Beyond Classical Search: Local Search

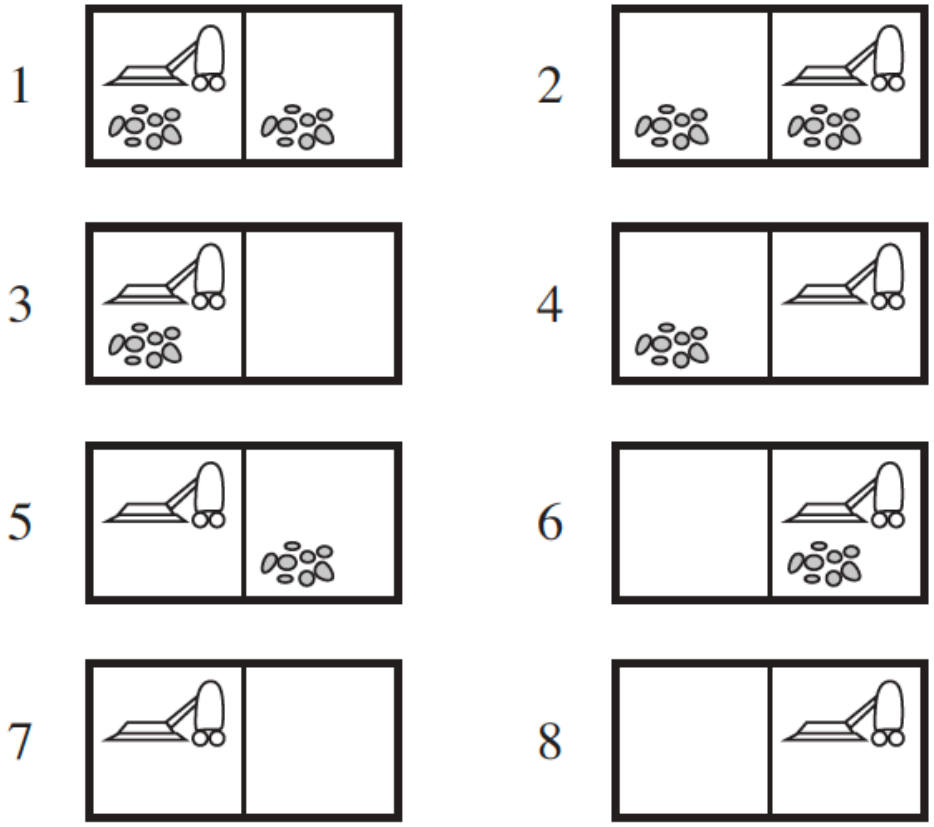
Part 2: Nondeterministic actions, partial observability,online search

CMPSCI 383
September 27, 2011

Today's lecture

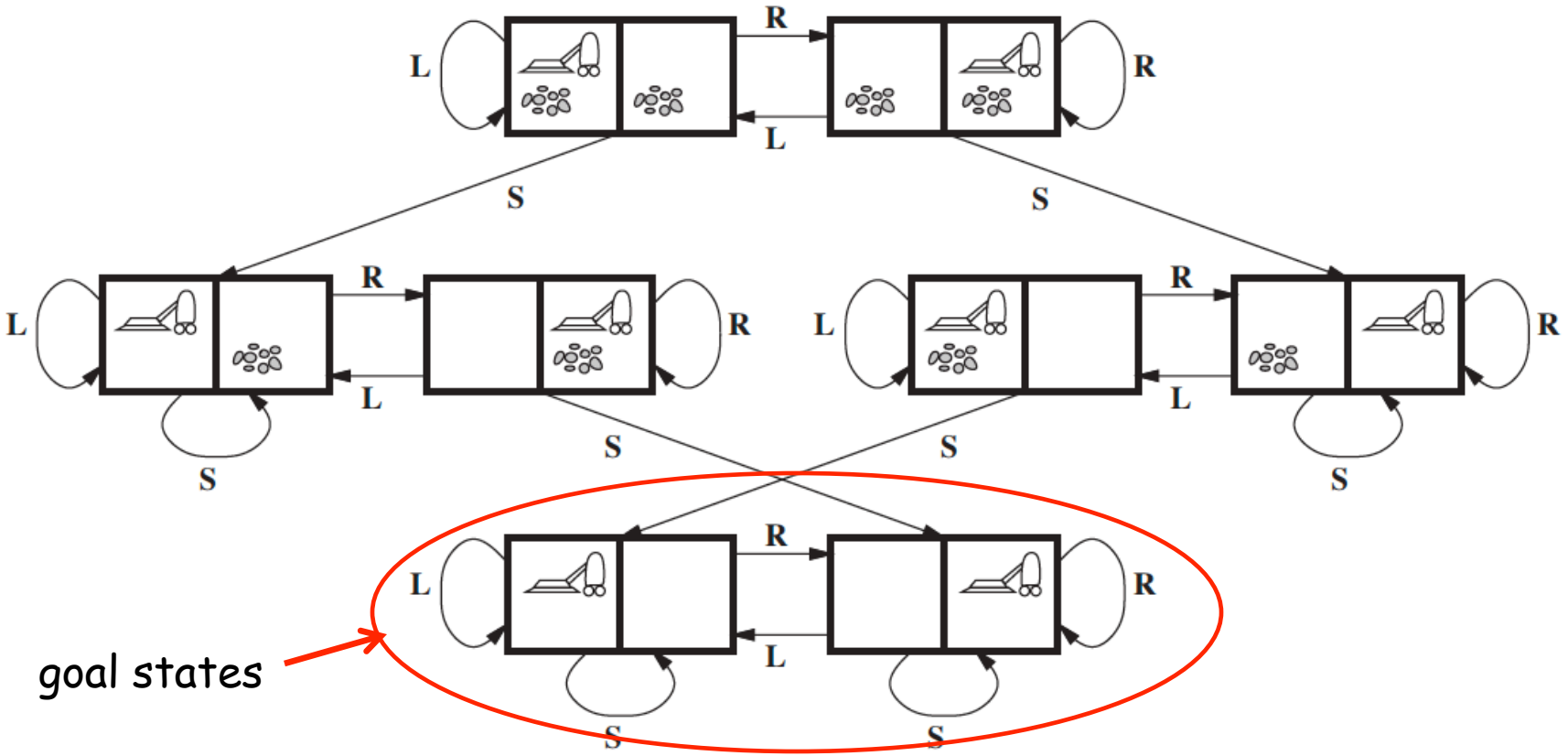
- Nondeterministic actions
- Partial Observations
- Online Search

