CSC242: Introduction to Artificial Intelligence

Lecture 1.6

Please put away all electronic devices

Announcements

- Unit 1 Exam: Next class
- Project 1 due that day 1159PM

Systematic Search

- Enumerates paths from initial state
- Records what alternatives have been explored at each point in the path

Good: Systematic $\rightarrow$ Exhaustive

Bad: Exponential time and/or space

Local Search

- Evaluates and modifies a small number of current states
- Does not record history of search (paths, explored set, etc.)

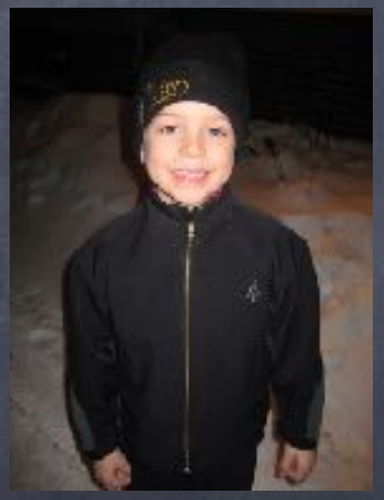
Good: Very little (constant) memory

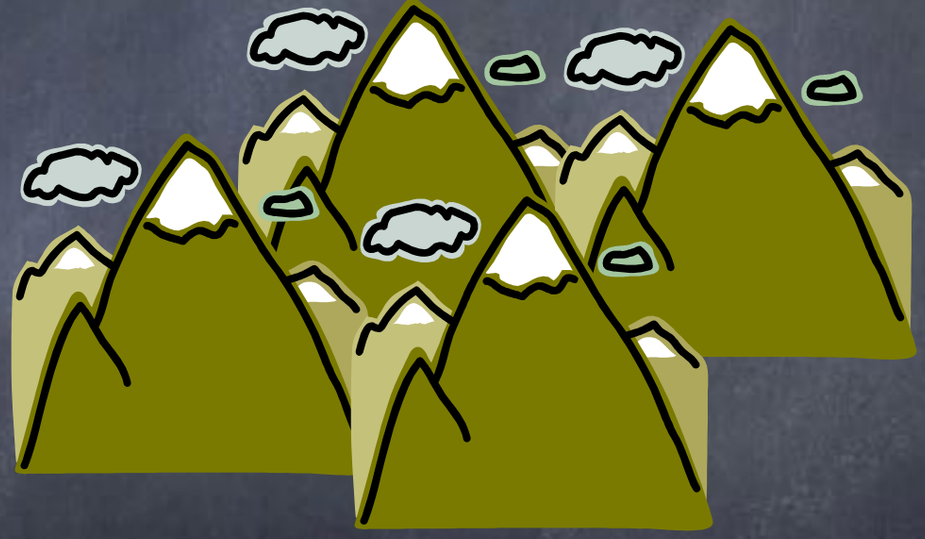
Bad: May not explore all alternatives

=> Incomplete

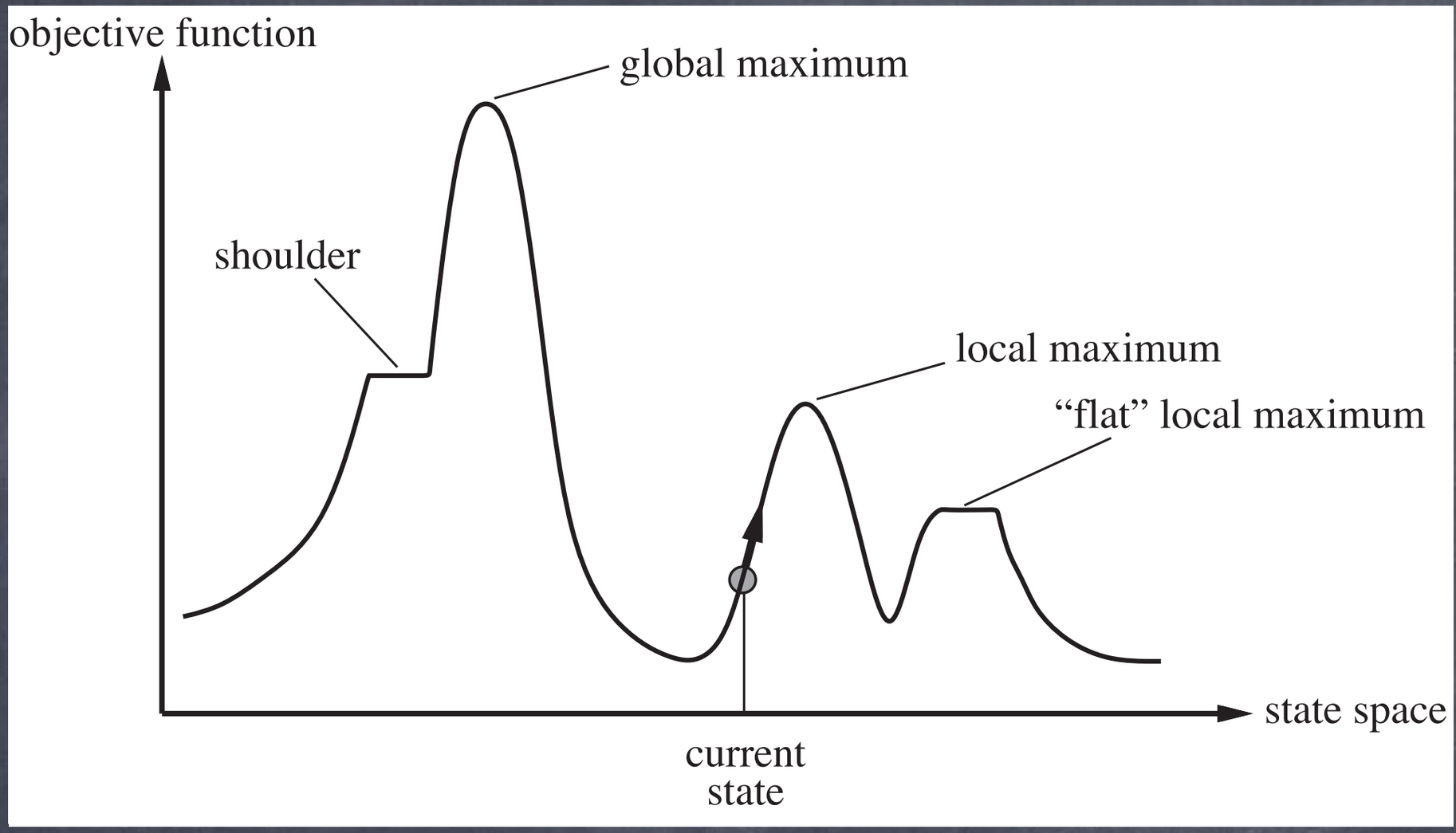
Hill-climbing a.k.a. Greedy Local Search

- Move through state space in the direction of increasing value ("uphill")





State-space landscape


objective function
global maximum
shoulder
local maximum
"flat" local maximum
state space
current
state

Random Restart Strategy

```
State randomRestart(Problem p) {
  while (true) {
    p.setInitialState(new random State);
    State solution = hillClimb(p);
    if (p.isGoal(solution)) {
      return solution;
    }
  }
}
```

Does it work? Yes (but)

How well does it work?

Prob of success = p Expected # of tries = $1/p$

= 0.14 ≈ 7

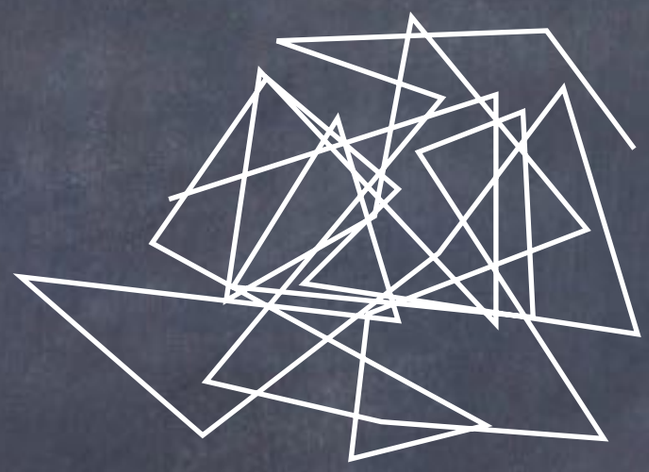
```
State randomRestart(Problem p) {
  while (true) {
    p.setInitialState(new random State);
    State solution = hillClimb(p);
    if (p.isGoal(solution)) {
      return solution;
    }
  }
}
```

Randomness

Stochastic hill climbing

Randomness in Search

Pure random walk



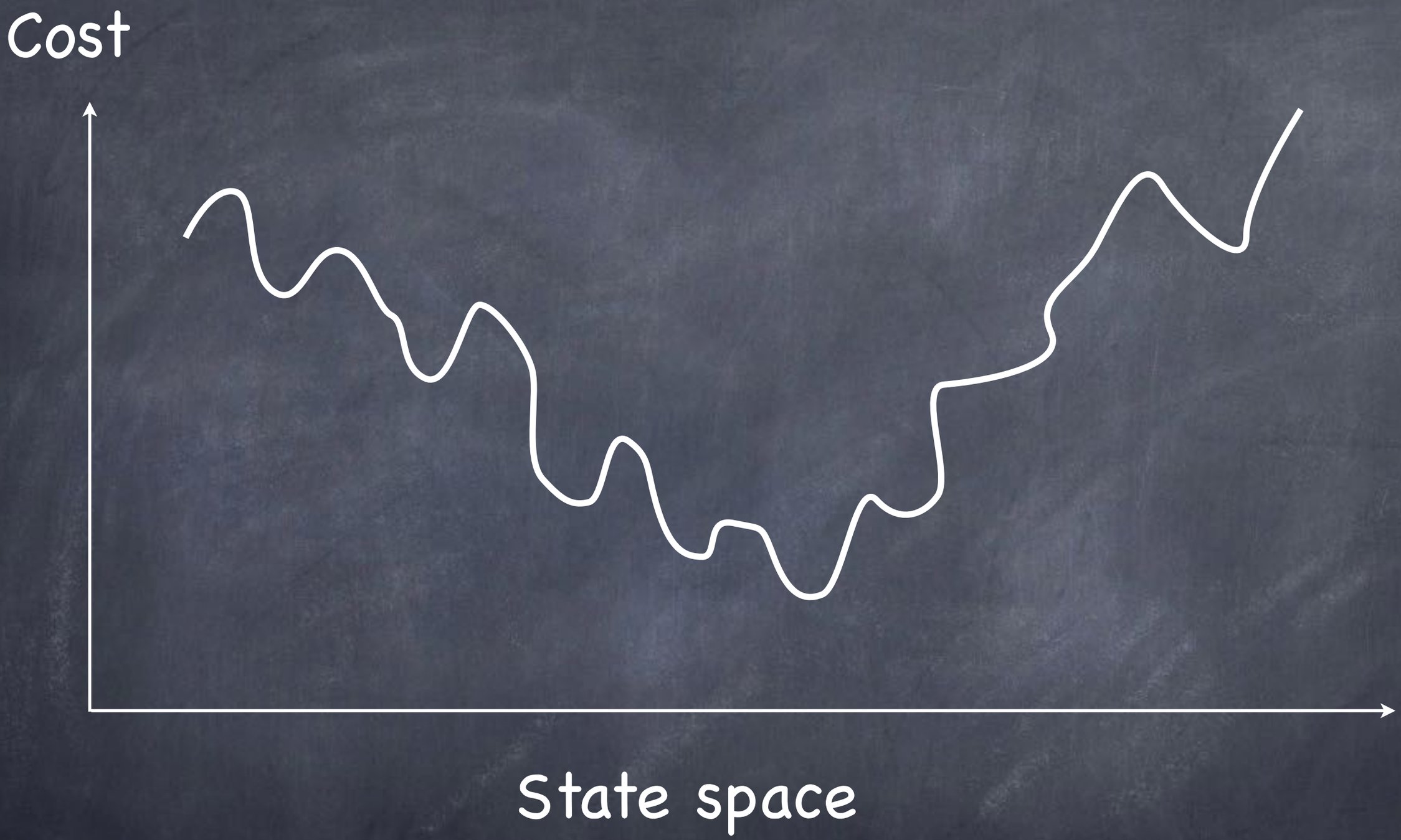
Complete,
but horribly slow

Greedy local search

Incomplete,
but fast

Simulated Annealing

- Greedy local search (Hill-descending)
- Select states with lower cost
- OR with some probability even if higher cost
- "High temperature": higher probability
- "Low temperature": lower probability
- "Cool" according to schedule


Cost
State space

Simulated Annealing

Complete? No.

Optimal? No.

"But if the schedule lowers T slowly enough, simulated annealing will find a global minimum with probability approaching one."

Local Search (cont.)

Local Search

- Evaluates and modifies a small number of current states
- Does not record history of search

