# Introduction to Artificial Intelligence

## Local search

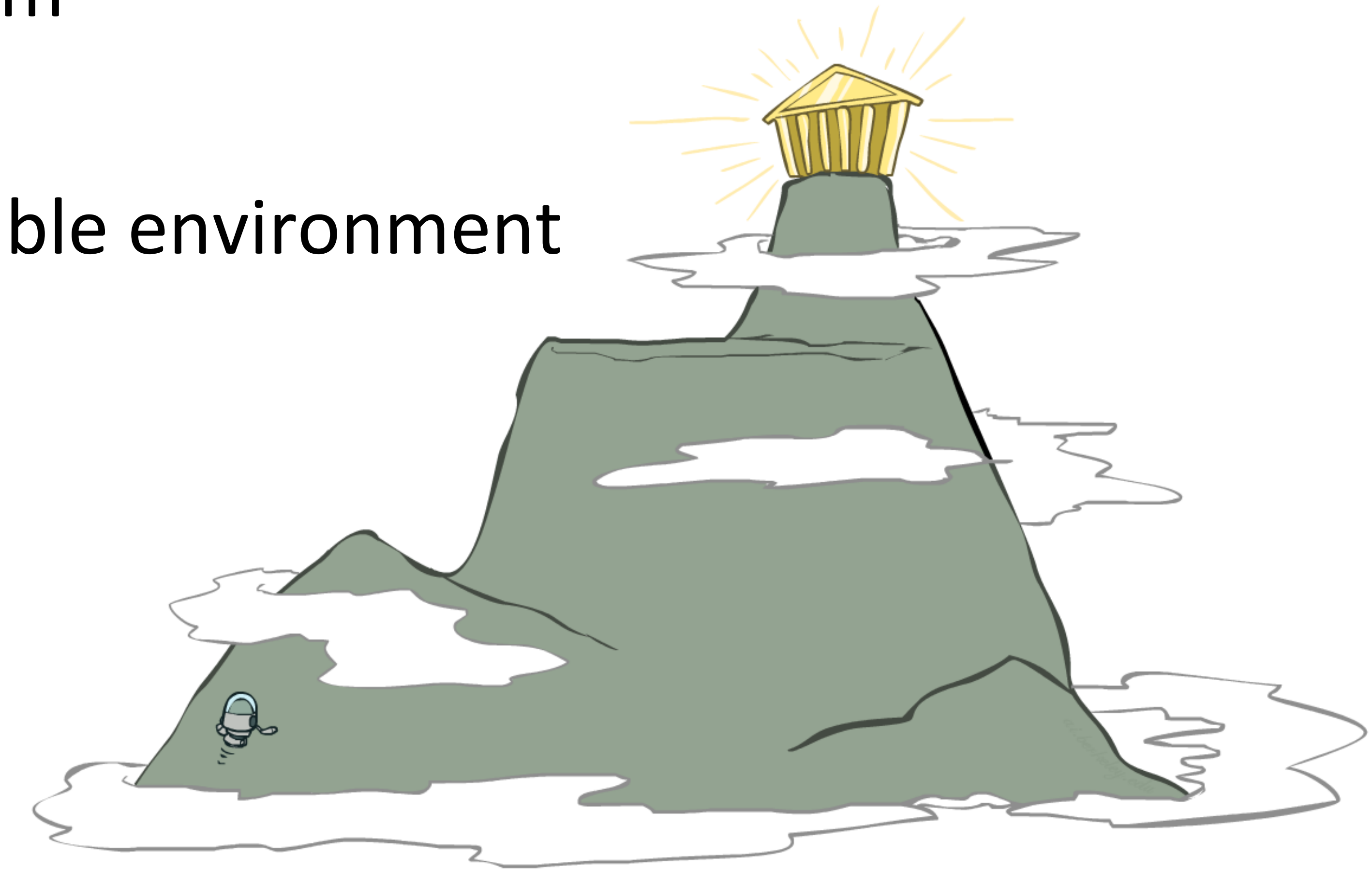
Instructor: Tatjana Petrov

University of Trieste, Italy

[slides adapted from Dan Klein, Pieter Abbeel, Stuart Russell, et al for CS188 Intro to AI at UC Berkeley. All materials available at http://ai.berkeley.edu.]

