Non Classical Planning

Lecture 4

Outline

- ♢ Planning in the real world
- ♢ Belief states sensing and non deterministi actions (RN 4.3)
- ♢ Search in Partially observable environments (RN 4.4)
- ♢ Belief states planning representation (RN 11.5,11.5.1,11.5.2)
- ♢ Monitoring and replanning (RN 11.5.3)

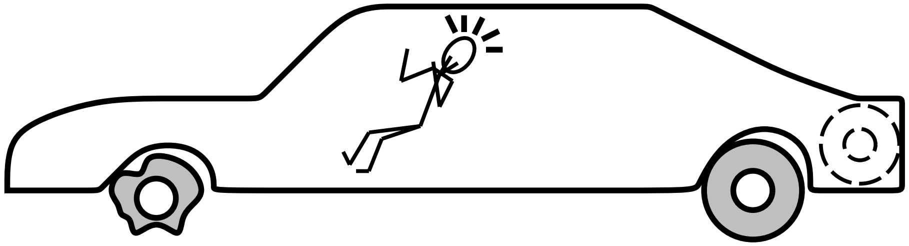
The real world

Classical search/planning not adequate: agents have to deal with **incomplete** and **incorrect** information

bounded indeterminacy (e.g. tossing a coin): the agent can deal with all possible cases

unbounded indeterminacy: no hope that the agent can deal with all possible cases The agent must be ready to revise its plan and/or its knowledge base

The real world



On(x) ~Flat(x)

START

~Flat(Spare) Intact(Spare) Off(Spare)
On(Tire1) Flat(Tire1)

FINISH

On(x)

Remove(x)

Off(x) ClearHub

Off(x) ClearHub

Puton(x)

On(x) ~ClearHub

Intact(x) Flat(x)

Inflate(x)

~Flat(x)

Things go wrong

Incomplete information

Unknown preconditions, e.g., $Intact(Spare)$?

Disjunctive effects, e.g., $Inflate(x)$ causes

$Inflated(x) \lor SlowHiss(x) \lor Burst(x) \lor BrokenPump \lor$

...

Qualification problem:

can never finish listing all the required preconditions
and possible conditional outcomes of actions

Incorrect information

Current state incorrect, e.g., spare NOT intact

$Intact(Spare)$: unknown, missing, wrongly perceived

Uncertainty in the state

Due to:

- ♢ non determinism
- ♢ partial observability
- ♢ both

Search in the space of **belief states** (sets of possible actual states)

Agent does not whether in s_1 or s_2 is represented by the belief set: $\{s_1, s_2\}$

Solutions

Conformant or sensorless planning

Devise a plan that works regardless of state

Not always these plans exist

Replace tire if can not observe whether flat

(The difference between search and planning is only about the state representation)

Solutions cntd

Conditional planning or contingency planning

Plan to obtain information (**observation actions**)

Subplan for each contingency, e.g.,

$[Check(Tire1), \textbf{if}\ Intact(Tire1)\ \textbf{then}\ Inflate(Tire1)$
$\textbf{else}\ CallAAA$

Expensive because it takes into account many unlikely cases

Solutions cntd

Monitoring/Replanning

Assume normal states, outcomes

Check progress *during execution*, replan if necessary

Unanticipated outcomes may lead to failure (e.g., no AAA card)

Really need a combination; plan for likely/serious eventualities, deal with others when they arise, as they must eventually.

Solutions cntd

Continuous Planning

Goal can change
It deals with goal formulation, planning and acting phases

A partial order planner can be suitably modified to embody:

- ♢ check the progress of the plan based on environment observation
- ♢ update the goal

At each step the current plan is adjusted based on the updated scenario and the next action is selected.

Another Example: painting

Chair, table and several cans of paint.
Our agent, who does not know the initial colours, has to make chair and table of the same color.

ACTION: $RemoveLid(can)$
PRECONDITION: $Can(can)$
EFFECT: $Open(can)$

ACTION: $Paint(x, can)$
PRECONDITION: $Object(x), Can(can), Color(can, c), Open(can)$
EFFECT: $Color(x, c)$

Perception Action (painting)

Our agent can perceive the color of an open can.
And, it can switch view between table and chair.

