Introduction to Planning

Malte Helmert

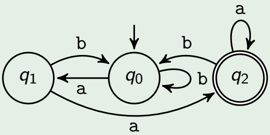
University of Basel

CP 2018 Workshop on Constraints and AI Planning

POP QUIZ: Automata Theory

DFA Pop Quiz: 1

Example



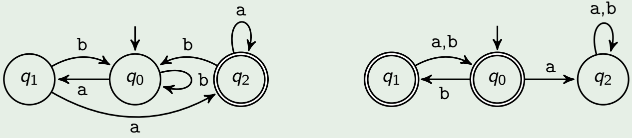
Given: a DFA M

Question: Is $L(M)$ empty?

How can we decide this? How difficult is it?

DFA Pop Quiz: 2

Example



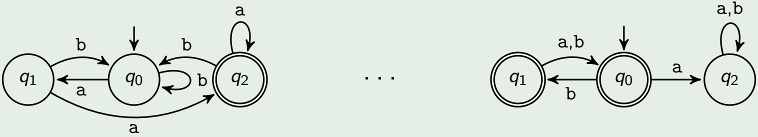
Given: two DFAs M_1 and M_2

Question: Is $L(M_1) \cap L(M_2)$ empty?

How can we decide this? How difficult is it?

DFA Pop Quiz: n

Example



Given: DFAs $M_1, \ldots, M_n$

Question: Is $\bigcap_{i=1}^{n} L(M_i)$ empty?

How can we decide this? How difficult is it?

The Complexity of Classical Planning

- Empty intersection for n DFAs is PSPACE-complete (Kozen 1977).
- It is a trivial syntactic variant of the plan existence problem for classical planning.
- Are we warmed up? Let's talk about planning!

What is Planning?

Planning

Planning (pithy definition)

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

— Patrik Haslum

Planning

Planning (pithy definition)

“Planning is the art and practice of thinking before acting.”

— Patrik Haslum

Planning (more technical definition)

“Selecting a goal-leading course of action
based on a high-level description of the world.”

— Jörg Hoffmann

The Planning Research Landscape

- one of the major subfields of artificial intelligence

⇝ represented at major AI conferences (IJCAI, AAAI, ECAI)

- annual specialized conference ICAPS
    - ≈ 200–250 participants
    - before 2003: ECP (odd years) + AIPS (even years)
- major journals: general AI journals (JAIR, AIJ)

Advertisement: ICAPS 2019

Come to ICAPS 2019!

- ICAPS wants to strengthen its ties to CP and OR
- submission deadline: Nov 16 (abstracts), Nov 21 (papers)
- Journal Presentation Track seeks planning-related papers from other research communities
  $\rightsquigarrow$ submission deadline TBA, most likely in early 2019
- conference in July 2019 in Berkeley, CA, USA

Many Flavors of Planning (1)

What happens when we act?

- deterministic planning
  $\rightsquigarrow$ action sequences in perfectly predictable environments
- nondeterministic planning
  $\rightsquigarrow$ uncertain action outcomes (qualitative uncertainty)
- probabilistic planning
  $\rightsquigarrow$ ditto with quantitative uncertainty

. . . and others (e.g., adversarial and multi-agent)

This talk focuses on deterministic planning.

Many Flavors of Planning (2)

Modelling language complexity:

- classical: finite-domain state variables
- numerical: + real-valued state variables
- temporal: + concurrent and overlapping actions

. . . and many fine-grained modelling features, normal forms, etc.

Many Flavors of Planning (3)

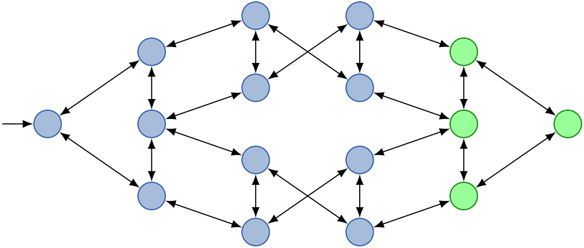
Solution quality requirements:

- optimal planning: only best possible solutions will do
- satisficing planning: optimality not mandatory; better quality preferred

. . . and many further variations (bounded suboptimal, cost-bounded, anytime, oversubscription, net benefit. . .)