The Vacuum-Cleaner World



Vacuum-Cleaner World Transition Model

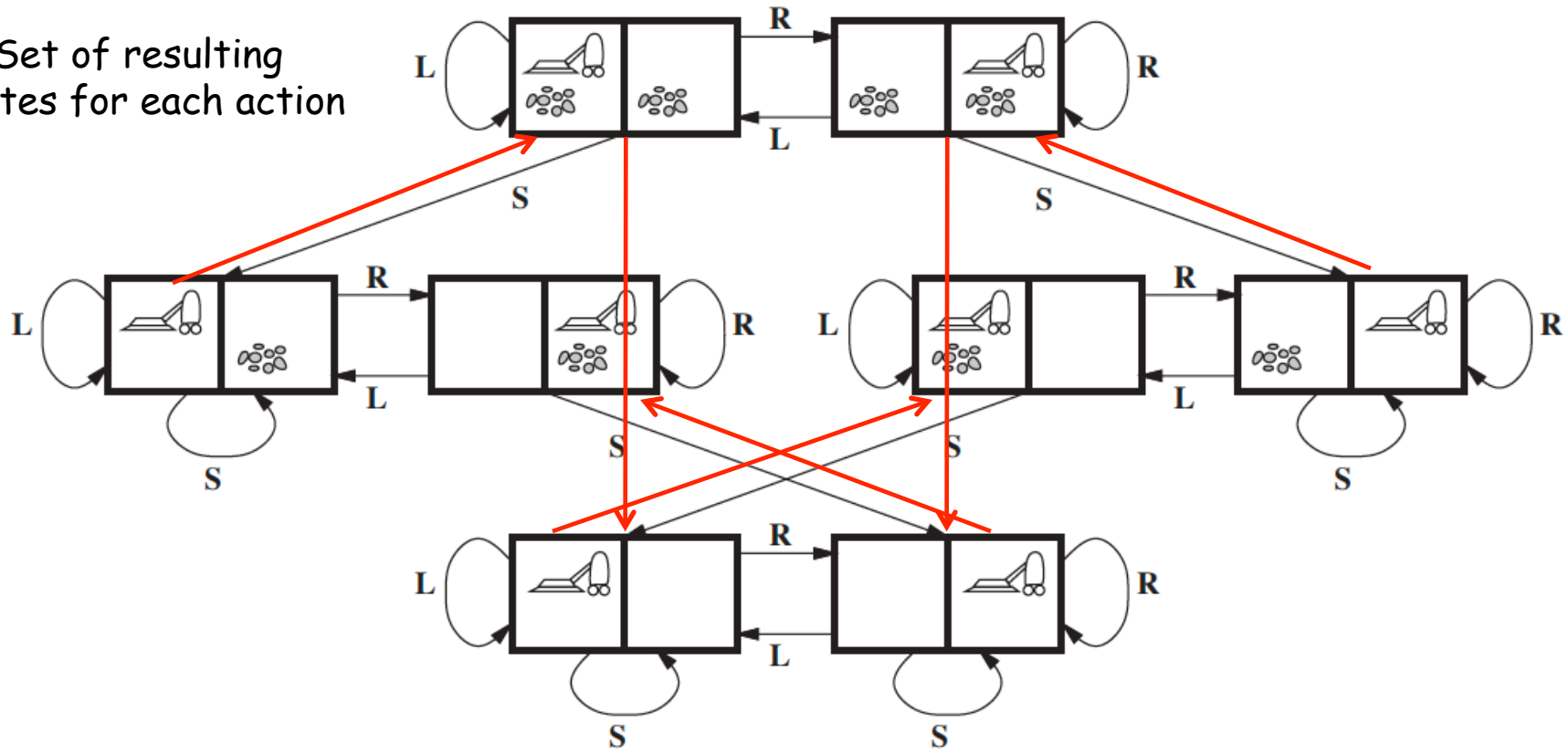
Deterministic actions, observable, known



Erratic Vacuum World

Suck action: applied to a dirty square, cleans the square but sometimes cleans adjacent square; applied to a clean square, sometimes deposits dirt.