ACTION: $Percept(Color(can, c))$
PRECONDITION: $Can(can), InView(can), Open(can)$

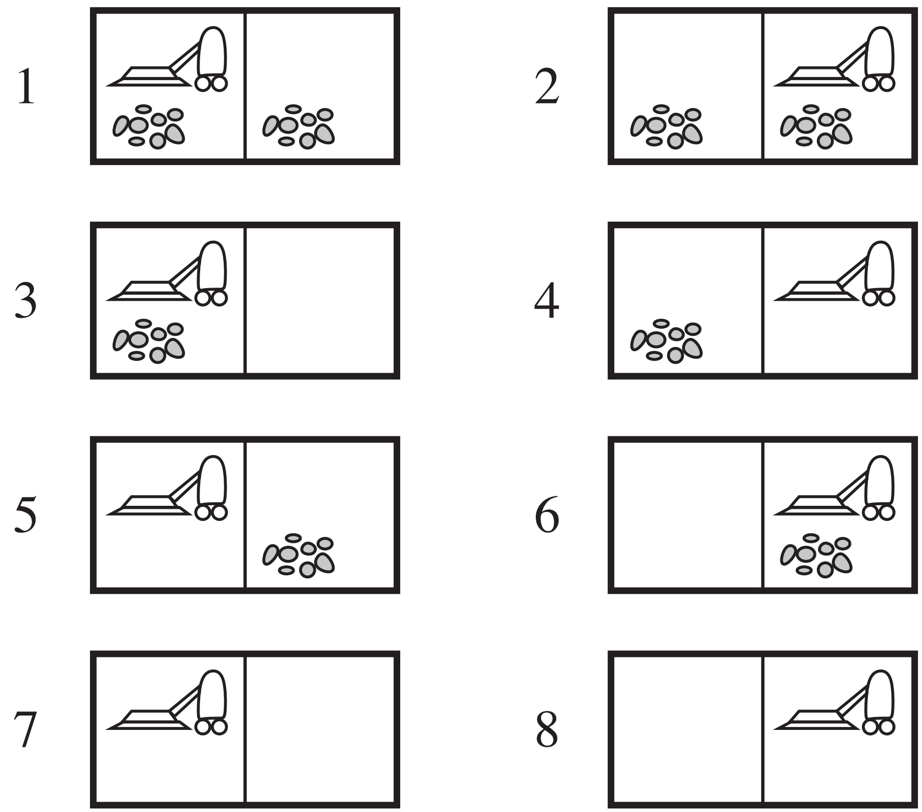
ACTION: $LookAt(x)$
PRECONDITION: $InView(y), y \neq x$
EFFECT: $InView(y), \neg InView(x)$

Possible approaches

♢ **classical**: not doable, incomplete info in initial state
♢ **sensorless**: take any can and paint both
♢ **conditional**: sense the color of table and chair, if they are equal ...
♢ **replanning**: can check whether the painting was effective, and, look for another color if necessary ...
♢ **continous**: can cope with new goals, constrains (e.g. I need the table for a dinner, so postpone painting it)

