1 Motivation

How hard is planning?

- We have seen that planning can be done in time polynomial in the size of the transition system.
- However, we have not seen algorithms which are polynomial in the input size (size of the task description).
- What is the precise computational complexity of the planning problem?
Motivation

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?
  - Is planning for FDR tasks harder than for propositional tasks?

2 Background

- Turing machines
- Complexity classes

Background

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 \( \delta \subseteq (Q' \times \Sigma \square) \times (Q \times \Sigma \square \times \{-1, +1\}) \)

Deterministic Turing machines

Definition (deterministic Turing machine)

A deterministic Turing machine (DTM) is an NTM where the transition relation is functional, i.e., for all \( \langle q, a \rangle \in Q' \times \Sigma \square \), there is exactly one triple \( \langle q', a', \Delta \rangle \) with \( \langle \langle q, a \rangle, \langle q', a', \Delta \rangle \rangle \in \delta \).

Notation: We write \( \delta(q, a) \) for the unique triple \( \langle q', a', \Delta \rangle \) such that \( \langle \langle q, a \rangle, \langle q', a', \Delta \rangle \rangle \in \delta \).
Turing machine configurations

Definition (Configuration)
Let $M = (\Sigma, \Box, Q, q_0, q_Y, \delta)$ be an NTM.
A configuration of $M$ is a triple $(w, q, x) \in \Sigma^* \times Q \times \Sigma^*$.

- $w$: tape contents before tape head
- $q$: current state
- $x$: tape contents after and including tape head

Let $c = (w, q, x)$ be a configuration of $M$, and let $n \in \mathbb{N}_0$.

- If $q = q_Y$, $M$ accepts $c$ in time $n$.
- If $q \neq q_Y$ and $M$ accepts some $c'$ with $c \vdash c'$ in time $n$, then $M$ accepts $c$ in time $n + 1$.

Definition (accepting configuration, time)
Let $c = (w, q, x)$ be a configuration of $M$, and let $n \in \mathbb{N}_0$.

- If $q = q_Y$, $M$ accepts $c$ in time $n$.
- If $q \neq q_Y$ and $M$ accepts some $c'$ with $c \vdash c'$ in time $n$, then $M$ accepts $c$ in space $n$.

Definition (accepting configuration, space)
Let $c = (w, q, x)$ be a configuration of $M$, and let $n \in \mathbb{N}_0$.

- If $q = q_Y$ and $|w| + |x| \leq n$, $M$ accepts $c$ in space $n$.
- If $q \neq q_Y$ and $M$ accepts some $c'$ with $c \vdash c'$ in space $n$, then $M$ accepts $c$ in space $n$.

Accepting words and languages

Definition (accepting words)
Let $M = (\Sigma, \Box, Q, q_0, q_Y, \delta)$ be an NTM.
$M$ accepts the word $w \in \Sigma^*$ in time (space) $n \in \mathbb{N}_0$ iff $M$ accepts $(\epsilon, q_0, w)$ in time (space) $n$.

- Special case: $M$ accepts $\epsilon$ in time (space) $n \in \mathbb{N}_0$ iff $M$ accepts $(\epsilon, q_0, \Box)$ in time (space) $n$.

Definition (accepting languages)
Let $M = (\Sigma, \Box, Q, q_0, q_Y, \delta)$ be an NTM, and let $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$.
$M$ accepts the language $L \subseteq \Sigma^*$ in time (space) $f$ iff $M$ accepts each word $w \in L$ in time (space) $f(|w|)$, and $M$ does not accept any word $w \notin L$ (in any time/space).
Background Complexity classes

Time and space complexity classes

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

Let \( f : \mathbb{N}_0 \rightarrow \mathbb{N}_0 \).

- Complexity class \( \text{DTIME}(f) \) contains all languages accepted in time \( f \) by some DTM.
- Complexity class \( \text{NTIME}(f) \) contains all languages accepted in time \( f \) by some NTM.
- Complexity class \( \text{DSPACE}(f) \) contains all languages accepted in space \( f \) by some DTM.
- Complexity class \( \text{NSPACE}(f) \) contains all languages accepted in space \( f \) by some NTM.

Polynomial time and space classes

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

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

\[
\begin{align*}
P &= \bigcup_{p \in P} \text{DTIME}(p) \\
NP &= \bigcup_{p \in P} \text{NTIME}(p) \\
PSPACE &= \bigcup_{p \in P} \text{DSPACE}(p) \\
NPSPACE &= \bigcup_{p \in P} \text{NSPACE}(p)
\end{align*}
\]

Polynomial complexity class relationships

**Theorem (complexity class hierarchy)**

\( P \subseteq NP \subseteq PSPACE = NPSPACE \)

**Proof.**

- \( P \subseteq NP \) and \( PSPACE \subseteq NPSPACE \) is 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).