Good: Very little (constant) memory

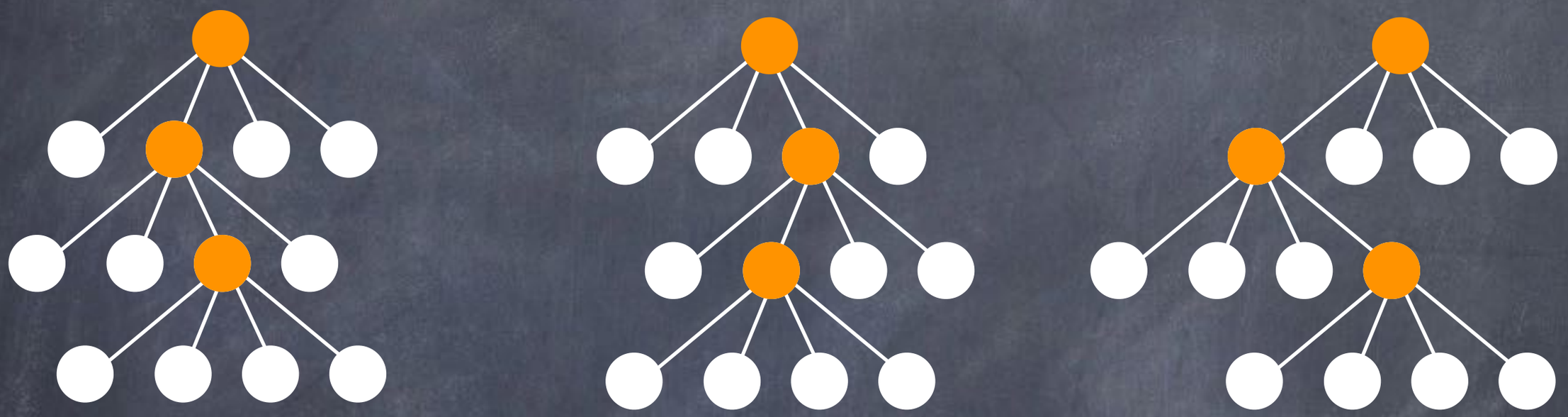
Bad: May not explore all alternatives

=> Incomplete

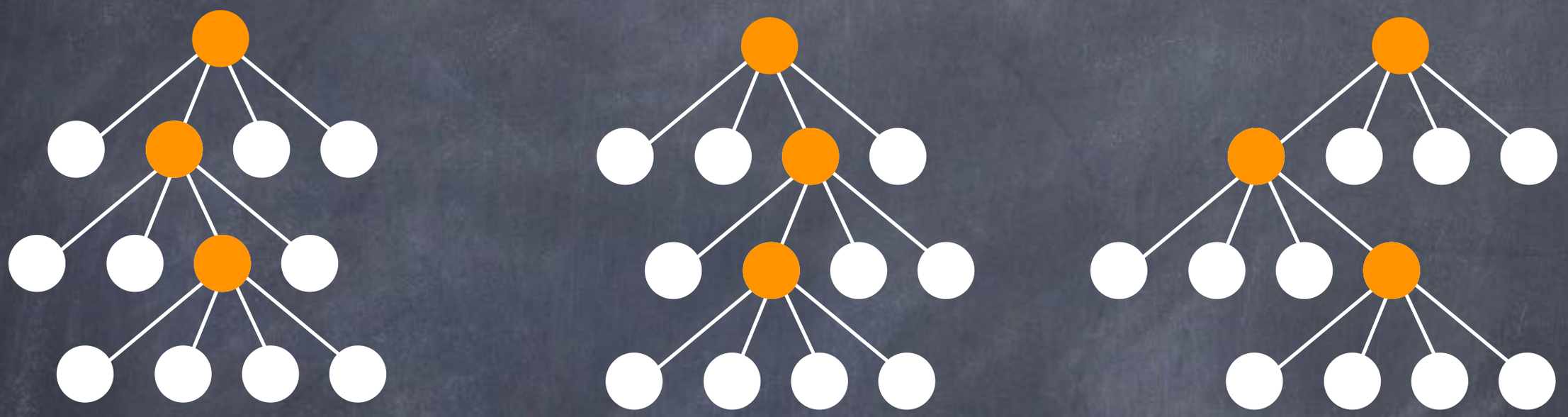
Local Search++

- Keep track of k states rather than just one

Parallel Local Search

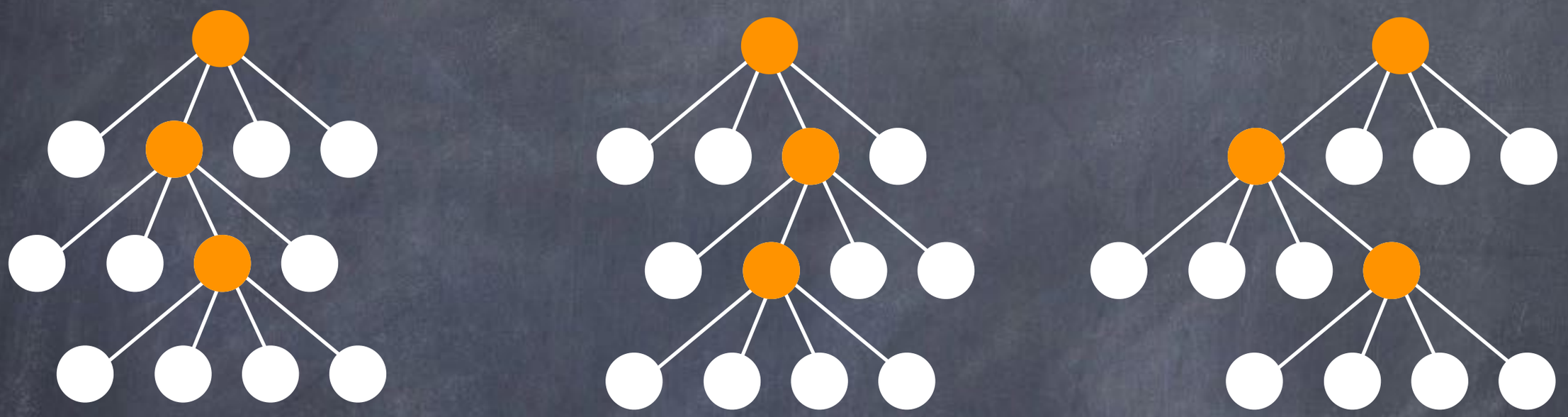


Parallel Local Search



Sequential: $1/k$

Parallel Local Search



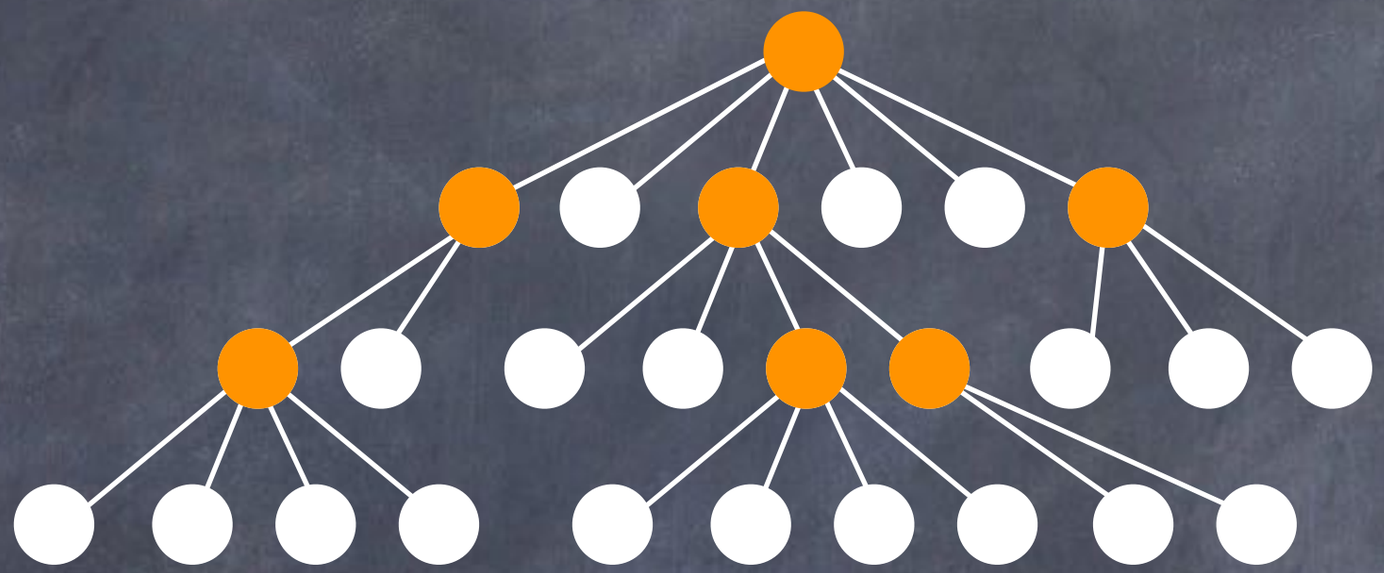
Sequential: $1/k$

Parallel: No extra cost

Local Search++

- Keep track of k states rather than just one
- At each step, generate all successors of all k states ($k \times b$ of them)
- Keep the most promising k of them

Local Search++



Local Beam Search

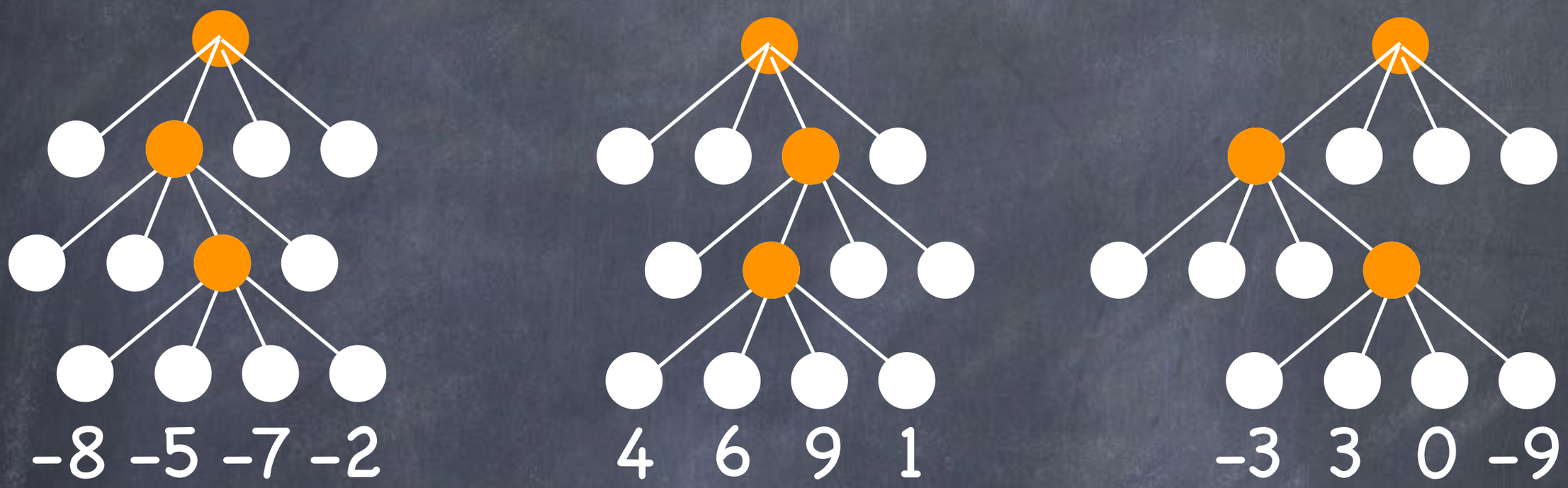
- Keep track of k states rather than just one
- At each step, generate all successors of all k states ($k \times b$ of them)
- Keep the most promising k of them
- k = "width of the beam"

Local Beam Search

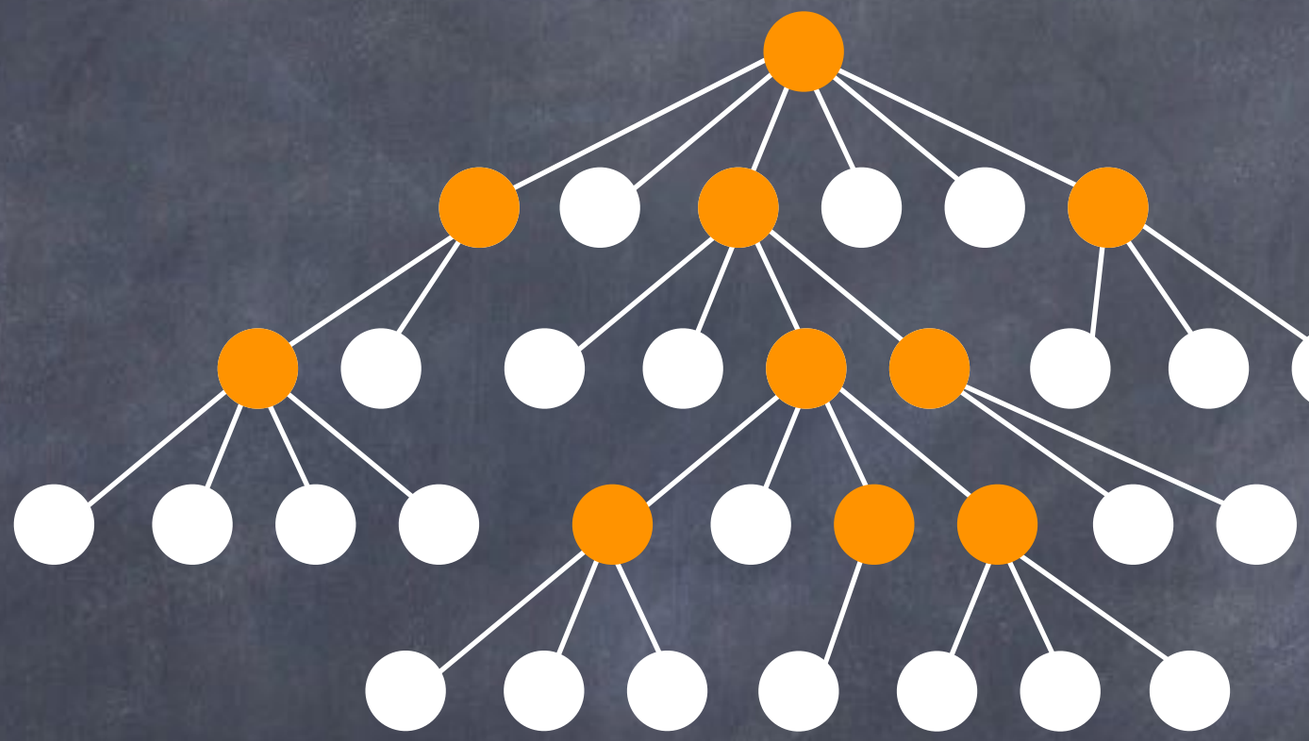
Local Beam Search

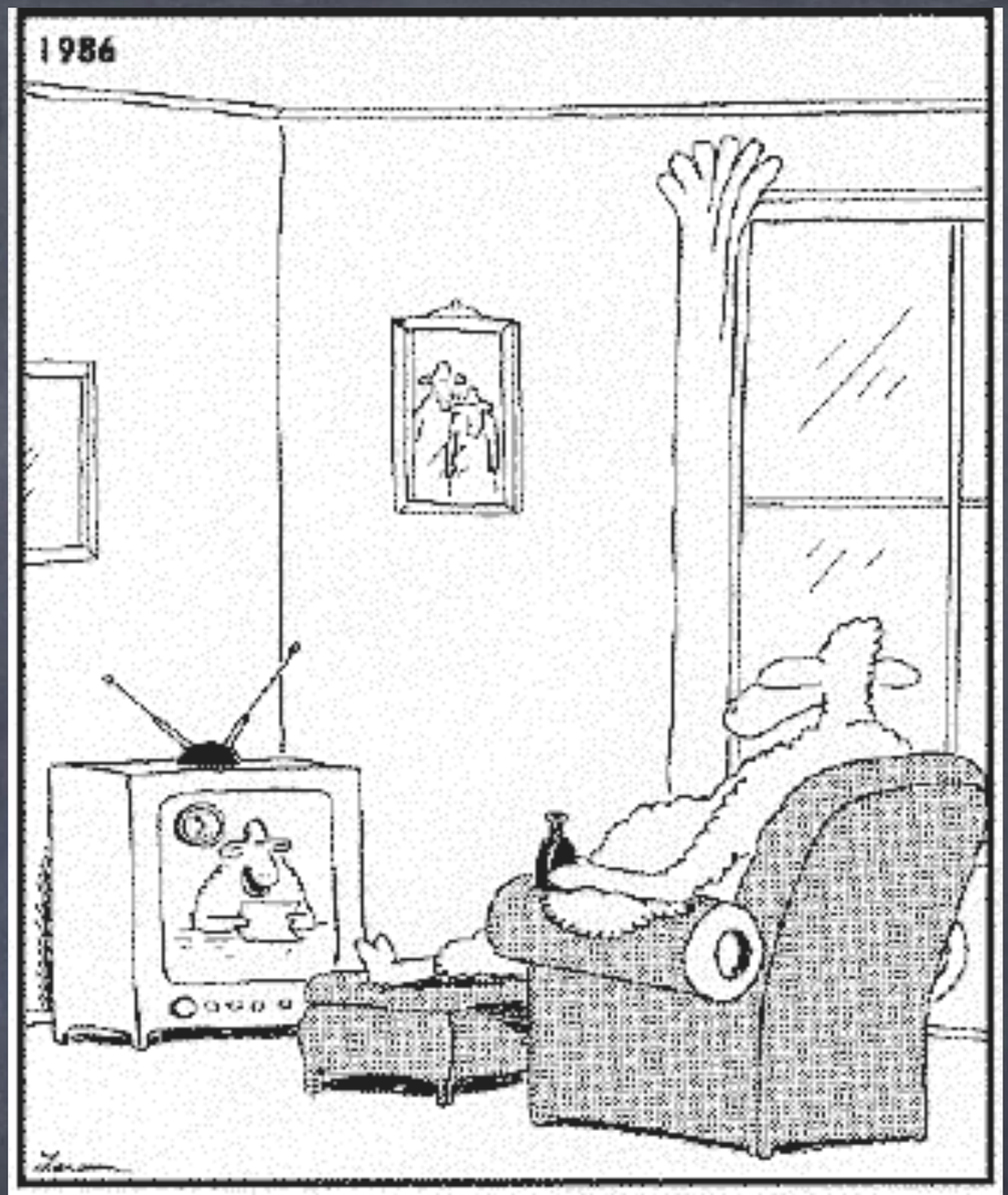
- Keep track of k states rather than just one
- At each step, generate all successors of all k states ($k \times b$ of them)
- Keep the most promising k of them
- k = "width of the beam"

