Symbolic breadth-first planning algorithms

symbolic ~ logical/formula-based

1. Breadth-first traversal of the state space (forward or backward)
   = computation of exact distances of (all) states

2. Sets of states and transition relations are formulae.

3. Implementation typically with binary decision diagrams BDDs.

A symbolic bread-first planning algorithm

Compute sets of states reachable in \( i \) time steps from \( I \) and test whether \( G \) intersects these sets:

\[
\begin{align*}
\epsilon & := \bigwedge \{p | p \in P, I(p) = 1\} \cup \{\neg p | p \in P, I(p) = 0\}; \\
D_0 & := \epsilon; i := 0; \\
\text{REPEAT} & \\
& \quad i := i + 1; \\
& \quad D_i := D_{i-1} \lor (\exists P. (D_{i-1} \land R(P, P'))) [p_1/p_1', p_2/p_2', \ldots, p_n/p_n']); \\
\text{UNTIL} & \\
& \quad D_i \equiv D_i \lor G \in \text{SAT}; \\
\text{IF} & \\
& \quad D_i \land G \in \text{SAT} \text{ THEN plan exists;}
\end{align*}
\]

Image of states w.r.t. an operator/relation

The image of a set \( S \) of states w.r.t. a transition relation \( R \):

\[
\text{img}_R(S) = \{s' | s \in S, \langle s, s' \rangle \in R\}
\]

Computation in the propositional logic:

\[
\text{img}_{R(P, P')}(\phi) = (\exists P. (\phi \land R(P, P'))) [p_1/p_1', p_2/p_2', \ldots, p_n/p_n']
\]

Pre-image of states w.r.t. an operator/relation

The (weak) preimage of a set \( S \) of states w.r.t. a transition relation \( R \):

\[
\text{wpreimg}_R(S) = \{s' | s \in S, \langle s, s' \rangle \in R\}
\]

Computation in the propositional logic:

\[
\text{wpreimg}_{R(P, P')}(\phi) = \exists P'. (\phi[p_1/p_1', p_2/p_2', \ldots, p_n/p_n'] \land R(P, P'))
\]
Preimages as matrix multiplication

Images = products $S_{1 \times n} \times M_{n \times n}$
Preimages = product $M_{n \times n} \times (S_{1 \times n})^T$

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

The states $\{1, 3\}$ are reachable from the states $\{2, 3\}$.

Extraction of plans from exact distances

$$G_j := D_j \land G;$$
FOR $j := i - 1$ DOWN TO 0
  FOREACH $o \in \Omega$
    IF $\text{wpreimg}_{r_o}(G_{j+1}) \land D_j \in \text{SAT}$
    THEN GOTO operatorok;
  END DO
operatorok:
  output $o$;
  $G_j := \text{wpreimg}_{r_o}(G_{j+1}) \land D_j$;
END FOR

Preimages vs. regression

Let $\mathcal{R}(P, P')$ be the translation of $\phi$ to the propositional logic.
Then $\text{wpreimg}_{\mathcal{R}(P, P')}(\phi) \equiv \text{regr}_o(\phi)$.

1. Regression = computation of preimages for deterministic operators directly, without existential abstraction.
2. Progression (image computation) for formulae without existential abstraction? Does not seem to exist: value of a state variable at $t$ cannot be expressed in terms of state variables at $t + 1$.

Preimages vs. regression: an example

$$o = \langle C, A \land (A \Rightarrow B) \rangle$$
$$\text{regr}_o(A \land B) = C \land (\top \land (B \lor A)) \equiv C \land (B \lor A)$$
$$\tau_o = C \land A' \land ((B \lor A) \leftrightarrow B') \land (C \leftrightarrow C')$$

The preimage of $A \lor B$ with respect to $o$ is represented by

$$\exists A'B'C'.((A' \land B') \land \tau_o) \equiv \exists A'B'C'.(A' \land B' \land C \land (B \lor A) \land C') \equiv \exists B'C'.(B' \land C \land (B \lor A) \land C') \equiv \exists C'.(C \land (B \lor A) \land C') \equiv C \land (B \lor A)$$
(Ordered) Binary decision diagrams (OBDDs)