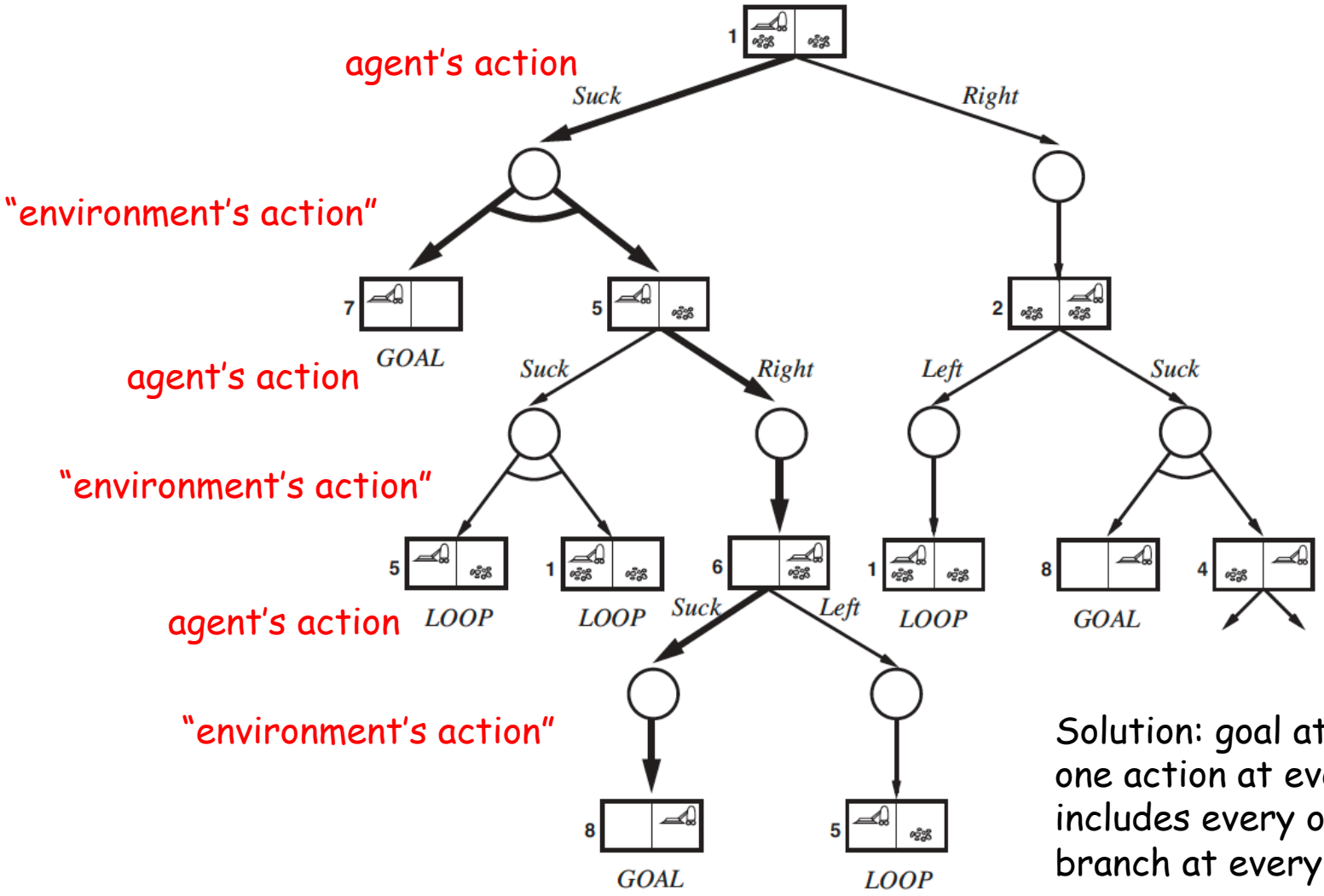
Set of resulting states for each action



Contigency Plan

- In deterministic case, solution is a sequence of actions.
- In nondeterminsitic (and stochastic) case, solution is a strategy or policy (or contingency plan, or control policy):
  - If states can be observed: a function from relevant states to actions:
    Π : States → Actions
  - If only "percepts" available:
    Π : Percepts → Actions
  - Can be expressed using **if-then-else** statements

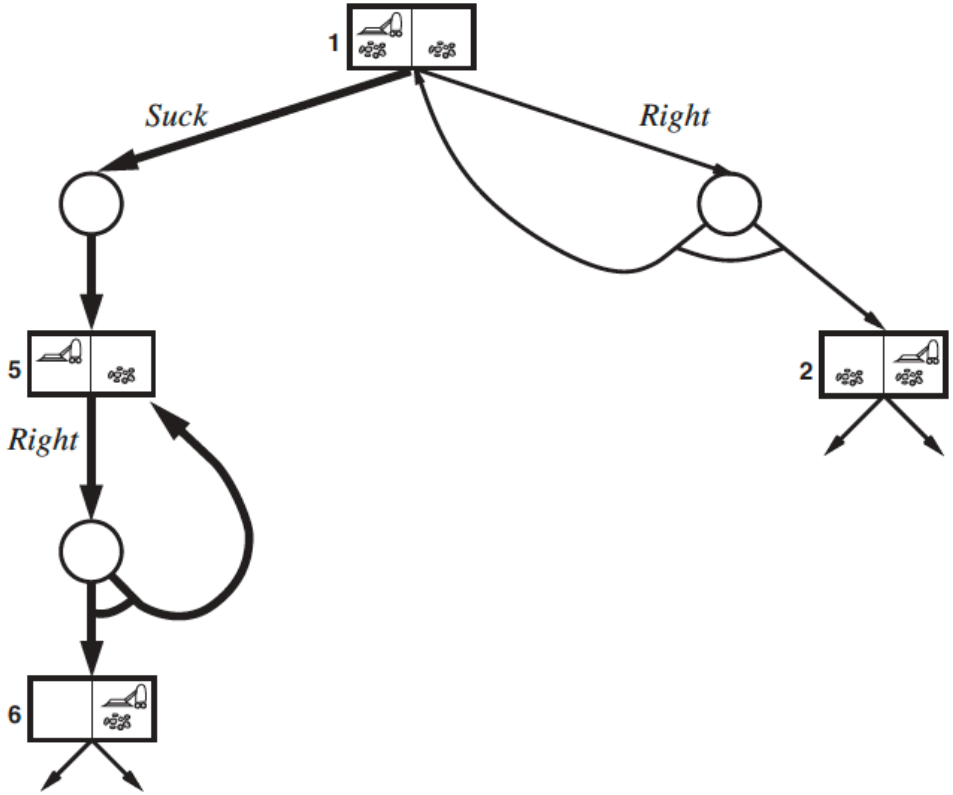
And-Or Tree



Solution: goal at every leaf, one action at every OR node, includes every outcome branch at every AND node.

Slippery Vacuum World

- Actions sometimes fail
- Need a cyclic solution: keep trying!



