Lecture 9

CS 6380 Artificial Intelligence

Search in Partially Observable Settings and Online Search

Department of Computer Science and Engineering

Indian Institute of Technology Madras (IITM), India

Logistics and Recap

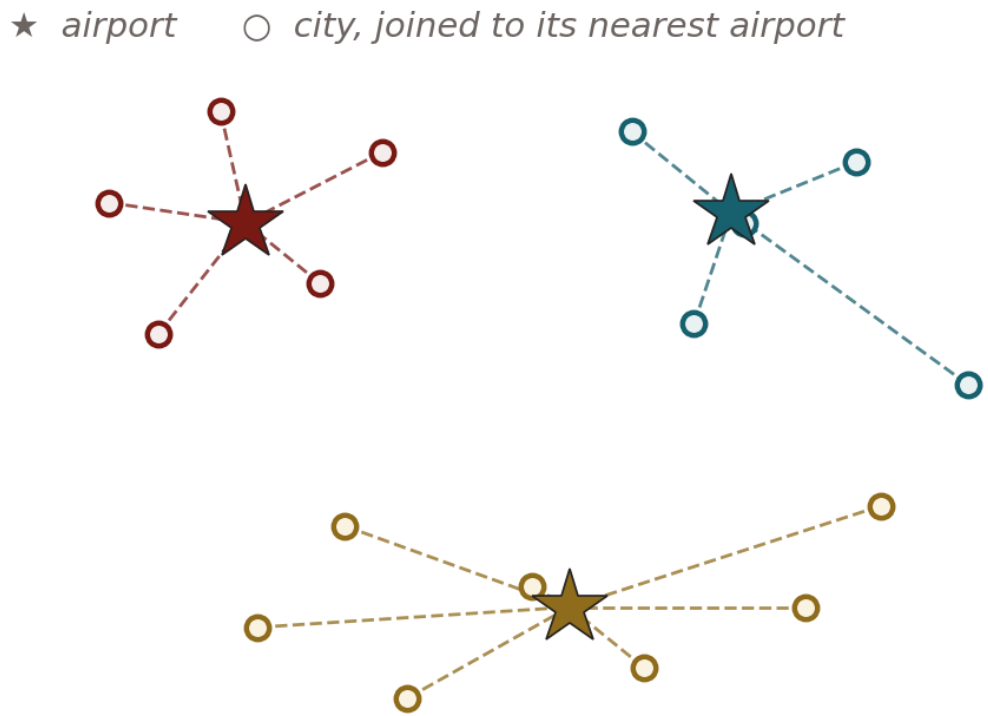
Talk Announcement

- **Title:** Towards Reliable Autonomy: From Individual Agents to Autonomous Teams
- **Date:** August 21, 2026
- **Time: 11**:00 AM – 12:00 PM
- Location: SSB 334

Dr. Sandhya Saisubramanian
Oregon State University, USA
https://www.sandhyasai.com/

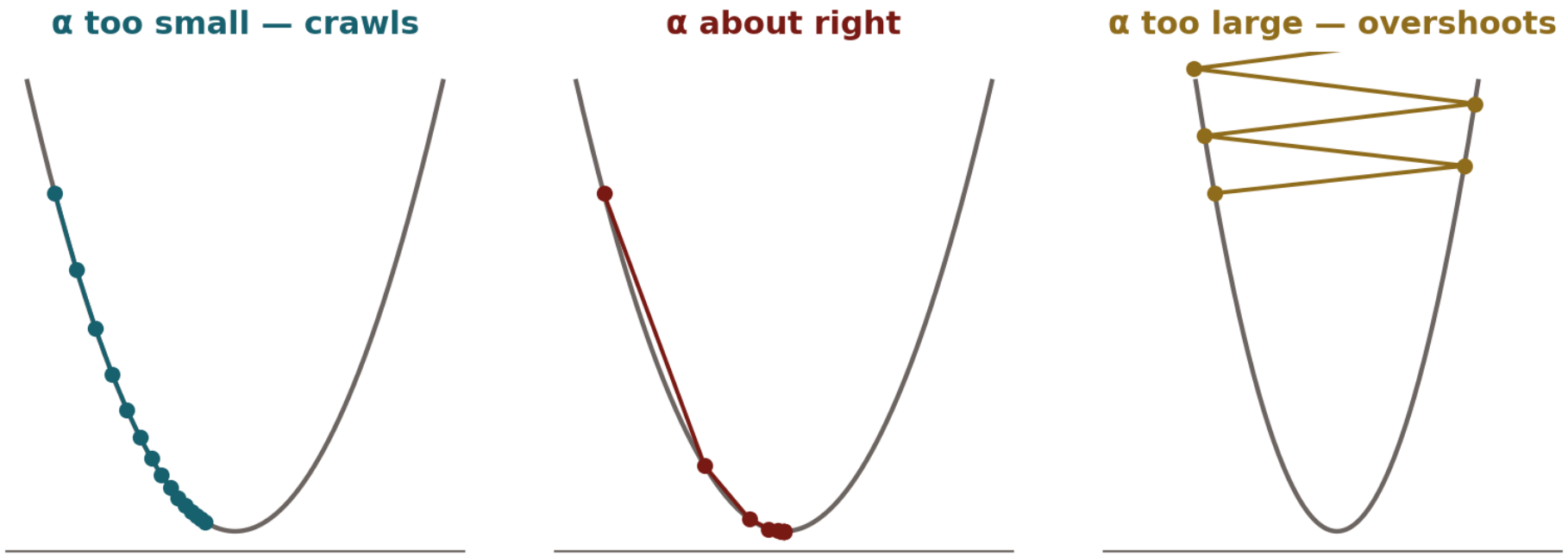
Why Continuous Spaces Break Our Machinery

