for optimal planning
- many tasks that are trivial for satisficing planners are impossibly hard for optimal planners

SAT Planning

SAT Planning: Basic Idea

- formalize problem of finding plan with a given horizon (length bound) as a propositional satisfiability problem and feed it to a generic SAT solver
- to obtain a (semi-) complete algorithm, try with increasing horizons until a plan is found (= the formula is satisfiable)
- important optimization: allow applying several non-conflicting actions "at the same time" so that a shorter horizon suffices

SAT Encodings: Variables

- given propositional planning task $\Pi = \langle V, I, G, A \rangle$
- given horizon $T \in \mathbb{N}_0$

Variables of SAT Encoding

- propositional variables v^i for all $v \in V, 0 \leq i \leq T$ encode state after i steps of the plan
- propositional variables a^i for all $a \in A, 1 \leq i \leq T$ encode action a applied in i-th step of the plan

SAT Planning: Design Choices

SAT Encoding

- sequential or parallel
- many ways of modeling planning semantics in logic

$\rightsquigarrow$ main focus of research on SAT planning

Evaluation Strategy

- always advance horizon by $+1$ or more aggressively
- possibly probe multiple horizons concurrently

SAT Solver

- out-of-the-box like MiniSAT, Glucose, Lingeling
- planning-specific modifications

Symbolic Search

Symbolic Search Planning: Basic Ideas

- search processes sets of states at a time
- operators, goal states, state sets reachable with a given cost represented by binary decision diagrams (BDDs) (or similar data structures)
- hope: exponentially large state sets can be represented as polynomially sized BDDs, which can be efficiently processed
- perform symbolic breadth-first search (or something more sophisticated) on these set representations

Symbolic Breadth-First Progression Search

prototypical algorithm:

Symbolic progression breadth-first search

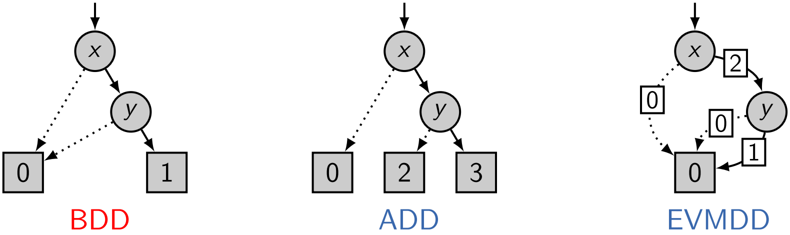
def bfs-progression(V, I, G, A):
    $goal := models(G)$
    $reached := \{I\}$
    loop:
        if $reached \cap goal \neq \emptyset$:
            return solution found
        $new\text{-}reached := reached \cup image(reached, A)$
        if $new\text{-}reached = reached$:
            return no solution exists
        $reached := new\text{-}reached$

$\rightsquigarrow$ If we can implement operations *models*, $\{I\}$, $\cap$, $\neq \emptyset$, $\cup$, *img* and $=$ efficiently, this is a reasonable algorithm.

Design Choice: Symbolic Data Structure

Which data structure should be used for the compact representation of sets of states?

- binary decision diagrams (BDDs)
- algebraic decision diagrams (ADDs)
- edge-valued multi-valued decision diagrams (EVMDDs)
- ...



Summary

Summary

big three classes of algorithms for classical planning:

- explicit search
  - design choices: search algorithm, heuristic
- SAT planning
  - design choices: SAT encoding, SAT solver, evaluation strategy
- symbolic search
  - design choices: symbolic data structure + same ones as for explicit search