Partial Observations

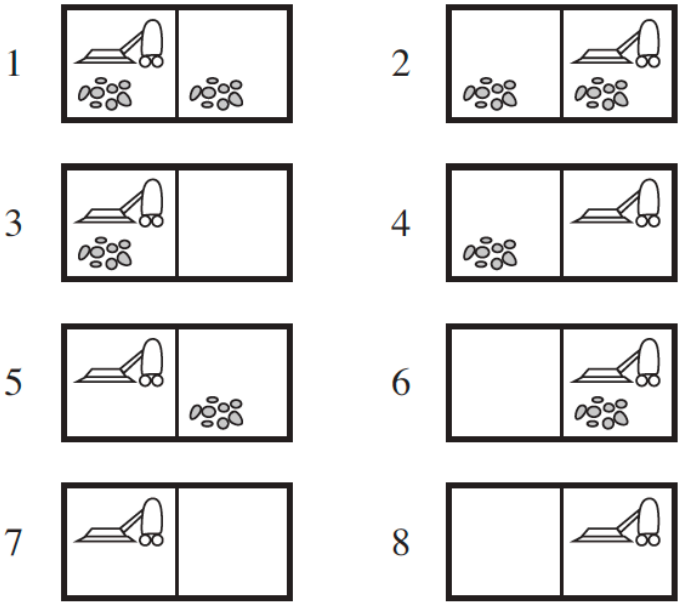
- E.g. No observations (!)
- Sensorless vacuum cleaner: has a model of the world but never knows what the state is.

Possible initial states {1, 2, 3, 4, 5, 6, 7, 8}

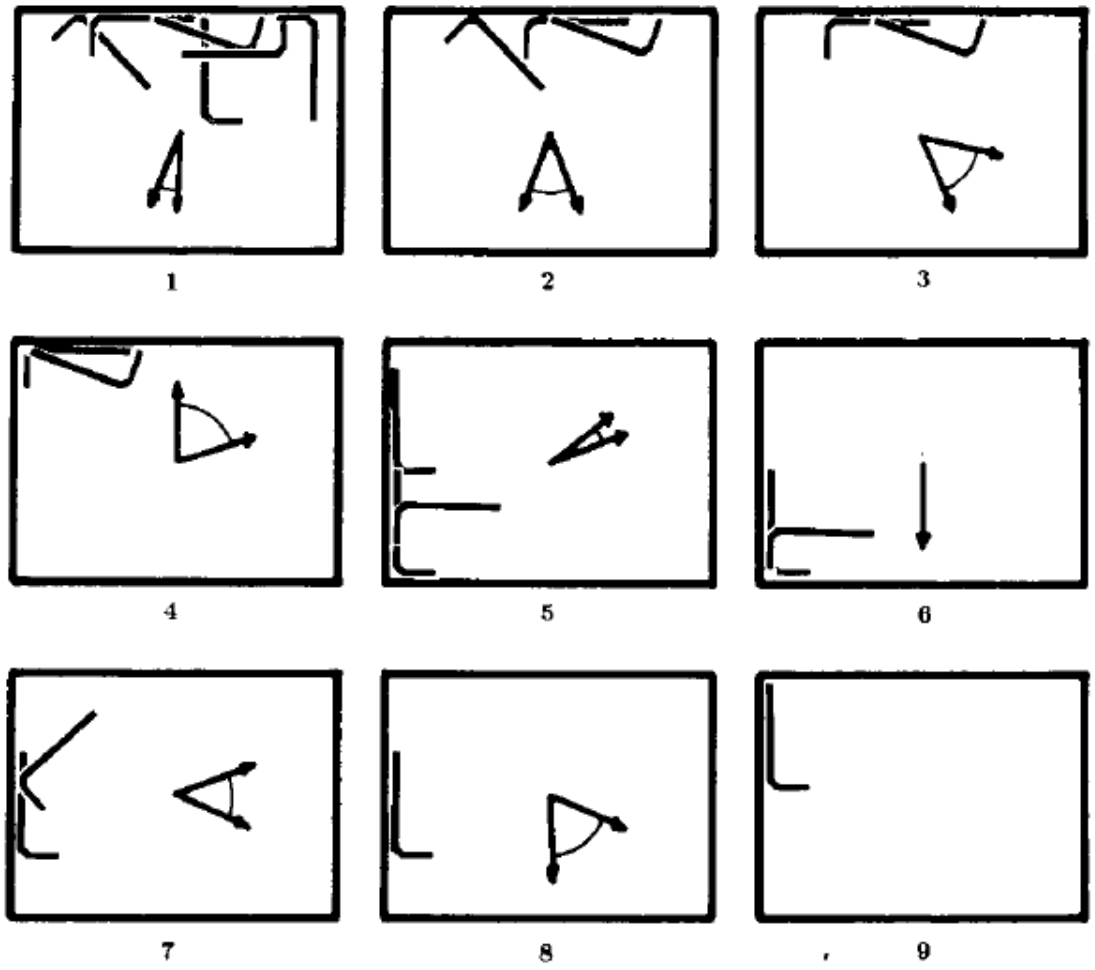
Possible states after action **right** {2, 4, 6, 8}