- Most real problems are continuous: positions, angles, prices, network weights
- The successor function collapses: a state has infinitely many neighbours, so "generate all successors" means nothing
- Running example: site 3 airports in Romania
  - state $x = (x_1, y_1, x_2, y_2, x_3, y_3)$: six real numbers
  - minimise $f(x) = \Sigma_a \Sigma_{c \in C_a} (x_a - x_c)^2 + (y_a - y_c)^2$
  - C_a = the cities whose nearest airport is a
- *Careful: C_a changes as the airports move, so f is smooth in patches — differentiable almost everywhere, not everywhere*



Getting the Step Size Right

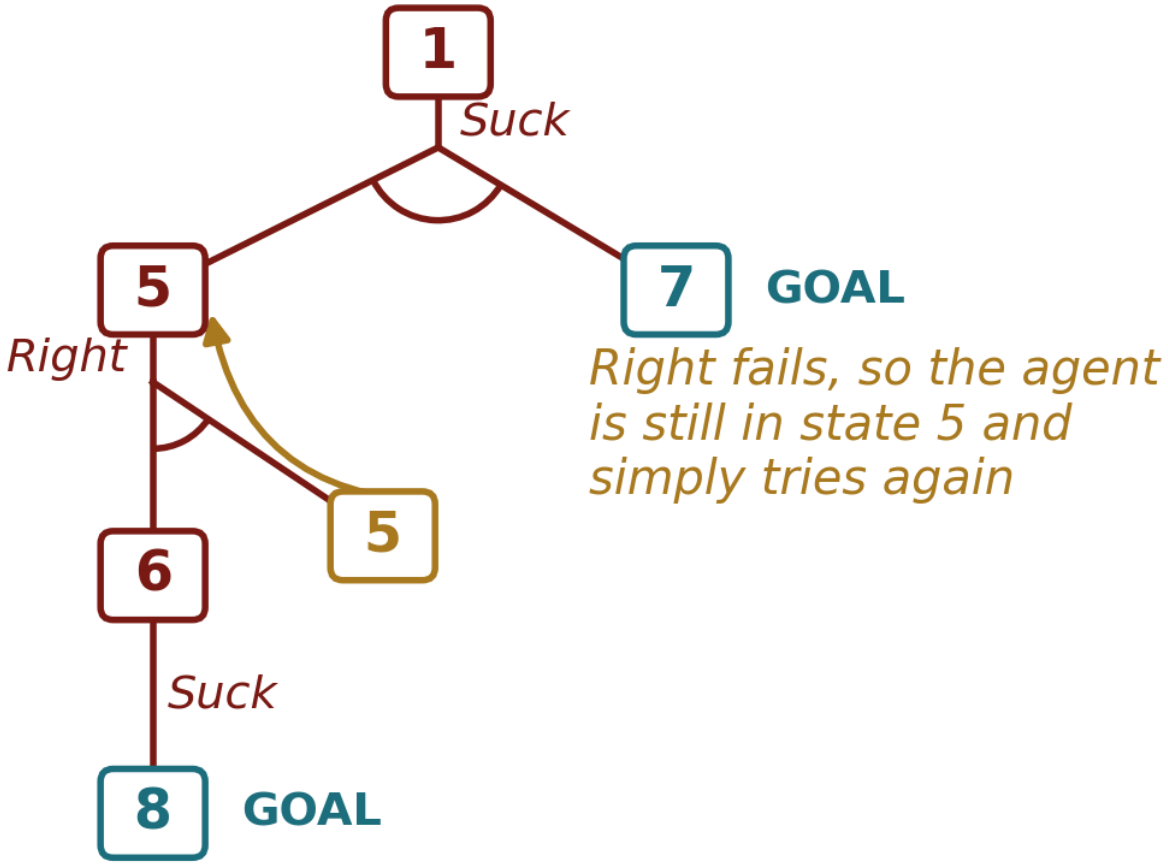
- α too small: correct but slow. α too large: overshoot, oscillate, diverge
- Line search: keep doubling α while f improves, then back off: no hand-tuning needed
- Newton–Raphson uses curvature, $x \leftarrow x - H^{-1}(x)\nabla f(x)$, with $H_{ij} = \partial^2 f / \partial x_i \partial x_j$: far fewer steps, but H is n × n (hence BFGS and L-BFGS)



Local maxima, plateaux and ridges all survive the move to continuous space: restarts and annealing still earn their keep.