Parallel Local Search



Local Beam Search



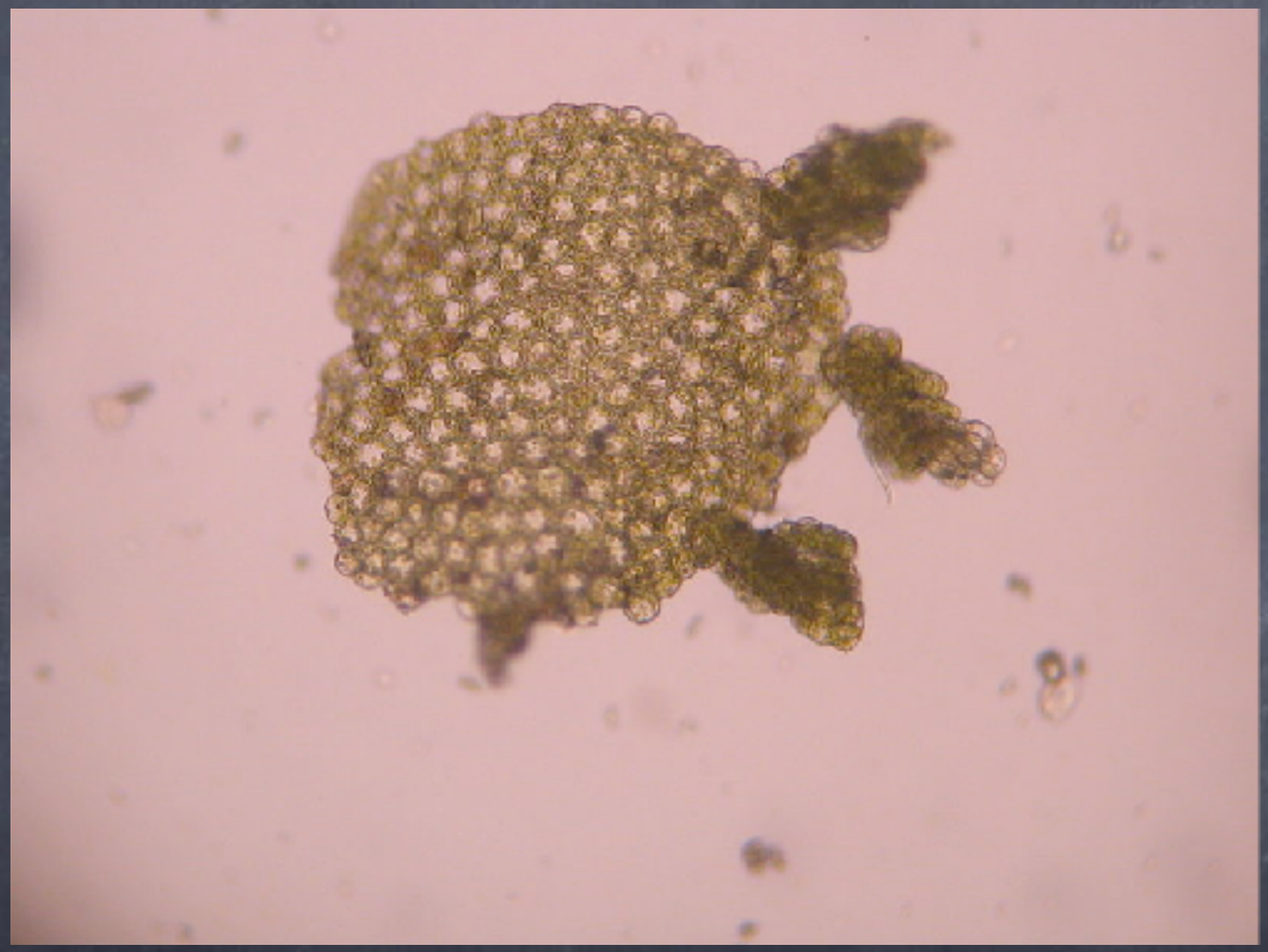


"And this report just in. ... Apparently, the grass *is* greener on the other side."

Local Beam Search

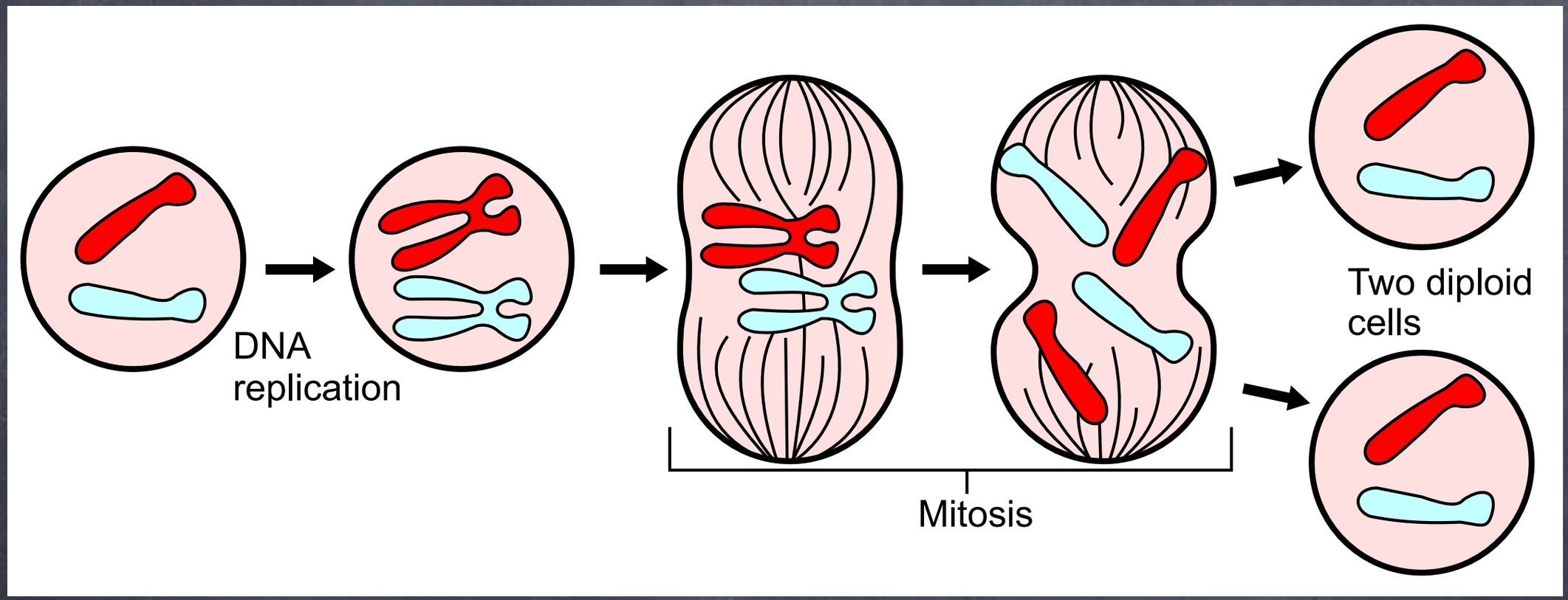
- Keep track of k states rather than just one
- At each step, generate all successors of all k states ($k \times b$ of them)
- Keep the most promising k of them
- k = "width of the beam"