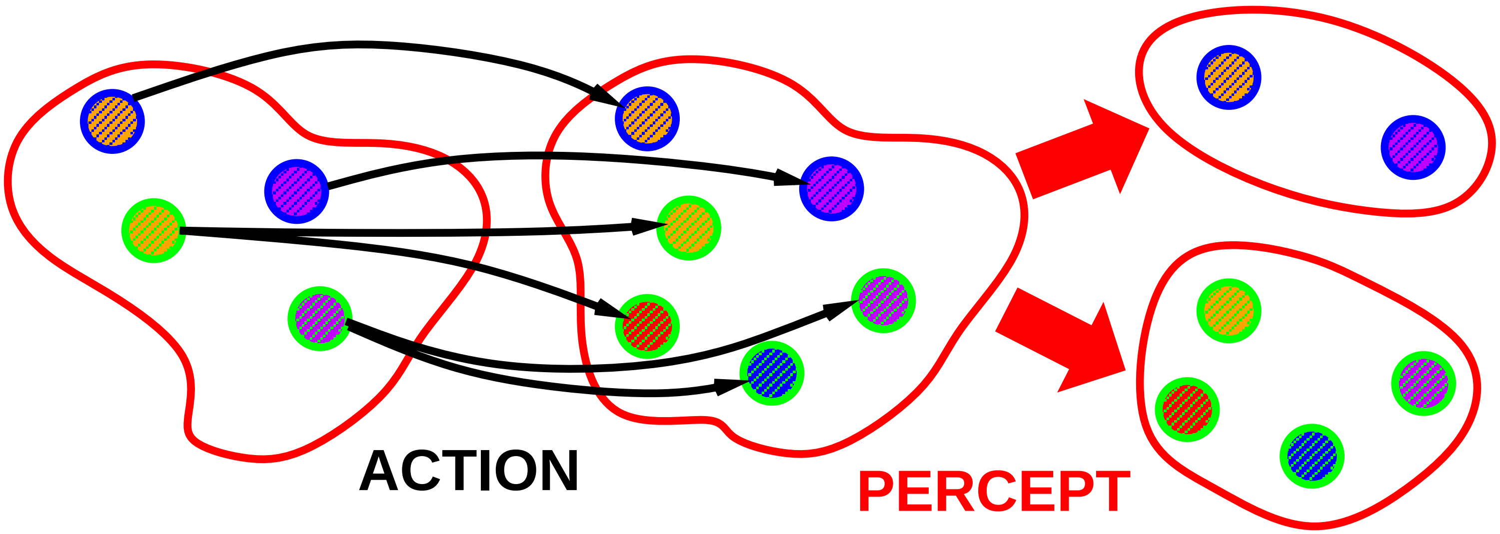
Belief states

◊ **Belief states** represent all possible world states:
Belief states: sets of *ordinary* states



Evolving Belief states

Actions determine the transition from belief states to other belief states



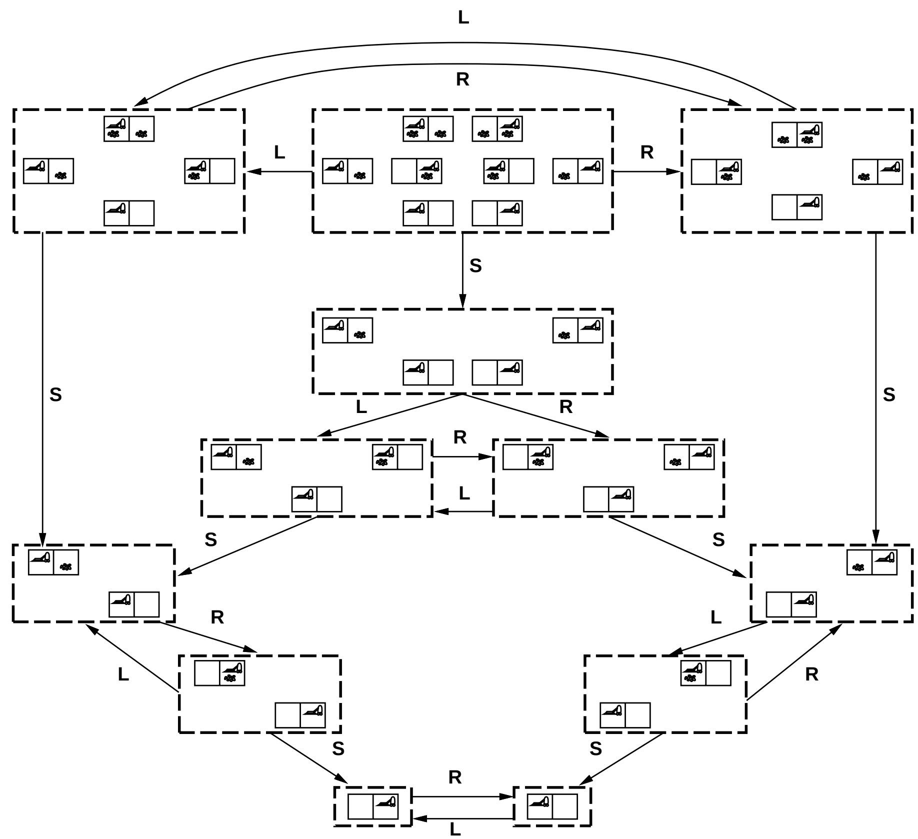
Terminology

- **Belief states**: set of 2^N set of states
- **Initial state**: a set of possible states
- **Actions**: all applicable actions (Deterministic and NON)
- **Successor state**:

$$b' = Result(b, a) = \{s' : s' = Result_P(s, a) \ and \ s \in b\}$$

- **Goal Test**: goal condition satisfied in every state of the belief set

Sensorless planning (cleaning)



Starting with complete uncertainty $Right, Suck, Left, Suck$ achieves the goal.

Sensorless planning (painting)

Starting with the following initial state:

$$Object(table), Object(Chair), Can(Can_1), Can(Can_2)$$

and the goal

$$Color(table, c), Color(Chair, c)$$

The plan:

$$RemoveLid[Can_1],$$
$$Paint[Chair, Can_1],$$
$$Paint[Table, Can_1]$$

achieves the goal.