# Today

- Local Search
  - With non-determinism
  - In a partially-observable environment



# Search with Uncertainty

- When the environment is nondeterministic or partially-observable, the agent doesn’t know what state it transitions to after taking an action

- **Belief state** = the set of physical states that the agent believes are possible

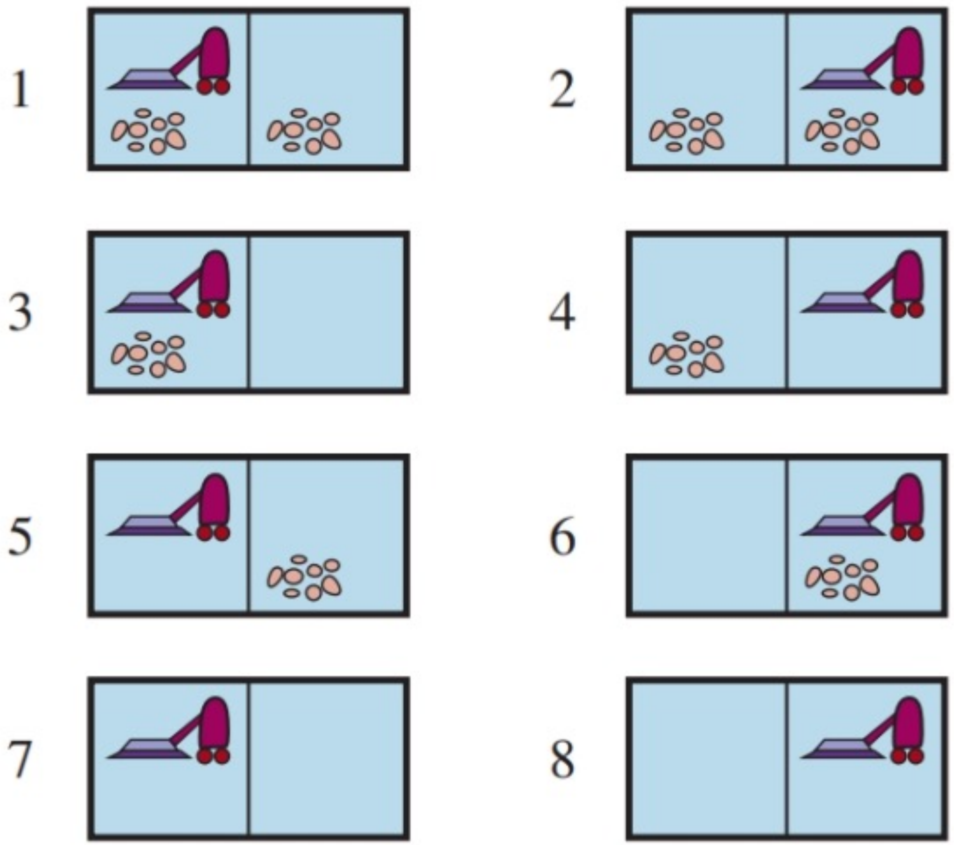
# Search with Uncertainty

- We cannot have as solution a sequence

- It has **conditional plan** (contingency plan or a strategy) that specifies what to do depending on what percepts agent receives while executing the plan.