Classical Planning as Reachability in Transition Systems

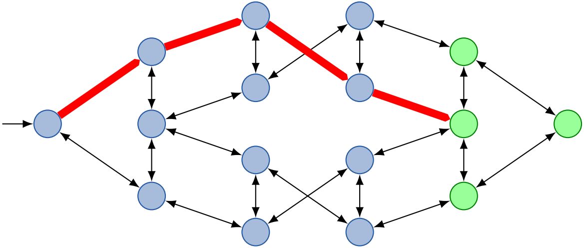
classical planning:
can be seen as finding paths in implicitly defined digraphs



Classical Planning as Reachability in Transition Systems

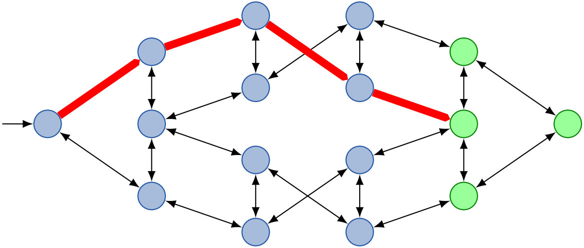
classical planning:

can be seen as finding paths in implicitly defined digraphs



Classical Planning as Reachability in Transition Systems

classical planning:
can be seen as finding paths in large implicitly defined digraphs



Example problem sizes:

- elevator control: $6.92 \cdot 10^{19}$ reachable states
- greenhouse automation: $1.68 \cdot 10^{21}$ reachable states
- transportation logistics: $6.31 \cdot 10^{218}$ reachable states

Example: FreeCell

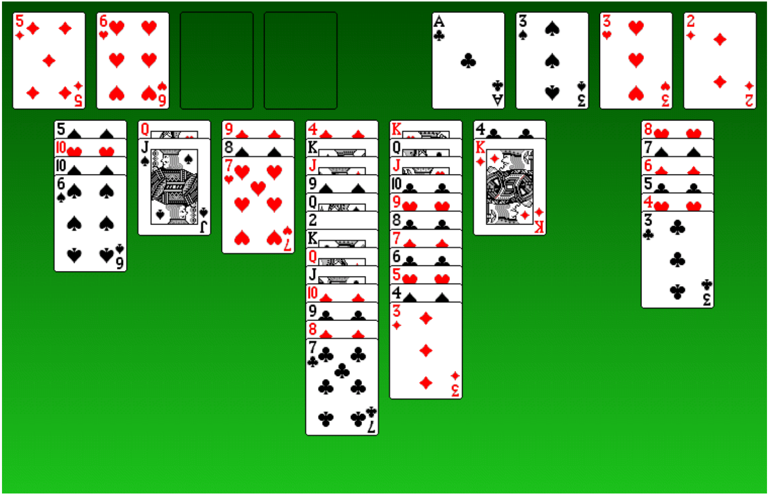


image credits: GNOME Project (GNU General Public License)

Classical Planning

Classical Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

Classical Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, z \mapsto \mathtt{a}\}$

Classical Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, z \mapsto \mathtt{a}\}$
- goal:
  $\{x \mapsto \mathtt{c}, z \mapsto \mathtt{b}\}$

Classical Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, z \mapsto \mathtt{a}\}$
- goal:
  $\{x \mapsto \mathtt{c}, z \mapsto \mathtt{b}\}$
- actions a.k.a. operators:

$$a_1 : \quad x \mapsto \mathtt{a},\ y \mapsto \mathtt{a} \quad \xrightarrow{4} \quad y := \mathtt{b},\ z := \mathtt{c}$$

$$a_2 : \quad x \mapsto \mathtt{a},\ z \mapsto \mathtt{b} \quad \xrightarrow{3} \quad z := \mathtt{b}$$