Contingent planning

The state is only partially observable.
a **conditional** plan that checks percepts is needed:

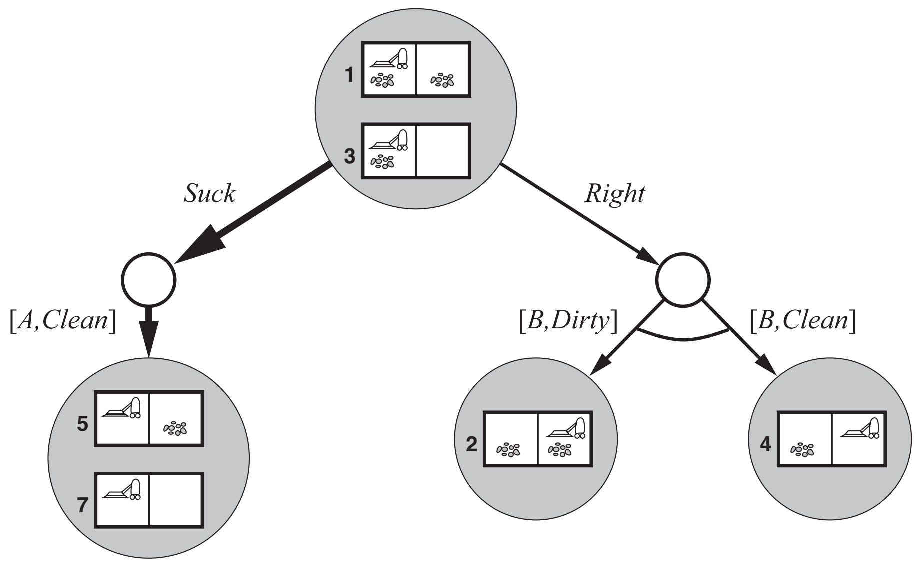
$$[\ldots, \textbf{if } C \textbf{ then } Plan_A \textbf{ else } Plan_B, \ldots]$$

Execution:
check C against current state, execute "then" or "else"

Remark 1: a plan must achieve the goal for *every* possible percept.
Remark 2: belief states allow to model sensing actions to check the status some of the state properties.

Contingent planning (cleaning)

Example: the agent can only check for dirt the current location (PO).



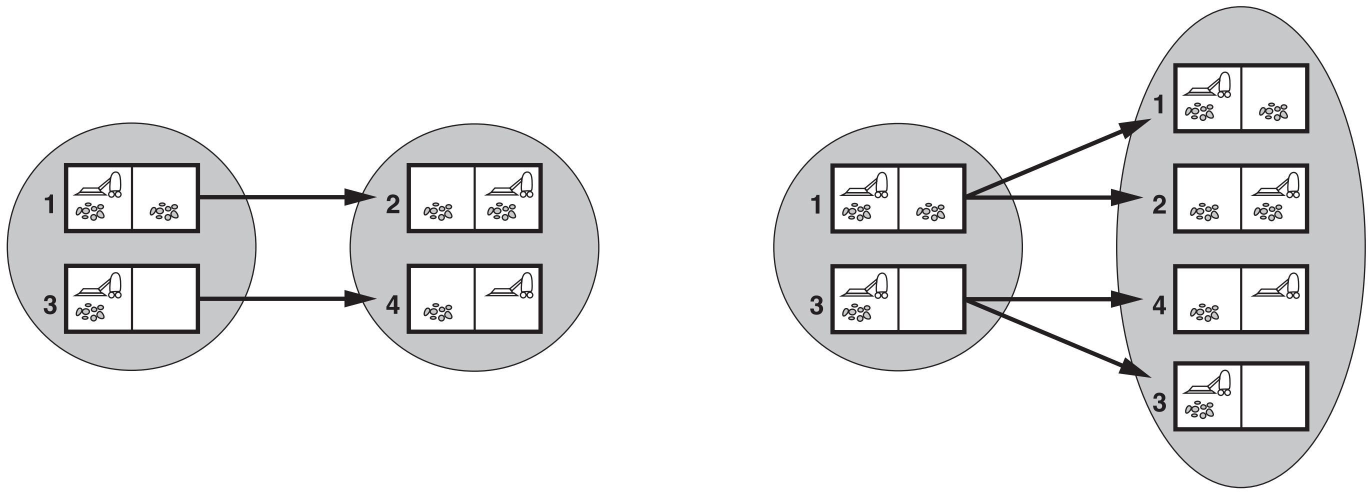
$$[Suck, Right, \textbf{if} \neg CleanR \textbf{ then } Suck, \textbf{else} No - op \ldots]$$

Contingent planning (painting)

$$
\begin{array}{l}
[LookAt(Table), LookAt(Chair), \\
\quad \textbf{if } Color(Table, c) \wedge Color(Chair, c) \textbf{ then } NoOp \\
\qquad \textbf{else } [RemoveLid(Can_1), LookAt(Can_1), RemoveLid(Can_2), LookAt(Can_2), \\
\qquad\qquad \textbf{if } Color(Table, c) \wedge Color(can, c) \textbf{ then } Paint(Chair, can) \\
\qquad\qquad \textbf{else if } Color(Chair, c) \wedge Color(can, c) \textbf{ then } Paint(Table, can) \\
\qquad\qquad \textbf{else } [Paint(Chair, Can_1), Paint(Table, Can_1)]]]
\end{array}
$$

Using perception actions the agent can do a much better job!