Asexual Reproduction



Wikipedia

Mitosis

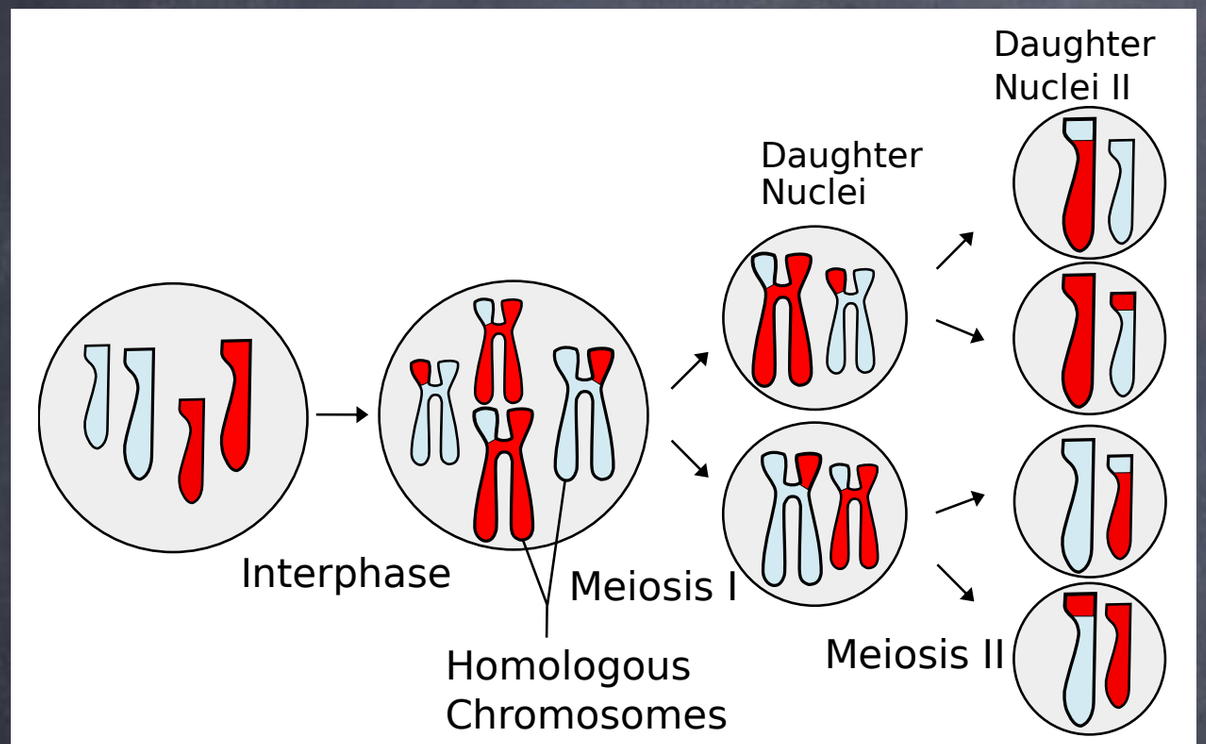


Wikipedia

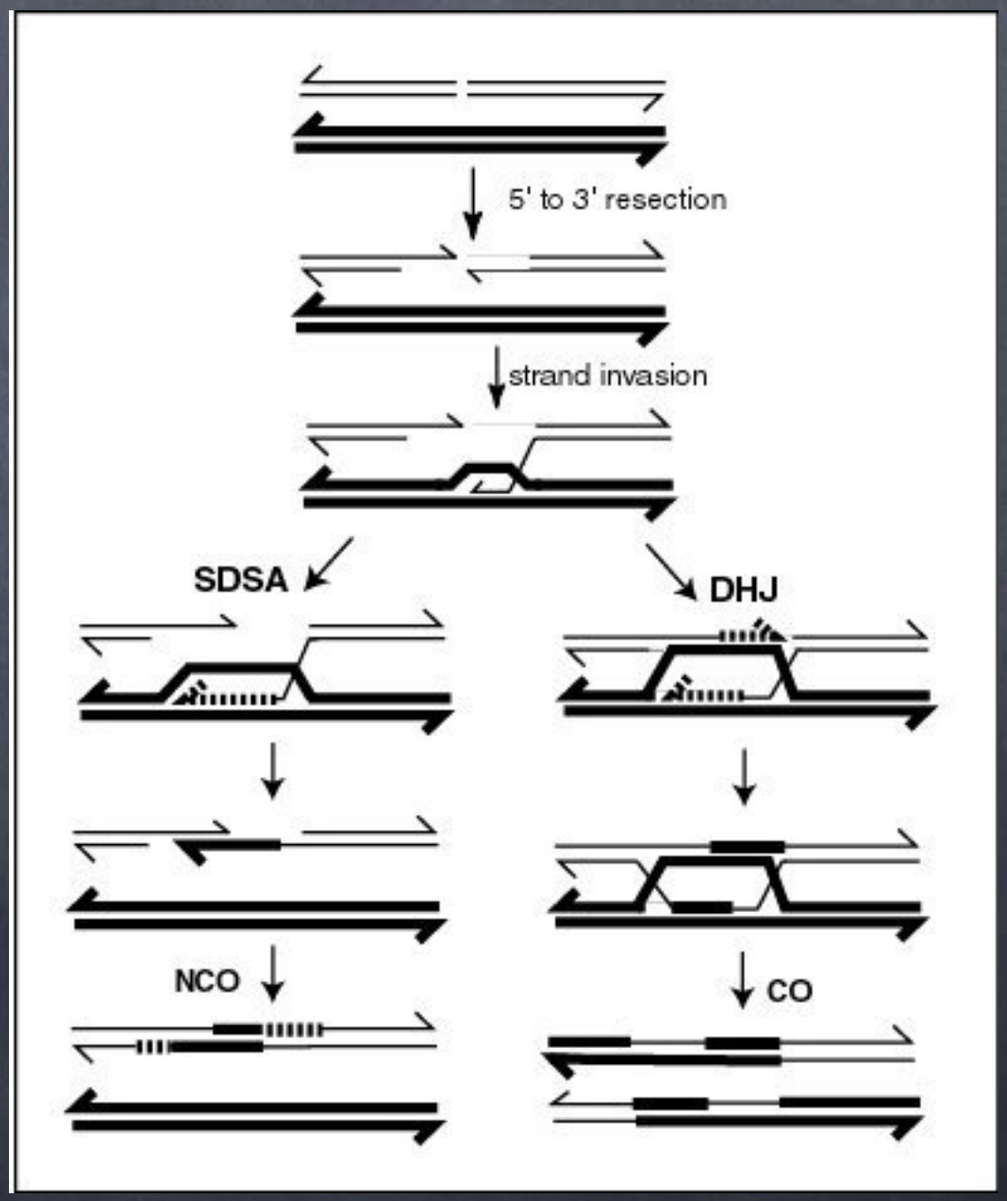
Sexual Reproduction

Wikipedia

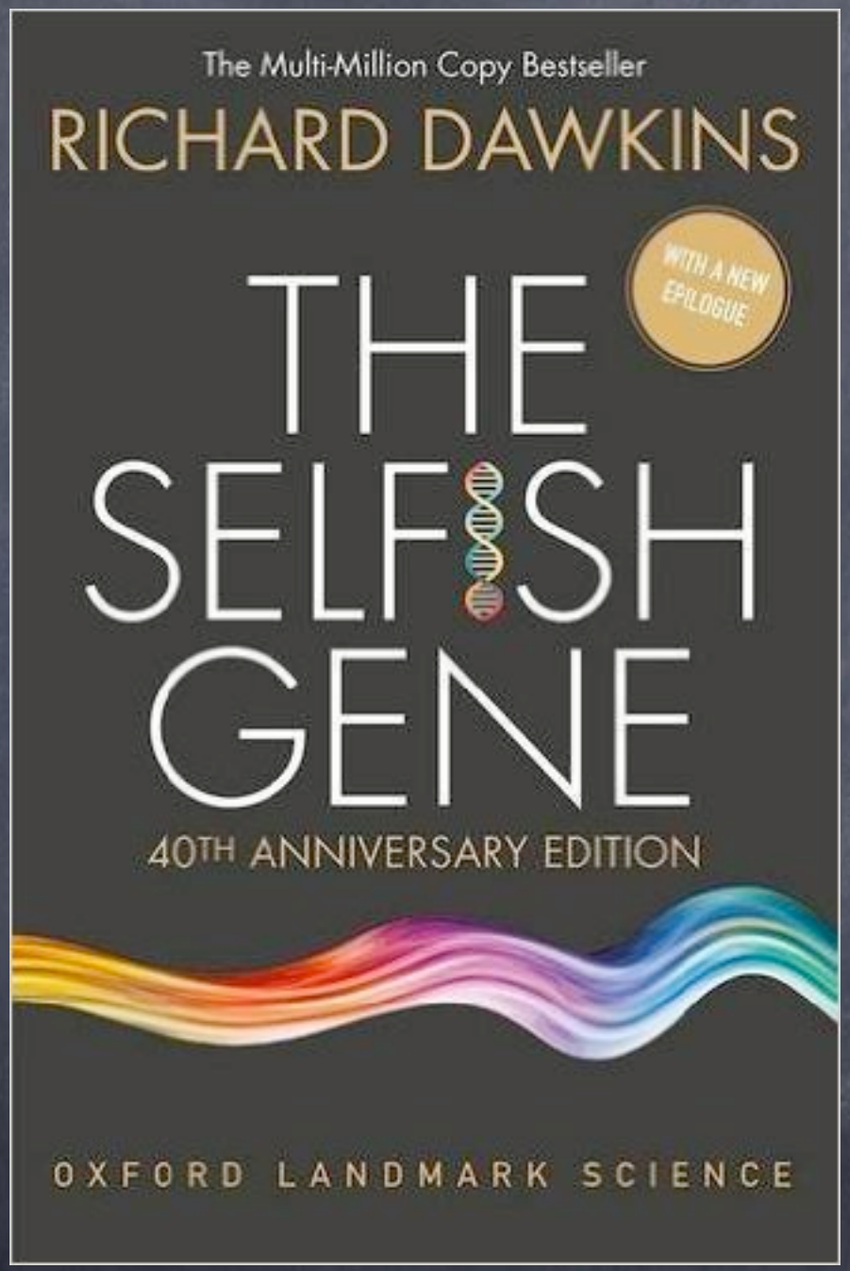
Meiosis



Recombination



Wikipedia


The Multi-Million Copy Bestseller
RICHARD DAWKINS
WITH A NEW EPILOGUE
THE SELFISH GENE
40TH ANNIVERSARY EDITION
OXFORD LANDMARK SCIENCE

Genetic Algorithms

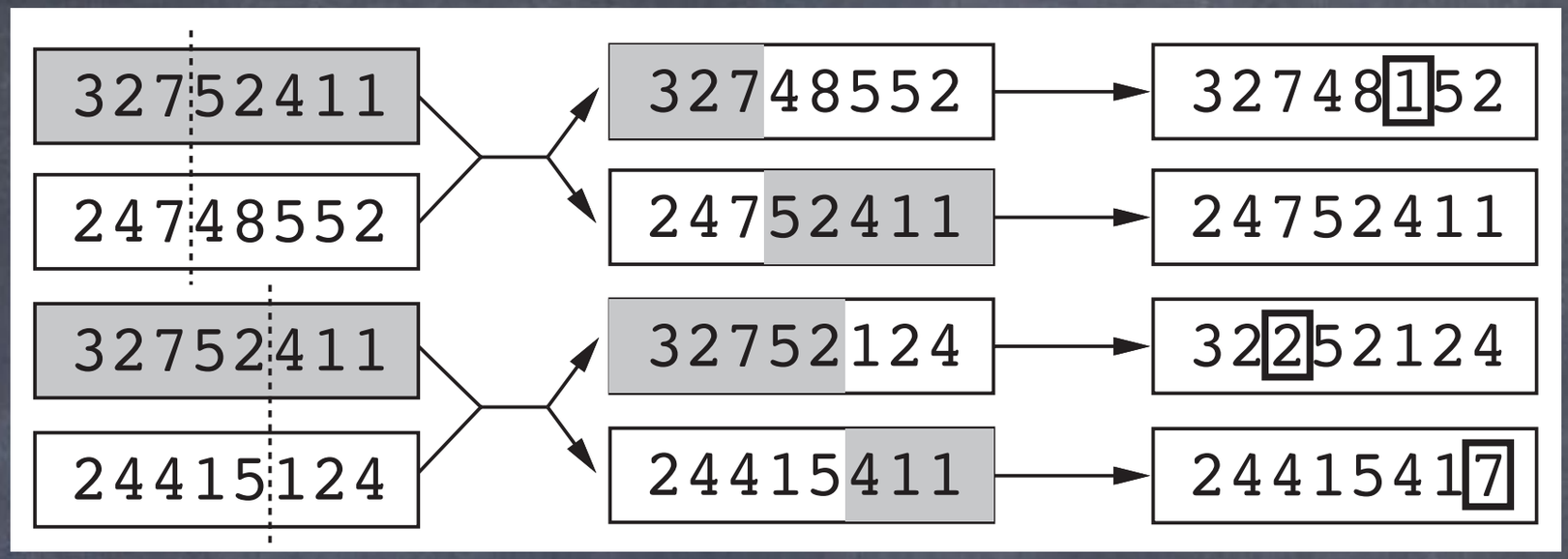
- Start with k random states
- Select pairs of states and have them "mate" to produce "offspring"
- Most fit (highest-scoring) individuals (states) reproduce more often

Genetic Algorithms

- States encoded as "chromosomes" (linear sequences, a.k.a. strings)

Genetic Algorithms

- States encoded as "chromosomes" (linear sequences, a.k.a. strings)
- During mating:
  - Crossover: swap chunks of state
  - Mutation: randomly change bits of state



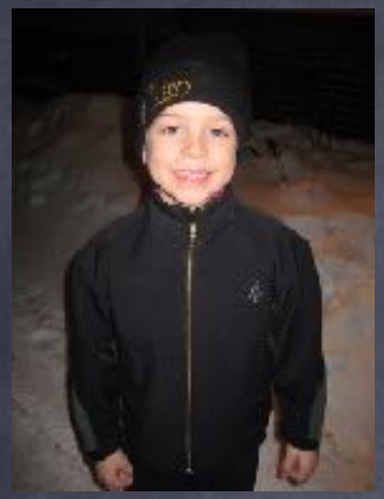
Crossover Mutation

Genetic Algorithms

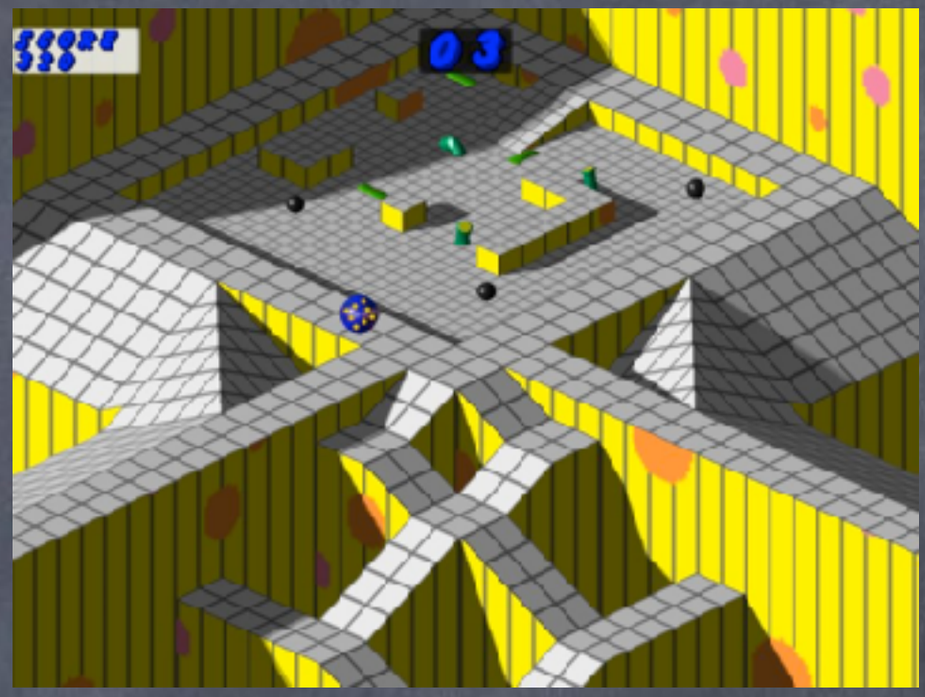
- States encoded as "chromosomes" (linear sequences, a.k.a. strings)
- During mating:
  - Crossover: swap chunks of state
  - Mutation: randomly change bits of state

GF on GAs

- A version of stochastic local beam search with a special way to generate successor states (motivated by a naive biology analogy)
- "Much work remains to be done to identify the conditions under which genetic algorithms perform well."

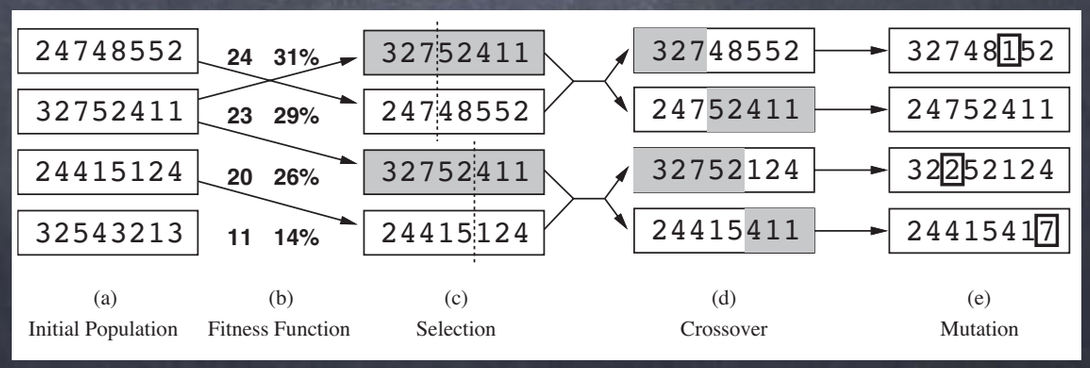


Hill-climbing
(Greedy Local Search)



Simulated Annealing

Local Beam Search



Genetic Algorithms

Local Search

- Evaluates and modifies a small number of current states
- Does not record history of search