$\ldots$

Classical Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, z \mapsto \mathtt{a}\}$
- goal:
  $\{x \mapsto \mathtt{c}, z \mapsto \mathtt{b}\}$
- actions a.k.a. operators:

$$\begin{array}{llcl} a_1 : & x \mapsto \mathtt{a},\ y \mapsto \mathtt{a} & \xrightarrow{4} & y := \mathtt{b},\ z := \mathtt{c} \\ a_2 : & x \mapsto \mathtt{a},\ z \mapsto \mathtt{b} & \xrightarrow{3} & z := \mathtt{b} \\ \dots & & & \end{array}$$

Problem:

- find sequence of actions transforming initial state to state consistent with the goal
- objective: minimize sum of action costs

Formalisms for Classical Planning Tasks (1)

Many formalism variants:

Finite-domain representation a.k.a. SAS^+

what we just saw

transition normal form (TNF)

- in each action, precondition variables = effect variables
- goal must describe single state

$\rightsquigarrow$ variants with and without action costs

Formalisms for Classical Planning Tasks (2)

Many formalism variants:

STRIPS

- all variable domains are $\{\mathbf{T}, \mathbf{F}\}$
- only $v \mapsto \mathbf{T}$ in action preconditions and goal
- set-based notations:
  $a: \quad x \mapsto \mathbf{T},\ y \mapsto \mathbf{T} \quad \xrightarrow{5} \quad w := \mathbf{F},\ y := \mathbf{F},\ z := \mathbf{T}$
  written as
  $\mathit{pre}(a) = \{x, y\},\ \mathit{add}(a) = \{z\},\ \mathit{del}(a) = \{w, y\},\ \mathit{cost}(a) = 5$

$\rightsquigarrow$ variants with and without action costs

Formalisms for Classical Planning Tasks (3)

Many formalism variants:

ADL

- all variable domains are $\{\mathbf{T}, \mathbf{F}\}$
- preconditions and goal are logical formulae over state variables
  - precondition $x \mapsto \mathbf{T}, \; y \mapsto \mathbf{T}$ becomes $x \wedge y$
  - but not limited to conjunctions: $(x \vee \neg(y \vee \neg z))$ etc.
- conditional (state-dependent) effects
  - $(y \vee z) \rhd (x := \mathbf{T}), \; (\neg x) \rhd (y := \mathbf{F})$

$\rightsquigarrow$ variants with and without action costs

Algorithms for Classical Planning

main algorithmic approaches for classical planning:

- heuristic state-space search (A^* etc.)
- symbolic search (BDDs etc.)
- compilation to SAT

Numerical and Temporal Planning

Numerical Planning

differences to classical planning:

- may have numerical state variables (usually in $\mathbb{R}$, unbounded)
- numerical conditions in action preconditions and goal
  - often restricted to comparisons to constants
  - examples: $v < 10$, $v \leq 8.5$, $v = 0$, $v \geq 7$, $v > -4$
- numerical action effects
  - often restricted to assignment and addition of constants
  - examples: $v := 5$, $v := v + 3$, $v := v - 0.3$
- numerical state variables used in objective functions
  - replacing/generalizing action costs
  - in planning lingo: metrics

$\rightsquigarrow$ again, many variations (most of them undecidable)

Numerical Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $u \in \mathbb{R}$, $v \in \mathbb{R}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, u \mapsto 0, v \mapsto -4\}$
- goal:
  $\{x \mapsto \mathtt{c}, v \leq 40\}$
- actions a.k.a. operators:
  $a_1: \quad x \mapsto \mathtt{a},\ v = 10 \quad \rightarrow \quad y := \mathtt{b},\ u := 3$
  $a_2: \quad x \mapsto \mathtt{a},\ u \leq 10 \quad \rightarrow \quad u := u + 2$
  $\ldots$

Problem:

- find sequence of actions transforming initial state to state consistent with the goal
- objective: minimize/maximize the given metric (e.g., $3u - v$)

Temporal Planning

differences to classical planning:

- actions have (real-valued) durations
- solutions are not sequences, but schedules of actions
  - actions may overlap
  - objective usually to minimize makespan
- actions have three sets of preconditions:
  - at start, over all, at end
- actions have two sets of effects:
  - at start, at end

