## Symbolic breadth-first planning algorithms

symbolic $\sim$ 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 $\leq i$ time steps from $I$ and test whether $G$ intersects these sets:

$\iota$ := $\bigwedge\{p|p \in P, I(p) = 1\} \cup \{\neg p|p \in P, I(p) = 0\}$;
$D_0$ := $\iota$; $i$ := 0;
REPEAT
  $i$ := $i + 1$;
  $D_i$ := $D_{i-1} \vee ((\exists P.(D_{i-1} \wedge \mathcal{R}_1(P, P')))[p_1/p'_1, p_2/p'_2, \ldots, p_n/p'_n])$;
UNTIL $D_{i-1} \equiv D_i$ OR $D_i \wedge G \in$ SAT;
IF $D_i \wedge G \in$ SAT THEN plan exists;



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

*The image* of a set $S$ of states w.r.t. a transition relation $R$:

$$\textit{img}_R(S) = \{s'|s \in S, \langle s, s'\rangle \in R\}$$

Computation in the propositional logic:

$$\textit{img}_{\mathcal{R}(P,P')}(\phi) = (\exists P.(\phi \wedge \mathcal{R}(P, P')))[p_1/p'_1, \ldots, p_n/p'_n]$$



## 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 $R$:

$$\textit{wpreimg}_R(S) = \{s|s' \in S, \langle s, s'\rangle \in R\}$$

Computation in the propositional logic:

$$\textit{wpreimg}_{\mathcal{R}(P,P')}(\phi) = \exists P'.(\phi[p'_1/p_1, \ldots, p'_n/p_n] \wedge \mathcal{R}(P, P'))$$

## Preimages as matrix multiplication

Images = products $S_{1\times n} \times M_{n\times n}$

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

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

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



## Extraction of plans from exact distances

$G_i$ := $D_i \wedge G$;
FOR $j$ := $i - 1$ DOWN TO 0
  FOREACH $o \in O$ DO
    IF *wpreimg*$_{\tau_o}(G_{j+1}) \wedge D_j \in$ SAT
    THEN GOTO operatorok;
  END DO
operatorok:
  output $o$;
  $G_j$ := *wpreimg*$_{\tau_o}(G_{j+1}) \wedge D_j$;
END FOR



## Preimages vs. regression

Let $\mathcal{R}(P, P')$ be the translation of $o$ to the propositional logic.

Then *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

$$\begin{aligned} o &= \langle C, A \wedge (A \rhd B)\rangle \\ \text{regr}_o(A \wedge B) &= C \wedge (\top \wedge (B \vee A)) \equiv C \wedge (B \vee A) \\ \tau_o &= C \wedge A' \wedge ((B \vee A) \leftrightarrow B') \wedge (C \leftrightarrow C') \end{aligned}$$

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

$$\begin{aligned} \exists A'B'C'.((A' \wedge B') \wedge \tau_o) &\equiv \exists A'B'C'.(A' \wedge B' \wedge C \wedge (B \vee A) \wedge C') \\ &\equiv \exists B'C'.(B' \wedge C \wedge (B \vee A) \wedge C') \\ &\equiv \exists C'.(C \wedge (B \vee A) \wedge C') \\ &\equiv C \wedge (B \vee A) \end{aligned}$$

# (Ordered) Binary decision diagrams (OBDDs)

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

$$\text{ite}(p, \phi_1, \phi_2) = (p \wedge \phi_1) \vee (\neg p \wedge \phi_2)$$

Shannon expansion:

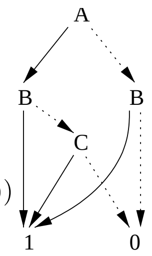
$$\phi \equiv (p \wedge \phi[\top/p]) \vee (\neg p \wedge \phi[\bot/p]) = \text{ite}(p, \phi[\top/p], \phi[\bot/p])$$



# Binary decision diagrams: example

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

$$
\begin{aligned}
& (A \vee B) \wedge (B \vee C) \\
\equiv\ & ite(A, (\top \vee B) \wedge (B \vee C), (\bot \vee B) \wedge (B \vee C)) \\
\equiv\ & ite(A, B \vee C, B) \\
\equiv\ & ite(A, ite(B, \top \vee C, \bot \vee C), ite(B, \top, \bot)) \\
\equiv\ & ite(A, ite(B, \top, C), ite(B, \top, \bot)) \\
\equiv\ & ite(A, ite(B, \top, ite(C, \top, \bot)), ite(B, \top, \bot))
\end{aligned}
$$






# Satisfiability algorithms vs. OBDDs

| algorithm | size of $\mathcal{R}_1(P, P')$ | runtime vs. plan length $n$ |
|---|---|---|
| satisfiability planning | not a problem | exponential on $n$ |
| OBDDs | major problem | much less dependent on $n$ |

| algorithm | critical resource |
|---|---|
| satisfiability planning | runtime |
| OBDDs | memory |

| algorithm | types of problems |
|---|---|
| satisfiability planning | lots of state variables, short plans |
| OBDDs | fewer state variables, long plans |