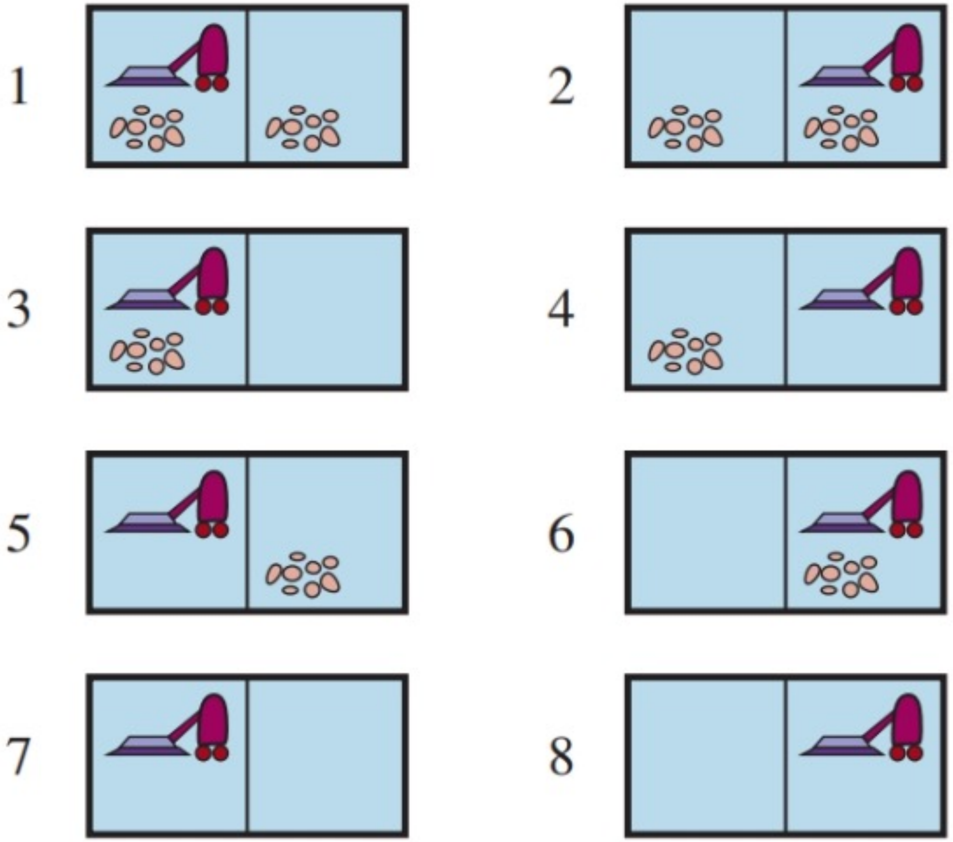
# An erratic vacuum world

- Goal : clean up all the dirt (states 7 and 8)
- Three actions: Right, Left, and Suck
- In the erratic vacuum world, the Suck action works as follows:
  - When applied to a dirty square the action cleans the square and **sometimes** cleans up dirt in an adjacent square, too.
  - When applied to a clean square the action sometimes deposits dirt on the carpet



# Belief State and Conditional Plan

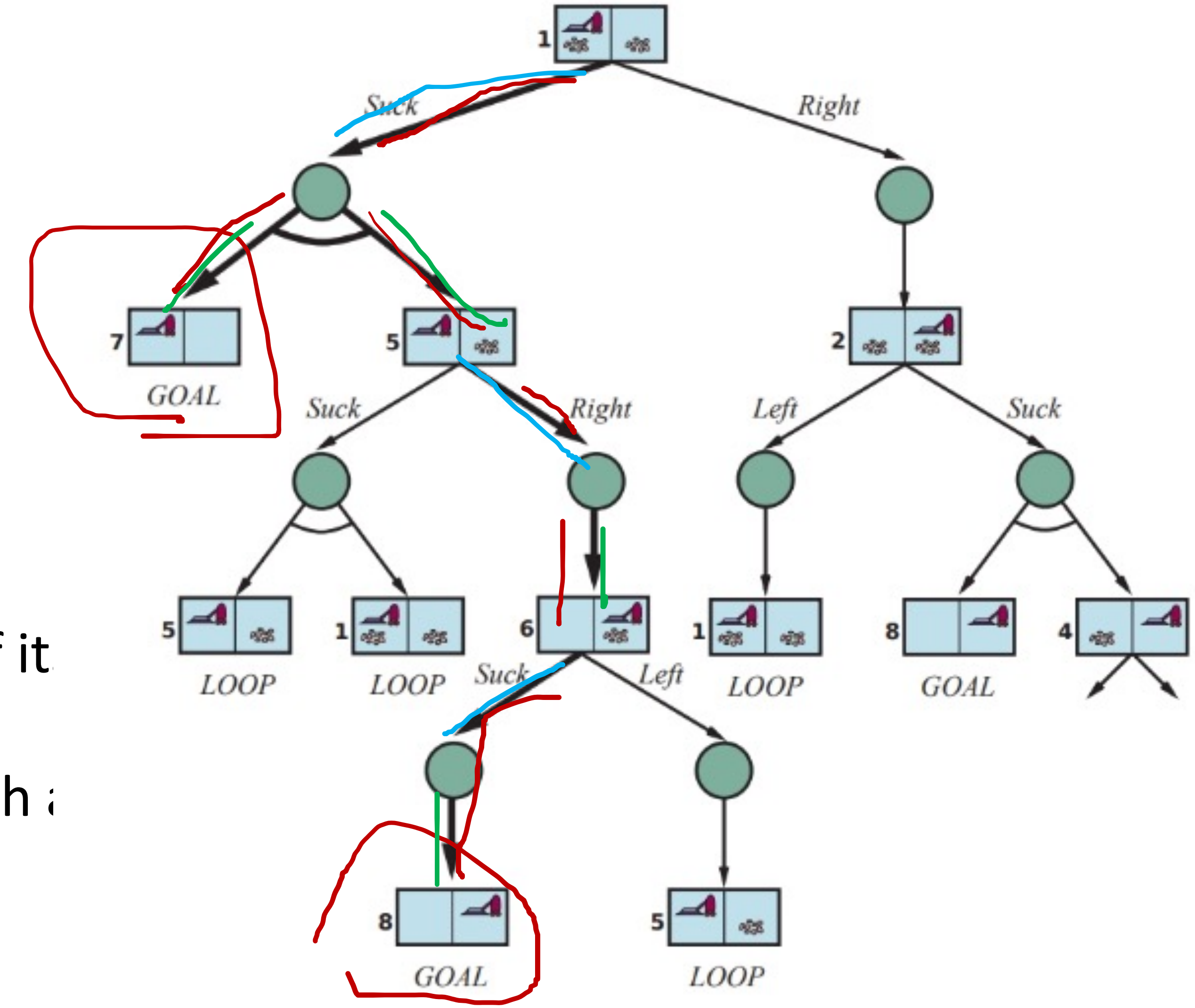
- A RESULTS function that returns a set of possible outcome states
  - E.g., RESULTS(1, Suck) = {5,7}
- Conditional plan gives the condition to solve the problem
  - E.g. [Suck, if State=5 then [Right,Suck] else [ ]]



