

# Planning and Optimization

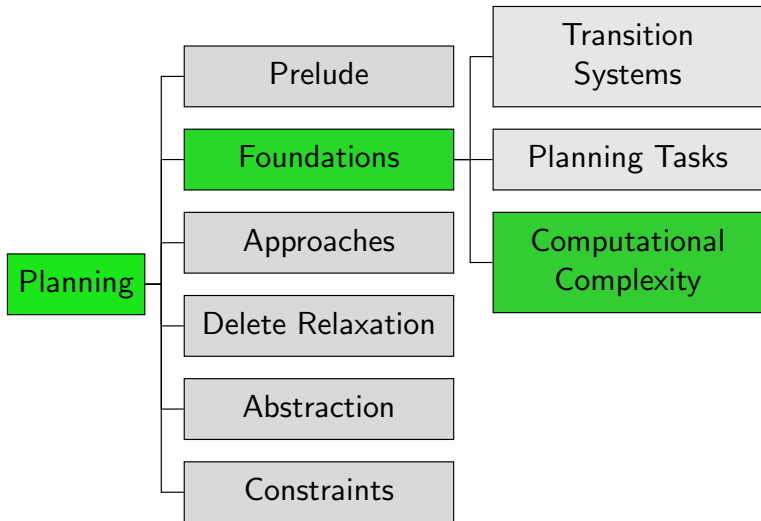
## B6. Computational Complexity of Planning

Malte Helmert and Gabriele Röger

Universität Basel

October 2, 2024

# Content of the Course



# Motivation

# How Difficult is Planning?

- Using **state-space search** (e.g., using Dijkstra's algorithm on the transition system), planning can be solved in **polynomial time** in the **number of states**.
- However, the number of states is **exponential** in the number of **state variables**, and hence in general exponential in the size of the input to the planning algorithm.
- ↪ Do non-exponential planning algorithms exist?
- ↪ What is the precise **computational complexity** of planning?

# Why Computational Complexity?

- **understand** the problem
- know what is **not** possible
- find interesting **subproblems** that are easier to solve
- distinguish **essential features** from **syntactic sugar**
  - Is STRIPS planning easier than general planning?

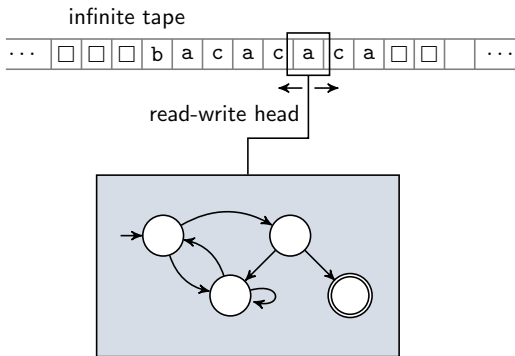
# Background: Complexity Theory

# Reminder: Complexity Theory

## Need to Catch Up?

- We assume knowledge of complexity theory:
  - languages and decision problems
  - Turing machines: NTMs and DTMs;  
polynomial equivalence with other models of computation
  - complexity classes: P, NP, PSPACE
  - polynomial reductions
- If you are not familiar with these topics, we recommend **Chapters B11, D1–D3, D6** of the **Theory of Computer Science** course at <https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-fs24/10948-main-lecture-theory-of-computer-science/>

# Turing Machines: Conceptually





# Turing Machines

## Definition (Nondeterministic Turing Machine)

A **nondeterministic Turing machine (NTM)** is a 6-tuple  $\langle \Sigma, \square, Q, q_0, q_Y, \delta \rangle$  with the following components:

- **input alphabet**  $\Sigma$  and **blank symbol**  $\square \notin \Sigma$ 
  - alphabets always nonempty and finite
  - **tape alphabet**  $\Sigma_{\square} = \Sigma \cup \{\square\}$
- finite set  $Q$  of **internal states** with **initial state**  $q_0 \in Q$  and **accepting state**  $q_Y \in Q$ 
  - **nonterminal states**  $Q' := Q \setminus \{q_Y\}$
- **transition relation**  $\delta : (Q' \times \Sigma_{\square}) \rightarrow 2^{Q \times \Sigma_{\square} \times \{-1, +1\}}$

**Deterministic Turing machine (DTM):**

$$|\delta(q, s)| = 1 \text{ for all } \langle q, s \rangle \in Q' \times \Sigma_{\square}$$

# Turing Machines: Accepted Words

## ■ Initial configuration

- state  $q_0$
- input word on tape, all other tape cells contain  $\square$
- head on first symbol of input word

## ■ Step

- If in state  $q$ , reading symbol  $s$ , and  $\langle q', s', d \rangle \in \delta(q, s)$  then
  - the NTM **can** transition to state  $q'$ , replacing  $s$  with  $s'$  and moving the head one cell to the left/right ( $d = -1/+1$ ).
- Input word ( $\in \Sigma^*$ ) is **accepted** if **some** sequence of transitions brings the NTM from the initial configuration into state  $s_Y$ .

# Acceptance in Time and Space

## Definition (Acceptance of a Language in Time/Space)

Let  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ .

A NTM **accepts** language  $L \subseteq \Sigma^*$  **in time  $f$**  if it accepts each  $w \in L$  within  $f(|w|)$  steps and does not accept any  $w \notin L$  (in any time).

It **accepts** language  $L \subseteq \Sigma^*$  **in space  $f$**  if it accepts each  $w \in L$  using at most  $f(|w|)$  tape cells and does not accept any  $w \notin L$ .

# Time and Space Complexity Classes

## Definition (DTIME, NTIME, DSPACE, NSPACE)

Let  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ .

Complexity class **DTIME**( $f$ ) contains all languages accepted in time  $f$  by some DTM.

Complexity class **NTIME**( $f$ ) contains all languages accepted in time  $f$  by some NTM.

Complexity class **DSPACE**( $f$ ) contains all languages accepted in space  $f$  by some DTM.

Complexity class **NSPACE**( $f$ ) contains all languages accepted in space  $f$  by some NTM.

# Polynomial Time and Space Classes

Let  $\mathcal{P}$  be the set of polynomials  $p : \mathbb{N}_0 \rightarrow \mathbb{N}_0$  whose coefficients are natural numbers.

## Definition (P, NP, PSPACE, NPSPACE)

$$P = \bigcup_{p \in \mathcal{P}} \text{DTIME}(p)$$

$$NP = \bigcup_{p \in \mathcal{P}} \text{NTIME}(p)$$

$$\text{PSPACE} = \bigcup_{p \in \mathcal{P}} \text{DSPACE}(p)$$

$$\text{NPSPACE} = \bigcup_{p \in \mathcal{P}} \text{NSPACE}(p)$$

# Polynomial Complexity Class Relationships

## Theorem (Complexity Class Hierarchy)

$$P \subseteq NP \subseteq PSPACE = NPSPACE$$

## Proof.

$P \subseteq NP$  and  $PSPACE \subseteq NPSPACE$  are obvious because deterministic Turing machines are a special case of nondeterministic ones.

$NP \subseteq NPSPACE$  holds because a Turing machine can only visit polynomially many tape cells within polynomial time.

$PSPACE = NPSPACE$  is a special case of a classical result known as Savitch's theorem (Savitch 1970). □

# (Bounded-Cost) Plan Existence

# Decision Problems for Planning

## Definition (Plan Existence)

**Plan existence (PLANEX)** is the following decision problem:

GIVEN:            planning task  $\Pi$

QUESTION:    Is there a plan for  $\Pi$ ?

↪ decision problem analogue of **satisficing planning**

## Definition (Bounded-Cost Plan Existence)

**Bounded-cost plan existence (BCPLANEX)**

is the following decision problem:

GIVEN:            planning task  $\Pi$ , cost bound  $K \in \mathbb{N}_0$

QUESTION:    Is there a plan for  $\Pi$  with cost at most  $K$ ?

↪ decision problem analogue of **optimal planning**



# Plan Existence vs. Bounded-Cost Plan Existence

## Theorem (Reduction from PLANEX to BCPLANEX)

$\text{PLANEX} \leq_p \text{BCPLANEX}$

### Proof.

Consider a planning task  $\Pi$  with state variables  $V$ .

Let  $c_{\max}$  be the maximal cost of all operators of  $\Pi$ .

Compute the number of states of  $\Pi$  as  $N = 2^{|V|}$ .

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

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

$\rightsquigarrow$  polynomial reduction



# 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  $o$  is not applicable in  $s$ : 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 propositional planning task which is solvable iff  $M$  accepts  $w$  in space  $p(|w|)$ .
- Without loss of generality, we assume  $p(n) \geq n$  for all  $n$ .

## Reduction: State Variables

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

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

### State Variables

- $\text{state}_q$  for all  $q \in Q$
- $\text{head}_i$  for all  $i \in X \cup \{-p(n) - 1, p(n) + 1\}$
- $\text{content}_{i,a}$  for all  $i \in X, a \in \Sigma_\square$

$\rightsquigarrow$  allows encoding a Turing machine configuration

## Reduction: Initial State

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

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

### Initial State

Initially true:

- state  $q_0$
- head<sub>1</sub>
- content <sub>$i, w_i$</sub>  for all  $i \in \{1, \dots, n\}$
- content <sub>$i, \square$</sub>  for all  $i \in X \setminus \{1, \dots, n\}$

Initially false:

- all others

## Reduction: Operators

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

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

### Operators

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

- precondition:  $\text{state}_q \wedge \text{head}_i \wedge \text{content}_{i,a}$
- effect:  $\neg \text{state}_q \wedge \neg \text{head}_i \wedge \neg \text{content}_{i,a}$   
 $\wedge \text{state}_{q'} \wedge \text{head}_{i+d} \wedge \text{content}_{i,a'}$

Note that add-after-delete semantics are important here!

## Reduction: Goal

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

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

Goal

state <sub>$q_Y$</sub>



# PSPACE-Completeness of STRIPS Plan Existence

Theorem (PSPACE-Completeness; Bylander, 1994)

$\text{PLANEX}$  and  $\text{BCPLANEX}$  are PSPACE-complete.  
*This is true even if only STRIPS tasks are allowed.*

Proof.

Membership for  $\text{BCPLANEX}$  was already shown.

Hardness for  $\text{PLANEX}$  follows because we just presented a polynomial reduction from an arbitrary problem in PSPACE to  $\text{PLANEX}$ . (Note that the reduction only generates STRIPS tasks, after trivial cleanup to make them conflict-free.)

Membership for  $\text{PLANEX}$  and hardness for  $\text{BCPLANEX}$  follow from the polynomial reduction from  $\text{PLANEX}$  to  $\text{BCPLANEX}$ .  $\square$

# 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., 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:

- **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)

# Summary

# Summary

- **PSPACE**: decision problems solvable in **polynomial space**
- $P \subseteq NP \subseteq PSPACE = NPSPACE$ .
- **Classical planning** is **PSPACE-complete**.
- This is true both for **satisficing** and **optimal** planning (rather, the corresponding decision problems).
- 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  $NP = PSPACE$ .