3-place connective if-then-else \((p)\) is a proposition):

\[ \text{ite}(p, \phi_1, \phi_2) = (p \land \phi_1) \lor (\neg p \land \phi_2) \]

Shannon expansion:

\[ \phi \equiv (p \land \phi[T/p]) \lor (\neg p \land \phi[\bot/p]) = \text{ite}(p, \phi[T/p], \phi[\bot/p]) \]

Binary decision diagrams: example

Construct OBDD with variable ordering \(A, B, C\) by repeated Shannon expansion:

\[(A \lor B) \land (B \lor C)\]
\[\equiv \text{ite}(A, (\top \lor B) \land (B \lor C), (\bot \lor B) \land (B \lor C))\]
\[\equiv \text{ite}(A, B \lor C, B)\]
\[\equiv \text{ite}(A, \text{ite}(B, \top \lor C, \bot \lor C), \text{ite}(B, \top, \bot))\]
\[\equiv \text{ite}(A, \text{ite}(B, \top, \text{ite}(C, \top, \bot)), \text{ite}(B, \top, \bot))\]

Satisfiability algorithms vs. OBDDs

<table>
<thead>
<tr>
<th>algorithm</th>
<th>size of (R_1(P, P'))</th>
<th>runtime vs. plan length (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>satisfiability</td>
<td>not a problem</td>
<td>exponential on (n)</td>
</tr>
<tr>
<td>planning OBDDs</td>
<td>major problem</td>
<td>much less dependent on (n)</td>
</tr>
<tr>
<td>algorithm</td>
<td>critical resource</td>
<td></td>
</tr>
<tr>
<td>satisfiability</td>
<td>runtime</td>
<td></td>
</tr>
<tr>
<td>planning OBDDs</td>
<td>memory</td>
<td></td>
</tr>
<tr>
<td>algorithm</td>
<td>types of problems</td>
<td></td>
</tr>
<tr>
<td>satisfiability</td>
<td>lots of state variables, short plans</td>
<td></td>
</tr>
<tr>
<td>planning OBDDs</td>
<td>fewer state variables, long plans</td>
<td></td>
</tr>
</tbody>
</table>
Our roadmap (almost) half-way through the course

<table>
<thead>
<tr>
<th>form of planning</th>
<th>actions</th>
<th>initial states</th>
<th>observability</th>
</tr>
</thead>
<tbody>
<tr>
<td>classical (determ.)</td>
<td>deterministic</td>
<td>one</td>
<td>-</td>
</tr>
<tr>
<td>conditional</td>
<td>nondeterministic</td>
<td>several</td>
<td>full</td>
</tr>
<tr>
<td>probabilistic</td>
<td>nondeterministic</td>
<td>several</td>
<td>full</td>
</tr>
<tr>
<td>conditional</td>
<td>nondeterministic</td>
<td>several</td>
<td>partial</td>
</tr>
<tr>
<td>probabilistic</td>
<td>nondeterministic</td>
<td>several</td>
<td>partial</td>
</tr>
</tbody>
</table>

Nondeterminism

- The world cannot be completely modeled: we do not know what is going to happen next (missing information, even in problems that would otherwise be characterized completely deterministic.)
- Things just go wrong (and we might know everything about it!)
- Games: roulette, dice, chess (= opponent unpredictable!), ...

Nondeterministic actions

- Actions are **not** (partial) functions from states to states.
- Actions are binary relations on states. OR (equivalently)
- Actions are (partial) functions from states to sets of states. OR
- Actions are (partial) functions from states to probability distributions on the set of all states.
Nondeterministic actions as propositional formulae

1. Any Boolean (= 0, 1) matrix represents a nondeterministic action.

2. Any propositional formula on $P \cup P'$ represents a nondeterministic action.

3. Images and preimages can be computed with existential abstraction just like for formulae that represent deterministic actions.