# AND–OR search trees

- OR nodes:
  agent’s choice
- **AND nodes**: environment’s choice

- A solution is a subtree that:
  - has a goal node at every leaf
  - specifies one action at each of it OR nodes
  - includes every outcome branch each of its AND nodes.



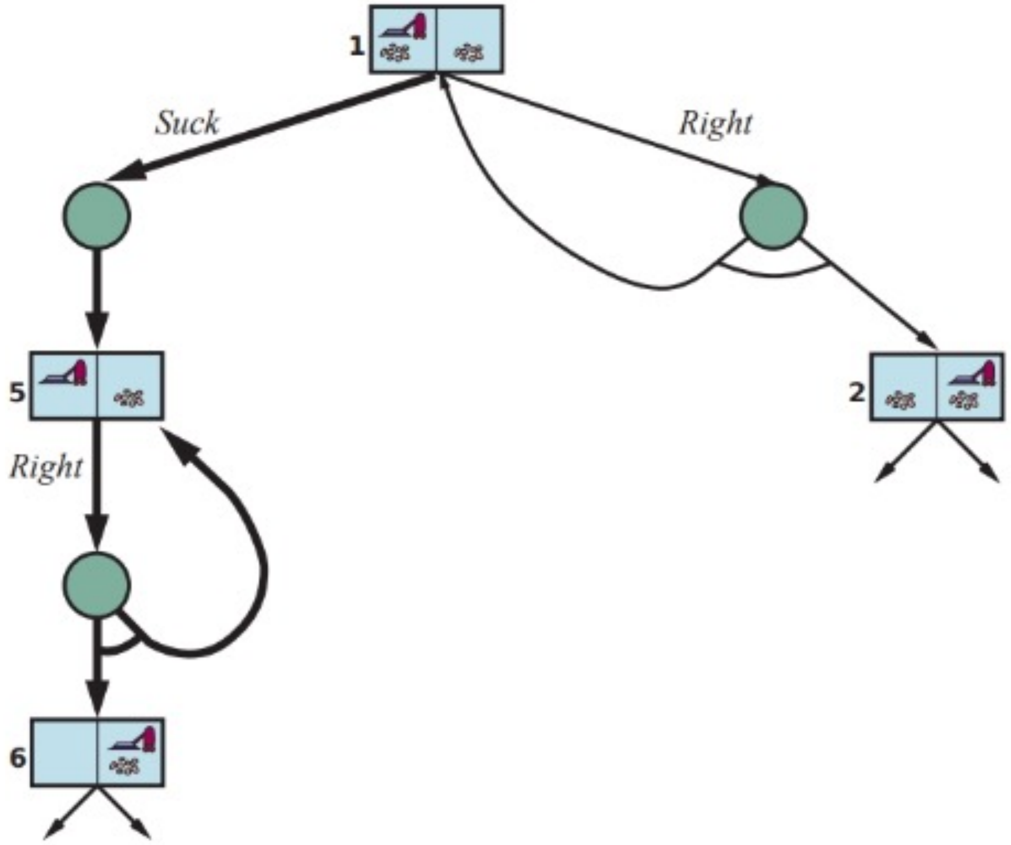
# Search Algorithms with non-determininsm

- AND–OR graphs can be explored with depth-first or breadth-first or best-first algorithms
- The concept of a heuristic function must be modified to estimate the cost of a contingent solution rather than a sequence
- The notion of admissibility carries over and there is an analog of the $A^*$ algorithm for finding optimal solutions

# A slippery vacuum world

- E.g., RESULTS(5, Right) = {5,6}
- Solution:
  [Suck,**while** State=5 **do** Right,Suck]

- When is a cyclic plan a solution?
  - every leaf is a goal state and that a leaf is reachable from every point in the plan
  - If random choice -> eventually it will work and the plan will succeed



