Lecture 12: Conditional planning

- stationary plans for planning with full observability
- alternating Turing machines
- Plan existence problem of planning with full observability is EXP-complete.
Memoryless/stationary plans

With full observability there always is a plan of form:

A plan of this form is called a stationary plan, because the choice of operator does not depend on the execution history i.e. what has happened in the plan execution before.

0: CASE
   \( \phi_G \): GOTO done
   \( \phi_1 \): GOTO 1
   \vdots
   \( \phi_n \): GOTO \( n \)
1: \( \phi_1 \): \( o_1 \)
   GOTO 0
   \vdots
n: \( \phi_n \): \( o_n \)
   GOTO 0
done:
Memoryless/stationary plans: plan construction

PROCEDURE FOplanconstruct2
let $S$ be the set of all states with a finite distance;
let $O = \{o_1, \ldots, o_n\}$;
FOR $j := 1$ TO $n$ DO
    $S' :=$ the maximal subset of $S$ such that progress($o_j, S'$);
    $S := S \setminus S'$;
    $l(0) := l(0) \cup \{\langle S', j \rangle\}$;
    $l(j) := \langle o_j, 0 \rangle$;
END
$l(0) := l(0) \cup \{\langle S, -1 \rangle\}; (*$ Goal states $*)
l(-1) := \emptyset$;
Memoryless/stationary plans

- A stationary plan is essentially a function from states to actions.

- When not everything is observable and the current state cannot be determined unambiguously, stationary plans in general are not sufficient.

1. Planning when nothing is observable (next lectures)
2. Planning when only some of the state variables are observable = partial observability.

It is important to remember what has happened before.
Complexity: Alternating TMs

An *alternating Turing machine* is described in terms of

1. an alphabet $\Sigma$ (a set of symbols),
2. a set $Q$ of internal states,
3. a transition function that maps $\langle q, s \rangle$ to a set
   $\{\langle s_1, q_1, m_1 \rangle, \ldots, \langle s_v, q_v, m_v \rangle\}$ of tuples $\langle s', q', m \rangle$ where
   $q, q' \in Q$, $s \in \Sigma \cup \{\|, \square\}$, $s' \in \Sigma$ and $m \in \{L, N, R\}$.
4. an initial state $q_0 \in Q$, and
5. a labeling $g : Q \rightarrow \{\text{accept, reject, } \exists, \forall\}$ of states.
Complexity: ATMs, accepting criterion

A configuration (state + tape contents) with state \( q \in Q \) is accepting if

1. \( g(q) = \text{accept} \),

2. \( g(q) = \forall \) and all successor configurations are accepting, or

3. \( g(q) = \exists \) and at least one successor configuration is accepting.

An ATM accepts if its initial configuration is accepting.
Complexity: ATMs, acceptance criterion
Complexity: ATMs, acceptance criterion
Complexity: ATMs, acceptance criterion
Complexity: ATMs, acceptance criterion
Complexity: Alternating TMs

Alternating TMs generalize both DTMs and NDTMs.

• DTM: any ATM with $|\delta(s, q)| = 1$ for all $s$ and $q$.

• NDTM: any ATM with $g(q) \in \{\exists, \text{accept, reject}\}$ for all $q \in Q$. 
Complexity: APSPACE, AEXPSPACE

$\text{ATIME}(f(n)) = \text{time consumption } O(f(n))$, alternating TM

$\text{ASPACE}(f(n)) = \text{space consumption } O(f(n))$, alternating TM

\[
\begin{align*}
\text{EXP} &= \bigcup_{k \geq 0} \text{DTIME}(2^{n^k}) \\
\text{EXPSPACE} &= \bigcup_{k \geq 0} \text{DSPACE}(2^{n^k}) \\
\text{2-EXP} &= \bigcup_{k \geq 0} \text{DTIME}(2^{2^{n^k}}) \\
\text{APSPACE} &= \bigcup_{k \geq 0} \text{ASPACE}(n^k) \\
\text{AEXPSPACE} &= \bigcup_{k \geq 0} \text{ASPACE}(2^{n^k})
\end{align*}
\]
Complexity: APSPACE, AEXPSPACE

THEOREM. APSPACE = EXP

THEOREM. AEXPSPACE = 2-EXP

See literature references in lecture notes.

We will use alternating Turing machines and APSPACE and AEXPSPACE in proofs of EXP-completeness and 2-EXP-completeness of conditional planning problems.
Complexity of planning with full observability

1. The plan existence problem is APSPACE-hard.

2. \( \text{EXP} = \text{APSPACE} \).

3. The plan existence problem is in EXP.

4. Hence the plan existence problem is EXP-complete.
APSPACE-hardness of planning with FO

Proof idea: extend the PSPACE-hardness proof for deterministic planning.

- ∃ states: one deterministic operator (for one transition) is chosen to the plan.

- ∀ states: one nondeterministic operator handles all transitions.

- ∀ states are followed by branches that choose how ATM execution continues.
Simulation of DTM\(s\) by deterministic planning

PSPACE-hardness proof of deterministic planning

one deterministic operator applicable

one deterministic operator applicable

one deterministic operator applicable

acc

Jussi Rintanen

November 25, AI Planning 16/34
Simulation of NDTMs by deterministic planning

PSPACE=NPSPACE-hardness proof of deterministic planning

two deterministic operators applicable

two deterministic operators applicable

two deterministic operators applicable

two deterministic operators applicable

acc rej rej acc acc acc acc rej
Simulation of ATMs by nondeterministic planning

APSPACE-hardness proof of nondeterministic planning

Jussi Rintanen

November 25, AI Planning  18/34
APSPACE-hardness of planning with FO

Proof utilizes two properties of nondeterministic planning not present in deterministic planning.

