Overview

Similar to the earlier analysis of deterministic planning, we will now study the computational complexity of nondeterministic planning with full observability.

We consider the case of strong planning.

The results for strong cyclic planning are identical.

As usual, the main motivation for such a study is to determine the limit of what is possible algorithmically: Should we try to develop a polynomial algorithm?

Comparison to deterministic planning

The basic proof idea is very similar to the PSPACE-completeness proof for deterministic planning.

The main difference is that we consider alternating Turing Machines (ATMs) instead of deterministic Turing Machines (DTMs) in the reduction.

Due to the similarity to the earlier proof, we first review some of the concepts introduced in the earlier lecture.
Alternating Turing Machines

Definition: Alternating Turing Machine

Alternating Turing Machine (ATM) \(\langle \Sigma, \square, Q, q_0, I, \delta \rangle\):

1. input alphabet \(\Sigma\) and blank symbol \(\square \notin \Sigma\)
2. alphas always non-empty and finite
3. tape alphabet \(\Sigma \cup \{\square\}\)
4. finite set \(Q\) of internal states with initial state \(q_0 \in Q\)
5. state labeling \(I : Q \rightarrow \{Y, N, \exists, \forall\}\)
   - accepting, rejecting, existential, universal states
   - terminal states \(Q_Y = Q_Y \cup Q_N\)
   - nonterminal states \(Q' = Q_\exists \cup Q_\forall\)
6. transition relation \(\delta \subseteq (Q' \times \Sigma^\bullet) \times (Q \times \Sigma \times \{-1, +1\})\)

Turing Machine configurations

Let \(M = \langle \Sigma, \square, Q, q_0, I, \delta \rangle\) be an ATM.

Definition: Configuration

A configuration of \(M\) is a triple \((w, q, x) \in \Sigma^\bullet \times Q \times \Sigma^\bullet\).

- \(w\): tape contents before tape head
- \(q\): current state
- \(x\): tape contents after and including tape head

Turing Machine transitions

Let \(M = \langle \Sigma, \square, Q, q_0, I, \delta \rangle\) be an ATM.

Definition: Yields relation

A configuration \(c\) of \(M\) yields a configuration \(c'\) of \(M\), in symbols \(c \vdash c'\), as defined by the following rules, where \(a, a', b \in \Sigma, w, x \in \Sigma^\bullet, q, q' \in Q\) and \(((q, a), (q', a', \Delta)) \in \delta\):

- \((w, q, ax) \vdash (w a', q', x)\) if \(\Delta = +1, |x| \geq 1\)
- \((w, q, a) \vdash (w a', q', \square)\) if \(\Delta = +1\)
- \((wb, q, ax) \vdash (w, q', ba'x)\) if \(\Delta = -1\)
- \((e, q, ax) \vdash (e, q', \square a'x)\) if \(\Delta = -1\)
Acceptance (space)

Let $M = \langle \Sigma, \square, Q, q_0, l, \delta \rangle$ be an ATM.

**Definition: Acceptance (space)**

Let $c = (w, q, x)$ be a configuration of $M$.

- $M$ accepts $c = (w, q, x)$ with $q \in Q_Y$ in space $n$ iff $|w| + |x| \leq n$.
- $M$ accepts $c = (w, q, x)$ with $q \in Q_2$ in space $n$ iff $M$ accepts some $c'$ with $c \vdash c'$ in space $n$.
- $M$ accepts $c = (w, q, x)$ with $q \in Q_Y$ in space $n$ iff $M$ accepts all $c'$ with $c \vdash c'$ in space $n$.

Accepting words and languages

Let $M = \langle \Sigma, \square, Q, q_0, l, \delta \rangle$ be an ATM.

**Definition: Accepting words**

$M$ accepts the word $w \in \Sigma^*$ in space $n \in \mathbb{N}_0$ iff $M$ accepts $(\varepsilon, q_0, w)$ in space $n$.

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

**Definition: Accepting languages**

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

$M$ accepts the language $L \subseteq \Sigma^*$ in space $f$ iff $M$ accepts each word $w \in L$ in space $f(|w|)$, and $M$ does not accept any word $w \notin L$.

Alternating space complexity

**Definition: ASPACE, APSPACE**

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

Complexity class ASPACE($f$) contains all languages accepted in space $f$ by some ATM.

Let $\mathcal{P}$ be the set of polynomials $p : \mathbb{N}_0 \rightarrow \mathbb{N}_0$.

$$\text{APSPACE} := \bigcup_{p \in \mathcal{P}} \text{ASPACE}(p)$$

Standard complexity classes relationships

**Theorem**

$P \subseteq NP \subseteq AP$  
$PSPACE \subseteq NPSPACE \subseteq APSPACE$  
$EXP \subseteq NEXP \subseteq AEXP$  
$EXPSPACE \subseteq NEXPSPACE \subseteq AEXPSPACE$  
$2 \cdot EXP \subseteq \ldots$
The power of alternation

Theorem (Chandra et al. 1981)

\[
\begin{align*}
AP &= \text{PSPACE} \\
\text{APSPACE} &= \text{EXP} \\
\text{AEXP} &= \text{EXPSPACE} \\
\text{AEXPSPACE} &= 2\text{-EXP}
\end{align*}
\]