Possible states after action **right**, **suck** {4, 8}

Possible states after action **right**, **suck**, **left**, **suck** {7}



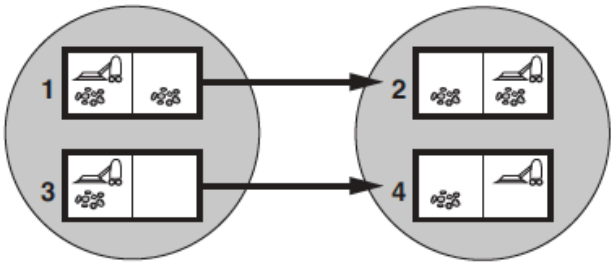
Sensorless Manipulation



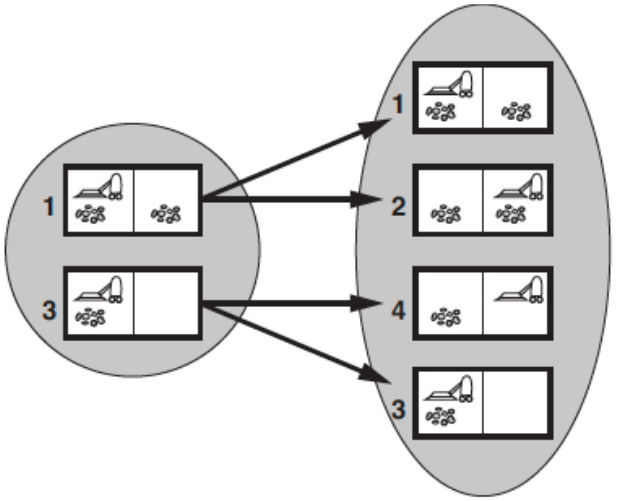
Erdmann and Mason 1988

Belief States

Predicting next belief state



sensorless vacuum with deterministic action **right**



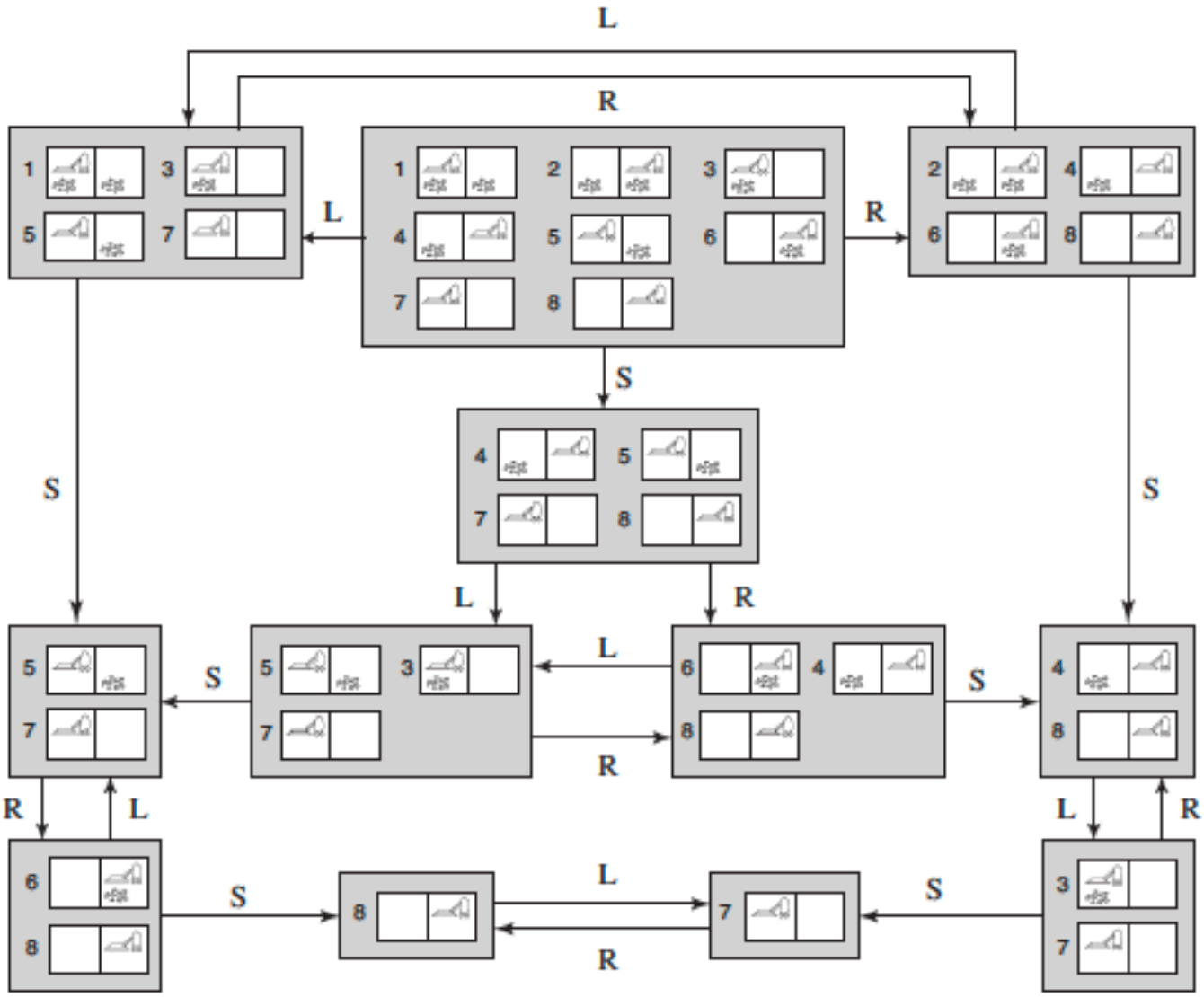
sensorless vacuum in slippery world

Reachable part of belief-state space
(deterministic, sensorless vacuum)

completely observable!

self-loops left out

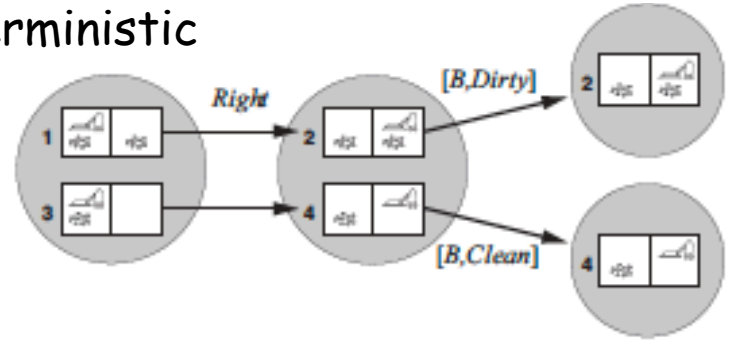
How big is the entire belief-state space?



