# Nondeterministic planning (May 25, 2005)







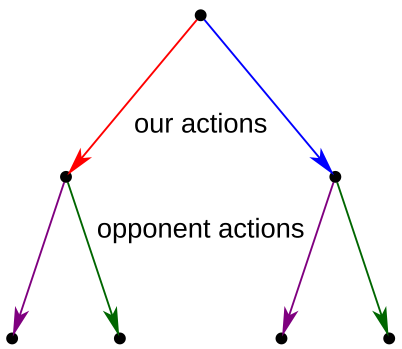
## Nondeterministic/conditional planning
### Motivation

- World is not predictable.
- AI robotics:
  - imprecise movement of the robot
  - other robots
  - human beings, animals
  - machines (cars, trains, airplanes, lawn-mowers, ...)
  - natural phenomena (wind, water, snow, temperature, ...)
- Games: other players are outside our control.
- To win a game (reaching a goal state) with certainty all possible actions by the other players have to be anticipated (a winning strategy of a game).
- World is not predictable because it is unknown: we cannot observe everything.





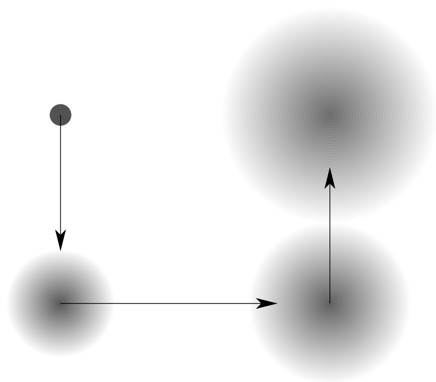
## Nondeterminism
### Example: several agents, games








## Nondeterminism
### Example: uncertainty in robot movement







## Nondeterministic/conditional planning
### Motivation

- In deterministic planning we have assumed that the only changes taking place in the world are those caused by us and that we can exactly predict the results of our actions.
- Other agents and processes, beyond our control, are formalized as nondeterminism.
- Implications:
  1. The future state of the world cannot be predicted.
  2. We cannot reliably plan ahead: no single action sequence achieves the goals.
  3. In some cases it is not possible to achieve the goals with certainty, only with some probability.





## Transition systems
### General definition with nondeterminism and observability

**Definition**
A transition system is a 5-tuple $\Pi = \langle S, I, O, G, P \rangle$ where
1. $S$ is a finite set of states,
2. $I \subseteq S$ is the set of initial states,
3. $O$ is a finite set of actions $o \subseteq S \times S$,
4. $G \subseteq S$ is the set of goal states, and
5. $P = (C_1, \ldots, C_n)$ is a partition of $S$ to classes of observationally indistinguishable states satisfying $\bigcup\{C_1, \ldots, C_n\} = S$ and $C_i \cap C_j = \emptyset$ for all $i, j$ such that $1 \leq i < j \leq n$.

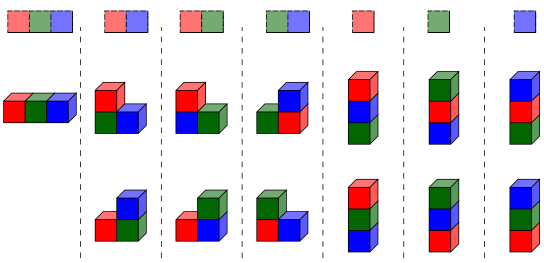
Making an observation tells which set $C_i$ the current state belongs to. Distinguishing states within a given $C_i$ is not possible by observations.





## Observability
### Example of partition of states into observational classes

Blocks world with 3 blocks and camera far above the table. State variables $V = \{Aclear, Bclear, Cclear\}$ are observable.



There are 8 valuations of $V$ but the valuation $v \models \neg Aclear \wedge \neg Bclear \wedge \neg Cclear$ does not correspond to a blocks world state.





## Observability
### Classification full, partial, no observability

Let $S = \{s_1, \ldots, s_n\}$ be the set of states.
Classification of planning problems in terms of observability:

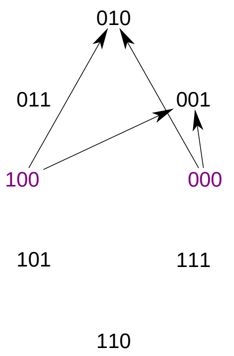
**Full** $P = (\{s_1\}, \{s_2\}, \ldots, \{s_n\})$
number of observational classes: $n$
Chess is a fully observable 2-person game.

**No** $P = (\{s_1, \ldots, s_n\})$
number of observational classes: 1

**Partial** No restrictions on $P$.
number of observational classes : between 1 and $n$
Poker is a partially observable 2-person game.
Mastermind is a partially observable 1-person game.

$n$-person games for $n \geq 2 \sim$ nondeterministic planning

# Nondeterministic actions as operators
## Example




| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 111 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In terms of state variables $A = \{a, b, c\}$ the action can be represented as operator

$$\langle \neg b \wedge \neg c, \neg a \wedge (b|c) \rangle$$





# Nondeterministic actions as operators
## Definition

### Definition
Let $A$ be a set of state variables. An operator is a pair $\langle c, e \rangle$ where $c$ is a propositional formula over $A$ (the precondition), and $e$ is an effect over $A$. Effects over $A$ are recursively defined as follows.