- nondeterministic operators
- branching in the plans
APSPACE-hardness of planning with FO

THEOREM. The problem of testing the existence of a plan for problem instances with full observability is EXP-hard.

PROOF: Simulate any ATM with a polynomial space bound $p(n)$ when input length is $n$.

Let $\langle \Sigma, Q, \delta, q_0, g \rangle$ be any alternating Turing machine with space bound $p(x)$. Let $\sigma$ be an input string of length $n$. 
The set $P$ of state variables in the problem instance consists of

1. $q \in Q$ for denoting the internal states of the TM,
2. $s_i$ for every symbol $s \in \Sigma \cup \{|, \square\}$ and tape cell $i \in \{0, \ldots, p(n)\}$, and
3. $h_i$ for the positions of the R/W head $i \in \{0, \ldots, p(n) + 1\}$.
The initial state formula is conjunction of
1. $q_0$
2. $\neg q$ for all $q \in Q \setminus \{q_0\}$.
3. $s_i$ for all $s \in \Sigma$ and $i \in \{1, \ldots, n\}$ s.t. $i$th input symbol is $s$.
4. $\neg s_i$ if $i$th input symbol is not $s$.
5. $\neg s_i$ for all $s \in \Sigma$ and $i \in \{0, n + 1, n + 2, \ldots, p(n)\}$.
6. $\Box_i$ for all $i \in \{n + 1, \ldots, p(n)\}$.
7. $\neg \Box_i$ for all $i \in \{0, \ldots, n\}$.
8. $|_0$
9. $\neg |_i$ for all $n \in \{1, \ldots, p(n)\}$
10. $h_1$
11. $\neg h_i$ for all $i \in \{0, 2, 3, 4, \ldots, p(n) + 1\}$
APSPACE-hardness of planning with FO

The goal is

\[ G = \bigvee \{ q \in Q | g(q) = \text{accept} \}. \]
APSPACE-hardness of planning with FO

For all \( \langle s, q \rangle \in (\Sigma \cup \{ |, \square, \} ) \times Q, \)
\( i \in \{ 0, \ldots, p(n) \}, \)
and \( \langle s', q', m \rangle \in (\Sigma \cup \{ || \} ) \times Q \times \{ L, N, R \} \) define

\[
\tau_{s,q,i}(s', q', m) = \alpha \land \kappa \land \theta
\]

\( \alpha = \text{effect for change in current tape cell} \)

\( \kappa = \text{effect for change in state of ATM} \)

\( \theta = \text{effect for tape movement} \)
APSPACE-hardness of planning with FO, $\alpha$

If $s \in \{|, s'\}$ then $\alpha = \top$.

Otherwise, $\alpha = \neg s_i \land s'_i$.
APSPACE-hardness of planning with FO, $\kappa$

\[ \kappa = \neg q \land q' \text{ if } q \neq q' \]

Otherwise, $\kappa = \top$.

Also $\kappa = \top$ when $i = p(n)$ and $m = R$ (space bound violated, no accepting state may be reached.)
APSPACE-hardness of planning with FO, $\theta$

\[
\theta = \begin{cases} 
- h_i \land h_{i-1} & \text{if } m = L \\
\top & \text{if } m = N \\
- h_i \land h_{i+1} & \text{if } m = R 
\end{cases}
\]

If $h_{p(n)+1}$ becomes true, no operator is applicable.
Let $\langle s, q \rangle \in (\Sigma \cup \{\|, \Box, \} \times Q,$
i $i \in \{0, \ldots, p(n)\}$ and
\begin{align*}
\delta(s, q) &= \{\langle s_1, q_1, m_1 \rangle, \ldots, \langle s_k, q_k, m_k \rangle\}.
\end{align*}

We next define

- one nondeterministic operator for any $s, q, i$ such that $g(q) = \forall$,
- $k$ deterministic operators for any $s, q, i$ such that $g(q) = \exists$. 
APSPACE-hardness of planning with FO

If $g(q) = \exists$, then define $k$ deterministic operators

\[
o_{s,q,i,1} = \langle h_i \land s_i \land q, \tau_{s,q,i}(s_1, q_1, m_1) \rangle \\
o_{s,q,i,2} = \langle h_i \land s_i \land q, \tau_{s,q,i}(s_2, q_2, m_2) \rangle \\
\vdots \\
o_{s,q,i,k} = \langle h_i \land s_i \land q, \tau_{s,q,i}(s_k, q_k, m_k) \rangle
\]

That is, the plan determines which transition is chosen.
APSPACE-hardness of planning with FO

If $g(q) = \forall$, then define one nondeterministic operator

$$o_{s,q,i} = \langle h_i \land s_i \land q, \frac{1}{k} \tau_{s,q,i}(s_1, q_1, m_1)|\frac{1}{k} \tau_{s,q,i}(s_2, q_2, m_2)| \ldots |\frac{1}{k} \tau_{s,q,i}(s_k, q_k, m_k)\rangle$$.

That is, the transition is chosen nondeterministically.
ATM executions and the corresponding plans

- One nondeterministic operator applicable
- Two deterministic operators applicable
- One nondeterministic operator applicable

Decision:
- Acceptance (acc)
- Rejection (rej)
ATM executions and the corresponding plans
APSPACE=EXPTIME membership of nondet. planning with FO

1. The size of the transition system in graph form is exponential in the size of the problem instance.

2. The algorithm for constructing plans with loops runs in polynomial time on the size of the graph representation of a problem instance.

3. \[ \Rightarrow \] Problem is in EXP

Result: the problem is EXP-complete.