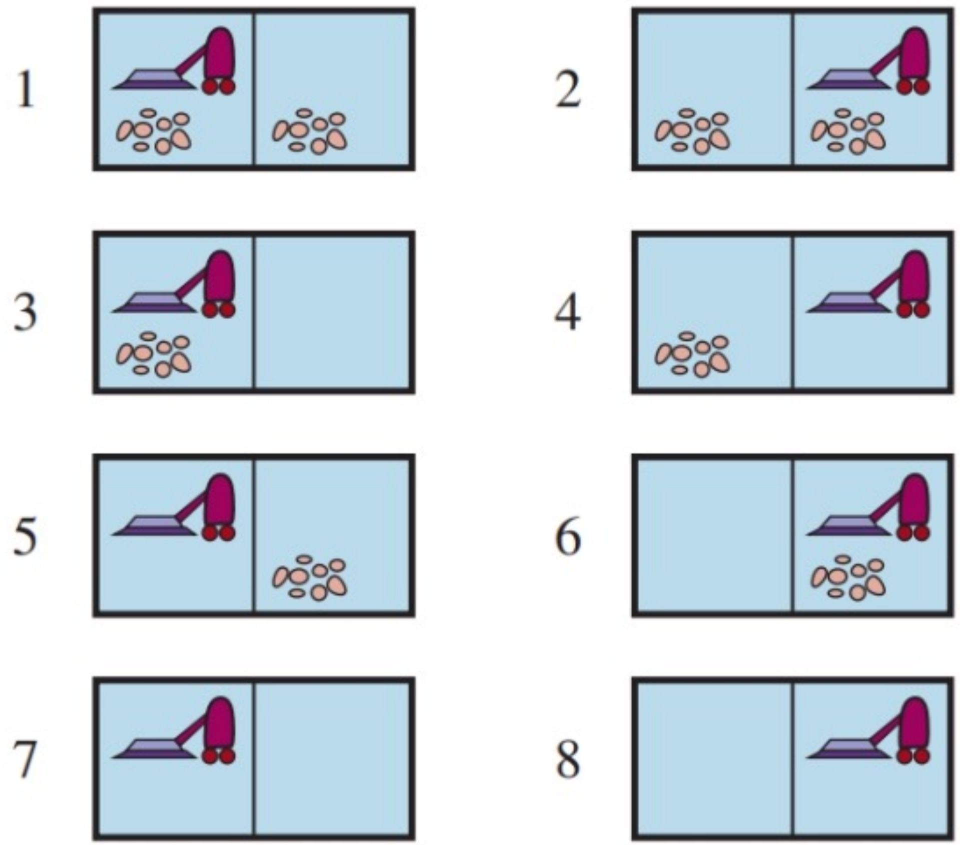
# Search in Partially Observable Environments

- A **sensorless** problem (or a conformant problem): when the agent's percepts provide *no information at all*

# A sensorless version of the vacuum world.

- Assume that the agent knows the geography of its world, but not its own location or the distribution of dirt.

- Its initial belief state is {1,2,3,4,5,6,7,8}
- Moving Right it will be in one of the states {2,4,6,8}
- After [Right,Suck] the agent will always end up in one of the states {4,8}
- [Right,Suck,Left,Suck] the agent is guaranteed to reach the goal state 7



# The belief-state problem

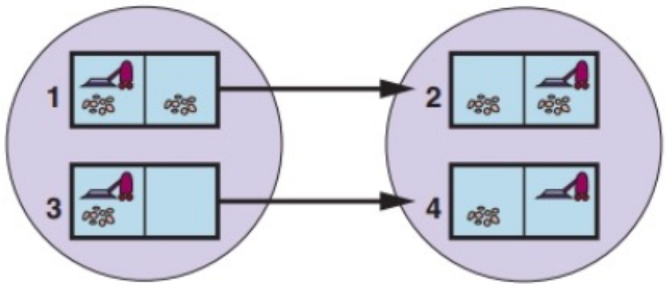
- States: $2^N$ belief states, if the original problem P has N states
- Initial state: typically all states in P
- Actions: $\text{ACTIONS}(b) \; = \cup_{s \in b} \; \text{ACTIONS}_P(s)$ . If an illegal action might lead to catastrophe, it is safer to allow only the intersection
- Transition model: $\text{RESULT}(b, a) \; = \; \{s' : \; s' \; = \text{RESULT}_P(s, a) \text{ and } s \; \in \; b\}$
- Goal test: The agent necessarily achieves the goal if every state satisfies $\text{IS} - \text{GOAL}_\text{P}$ (s).
- Action cost: we assume that the cost of an action is the same in all states