Actions in Belief sets



(a) (b)

(a) deterministic actions

(b) non deterministic actions (may increase the uncertainty)

Also with non deterministic actions the environment can be **fully or partially** observable

Non deterministic actions (with FO)

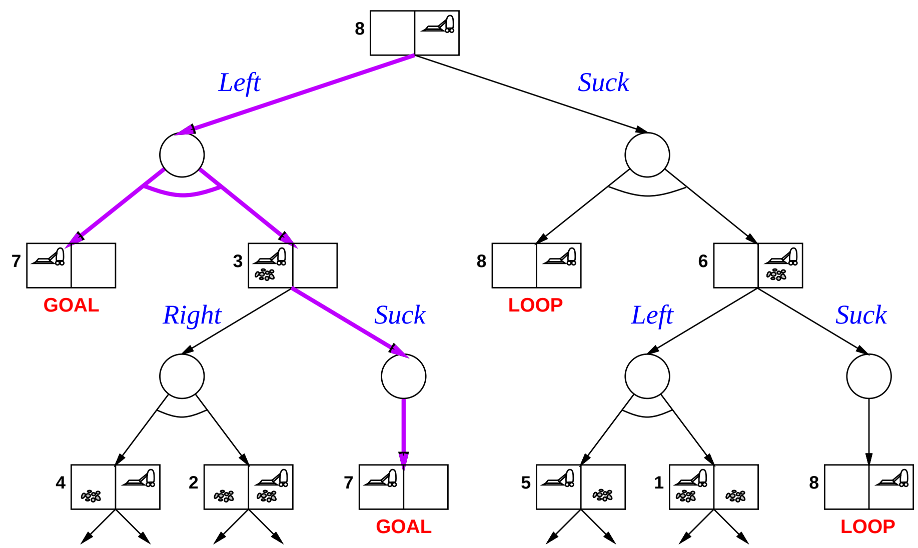
♢ Nondeterministic actions as **disjunctive effects**
i.e. $Suck$ on a clean square may leave a dirty square ...

♢ Percepts needed to determine the value of a property
i.e. check whether the current location is clean

AND–OR tree search: Each non deterministic action is followed by a sensing action, which eliminates the uncertainty caused by action execution.

Example

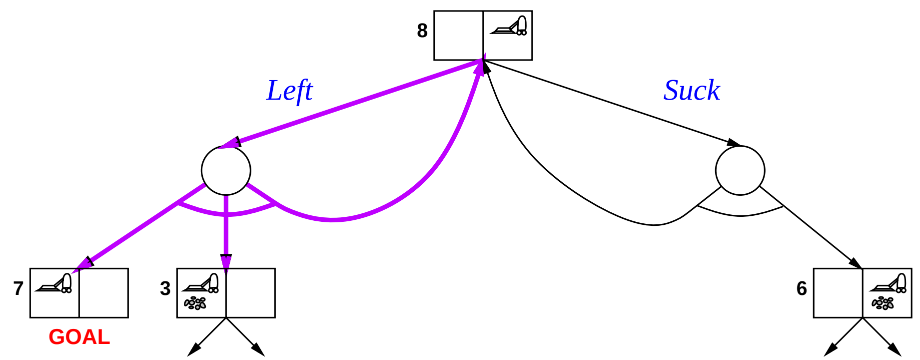
NDA (+ DFO): sucking or arriving may dirty a clean square



$$Left, \textbf{if } AtR \land CleanL \land CleanR \textbf{ then } [] \textbf{ else } Suck$$

Example

More NDA: also sometimes fails to move (slips)