Good: Very little (constant) memory

Bad: May not explore all alternatives

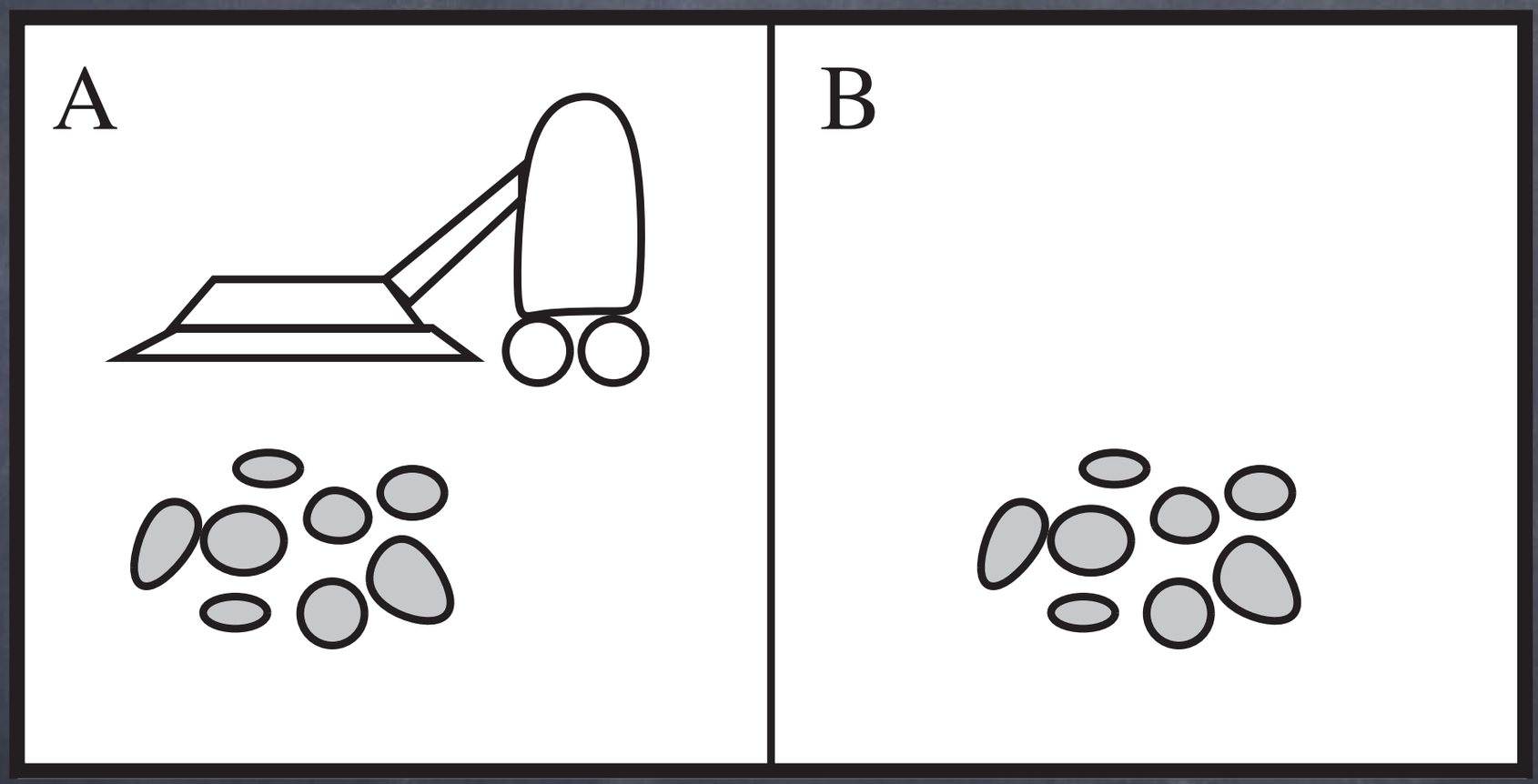
=> Incomplete

Local Search in Continuous Spaces

- FYI: Section 4.2

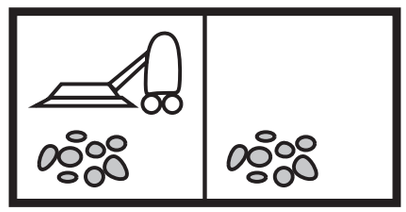
Searching with Nondeterministic Actions

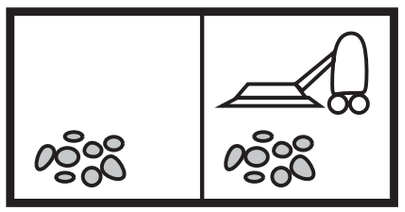
Vacuum World

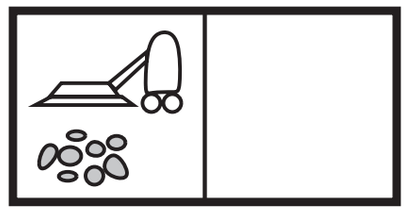


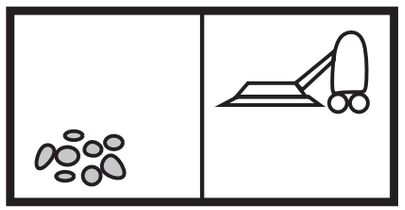
Actions: *Left*, *Right*, *Suck*

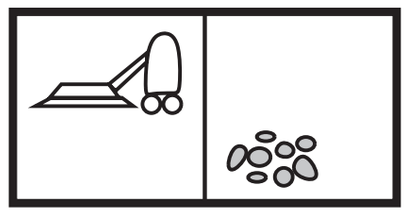
Vacuum World

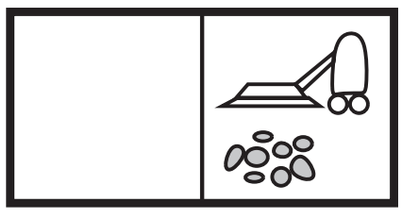
1 

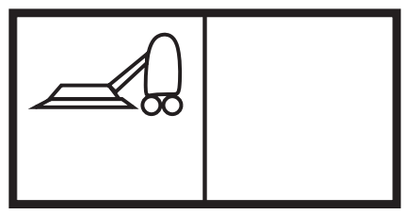
2 

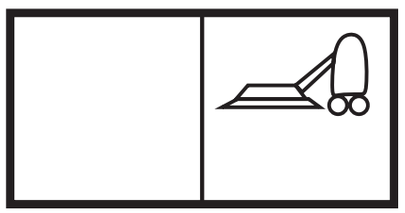
3 

4 

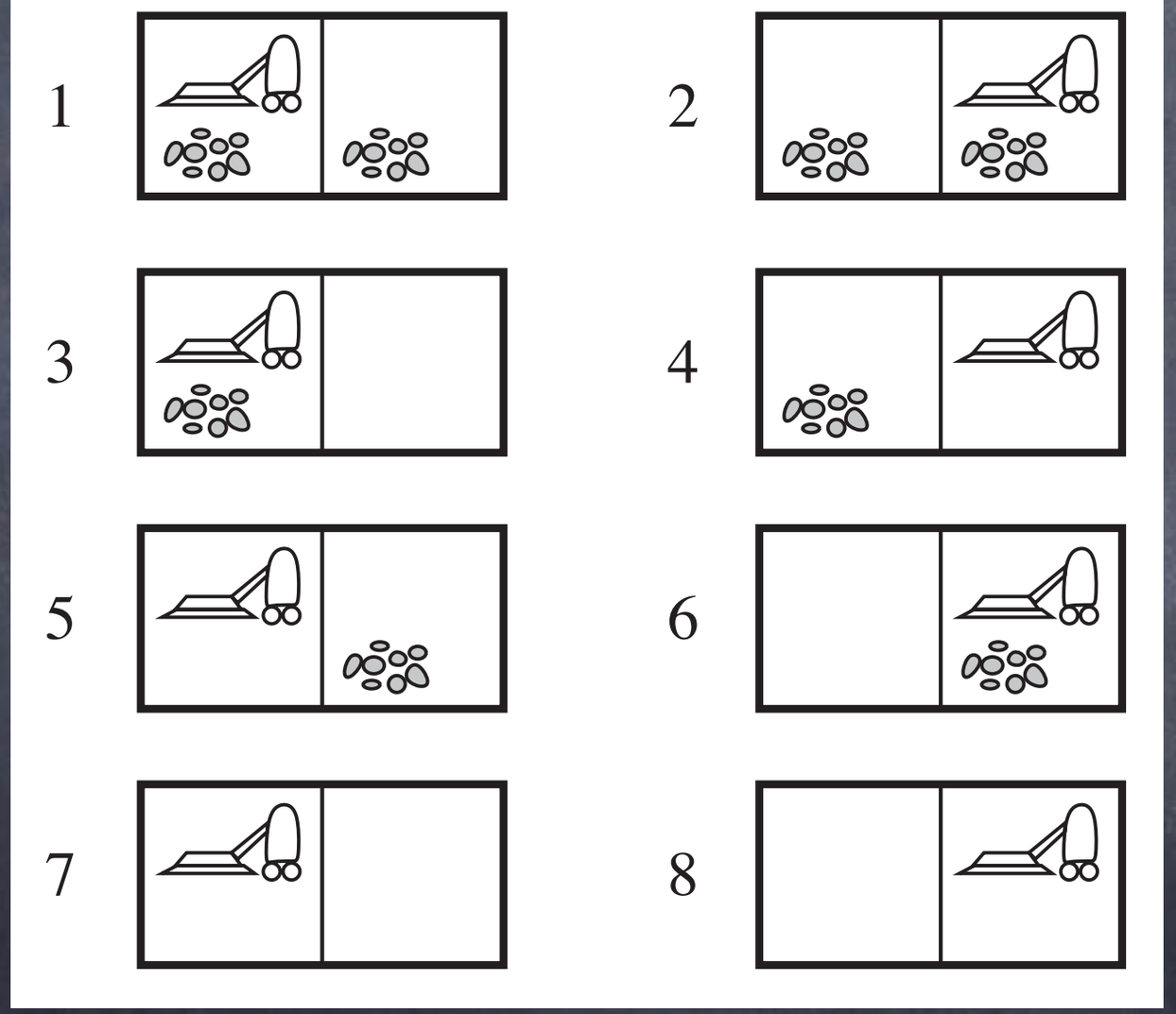
5 

6 

7 

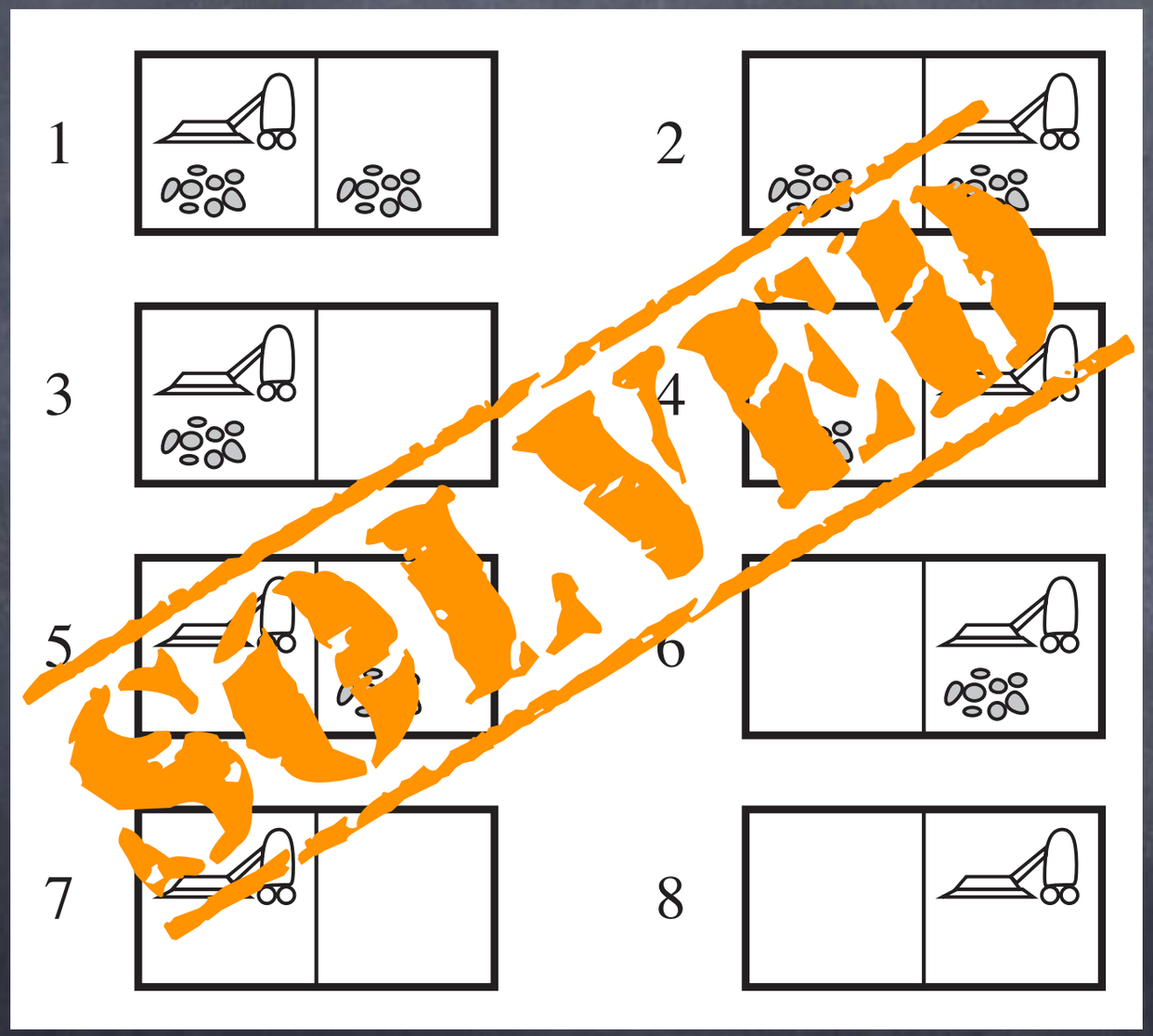
8 

Vacuum World



Fully Observable

Deterministic, Observable Vacuum World



Fully Observable

Nondeterministic Vacuum World

Nondeterministic Vacuum World

- "Erratic vacuum cleaner": bad $Suck$
  - $Left$ & $Right$ work as expected
  - $Suck$ on dirty square may clean other square also
  - $Suck$ on clean square may make it dirty

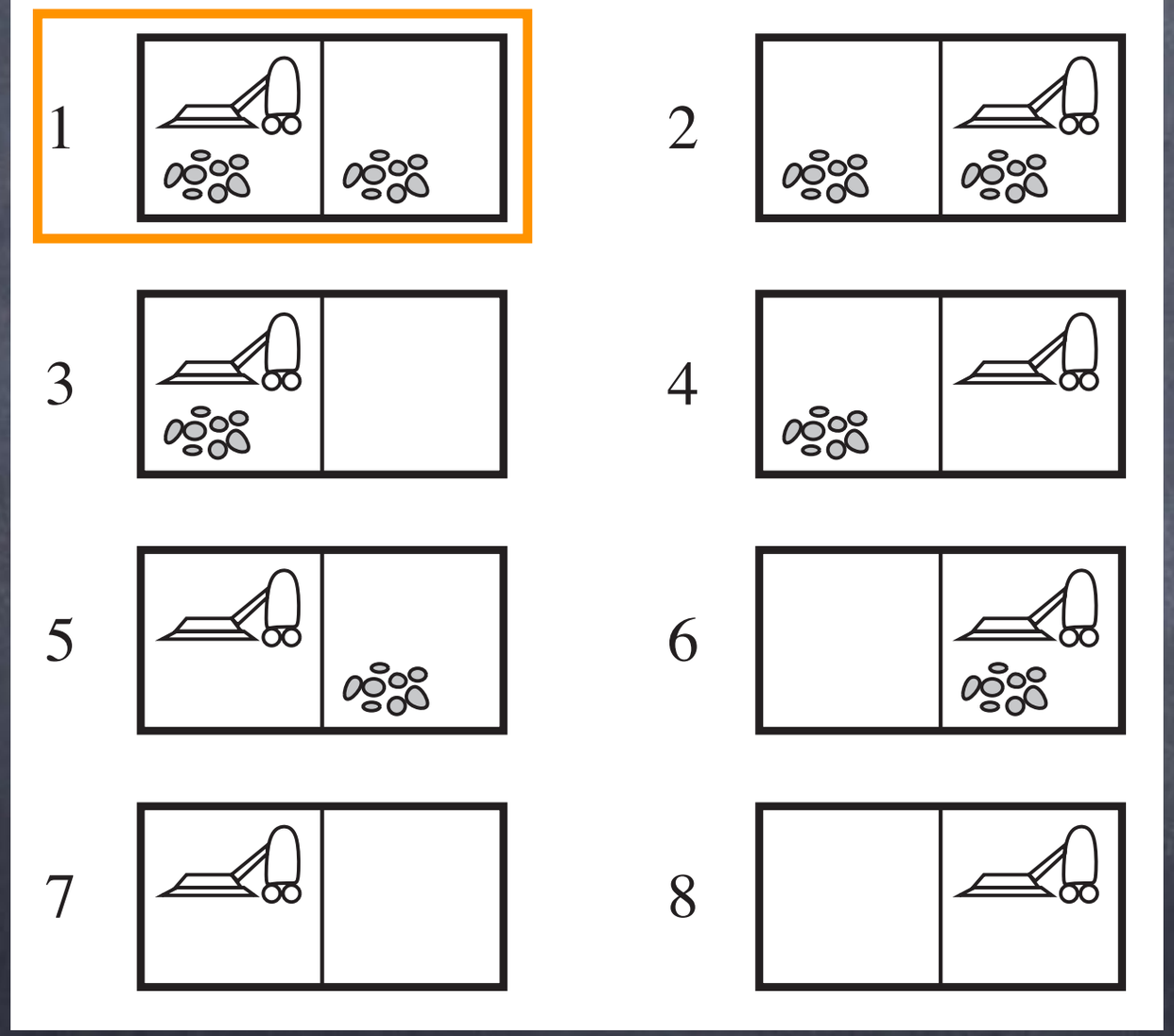
Nondeterministic Vacuum World

- "Erratic vacuum cleaner"
- Nondeterministic transition model: $\textsc{Result}(s,a)$ returns a set of possible outcome states

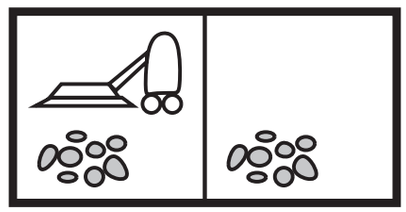
Nondeterministic Vacuum World

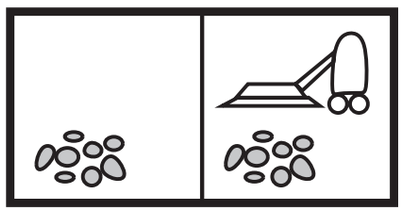
- "Erratic vacuum cleaner"
- Nondeterministic transition model: $\textsc{Result}(s,a)$ returns a set of possible outcome states
- Solution is a set of nested if-then statements (conditional plan)