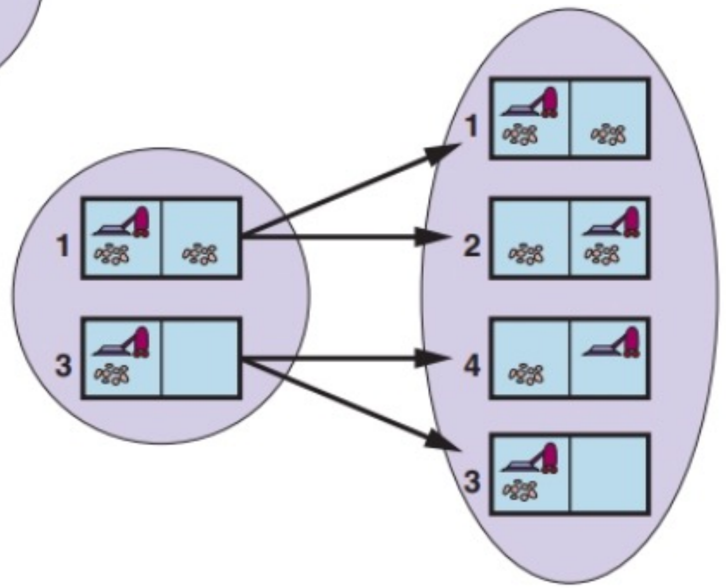
# The belief-state problem

$b' = \text{RESULT}(b, a) = \{s' : s' = \text{RESULT}_P(s, a) \text{ and } s \in b\}$

$\{b'\}^{\#} > \{b\}^{\#}$ only with nondeterministic actions

- $\text{RESULT}(\{1,3\}, right) = \{2,4\}$



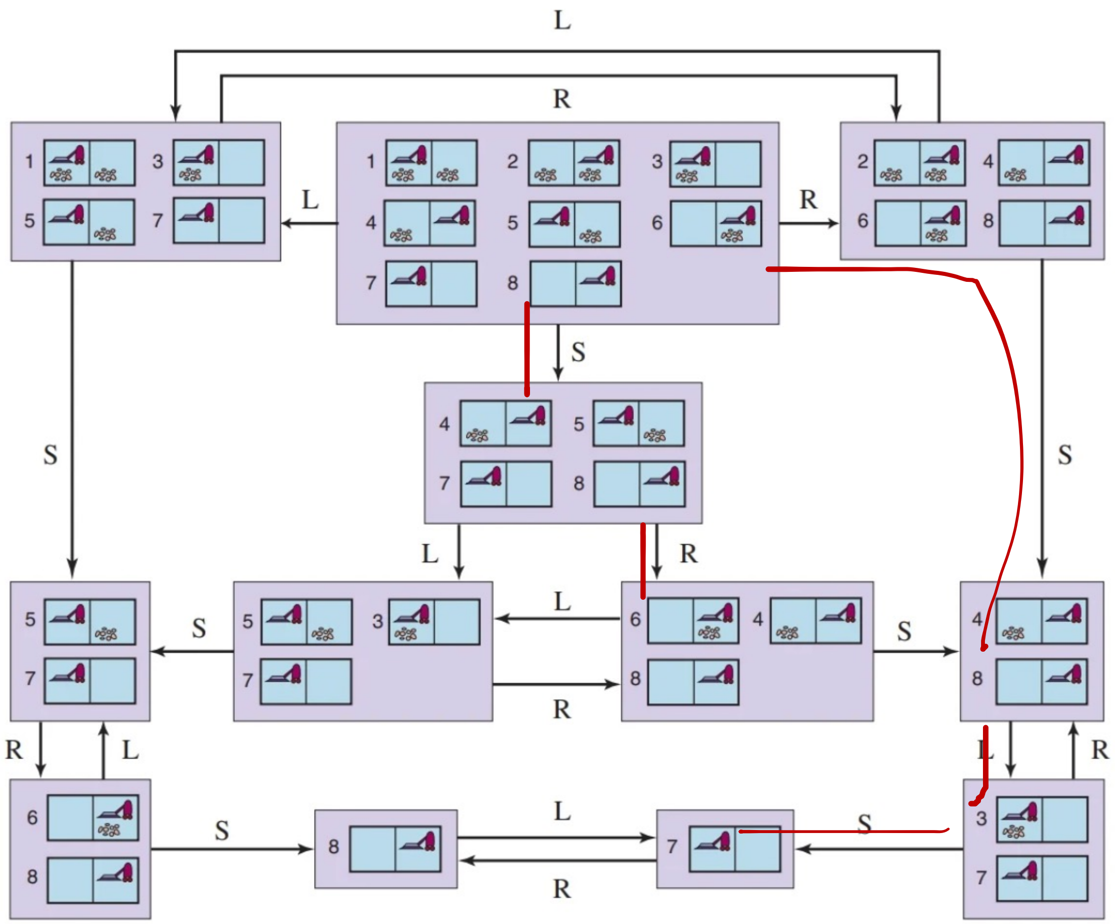




- $\text{RESULT}(\{1,3\}, right) = \{1,2,3,4\}$ in the slippery version

# Reachable belief-state space for the det world

- There are only 12 reachable belief states out of $2^8$=256 possible belief states

