Planning and Optimization

B6. Computational Complexity of Planning: Results

Malte Helmert and Gabriele Röger

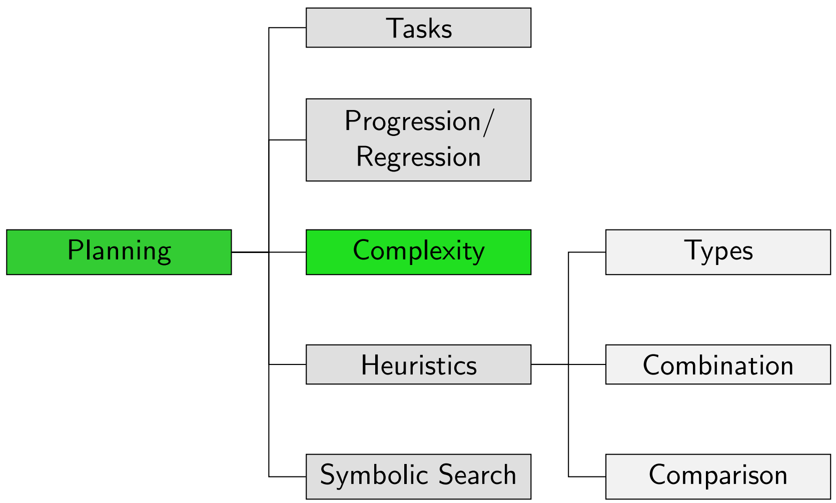
Universität Basel

October 18, 2017

Planning and Optimization

October 18, 2017 — B6. Computational Complexity of Planning: Results

Content of this Course



B6.1 (Bounded-Cost) Plan Existence

The Propositional Planning Problem

Definition (Plan Existence)

The plan existence problem (PLANEX) is the following decision problem:

GIVEN: propositional planning task Π
QUESTION: Is there a plan for Π?

$\rightsquigarrow$ decision problem analogue of satisficing planning

Definition (Bounded-Cost Plan Existence)

The bounded-cost plan existence problem (BCPLANEX) is the following decision problem:

GIVEN: propositional planning task Π, cost bound $K \in \mathbb{N}_0$
QUESTION: Is there a plan for Π with cost at most K?

$\rightsquigarrow$ decision problem analogue of optimal planning

Plan Existence vs. Bounded-Cost Plan Existence

Theorem (Reduction from PLANEX to BCPLANEX)

PLANEX $\leq_p$ BCPLANEX

Proof.

Consider a propositional planning task Π with n state variables.
Let $c_{\max}$ be the maximal cost of all actions of Π.

Π is solvable iff there is solution with cost at most $c_{\max} \cdot (2^n - 1)$ because a solution need not visit any state twice.

$\rightsquigarrow$ map instance Π of PLANEX to instance $\langle \Pi, c_{\max} \cdot (2^n - 1) \rangle$ of BCPLANEX

$\rightsquigarrow$ polynomial reduction □

B6.2 PSPACE-Completeness of Planning

Membership in PSPACE

Theorem

BCPLANEX $\in$ PSPACE

Proof.

Show BCPLANEX $\in$ NPSPACE and use Savitch's theorem.

Nondeterministic algorithm:

def plan($\langle V, I, O, \gamma \rangle$, K):
    $s := I$
    $k := K$
    loop forever:
        if $s \models \gamma$: **accept**
        guess $o \in O$
        if $s \not\models pre(o)$: **fail**
        if $cost(o) > k$: **fail**
        $s := s[\![o]\!]$
        $k := k - cost(o)$

□

PSPACE-Hardness

Idea: generic reduction

- For an arbitrary fixed DTM M with space bound polynomial p and input w, generate planning task which is solvable iff M accepts w in space $p(|w|)$.
- For simplicity, restrict to TMs which never move to the left of the initial head position (no loss of generality).

Reduction: State Variables

Let $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$ be the fixed DTM, and let p be its space-bound polynomial.

Given input $w_1 \dots w_n$, define relevant tape positions $X := \{1, \dots, p(n)\}$.