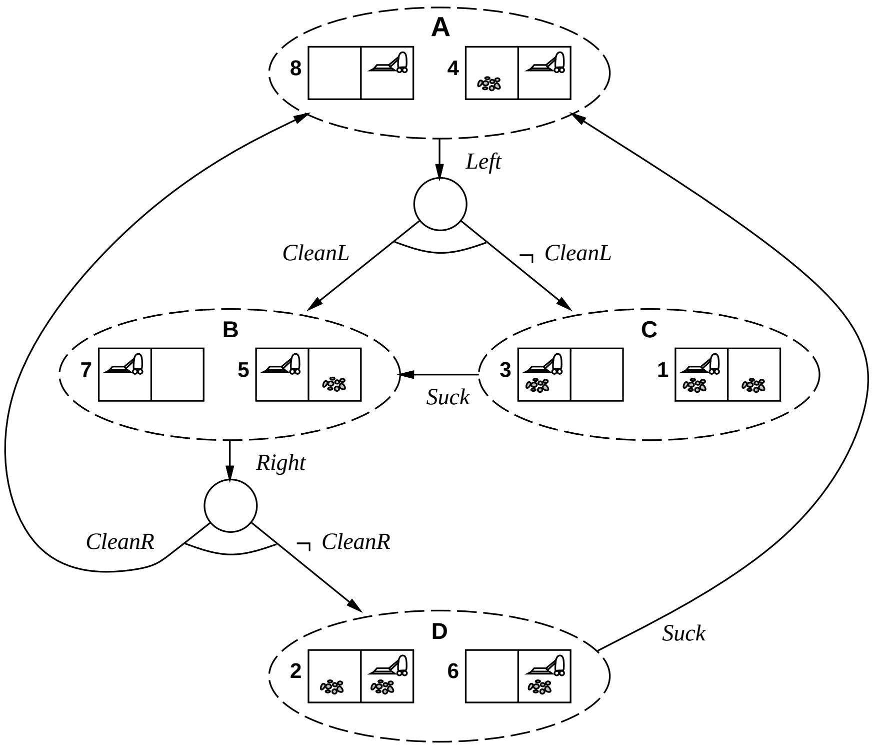
$$[L_1 : \ Left, \textbf{if } AtR \textbf{ then } L_1 \textbf{ else } [\textbf{if } CleanL \textbf{ then } [\,] \textbf{ else } Suck]]$$

or $[\textbf{while } AtR \textbf{ do } [Left], \textbf{if } CleanL \textbf{ then } [\,] \textbf{ else } Suck]$

With a "loop", plan may not terminate, but will eventually work unless action always fails!

Example

Sometimes a plan cannot be obtained: moving may dirty a clean square



Belief state representation for planning

Open world assumption:
efficient representation unknown properties

Example: AtR

$$\{AtR \land \neg AtL \land CleanR \land CleanL,$$
$$AtR \land \neg AtL \land \neg CleanR \land CleanL,$$
$$\ldots,$$
$$\}$$

Compact Representation of Belief states

The belief state can be represented by the conjunction of properties (i.e. literals) that are true in **every** possible world iff:

- effects are the same in every state (i.e. Basic PDDL action schemas)

The construction of the successor state can then be done as in classical planning.

Computing the successor state

$$b' = Result(b, a) = (b - DEL(a)) \cup ADD(a)$$

In particular, for the unknown literals in s:

- unknowns that are added will be $true$
- unknowns that are deleted will be $false$
- other unknowns remain unknown

checking that l and $\neg l$ can not both be true ...

General case

PDDL **Conditional effects**:

$$Action(Suck, EFFECT:$$
$$\textbf{when } AtL : CleanL \wedge \textbf{when } AtR : CleanR)$$

When $Suck$ is executed the belief state includes a state where $CleanL$ and a state where $CleanR$.

General settings with **Sensing actions/NDA with PO**

In these cases, the state must be represented with a disjuntion, and the computation becomes more complex.

Epistemic representation of belief states

Belief states can be represented as knowledge propositions:

$\mathbf{K}(AtR \wedge CleanR)$

Principle of minimal knowledge or maximal ignorance: everything unknown is $true$ in some models and $false$ in some models

$\mathbf{K}(AtR \vee CleanR)$ is different from $\mathbf{K}AtR \vee \mathbf{K}CleanR$

Computing with belief sets

Approximations (as in the case of reachable sets)

$\diamondsuit$ Belief state represented by the intersection of all states (sound but incomplete).

Heuristics for subsets of a belief state are always admissible.

General computation of the successor state

1. Predict (Compute the states that can result from ND-action execution)
2. Observation (Acquire info from ND-sensors – return a set of percepts)
3. Update