- we can solve sensorless problems with any of the ordinary search algorithms



# Decreasing the number of states

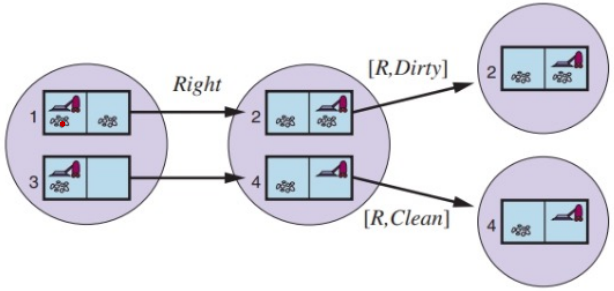
- Compact description
  - E.g. “ nothing ”, “Not in the rightmost column”
- Incremental belief-state search:finding a solution that works for state 1; then we check if it works for state 2, …
  - Single solution
  - Detect failure quickly

# Searching in partially observable environments

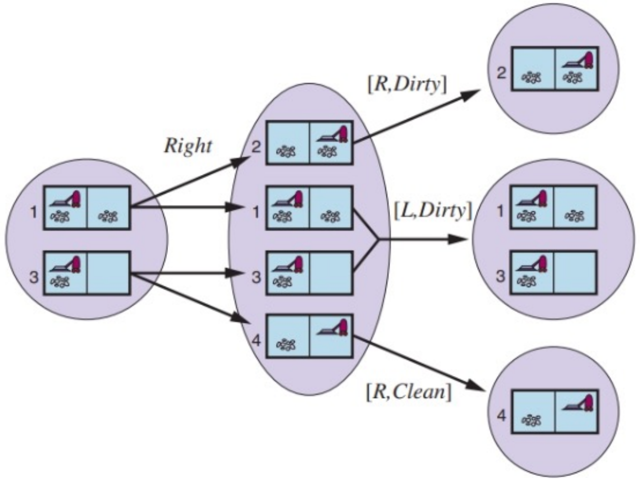
- Many problems cannot be solved without sensing.
- A PERCEPT(s) function that returns the percept received by the agent in a given state.
  - PERCEPT(s)={possible percepts.} , non-deterministic problems
  - PERCEPT(s)=s , fully observable problems
  - PERCEPT(s)= Ø for sensorless problem

# Local-sensing vacuum worlds

- A position sensor that yields the percept L in the left square, and R in the right square
- A dirt sensor that yields Dirty when the current square is dirty and Clean when it is clean




- In the deterministic world
  - PERCEPT(1)=[L,Dirty]

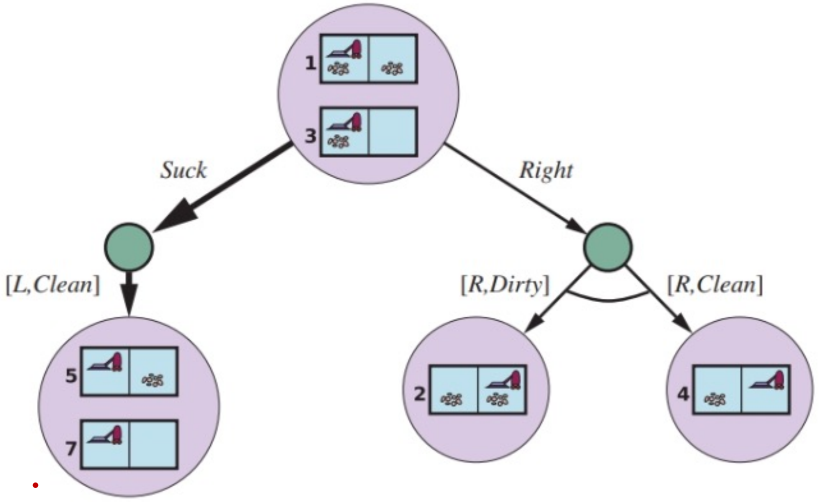



- In the slippery world

# Solving partially observable problems

- Direct way to redefine the RESULT function considering the PERCEPT fcn
- We can apply directly the AND–OR search algorithm

initial percept [L, Dirty]




- Solution is:
  - [Suck,Right, if R state={6} then Suck else [ ]]