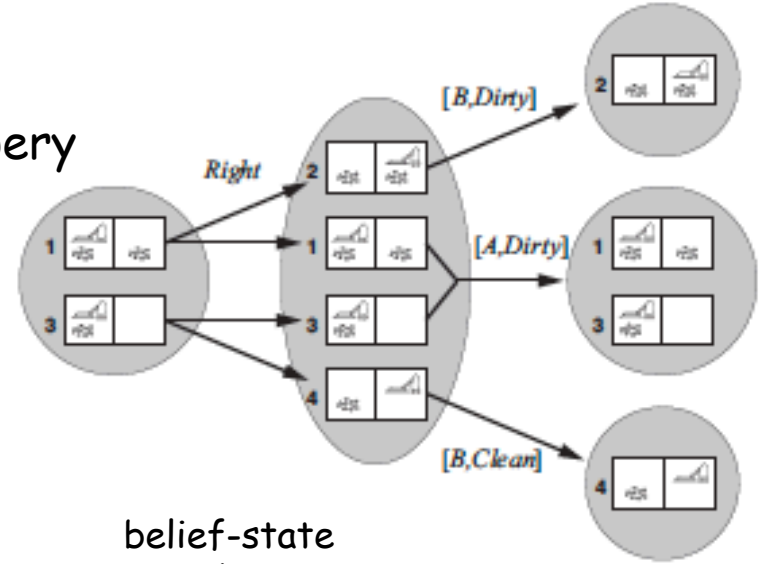
Add some observations

Percept(s) =
[position, local dirt]