# Normal forms for propositional formulae

| | $\vee$ | $\wedge$ | $\neg$ | $\phi \in$TAUT? | $\phi \in$SAT? | $\phi \equiv \phi'$? |
|---|---|---|---|---|---|---|
| circuits | poly | poly | poly | co-NP-hard | NP-hard | co-NP-hard |
| formulae | poly | poly | poly | co-NP-hard | NP-hard | co-NP-hard |
| DNF | poly | exp | exp | co-NP-hard | in P | co-NP-hard |
| CNF | exp | poly | exp | in P | NP-hard | co-NP-hard |
| OBDD | exp | exp | poly | in P | in P | in P |
| DNNF | poly | exp | exp | co-NP-hard | in P | co-NP-hard |

## Our roadmap (almost) half-way through the course

| form of planning | actions | initial states | observability |
|---|---|---|---|
| classical (determ.) | deterministic | one | - |
| conditional | nondeterministic | several | full |
| probabilistic | nondeterministic | several | full |
| conditional | nondeterministic | several | partial |
| probabilistic | nondeterministic | several | partial |



## Nondeterminism

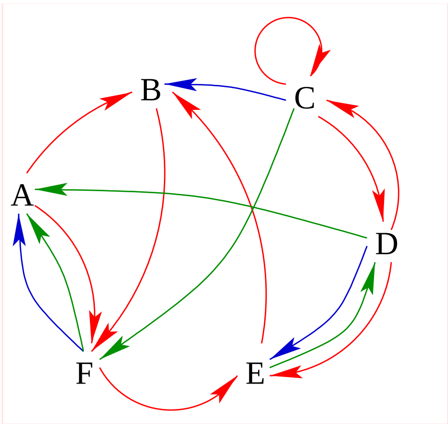
- The world cannot be completely modeled: we do not know what is going to happen next (missing information, even in problems that would otherwise be characterized completely deterministic.)
- Things just go wrong (and we might know everything about it!)
- Games: roulette, dice, chess (= opponent unpredictable!),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

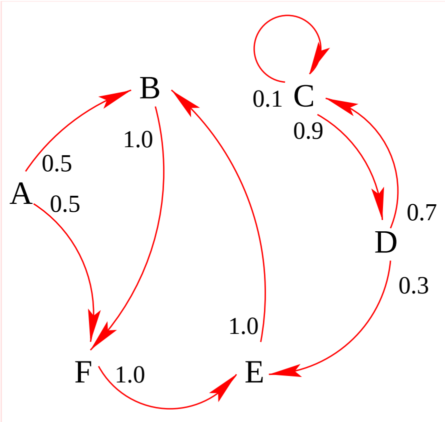


## Nondeterministic actions

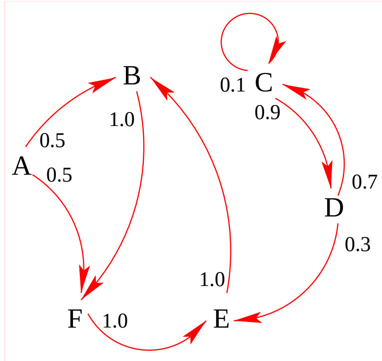
- Actions are not (partial) functions from states to states.
- Actions are binary relations on states. OR (equivalently)
- Actions are (partial) functions from states to sets of states. OR
- Actions are (partial) functions from states to probability distributions on the set of all states.





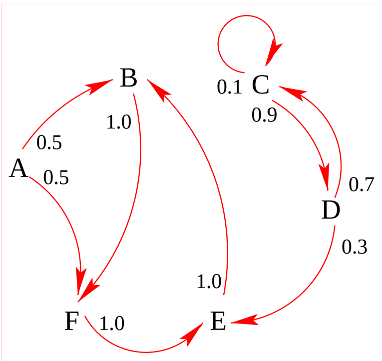





| | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $A$ | 0 | 0.5 | 0 | 0 | 0 | 0.5 |
| $B$ | 0 | 0 | 0 | 0 | 0 | 1.0 |
| $C$ | 0 | 0 | 0.1 | 0.9 | 0 | 0 |
| $D$ | 0 | 0 | 0.7 | 0 | 0.3 | 0 |
| $E$ | 0 | 1.0 | 0 | 0 | 0 | 0 |
| $F$ | 0 | 0 | 0 | 0 | 1.0 | 0 |






| | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $A$ | 0 | 1 | 0 | 0 | 0 | 1 |
| $B$ | 0 | 0 | 0 | 0 | 0 | 1 |
| $C$ | 0 | 0 | 1 | 1 | 0 | 0 |
| $D$ | 0 | 0 | 1 | 0 | 1 | 0 |
| $E$ | 0 | 1 | 0 | 0 | 0 | 0 |
| $F$ | 0 | 0 | 0 | 0 | 1 | 0 |



## 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.