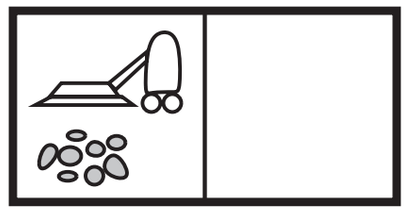
Vacuum World

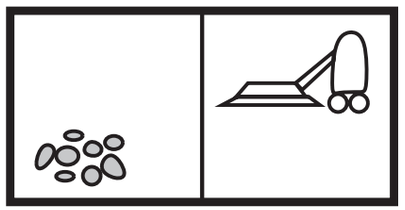


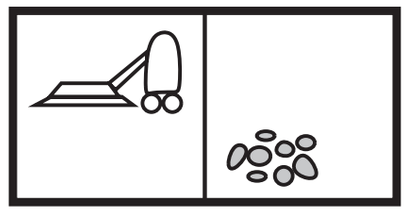
Vacuum World

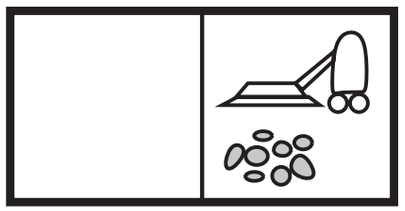
1 

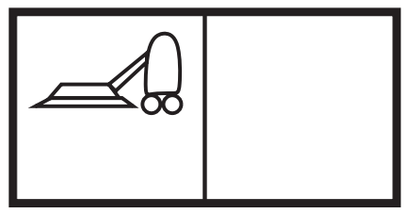
2 

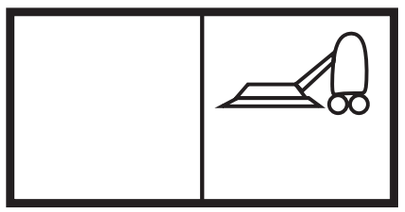
3 

4 

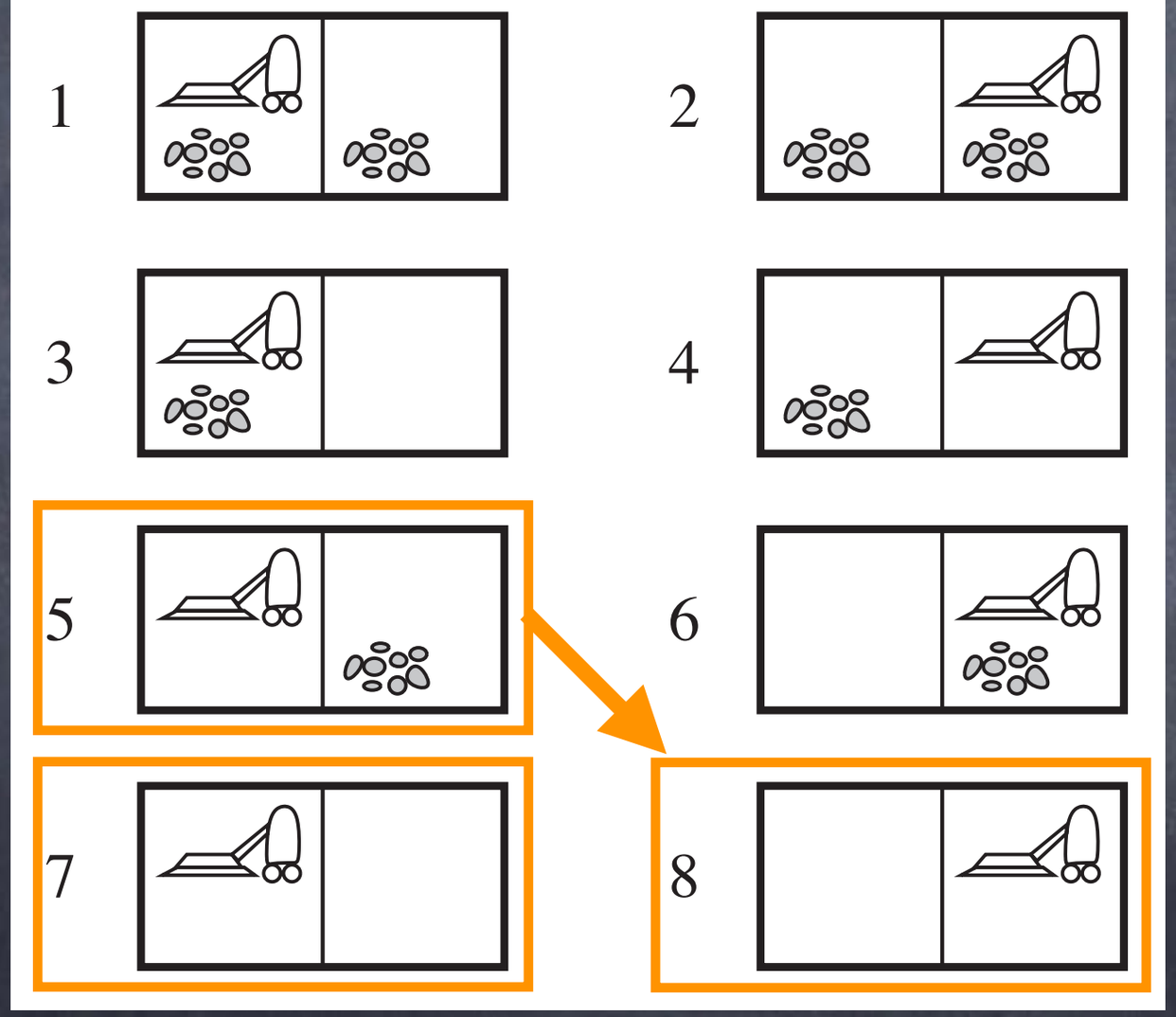
5 

6 

7 

8 

Vacuum World

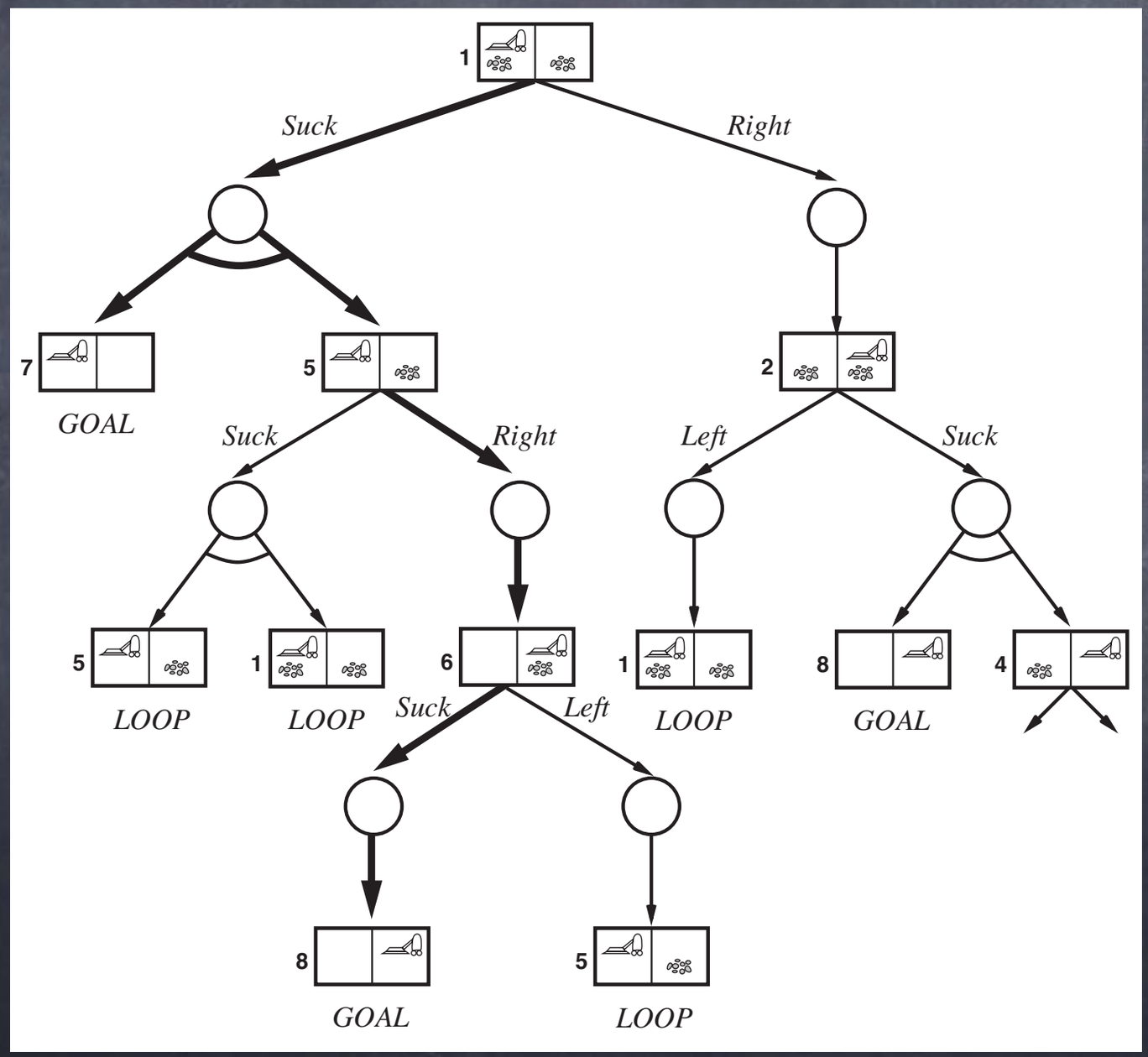


Nondeterministic Vacuum World

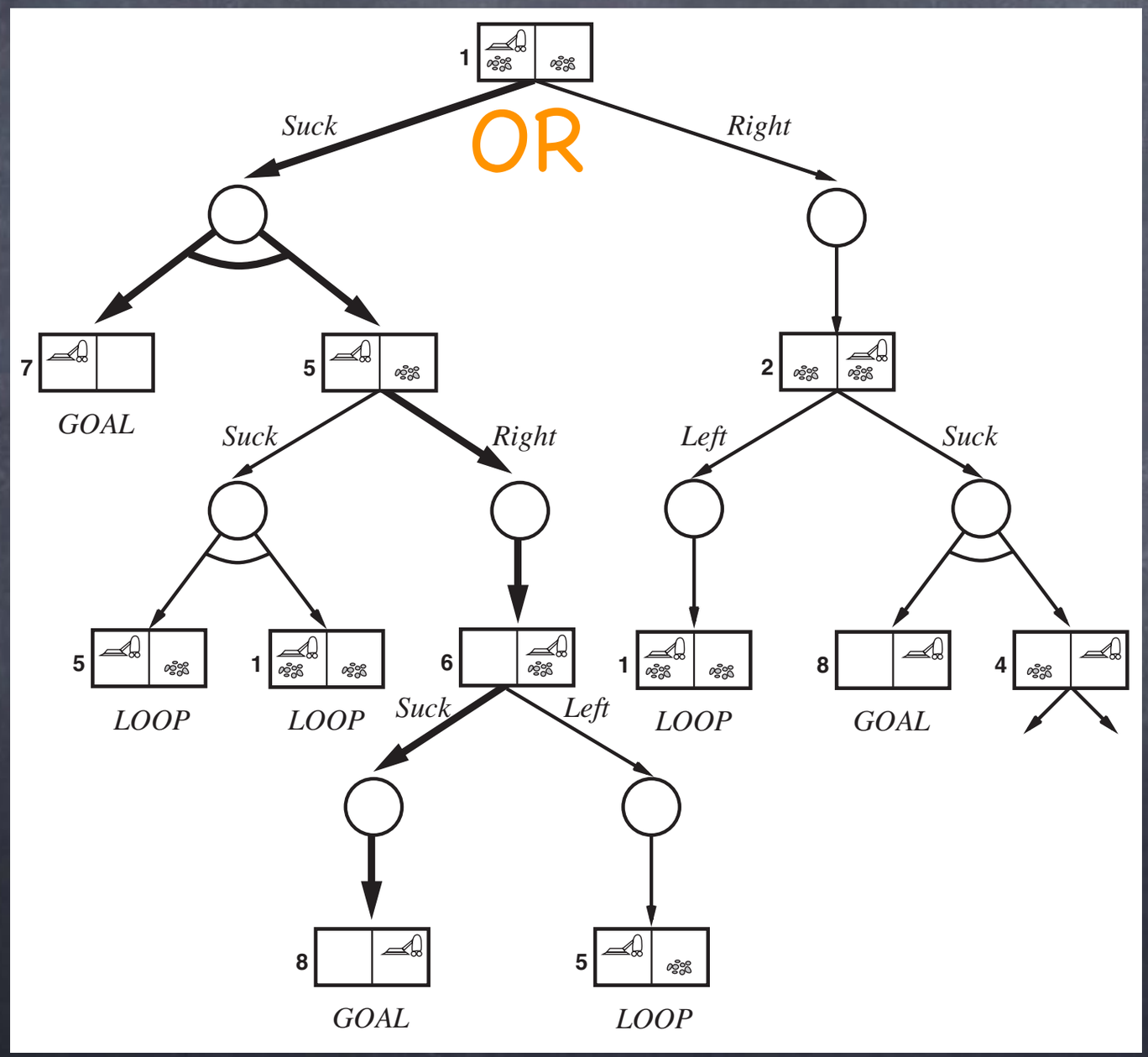
- "Erratic vacuum cleaner"
- Nondeterministic transition model: $\textsc{Result}(s, a)$ returns a set of possible outcome states
- Solution is a set of nested if-then statements (conditional plan):

$$[Suck, \textbf{if } State = 5 \textbf{ then } [Right, Suck] \textbf{ else } []]$$