deterministic



slippery



belief-state prediction

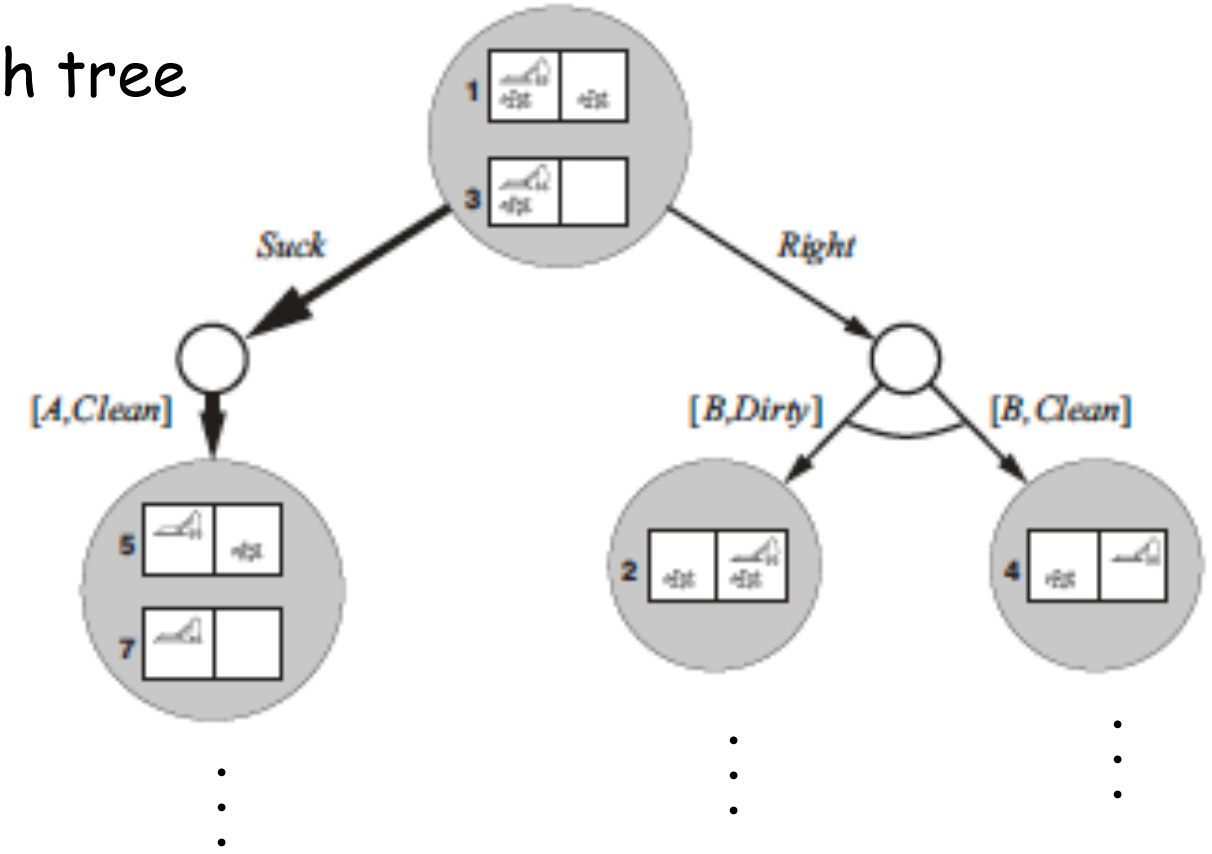
observation prediction

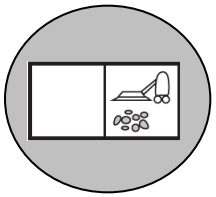
belief-state update

Note: for an agent, percept is given by the environment

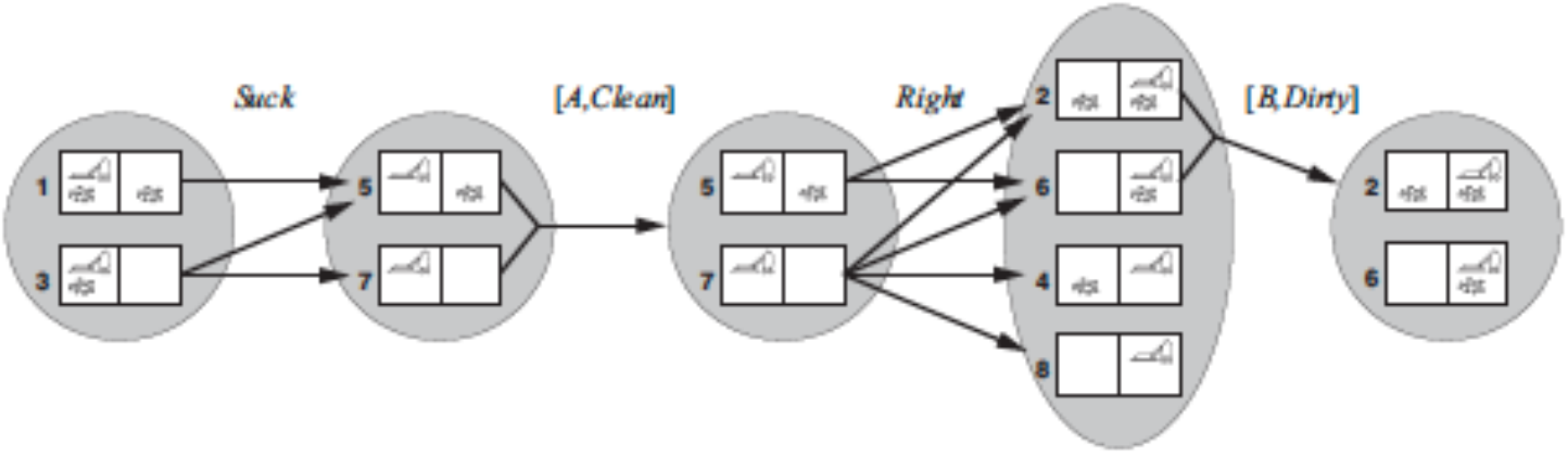
Solving Partially Observable Problems

And-Or search tree fragment



suck, right, if b=  then suck, else done

Belief-state maintenance for an agent



Two update cycles (for "kindergarten vacuum world":
any square can become dirty at any time, unless the vacuum is cleaning it)

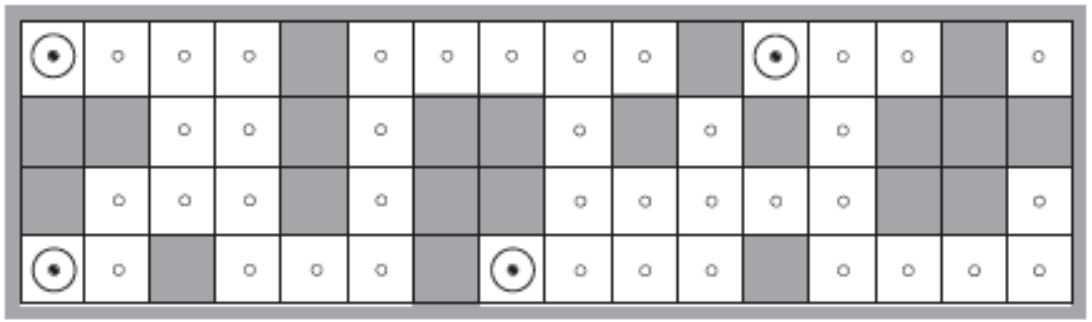
Note: b-s update depends only on previous b-s and current observation.

Robot Localization

senses obstacles in
4 adjacent squares

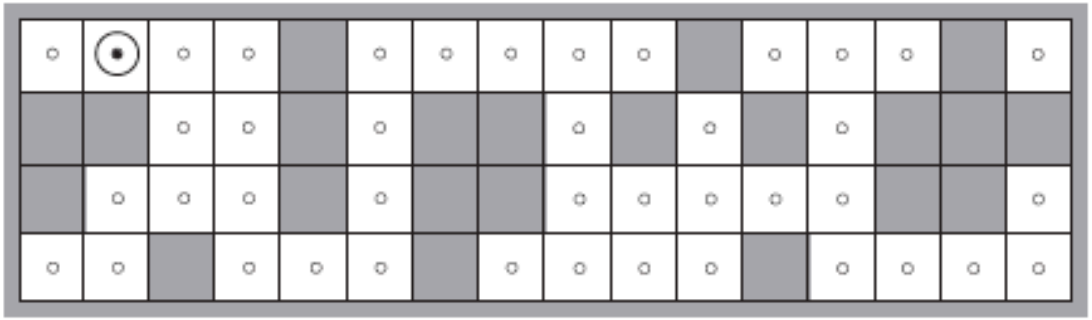
moves randomly

First percept is NSW



(a) Possible locations of robot after E_1 = NSW

What is b after one move?
What is b after second percept NS?



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

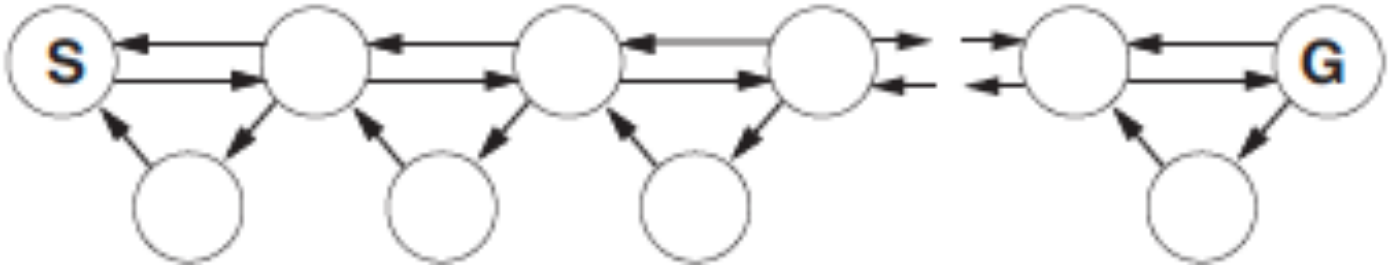
Online Search

- Agent is in the world ("situated")
- Exploration problem: agent may not know what states exist, how its actions affect states, etc.
- Assume full observability and agent knows:
  - Available actions in each state
  - Step costs
  - Goal test
- To determine next state, agent has to actually execute an action.
- E.g. "motor babbling"
- Deadends?