◇ **NDA + DFO**

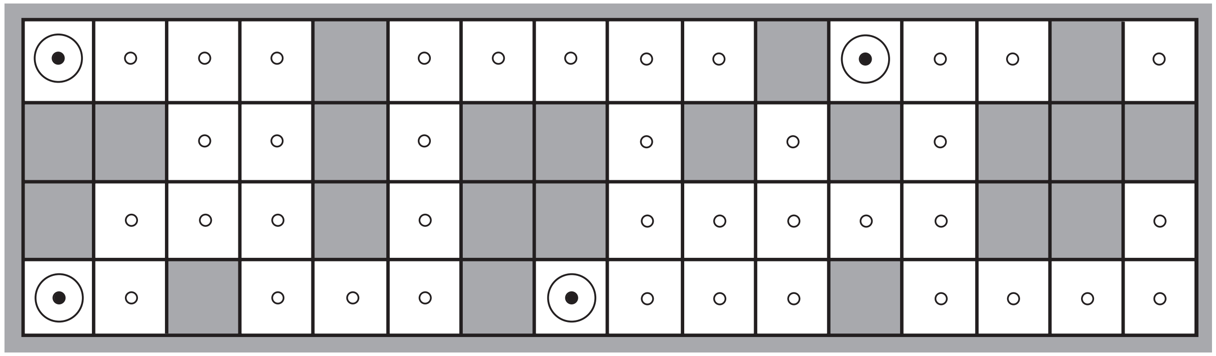
——————— require complex belief state representation

◇ **NDA + DPO**

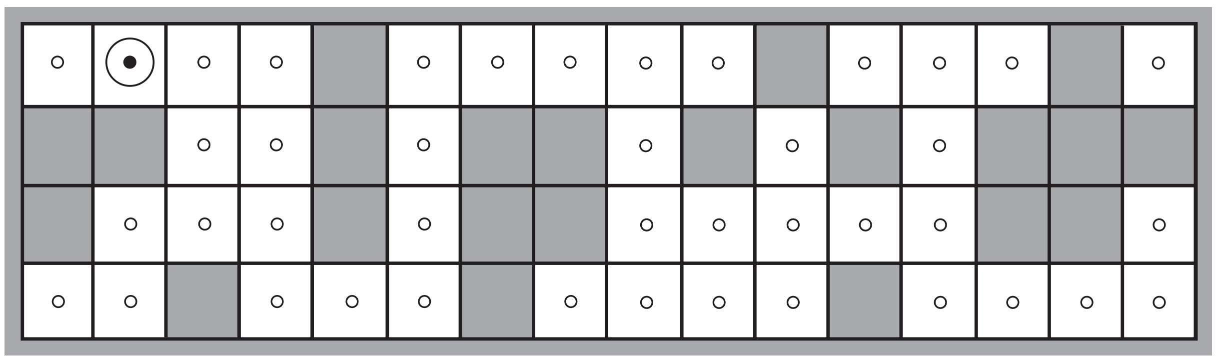
◇ **+ active sensing**

◇ **+ NDPO** (probabilistic model of sensing)

Localization



(a) Possible locations of robot after $E_1 = NSW$



(b) Possible locations of robot After $E_1 = NSW, E_2 = NS$

Sens: NSW, Move, Sens:NS

$$b' = UPDATE(PREDICT(b, a), o)$$

Plan failures

Action monitoring:

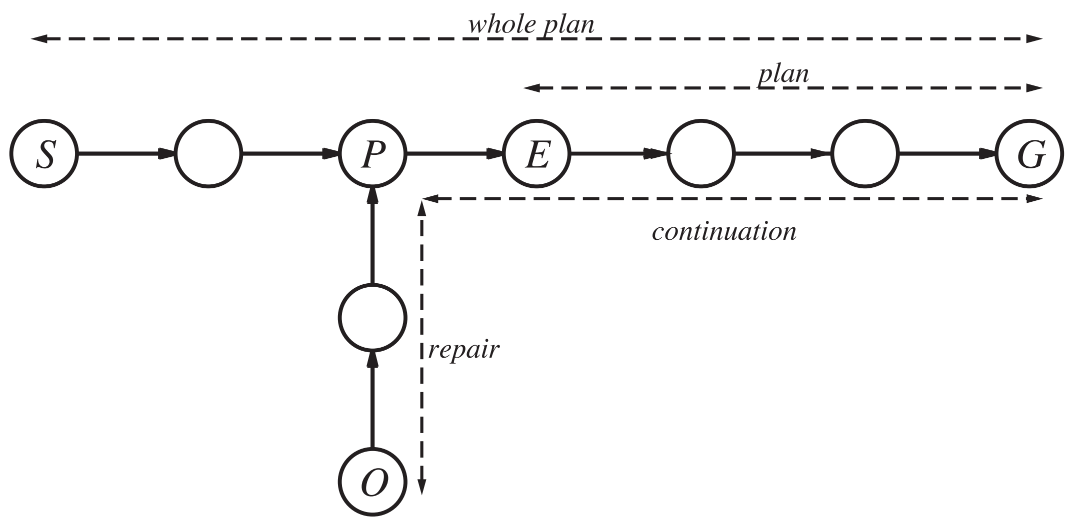
- ♢ missing preconditions (e.g. no can opener)
- ♢ missing effect (e.g. painting the chair does not spill the colour on the table)
- ♢ state incomplete (e.g. not enough painting)
- ♢ exogenous events (e.g. rain melts the colour)