The hierarchy of complexity classes

\[
\begin{align*}
2\text{-EXPSPACE} &= 2\text{-NEXPSPACE} \\
2\text{-NEXP} &= \text{APSPACE} \\
\text{EXPSPACE} &= \text{NEXPSPACE} = \text{AEXP} \\
\text{EXP} &= \text{APSPACE} \\
\text{PSPACE} &= \text{NPSPACE} = \text{AP} \\
\text{NP} &\neq \text{P}
\end{align*}
\]

3 Complexity results

- The strong planning problem
- APSPACE reduction
- EXP-completeness proof

The strong planning problem

**STRONGPlanEx** (strong plan existence)

**Given:** nondeterministic planning task \( \langle A, I, O, G, V \rangle \) with full observability \( A = V \)

**Question:** Is there a strong plan for the task?

- We do not consider a nondeterministic analog of the bounded plan existence problem (PLANLEN).
We will prove that StrongPlanEx is EXP-complete.
We already know that the problem belongs to EXP, because we have presented a dynamic programming algorithm that generates strong plans in exponential time.
We prove hardness for EXP by providing a generic reduction for alternating Turing Machines with polynomial space and use Chandra et al.’s theorem showing $\text{APSPACE} = \text{EXP}$.

For a fixed polynomial $p$, given ATM $M$ and input $w$, generate planning task which is solvable by a strong plan iff $M$ accepts $w$ in space $p(|w|)$.
For simplicity, restrict to ATMs which never move to the left of the initial head position (no loss of generality).
Existential states of the ATM are modeled by states of the planning task where there are several applicable operators to choose from.
Universal states of the ATM are modeled by states of the planning task where there is a single applicable operator with a nondeterministic effect.

Let $p$ be the space-bound polynomial.
Given ATM $\langle \Sigma, \square, Q, q_0, l, \delta \rangle$ and input $w_1 \ldots w_n$, define relevant tape positions $X = \{1, \ldots, p(n)\}$.
State variables
- $\text{state}_q$ for all $q \in Q$
- $\text{head}_i$ for all $i \in X \cup \{0, p(n) + 1\}$
- $\text{content}_{i,a}$ for all $i \in X$, $a \in \Sigma \square$

Let $p$ be the space bound polynomial.
Given ATM $\langle \Sigma, \square, Q, q_0, l, \delta \rangle$ and input $w_1 \ldots w_n$, define relevant tape positions $X = \{1, \ldots, p(n)\}$.
Initial state formula
Specify a unique initial state.
Initially true:
- $\text{state}_{q_0}$
- $\text{head}_1$
- $\text{content}_{i,w_i}$ for all $i \in \{1, \ldots, n\}$
- $\text{content}_{i,\square}$ for all $i \in X \setminus \{1, \ldots, n\}$
Initially false:
- all others
Reduction: goal

Let \( p \) be the space bound polynomial.

Given ATM \( \langle \Sigma, \square, Q, q_0, l, \delta \rangle \) and input \( w_1 \ldots w_n \), define relevant tape positions \( X = \{1, \ldots, p(n)\} \).

Goal

\[ V_{q \in Q} \text{state}_{q} \]

- Without loss of generality, we can assume that \( Q_Y \) is a singleton set so that we do not need a disjunctive goal.
- This way, the hardness result also holds for a restricted class of planning tasks (“nondeterministic STRIPS”).

Reduction: operators

Let \( p \) be the space bound polynomial.

Given ATM \( \langle \Sigma, \square, Q, q_0, l, \delta \rangle \) and input \( w_1 \ldots w_n \), define relevant tape positions \( X = \{1, \ldots, p(n)\} \).

Operators

For \( q, q' \in Q, a, a' \in \Sigma, \Delta \in \{-1, +1\}, i \in X \), define

- \( \text{pre}_{q,a,i} = \text{state}_q \wedge \text{head}_{i} \wedge \text{content}_{i,a} \)
- \( \text{eff}_{q,a,q',a',\Delta,i} = \neg \text{state}_q \wedge \neg \text{head}_{i} \wedge \neg \text{content}_{i,a} \wedge \text{state}_{q'} \wedge \text{head}_{i+\Delta} \wedge \text{content}_{i,a'} \)

  - If \( q = q' \), omit the effects \( \neg \text{state}_q \) and \( \text{state}_{q'} \).
  - If \( a = a' \), omit the effects \( \neg \text{content}_{i,a} \) and \( \text{content}_{i,a'} \).
EXP-completeness of strong planning with full observability

Theorem (Rintanen)

**STRON{G|P}LANEx** is EXP-complete.

This is true even if we only allow operators in unary nondeterminism normal form where all deterministic sub-effects and the goal satisfy the STRIPS restriction and if we require a deterministic initial state.

Proof.

Membership in EXP has been shown by providing exponential-time algorithms that generate strong plans (and decide if one exists as a side effect).

Hardness follows from the previous generic reduction for ATMs with polynomial space bound and Chandra et al.’s theorem.

Summary

- Nondeterministic planning is harder than deterministic planning.
- In particular, it is EXP-complete in the fully observable case, compared to the PSPACE-completeness of deterministic planning.
- The hardness result already holds if the operators and goals satisfy some fairly strong syntactic restrictions and there is a unique initial state.
- The introduction of nondeterministic effects corresponds to the introduction of alternation in Turing Machines.
- Later, we will see that restricted observability has an even more dramatic effect on the complexity of the planning problem.