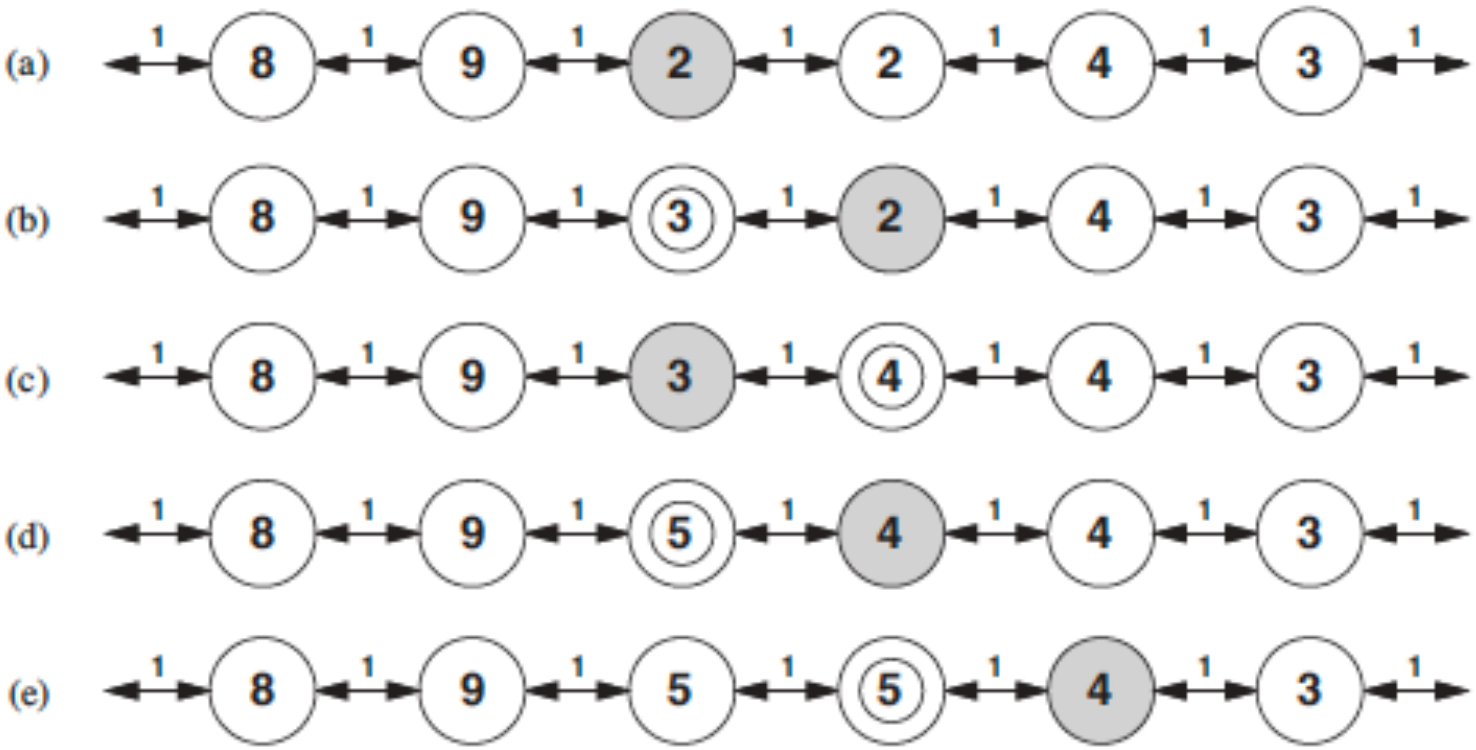
Online Search

- Good to expand nodes in local order: e.g. depth-first search
- Build a **map**, or **model**
  - `Result[s,a]`
- "Explore/Exploit" tradeoff
- "Optimism under uncertainty"

Random exploration can be very slow!



LRTA*



Summary Chapter 4

- Local search
- Hill climbing
- Local extrema. plateaux, etc.
- Simulated annealing
- Linear programming
- Convex optimization
- Genetic algorithm
  - Mutation
  - Crossover
- Genetic programming
- Nondeterministic environment
- Partial observability
- Contingency plans
- Belief state
- AND-OR search
- Sensorless problems
- Exploration problems
- Online Search

Exercise 3.16

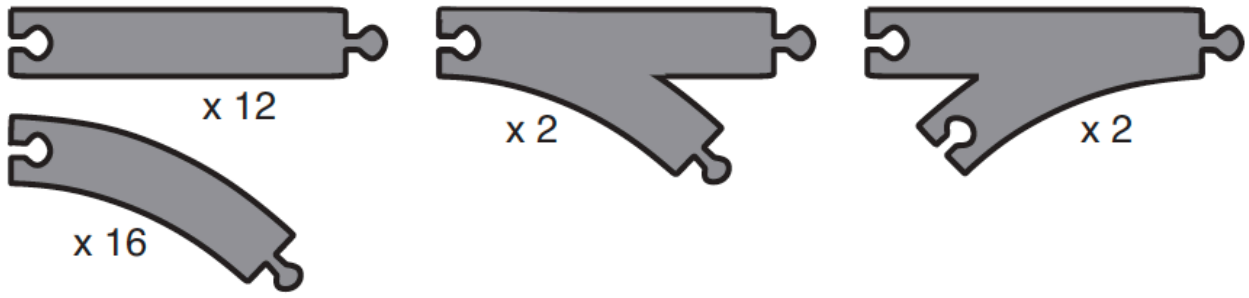


Figure 3.32 FILES: figures/brio.eps (Wed Nov 4 14:35:23 2009). The track pieces in a wooden railway set; each is labeled with the number of copies in the set. Note that curved pieces and "fork" pieces ("switches" or "points") can be flipped over so they can curve in either direction. Each curve subtends 45 degrees.

Connect into a railway that has no overlapping tracks and no loose ends.

Next Class

- Adversarial Search (Games)
- Secs. 5.1-5.4