The Slippery Vacuum World

- Movement is unreliable now, so Right sometimes leaves the agent where it was
- RESULTS(5, Right) = {5, 6}
- No acyclic solution exists, since every finite plan runs out of moves on the branch where the wheels keep slipping

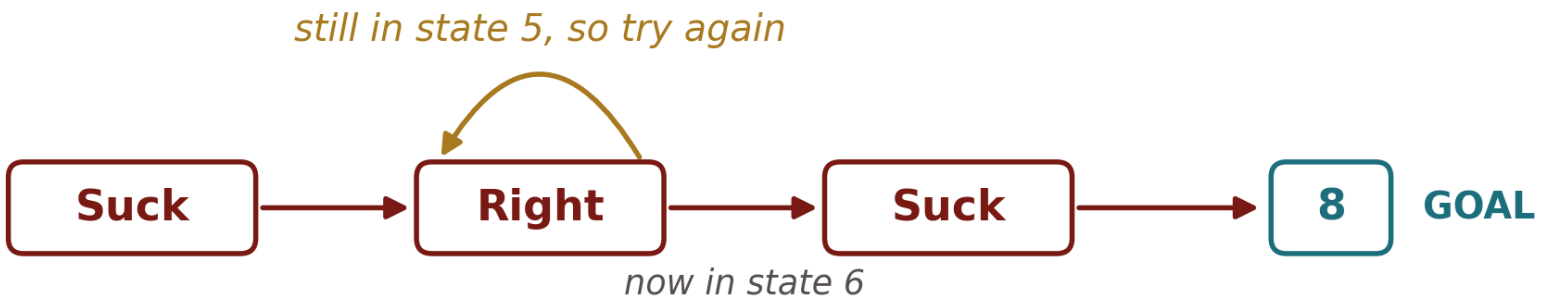


Cyclic Plans

- Let a plan point back into itself, so a failed attempt is simply repeated

```
[Suck, while State = 5 do Right, Suck]
```

- It is a solution when every leaf is a goal and every leaf is reachable from every point in the plan
- The guarantee holds only if the failure does not repeat forever, so the action must eventually work
- This is the strong cyclic plan we met earlier, now with the condition stated precisely

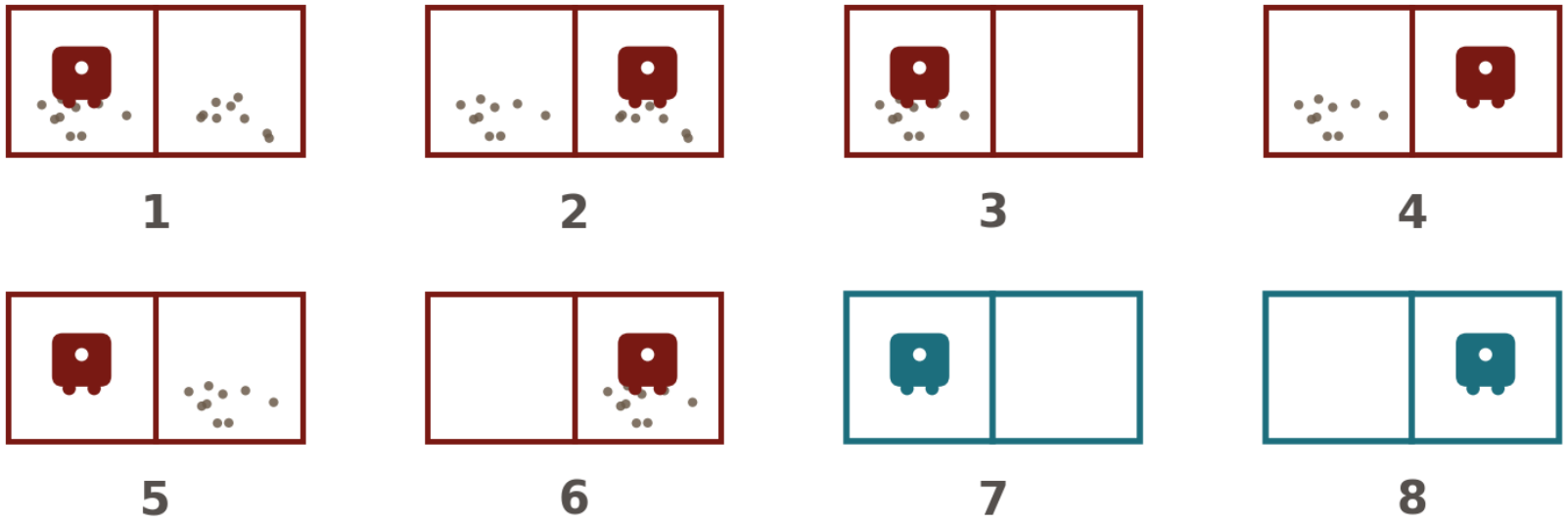


Search in Partially Observable Environments

When the agent cannot tell which state it is in

What the Agent Knows Instead

- With no sensors the agent still knows the map and what each action does
- What it does not know is which of these states it is actually in
- So it reasons about a belief state, the set of states it could be in