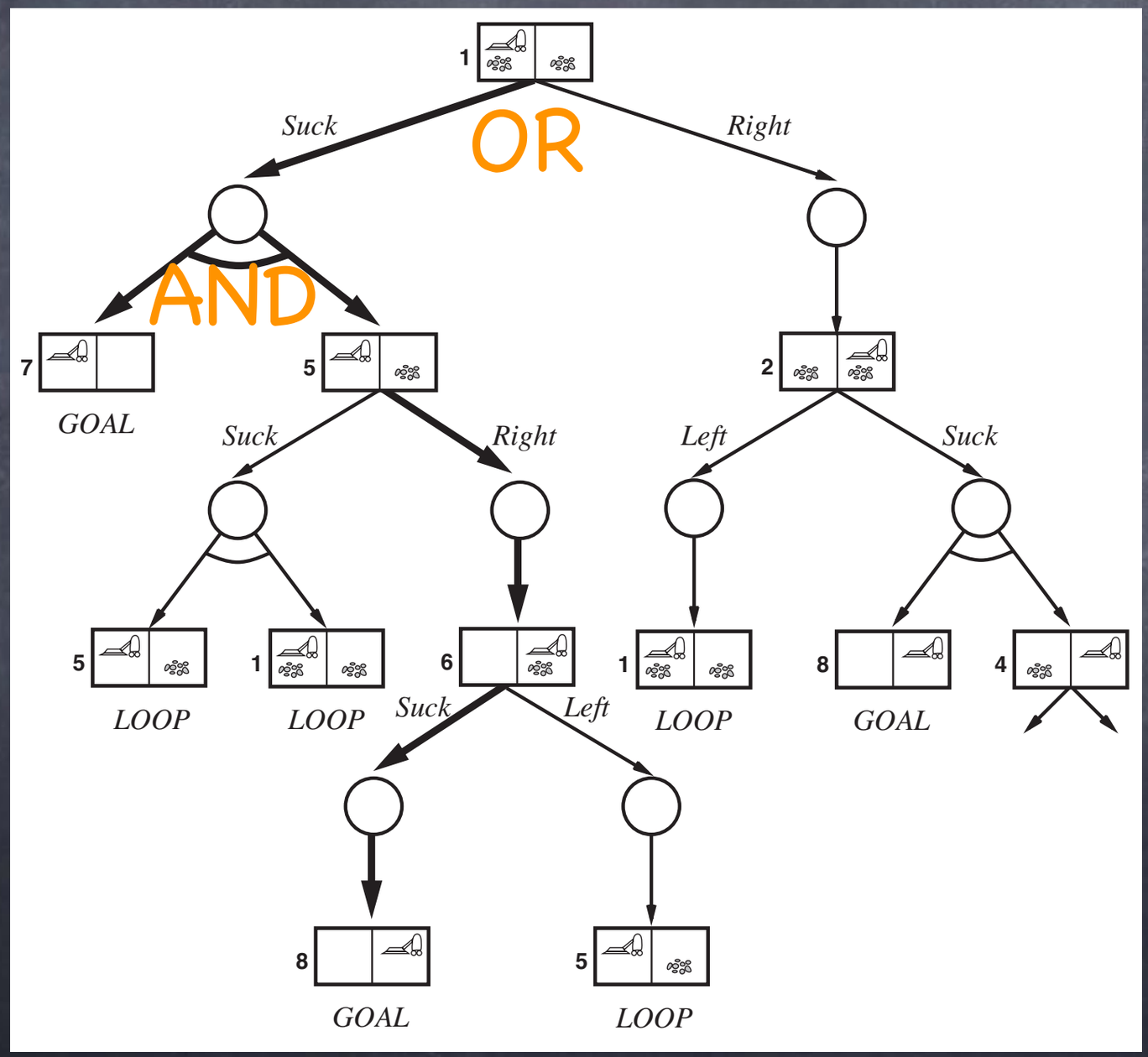
Nondeterministic Vacuum World



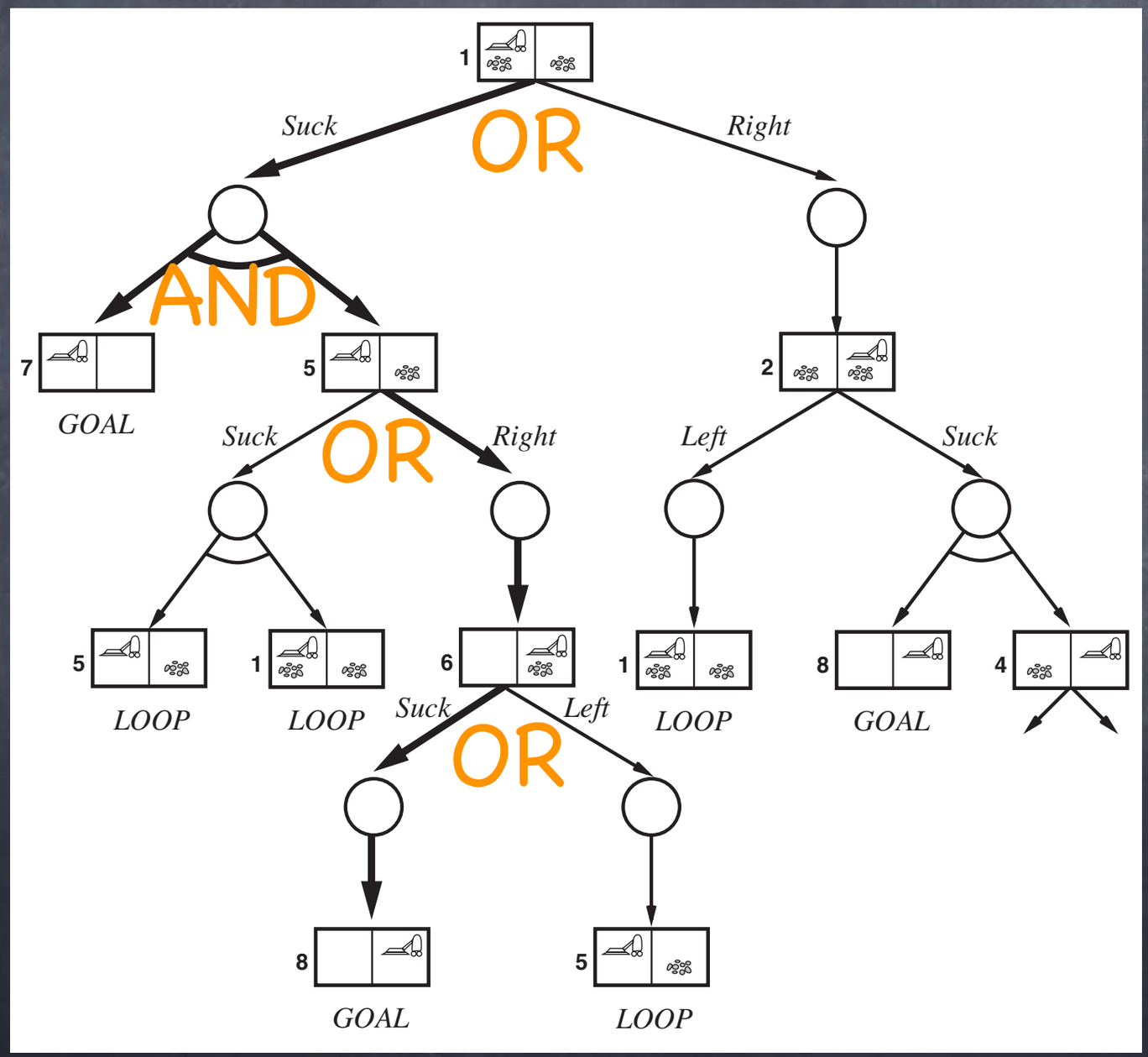
Nondeterministic Vacuum World



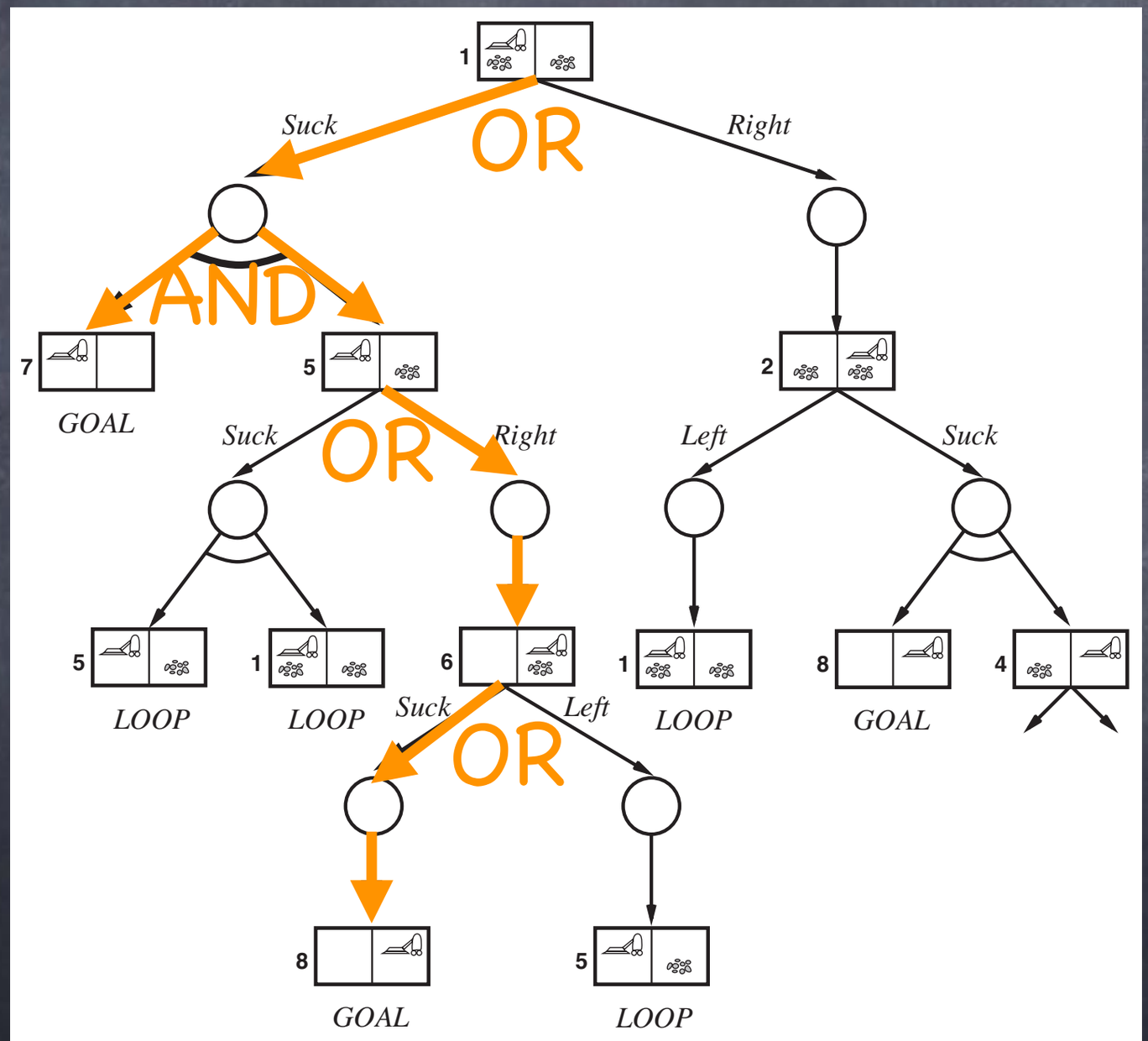
Nondeterministic Vacuum World



Nondeterministic Vacuum World



Nondeterministic Vacuum World



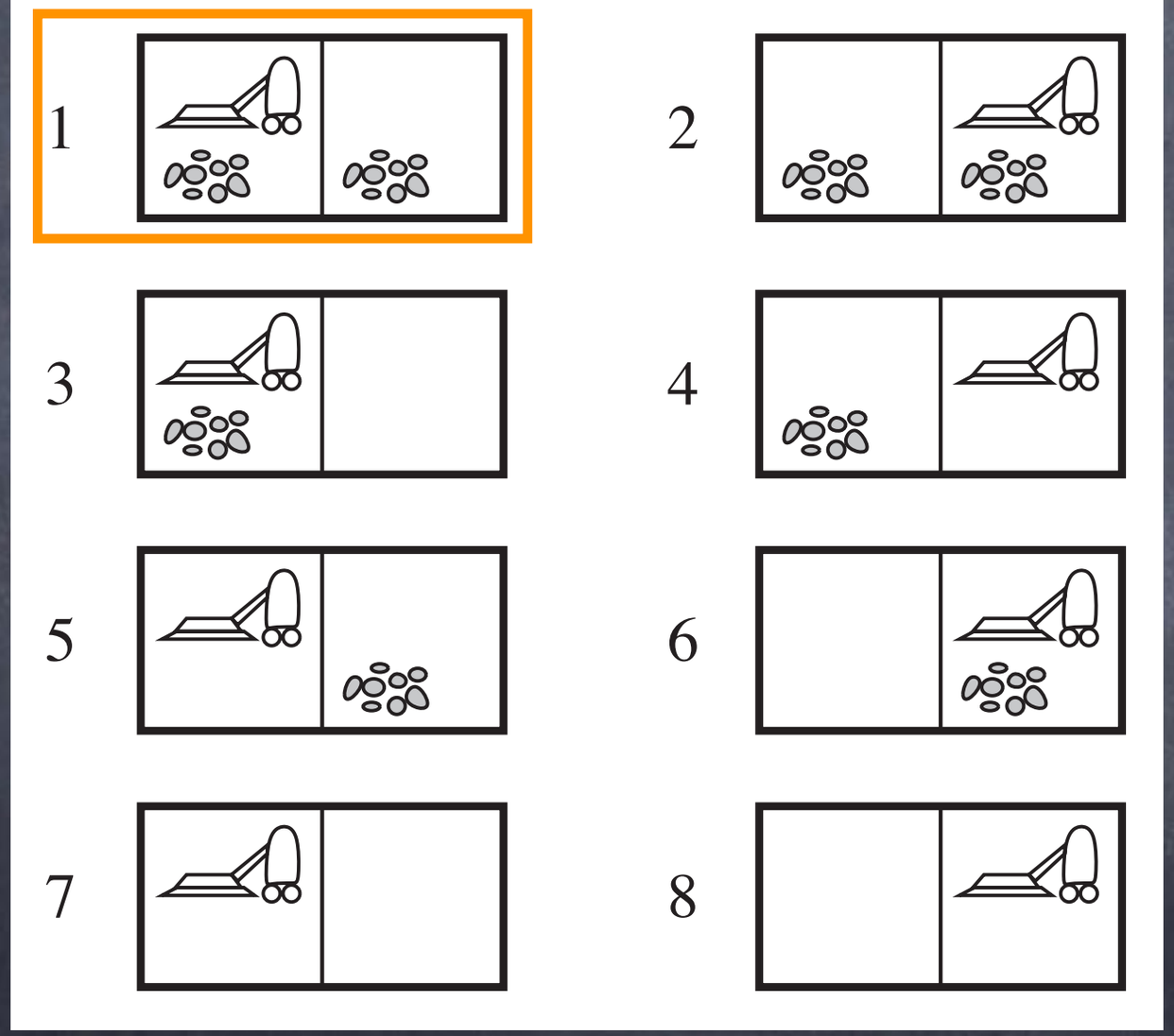
Searching with Nondeterministic Actions

- Depth-first algorithm: AIMA Fig. 4.11
- Also breadth-first, best-first, incl. A*

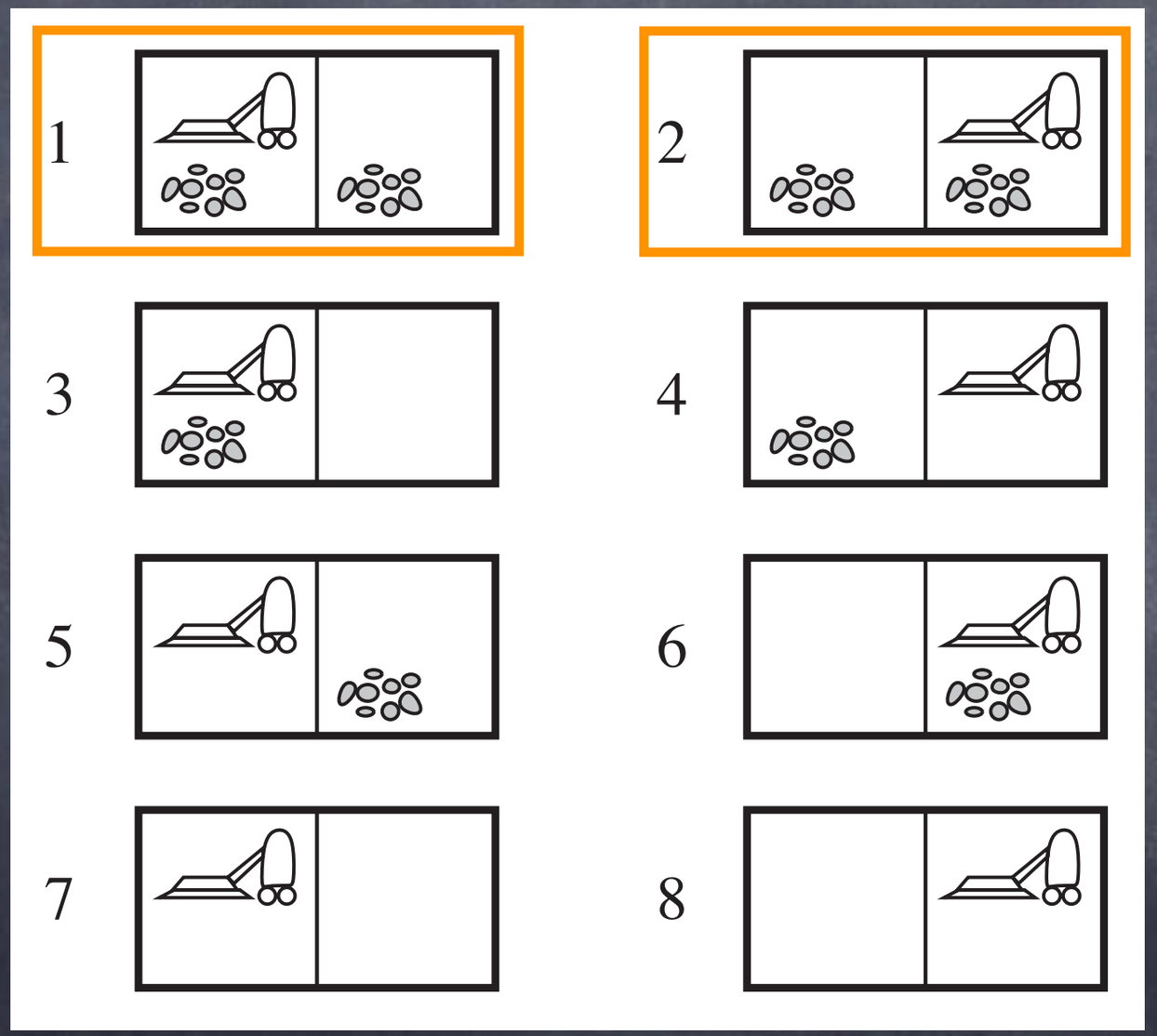
Nondeterministic Vacuum World

- "Slippery vacuum world":
  - bad $Left$/$Right$
- Transition model:
  - $\textsc{Result}(s,a)$ may include s