⇝ variations include temporally simple and temporally expressive

Temporal Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, z \mapsto \mathtt{a}\}$
- goal:
  $\{x \mapsto \mathtt{c}, z \mapsto \mathtt{b}\}$
- durative actions:
  $$a_1 : \quad x \overset{\text{at start}}{\mapsto} \mathtt{a},\ x \overset{\text{over all}}{\mapsto} \mathtt{b},\ y \overset{\text{at start}}{\mapsto} \mathtt{a} \quad \xrightarrow{10}$$
  $$x \overset{\text{at start}}{:=} \mathtt{b},\ x \overset{\text{at end}}{:=} \mathtt{a},\ z \overset{\text{at end}}{:=} \mathtt{c}$$
  $\dots$

Temporal Planning Tasks

- state variables:
  $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $y \in \{\mathtt{a}, \mathtt{b}\}$, $z \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- initial state:
  $\{x \mapsto \mathtt{a}, y \mapsto \mathtt{a}, z \mapsto \mathtt{a}\}$
- goal:
  $\{x \mapsto \mathtt{c}, z \mapsto \mathtt{b}\}$
- durative actions:

$$a_1 : \quad x \overset{\text{at start}}{\mapsto} \mathtt{a},\ x \overset{\text{over all}}{\mapsto} \mathtt{b},\ y \overset{\text{at start}}{\mapsto} \mathtt{a} \quad \xrightarrow{10}$$

$$x \overset{\text{at start}}{:=} \mathtt{b},\ x \overset{\text{at end}}{:=} \mathtt{a},\ z \overset{\text{at end}}{:=} \mathtt{c}$$

...

Problem:

- find schedule of actions transforming initial state to state consistent with the goal
- objective: minimize makespan of schedule

Combining Numerical and Temporal Planning

- numerical and temporal features often used together
- additional feature: continuous change
  - effect of the form $v := v + 0.1 \cdot \Delta t$

$\rightsquigarrow$ hybrid planning

PDDL & Domain-Independent Planning

PDDL

PDDL: Planning Domain Definition Language

- high-level representation of planning tasks
- standard used in planning community
- LISP-based syntax
- covers all formalism I described (and more)
- parametric description based on first-order logic
- almost all planning algorithms begin by grounding into one of the representations I showed

Domain-Independent Planning

strong focus of planning mainstream on domain-independence:

- one planning algorithm for all domains
- “stupid” models

Modeling Maxim

> “Physics, not advice.”
>
> — Drew McDermott

The International Planning Competition

IPC: International Planning Competition

- competition for planning systems
- focus on blind evaluation on new problems
- currently every 3–4 years
  - ⇝ most recently 2008, 2011, 2014, 2018
- different tracks for different levels of expressiveness
- different tracks for satisficing vs. optimal planning
- source of standard benchmark suite in planning
  - ⇝ 100s of domains, 1000s of tasks

PDDL and Planner Demo: FreeCell

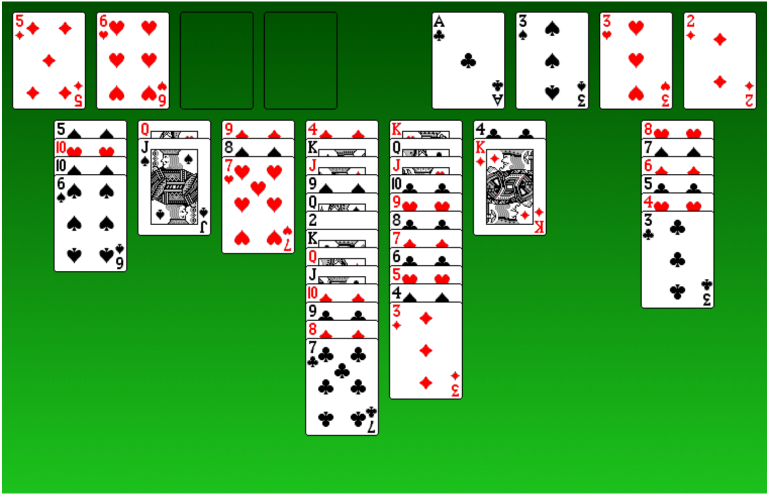


image credits: GNOME Project (GNU General Public License)

Thank you for your attention!