states 7 and 8 are the goal states

The Belief State Space

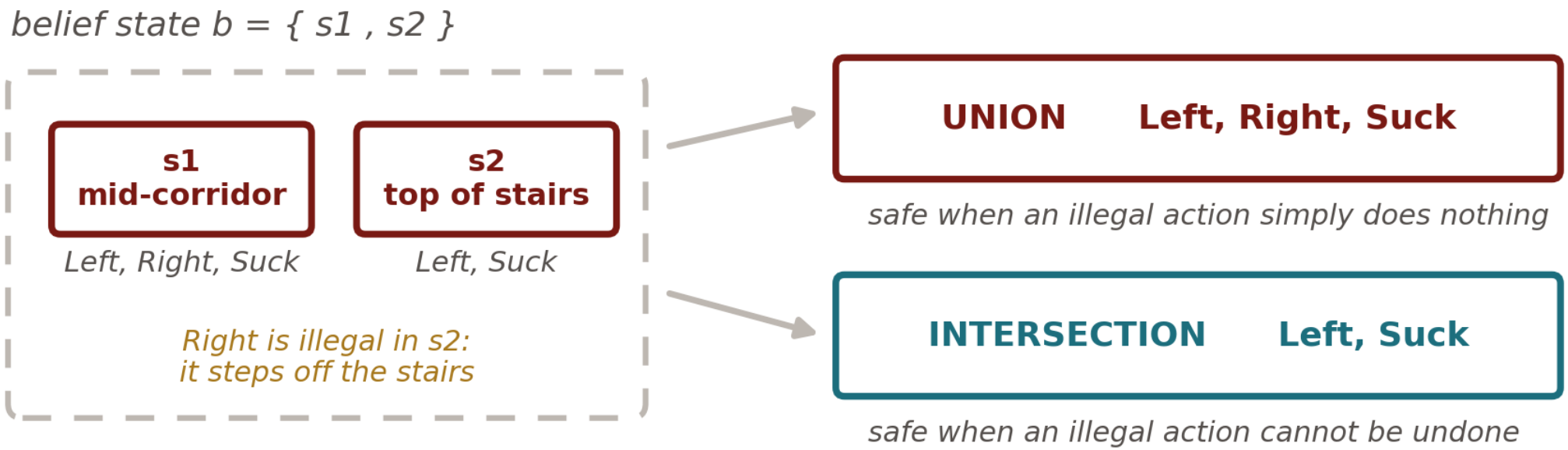
- Belief states inherit everything from the underlying problem, an initial state, actions, a transition model and a goal test

8 physical states, 256 belief states, 12 reachable

- The goal test asks whether every state in the belief state is a goal, since the agent has to be certain
- The blow up is the price of not knowing, and the reachable part is usually far smaller than the worst case

Which Actions Are Even Legal?

- Applying an action to a belief state assumes the action is available, but the states inside the set need not agree about that
- If ACTIONS(s1) and ACTIONS(s2) differ, the agent cannot tell whether the move it wants is legal, because it does not know which state it is in
- So ACTIONS(b) has to be defined for a set, and the two natural definitions disagree exactly here



Union or Intersection

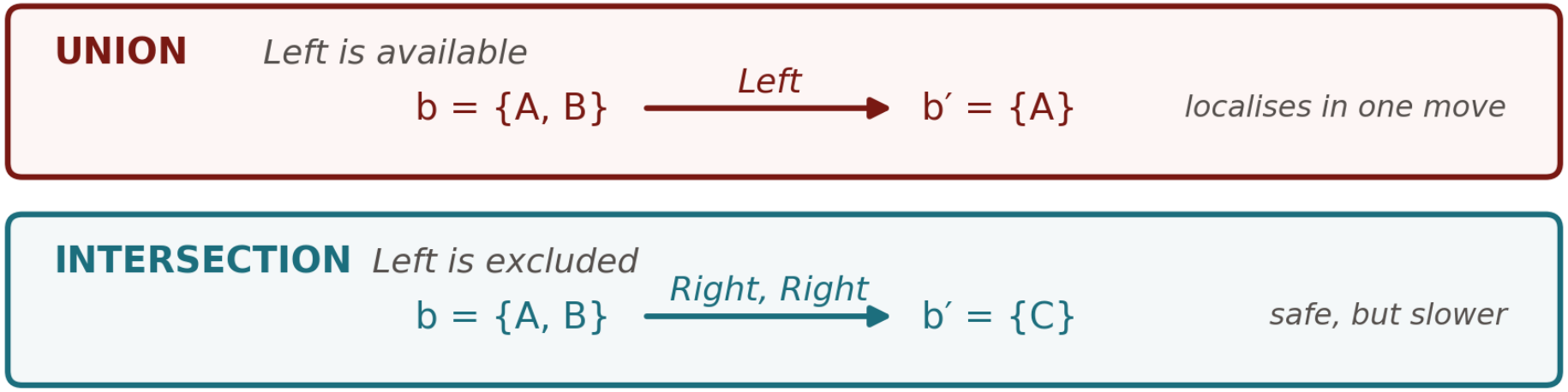
- Take the union when an illegal action simply has no effect, the way bumping into a wall does in the vacuum world

```
ACTIONS(b) = union of ACTIONS(s) over s in b
```