Monitor the execution (action, plan, goal) + replan

Execution Monitoring (action)

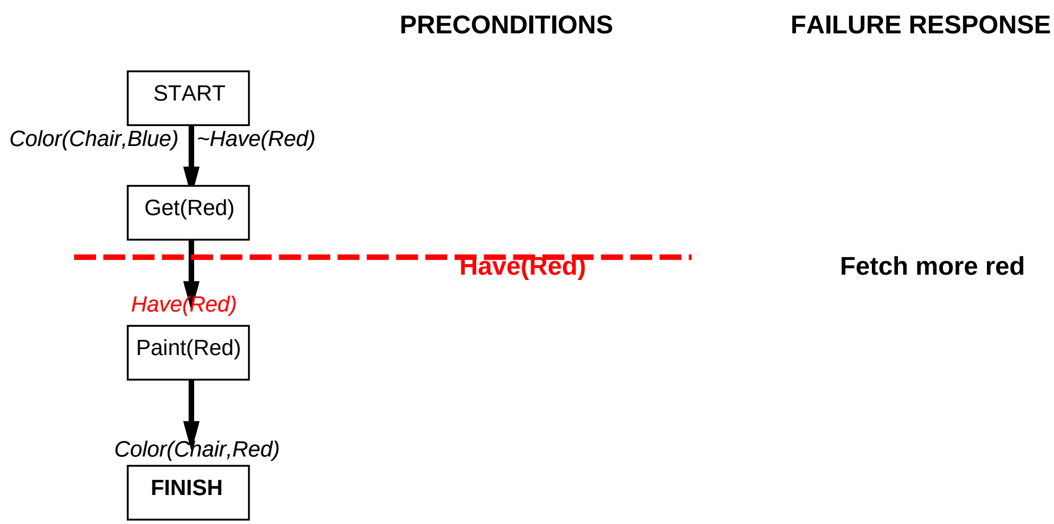
Action monitoring:

- before the next action the agent checks the preconditions
- if any precondition is no longer satisfied the agents replans

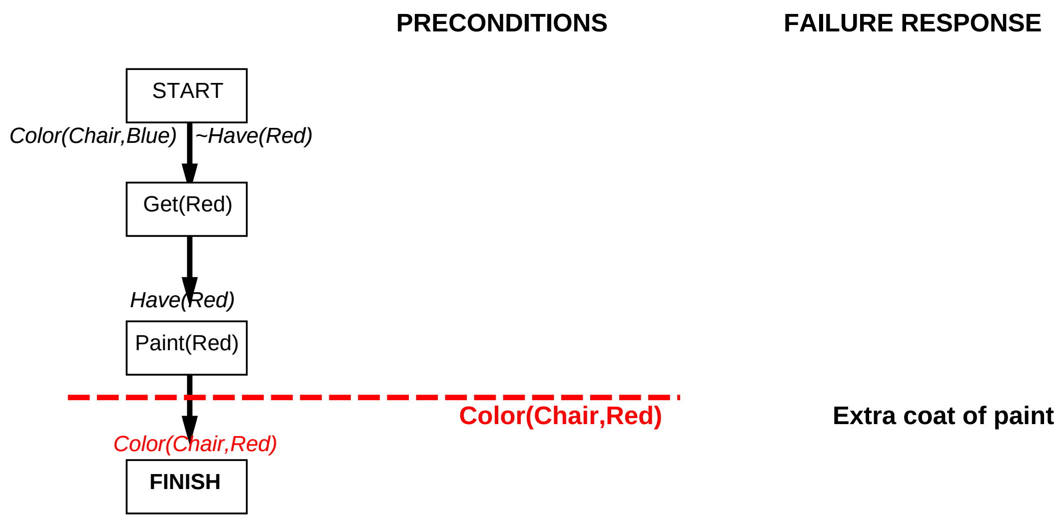


Checking failures during plan execution

Example: if an action fails the agent can go one step back in the plan and execute the action again



Checking failures during plan execution



"Loop until success" behavior *emerges* from interaction between monitor/replan agent and uncooperative environment

Execution Monitoring (plan)

Plan monitoring:
Check whether the goal is still achievable

e.g. not enough painting to finish the work!

"Failure" = preconditions of *remaining plan* not met

Execution Monitoring (goal)

Goal monitoring

Check whether the goal has to be changed

e.g. chair is discovered to be broken

$\longrightarrow$ Continuous Planning