3 Complexity of propositional planning

- Plan existence and bounded plan existence
- PSPACE-completeness
The propositional planning problem

Definition (plan existence)
The plan existence problem (\textsc{PlanEx})
is the following decision problem:
\begin{itemize}
\item \textbf{Given:} Planning task \(\Pi\)
\item \textbf{Question:} Is there a plan for \(\Pi\)?
\end{itemize}
\(\Rightarrow\) decision problem analogue of satisficing planning

Definition (bounded plan existence)
The bounded plan existence problem (\textsc{PlanLen})
is the following decision problem:
\begin{itemize}
\item \textbf{Given:} Planning task \(\Pi\), length bound \(K \in \mathbb{N}_0\)
\item \textbf{Question:} Is there a plan for \(\Pi\) of length at most \(K\)?
\end{itemize}
\(\Rightarrow\) decision problem analogue of optimal planning

Plan existence vs. bounded plan existence

Theorem (reduction from \textsc{PlanEx} to \textsc{PlanLen})
\(\text{PlanEx} \leq_p \text{PlanLen}\)

Proof.
A propositional planning task with \(n\) state variables has a plan
iff it has a plan of length at most \(2^n - 1\).
\(\Rightarrow\) map instance \(\Pi\) of \textsc{PlanEx} to instance \(\langle \Pi, 2^n - 1 \rangle\) of \textsc{PlanLen},
where \(n\) is the number of \(n\) state variables of \(\Pi\)
\(\Rightarrow\) polynomial reduction

Membership in PSPACE

Theorem (PSPACE membership for \textsc{PlanLen})
\(\text{PlanLen} \in \text{PSPACE}\)

Proof.
Show \(\text{PlanLen} \in \text{NPSPACE}\) and use Savitch’s theorem.
Nondeterministic algorithm:
\begin{verbatim}
def plan((A, I, O, G), K):
    s := I
    k := K
    while s \not\models G:
        guess o \in O
        fail if o not applicable in s or k = 0
        s := apply\(o\)(s)
        k := k - 1
    accept
\end{verbatim}

Hardness for PSPACE

Idea: generic reduction
\begin{itemize}
\item 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 \(p(|w|)\).
\item For simplicity, restrict to TMs which never move to the left of
the initial head position (no loss of generality).
\end{itemize}
Complexity of planning  PSPACE-completeness

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

State variables

- state$_q$ for all $q \in Q$
- head$_i$ for all $i \in X \cup \{0, p(n) + 1\}$
- content$_{i,a}$ for all $i \in X$, $a \in \Sigma^\square$

$\Rightarrow$ allows encoding a Turing machine configuration.

Complexity of planning  PSPACE-completeness

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

Initial state

Initially true:

- state$q_0$
- head$_1$
- content$_{i,w_i}$ for all $i \in \{1, \ldots, n\}$
- content$_{i,\square}$ for all $i \in X \setminus \{1, \ldots, n\}$

Initially false:

- all others

Complexity of planning  PSPACE-completeness

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

Operators

One operator for each transition rule $\delta(q, a) = \langle q', a', \Delta \rangle$

and each cell position $i \in X$:

- precondition: state$_q$ $\land$ head$_i$ $\land$ content$_{i,a}$
- effect: $\neg$state$_q$ $\land$ $\neg$head$_i$ $\land$ $\neg$content$_{i,a}$

$\land$ state$_{q'}$ $\land$ head$_{i+\Delta}$ $\land$ content$_{i,a'}$

- If $q = q'$ and/or $a = a'$, omit the effects on state$_q$ and/or content$_{i,a}$, to avoid consistency condition issues.

Complexity of planning  PSPACE-completeness

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

Goal

state$_{q_Y}$
PSPACE-completeness for STRIPS plan existence

Theorem (PSPACE-completeness; Bylander, 1994)

\textit{PlanEx and PlanLen are PSPACE-complete. This is true even when restricting to STRIPS tasks.}

\textbf{Proof.}

Membership for \textit{PlanLen} was already shown.

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

Membership for \textit{PlanEx} and hardness for \textit{PlanLen} follows from the polynomial reduction from \textit{PlanEx} to \textit{PlanLen}.

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 to certain classes of causal graphs
- particular planning domains
  - e.g., Blocksworld, Logistics, FreeCell

Complexity results for different planning formalisms

Some results for different planning formalisms:

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

- Propositional planning is PSPACE-complete.
- The hardness proof is a polynomial reduction that translates an arbitrary polynomial-space DTM into a STRIPS task:
  - Configurations of the DTM are encoded by propositional variables.
  - Operators simulate transitions of the DTM.
  - 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 classical planning is not polynomially reducible to any problem in NP unless NP=PSPACE.