- Take the intersection when an illegal action could end the episode, since the agent has to be safe in every state it might be in

```
ACTIONS(b) = intersection of ACTIONS(s) over s in b
```

- In the vacuum world every state permits Left, Right and Suck, so the two rules agree there and the choice never surfaces

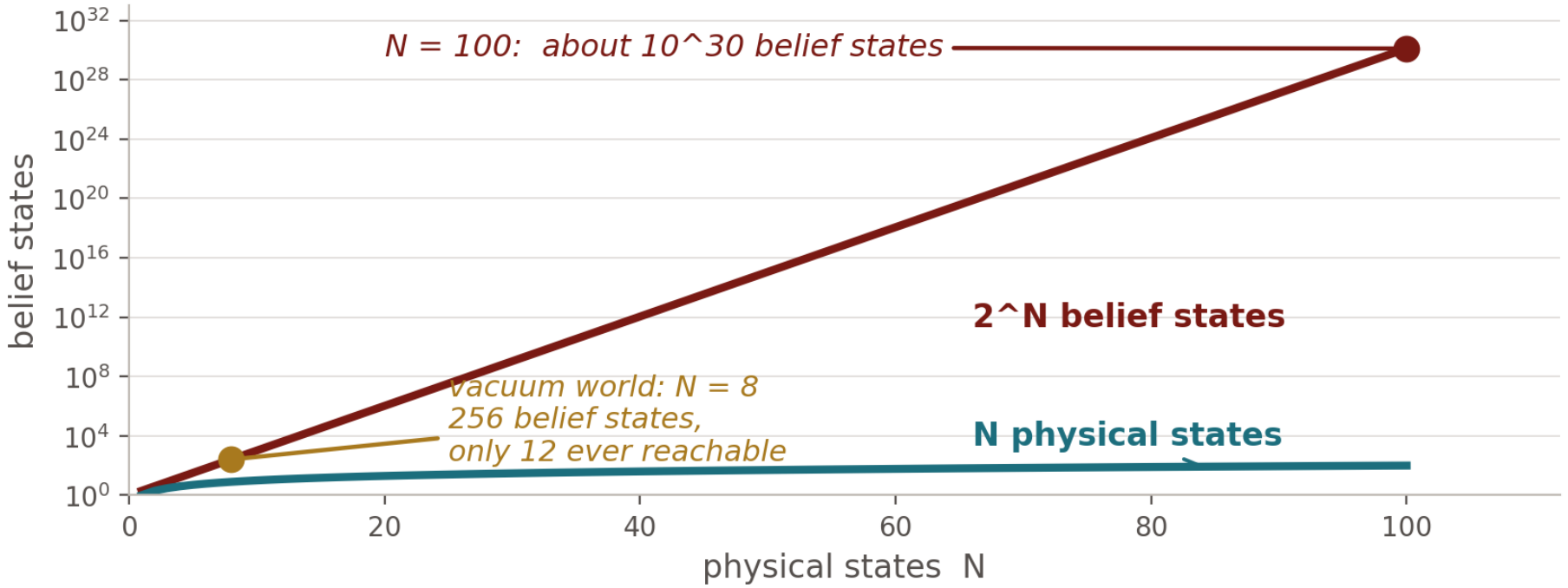


How Big Is Belief State Space?

- A belief state is any subset of the physical states, so N physical states give 2^N belief states

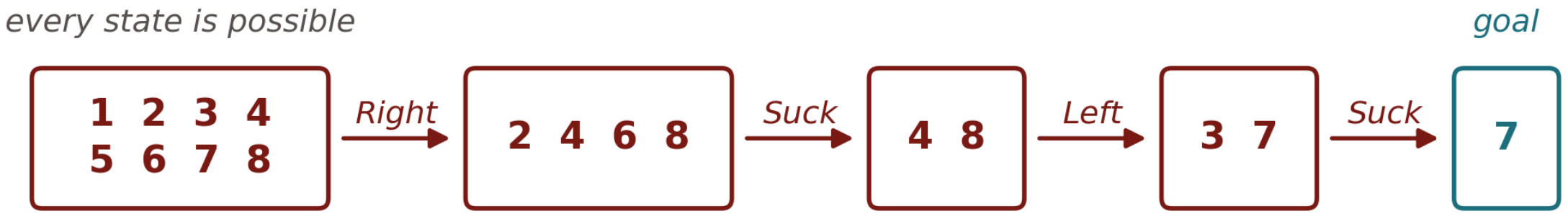
`N = 8  ->  256` `N = 20  ->  10^6` `N = 100  ->  10^30`