1. $a$ and $\neg a$ for state variables $a \in A$ are effects over $A$.
2. $e_1 \wedge \cdots \wedge e_n$ is an effect over $A$ if $e_1, \ldots, e_n$ are effects over $A$.
3. $c \rhd e$ is an effect over $A$ if $c$ is a formula over $A$ and $e$ is an effect over $A$.
4. $e_1 | \cdots | e_n$ is an effect over $A$ if $e_1, \ldots, e_n$ for $n \geq 2$ are effects over $A$.





# Nondeterministic operators
## Semantics, example

### Example

$$\langle a, (b|\neg b) \wedge (c|\neg c) \wedge (d|\neg d) \rangle$$

has $2^3$ alternative sets of effects, leading to 8 different successor states.

1. effects $\{b, c, d\}$ lead to state $s \models a \wedge b \wedge c \wedge d$
2. effects $\{\neg b, c, d\}$ lead to state $s \models a \wedge \neg b \wedge c \wedge d$
3. effects $\{b, \neg c, d\}$ lead to state $s \models a \wedge b \wedge \neg c \wedge d$
4. effects $\{\neg b, \neg c, d\}$ lead to state $s \models a \wedge \neg b \wedge \neg c \wedge d$
5. effects $\{b, c, \neg d\}$ lead to state $s \models a \wedge b \wedge c \wedge \neg d$
6. effects $\{\neg b, c, \neg d\}$ lead to state $s \models a \wedge \neg b \wedge c \wedge \neg d$
7. effects $\{b, \neg c, \neg d\}$ lead to state $s \models a \wedge b \wedge \neg c \wedge \neg d$
8. effects $\{\neg b, \neg c, \neg d\}$ lead to state $s \models a \wedge \neg b \wedge \neg c \wedge \neg d$





# Nondeterministic operators
## Semantics

### Definition (Operator application)
Let $\langle c, e \rangle$ be an operator over $A$ and $s$ a state.
The set $[e]_s$ of sets of literals is recursively defined as follows.

1. $[a]_s = \{\{a\}\}$ and $[\neg a]_s = \{\{\neg a\}\}$ for $a \in A$.
2. $[e_1 \wedge \cdots \wedge e_n]_s = \{\bigcup_{i=1}^{n} E_i | E_1 \in [e_1]_s, \ldots, E_n \in [e_n]_s\}$.
3. $[c' \rhd e]_s = [e]_s$ if $s \models c'$ and $[c' \rhd e]_s = \{\emptyset\}$ otherwise.
4. $[e_1 | \cdots | e_n]_s = [e_1]_s \cup \cdots \cup [e_n]_s$ .

### Definition
Operator $\langle c, e \rangle$ is applicable in $s$ if $s \models c$ and every set $E \in [e]_s$ is consistent.





# Nondeterministic operators
## Binary relation induced by an operator

### Definition
An operator $\langle c, e \rangle$ induces a binary relation $R\langle c, e \rangle$ on the states as follows: $sR\langle c, e \rangle s'$ if there is $E \in [e]_s$ such that

1. $s \models c$,
2. $s' \models E$, and
3. $s \models a$ iff $s' \models a$ for all $a \in A$ such that $\{a, \neg a\} \cap E = \emptyset$.

We also write simply $sos'$ instead of $sR(o)s'$.

### Definition
Let $s$ and $s'$ be states and $o$ an operator. If $sos'$ then $s'$ is a successor state of $s$.





# Succinct transition systems
## General definition

### Definition
A succinct transition system is a 5-tuple $\Pi = \langle A, I, O, G, V \rangle$ where

1. $A$ is a finite set of state variables,
2. $I$ is a formula over $A$ describing the initial states,
3. $O$ is a finite set of operators over $A$,
4. $G$ is a formula over $A$ describing the goal states, and
5. $V \subseteq A$ is the set of observable state variables.





# Succinct transition systems
## Observability

Let $A = \{a_1, \ldots, a_n\}$ be the state variables.
Classification of planning problems in terms of observability:

- **Full** observable state variables: $V = A$
  number of observational classes: $2^{|A|}$
- **No** observable state variables: $V = \emptyset$
  number of observational classes: 1
- **Partial** observable state variables: no restrictions, $\emptyset \subseteq V \subseteq A$
  number of observational classes: 1 to $2^{|A|}$





# Succinct transition system
## Translation into transition systems

We can associate a transition system with every succinct transition system.

### Definition
Given a succinct transition system $\Pi = \langle A, I, O, G, V \rangle$, construct the transition system $F(\Pi) = \langle S, I', O', G', P \rangle$ where

1. $S$ is the set of all Boolean valuations of $A$,
2. $I' = \{s \in S | s \models I\}$,
3. $O' = \{R(o) | o \in O\}$,
4. $G' = \{s \in S | s \models G\}$, and
5. $P = (C_1, \ldots, C_n)$ where $v_1, \ldots, v_n$ for $n = 2^{|V|}$ are all the Boolean valuations of $V$ and $C_i = \{s \in S | s(a) = v_i(a) \text{ for all } a \in V\}$ for all $i \in \{1, \ldots, n\}$.