# An agent for partially observable environments

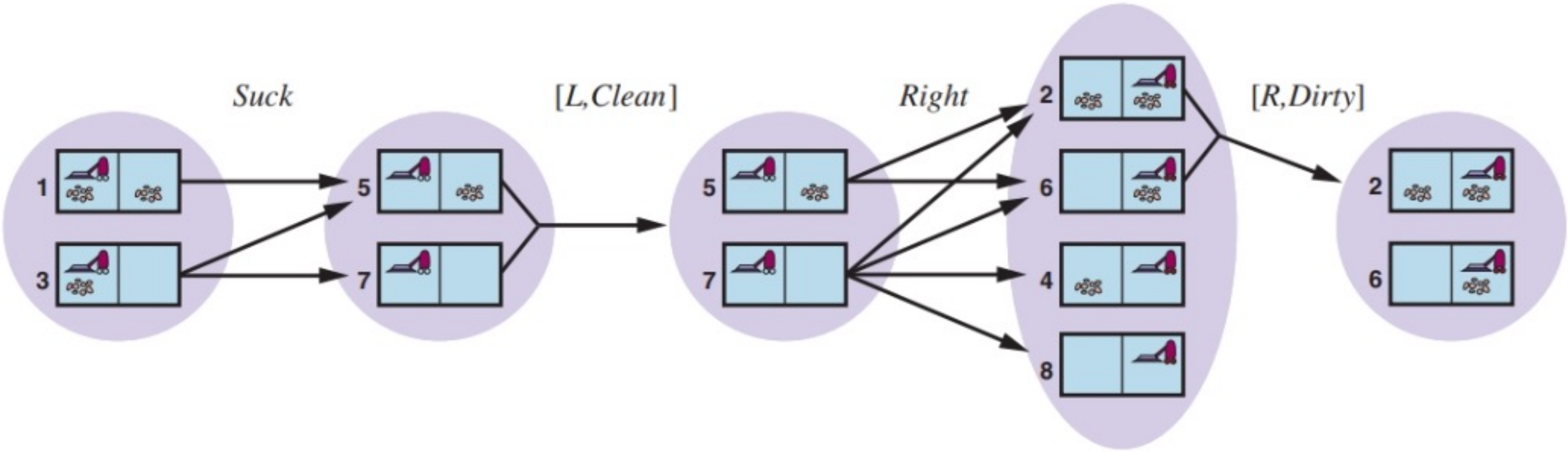
An agent for partially observable environments formulates a problem, calls a search algorithm (such as AND-OR-SEARCH) to solve it, and executes the solution

Two differences between this agent and the one for fully observable deterministic environments:

1. The solution will be a conditional plan rather than a sequence
2. The agent will need to maintain its belief state as it performs actions and receives percepts

# Kindergarten vacuum world

- Agents sense only the state of their current square
- Any square may become dirty at any time unless the agent is actively cleaning it at that moment

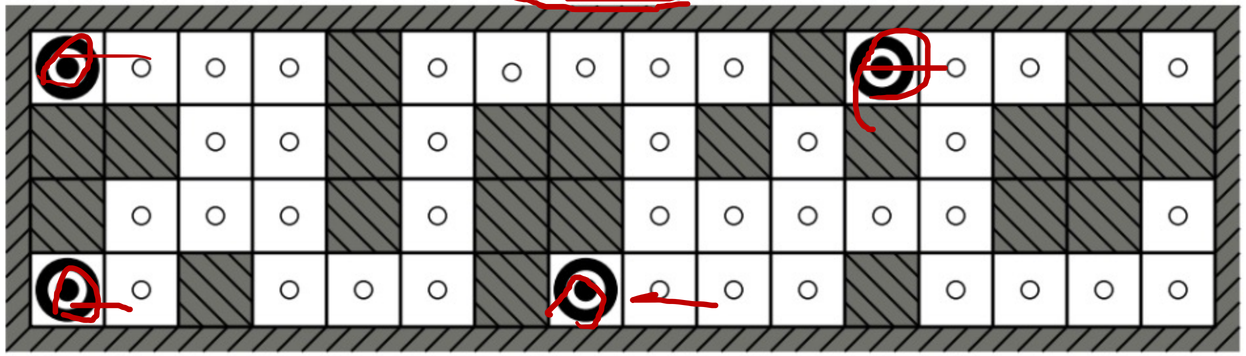


# Localization

Problem: given a map of the world and a sequence of percepts and actions working out where it is

- 4 sonar sensors that tell whether there is an obstacle
- Percept is in the form of a bit vector
- Right random action
- Initial belief state ={all states}

UPDATE(1011)



UPDATE(PREDICT(UPDATE(b,1011),Right),1010)