Lecture 13: Planning with restricted observability

- Observability
- Full observability, partial observability, unobservability
- Belief states, beliefspace, observational classes
Observability

- Robot can see and hear only the immediate surroundings.

- Poker player cannot see the opponents’ cards.

- Formalization: split state variables to those that can be, and to those that cannot be observed.

- More general formalization: set of atomic and compound formulas; truth-values of these formulas can be observed, but not necessarily the truth-values of their subformulas.
Problem definition

A 5-tuple \( \langle P, I, O, G, B \rangle \) consisting of

- a set \( P \) of state variables,
- a propositional formula \( I \) over \( P \),
- a set \( O \) of operators,
- a propositional formula \( G \) over \( P \), and
- a set \( B \subseteq P \) of observable state variables.

is a problem instance in nondeterministic planning with partial observability.
Conditional plans: definition

\( B = \text{the observable state variables}, \ O = \text{the operators}. \) A plan is \( \langle N, b, l \rangle \) where

- \( N \) is a finite set of nodes,
- \( b \in N \) is the initial node,
- \( l : N \rightarrow (O \times N) \cup 2^{\mathcal{L} \times N} \) assigns each node
  - an operator and a successor node \( \langle o, n \rangle \in O \times N \) or
  - a set of conditions and successor nodes \( \langle \phi, n \rangle \),
  where \( n \in N \) and \( \phi \) is a formula over \( B \).
Restrictions on observability

Let \( \langle P, I, O, G, B \rangle \) be a problem instance in conditional planning. If \( P = B \), the problem instance is *fully observable*. If \( B = \emptyset \), the problem instance is *unobservable*. If \( B \neq \emptyset \) and \( P \neq B \), the problem instance is *partially observable*. We also use this term for the general case that includes full observability and unobservability.
Observational classes

- If only $p_1$ and $p_2$ are observable, then distinguishing between states $s$ and $s'$ such that $s(p_1) = s'(p_1)$ and $s(p_2) = s'(p_2)$ is not possible.

- Observability partitions states to classes $S_1 \cup S_2 \cup \cdots \cup S_n$ of indistinguishable states. One class for every valuation of the observable state variables.

$$S = S_1 \cup S_2 \cup \cdots \cup S_n,$$

$$S_i \cap S_j = \emptyset \text{ for any } \{i, j\} \subset \{1, \ldots, n\} \text{ such that } i \neq j.$$
Observational classes: cardinality

Full observability: state space $S = \{s_1, \ldots, s_n\}$ is partitioned to singleton classes $S_1 = \{s_1\}, S_2 = \{s_2\}, \ldots, S_n = \{s_n\}$.

Unobservability: The partition has only one class $S_1 = S$ consisting of all the states.

Partial observability: the number and cardinality of $S_i$ may be anywhere between 1 and $n$. 

Jussi Rintanen

November 27, AI Planning 7/12
Belief states and belief space

- When there is no full observability, during plan execution only a set of possible current states is known.
- A set of states is a belief state.
- The set of all belief states is the belief space.

Current belief state is $F$.
Operator $o$ is executed: $E = \text{img}_o(F)$.
Observations are made: they select observational class $S_j$.
New belief state is $F' = E \cap S_j$. 
Unobservable planning

• Like fully observable planning, a simple special case of general partially observable planning.

• Practical applications: less than for FO planning because using sensors is usually possible and has a big pay-off.

Exception: finding synchronization/reset sequences of synchronous circuits (but this is not AI planning...)

Jussi Rintanen
Unobservable planning: plans

Because set $B$ of observable state variables is empty, branching is possible only on formulae equivalent to $\top$ or $\bot$: not useful.

Plans are finite sequences $o_1, \ldots, o_n$ of operators $o_i \in O$. 
Belief space: example

Belief space generated by states over two Boolean state variables.

\( n = 2 \) state variables, \( 2^n = 4 \) states, \( 2^{2n} = 16 \) states

red action: complement the value of the first state variable

blue action: randomly set the second state variable
Belief space: example, cont’d

The diagram illustrates the belief space with the following sets:

- {00}
- {00,01}
- {11}
- {00,01,10}
- {00,01,11}
- {00,10,11}
- {01,10}
- {01,11}
- {00,10,11}
- {00,11}
- {01,01,10}
- {01,10,11}
- {01,11}
- {00,01,10,11}
- {10,11}
- {10}
- {}