State Variables

- state_q for all $q \in Q$
- head_i for all $i \in X \cup \{0, p(n) + 1\}$
- $\mathrm{content}_{i,a}$ for all $i \in X$, $a \in \Sigma_\Box$

$\rightsquigarrow$ allows encoding a Turing machine configuration

Reduction: Initial State

Let $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$ be the fixed DTM, and let p be its space-bound polynomial.

Given input $w_1 \dots w_n$, define relevant tape positions $X := \{1, \dots, p(n)\}$.

Initial State

Initially true:

- state_{q_0}
- head_1
- $\mathrm{content}_{i,w_i}$ for all $i \in \{1, \dots, n\}$
- $\mathrm{content}_{i,\Box}$ for all $i \in X \setminus \{1, \dots, n\}$

Initially false:

- all others

Reduction: Operators

Let $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$ be the fixed DTM, and let p be its space-bound polynomial.

Given input $w_1 \dots w_n$, define relevant tape positions $X := \{1, \dots, p(n)\}$.

Operators

One operator for each transition rule $\delta(q, a) = \langle q', a', \Delta \rangle$ and each cell position $i \in X$:

- precondition: $\mathrm{state}_q \land \mathrm{head}_i \land \mathrm{content}_{i,a}$
- effect: $\neg\mathrm{state}_q \land \neg\mathrm{head}_i \land \neg\mathrm{content}_{i,a}$
  $\land\, \mathrm{state}_{q'} \land \mathrm{head}_{i+\Delta} \land \mathrm{content}_{i,a'}$

Reduction: Goal

Let $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$ be the fixed DTM, and let p be its space-bound polynomial.

Given input $w_1 \dots w_n$, define relevant tape positions $X := \{1, \dots, p(n)\}$.

Goal

state_{q_Y}

PSPACE-Completeness of STRIPS Plan Existence

Theorem (PSPACE-Completeness; Bylander, 1994)

PlanEx and BCPlanEx are PSPACE-complete. This is true even if only STRIPS tasks are allowed.

Proof.

Membership for BCPlanEx was already shown.

Hardness for PlanEx follows because we just presented a polynomial reduction from an arbitrary problem in PSPACE to PlanEx. (Note that the reduction only generates STRIPS tasks.)

Membership for PlanEx and hardness for BCPlanEx follow from the polynomial reduction from PlanEx to BCPlanEx. □

B6.3 More Complexity Results

More Complexity Results

In addition to the basic complexity result presented in this chapter, there are many special cases, generalizations, variations and related problems studied in the literature:

- different planning formalisms
  - e.g., finite-domain representation, nondeterministic effects, partial observability, schematic operators, numerical state variables
- syntactic restrictions of planning tasks
  - e.g., without preconditions, without conjunctive effects, STRIPS without delete effects
- semantic restrictions of planning task
  - e.g., restricting variable dependencies ("causal graphs")
- particular planning domains
  - e.g., Blocksworld, Logistics, FreeCell

Complexity Results for Different Planning Formalisms

Some results for different planning formalisms:

- FDR tasks:
  - same complexity as for propositional tasks ("folklore")
  - also true for the SAS^+ and TNF special cases
- nondeterministic effects:
  - fully observable: EXP-complete (Littman, 1997)
  - unobservable: EXPSPACE-complete (Haslum & Jonsson, 1999)
  - partially observable: 2-EXP-complete (Rintanen, 2004)
- schematic operators:
  - usually adds one exponential level to PLANEX complexity
  - e.g., classical case EXPSPACE-complete (Erol et al., 1995)
- numerical state variables:
  - undecidable in most variations (Helmert, 2002)

B6.4 Summary

Summary

- Propositional planning is PSPACE-complete.
- This is true both for satisficing and optimal planning.
- The hardness proof is a polynomial reduction that translates an arbitrary polynomial-space DTM into a STRIPS task:
  - DTM configurations are encoded by state variables.
  - Operators simulate transitions between DTM configurations.
  - The DTM accepts an input iff there is a plan for the corresponding STRIPS task.
- This implies that there is no polynomial algorithm for classical planning unless P = PSPACE.
- It also means that planning is not polynomially reducible to any problem in NP unless NP = PSPACE.