- The 10 by 10 vacuum world has about 10^32 physical states, so one belief state there is already too large to write down
- Only 12 of the 256 are reachable, so search meets far fewer than the worst case, and compact descriptions handle the rest



Coercing the World

- The agent cannot sense, so it acts to force the world into a configuration it can be sure of
- Right drives the agent to the right wall from every starting state at once
- The belief state shrinks until every state left in it is a goal



It Is Ordinary Search Again

- Belief states are just states, so breadth first, uniform cost and A star all apply unchanged
- The solution is a sequence of actions, because a sensorless agent has nothing to branch on

A plan for a belief state solves every subset of it

- So you can solve one physical state first, then test that plan against the rest and repair it as needed
- The real difficulty is size, since each node in the search is a set rather than a single state

Partial Observability

- Now the agent has sensors, but they do not identify the state
- In the local sensing vacuum world the agent sees its own square and nothing else

A percept rules possibilities out, it does not pin the state down

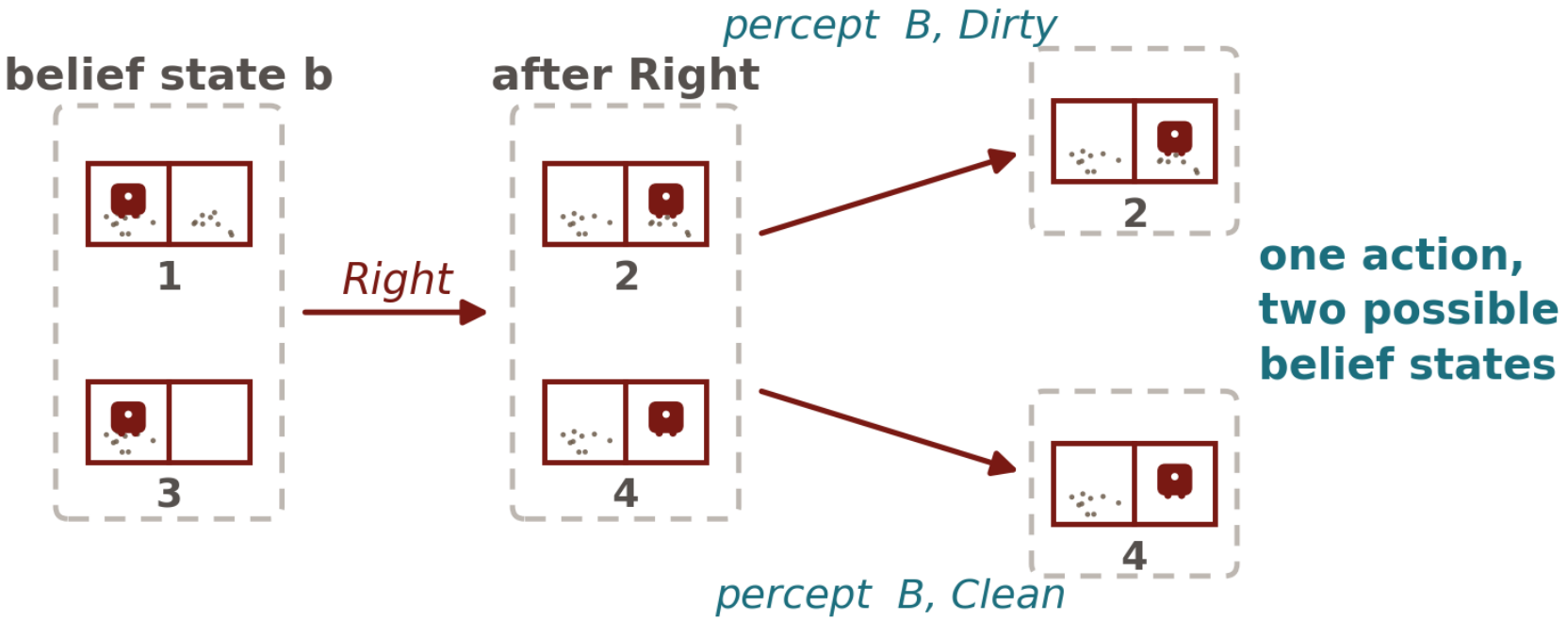
- Fully observable and sensorless are the two ends of one scale, and everything real lives in between

Predict, Possible Percepts, Update

- Predict, apply the action to every state in the current belief state
- Possible percepts, work out which observations could be received in that predicted set
- Update, for each possible percept keep only the states consistent with it

One Action, Several Belief States

- Start knowing you are in the left square with dirt, but not whether the right square is dirty
- Right predicts two states, and the two possible percepts separate them



Solving Partially Observable Problems

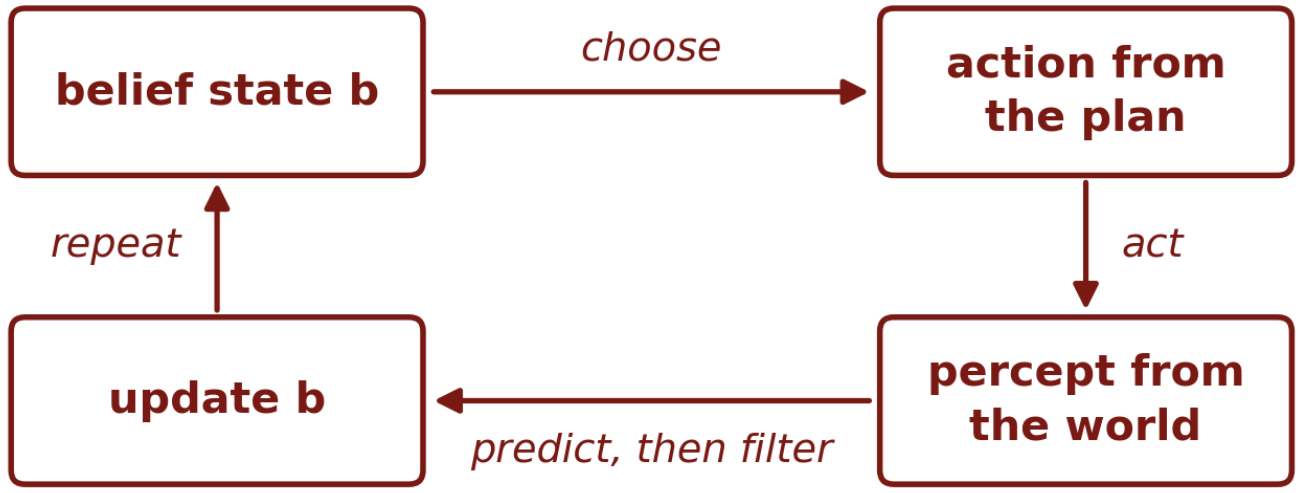
- Run AND-OR search over the belief state space, with the agent choosing actions and the environment choosing percepts

```
[Suck, Right, if State = 6 then Suck else [ ] ]
```

- The solution is a conditional plan again, but the tests are on percepts rather than on physical states
- Everything we said about strong plans, weak plans and cyclic plans carries over unchanged

An Agent for Partially Observable Environments

- The agent executes its plan and maintains its belief state as it goes
- After each action it predicts, then filters by the percept it actually received
- This is state estimation, and the same recursive update returns in filtering later in the book

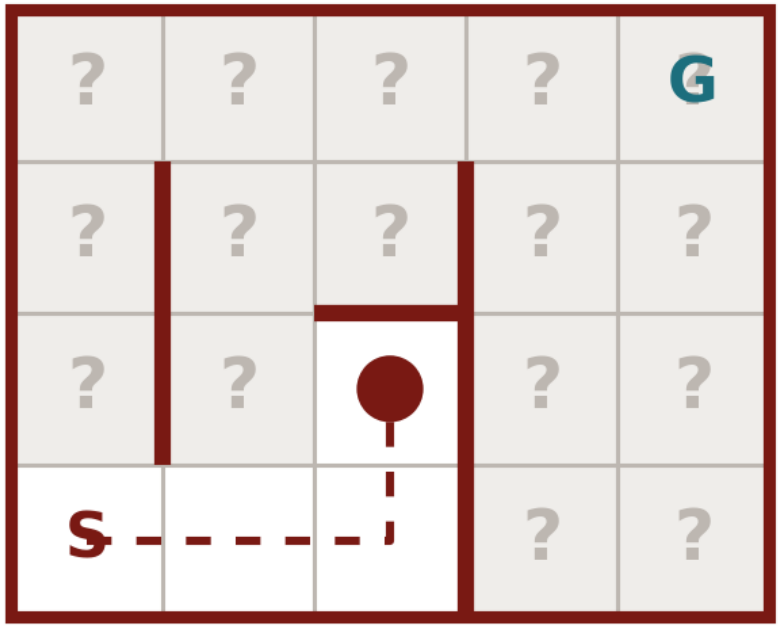


Online Search and Unknown Environments

Acting in order to find out

Online Search Problems

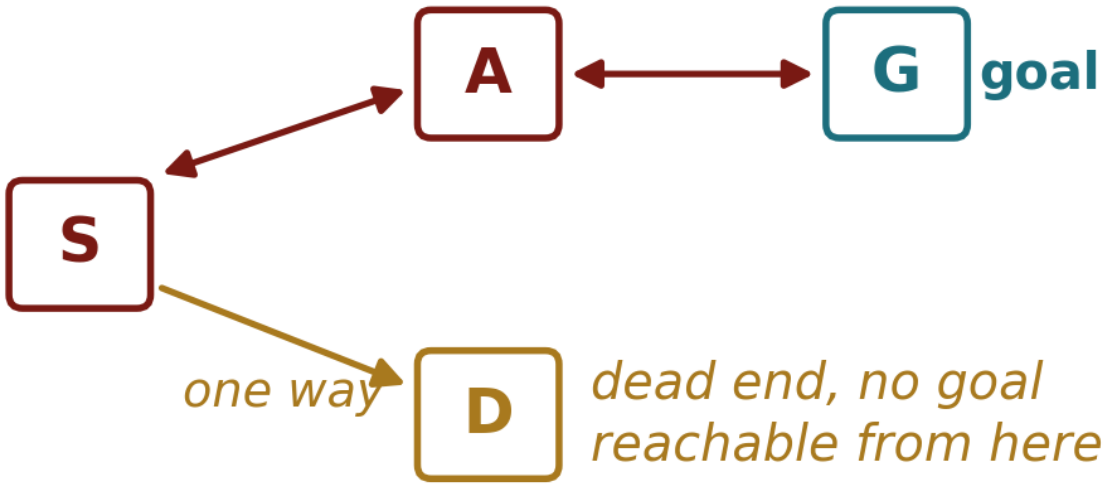
- The agent knows the actions available in the state it currently occupies
- It learns the cost of an action only after taking it
- It can tell whether the state it is in is a goal
- It cannot see where an action leads until it goes there, so it must act in order to learn



the agent knows only the squares it has walked

Competitive Ratio and Dead Ends

- The competitive ratio compares the cost of the path actually taken with the cost of the best path if the map had been known
- No algorithm can bound that ratio in the worst case, since an adversary can always hide the goal
- A dead end makes it unbounded, so we usually assume the state space is safely explorable



Online Depth First Search

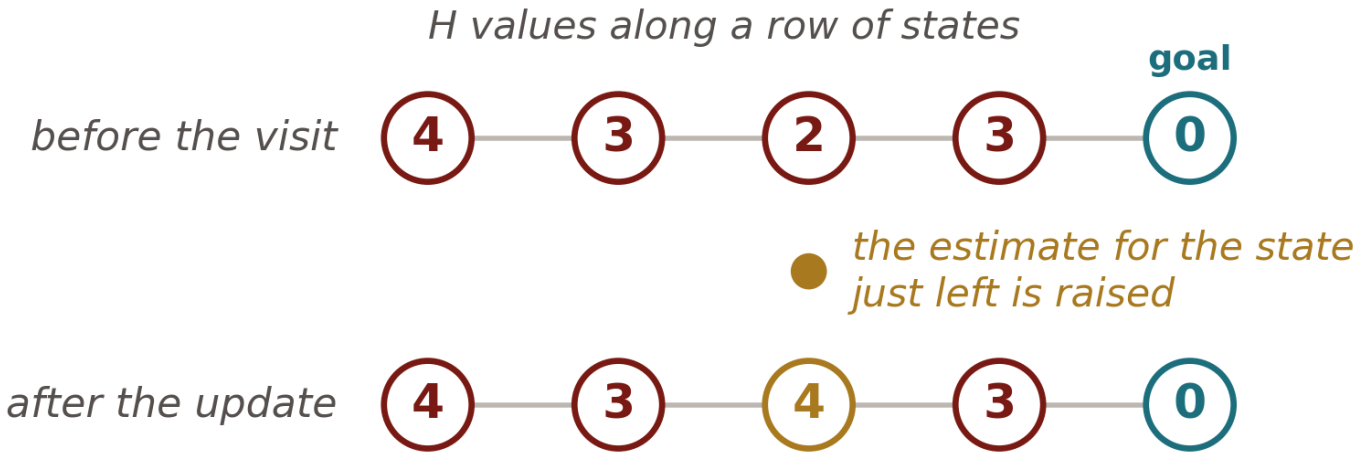
- The agent can only expand the state it is physically standing in
- It keeps a table of where each action from each visited state led, building the map as it walks
- Backtracking means walking back, not jumping back, so it needs a record of how it arrived
- It works only when every action can be undone, and it can wander a long way before finding a nearby goal

Online Local Search

- Hill climbing is already an online method, since it keeps only the current state and looks at its neighbours
- But a local minimum traps it, and it cannot use random restarts, because it cannot teleport to a new state
- A random walk does eventually escape, though it can take exponentially many steps to do so

Learning While Acting

- Store an estimate H(s) of the cost to reach a goal from each state visited
- On leaving a state, update its estimate to the cost of the best action plus the estimate of where that action leads
- Untried actions are assumed cheap, so the agent explores what it has not seen
- Repeated visits raise the estimates in a local minimum until it fills in and the agent walks out



Learning While Acting: Examples

- SLAM: Simultaneous Localization and Mapping
  - https://www.youtube.com/watch?v=34n1tF5OtQU
  - https://www.youtube.com/watch?v=_GdBWf0e8jU

- RL: Reinforcement Learning