Vacuum World



Vacuum World



Nondeterministic Vacuum World

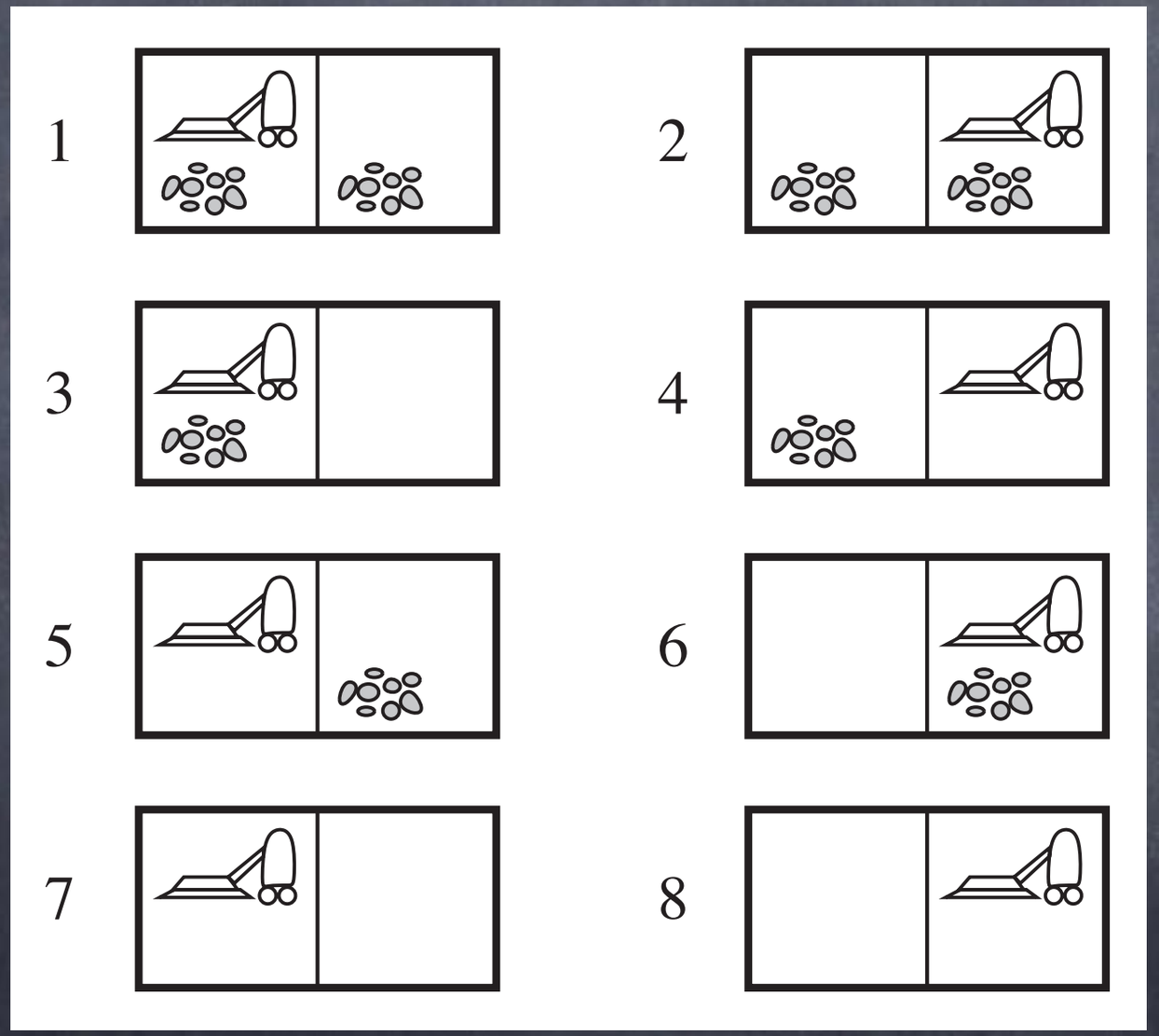
- "Erratic vacuum cleaner": bad $Left$ and/or $Right$
- Transition model: $\textsc{Result}(s, a)$ may include s
- Solution requires loops

Searching with Nondeterministic Actions

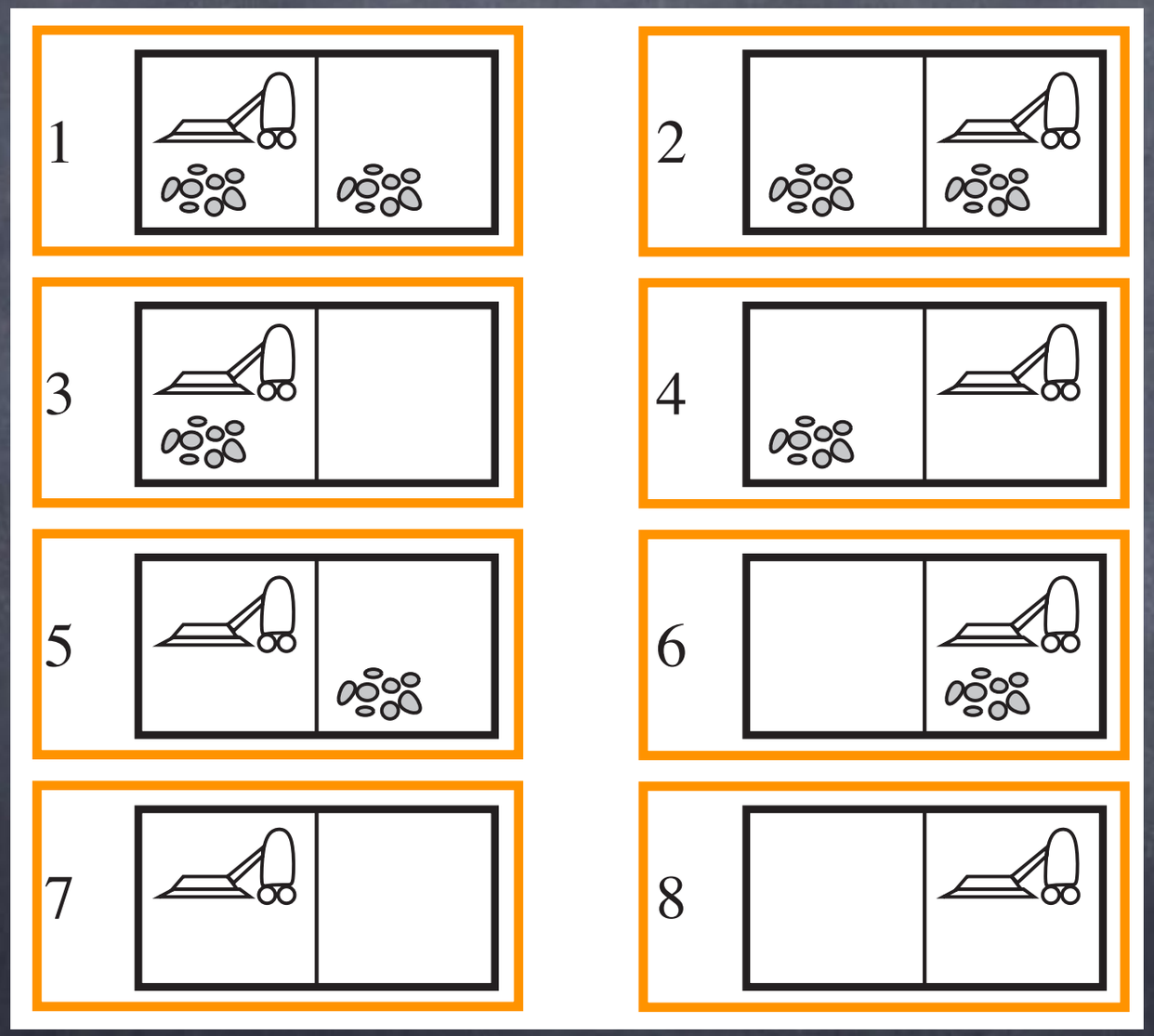
- Modified version of AND-OR graph search (AIMA Fig. 4.11)
- Will find cyclic solutions
  - "Provided that each outcome of a nondeterministic action eventually occurs."

Searching with Partial Observations

Sensorless Vacuum World

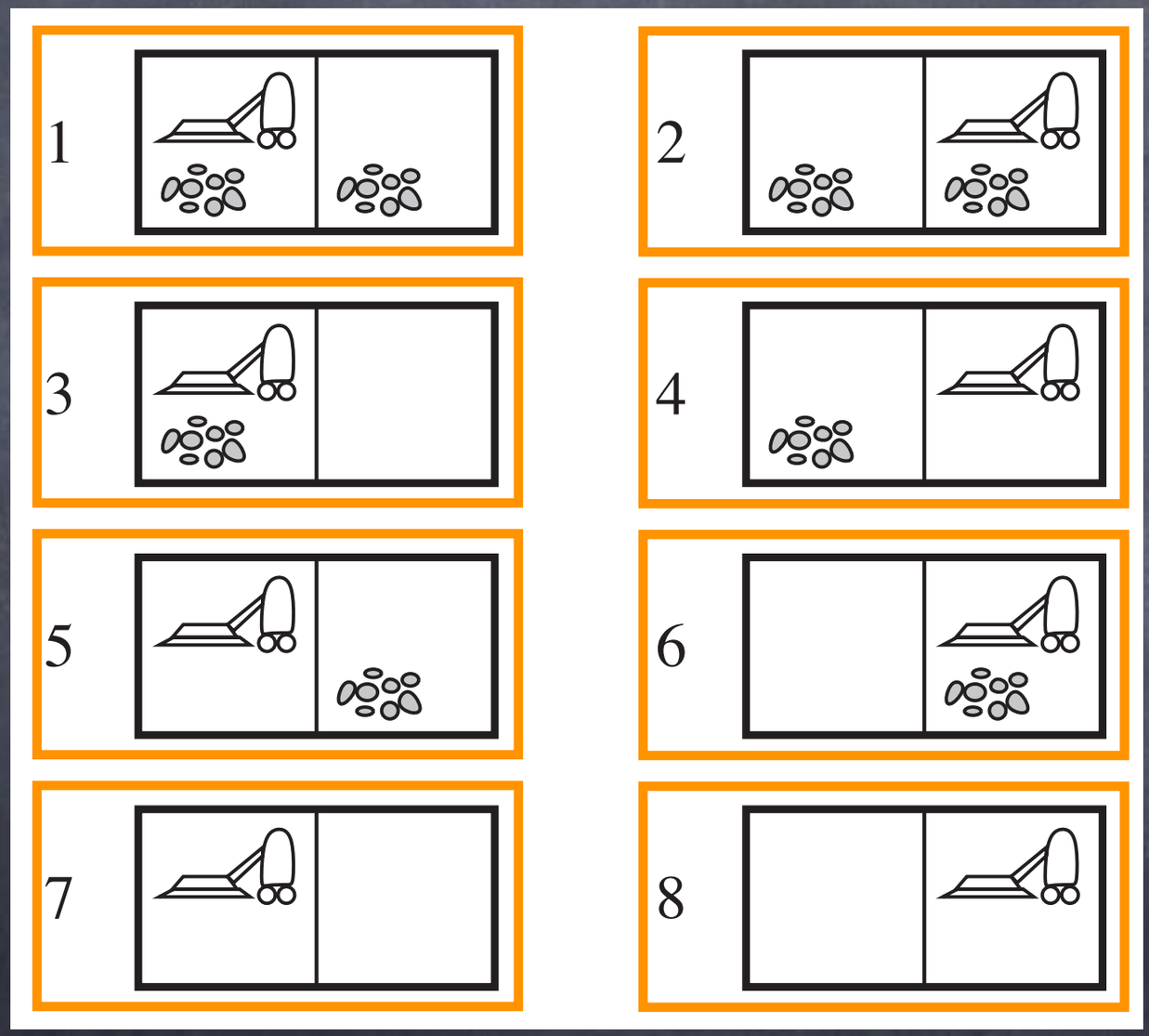


Sensorless Vacuum World



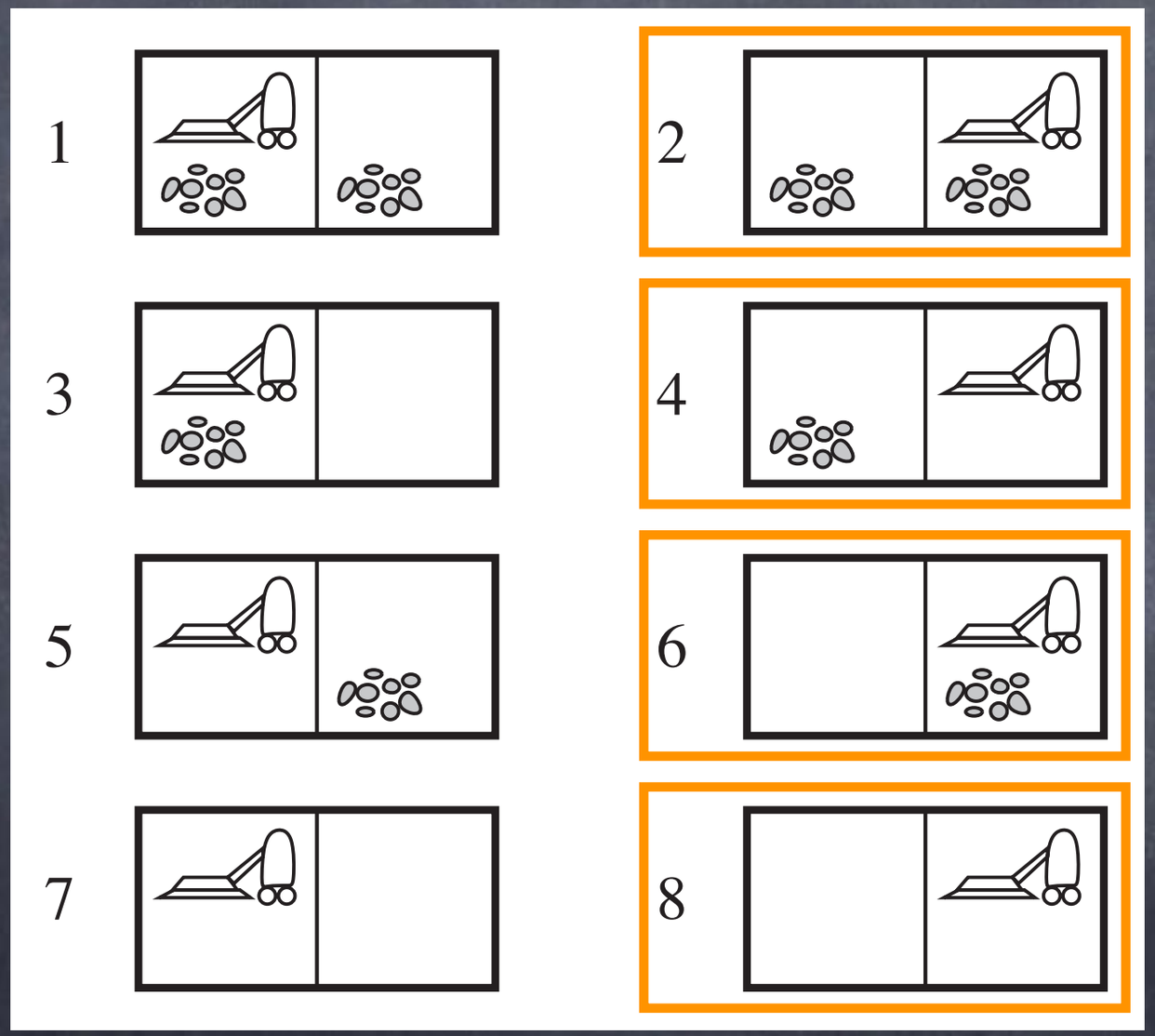
Sensorless Vacuum World

Right



Sensorless Vacuum World

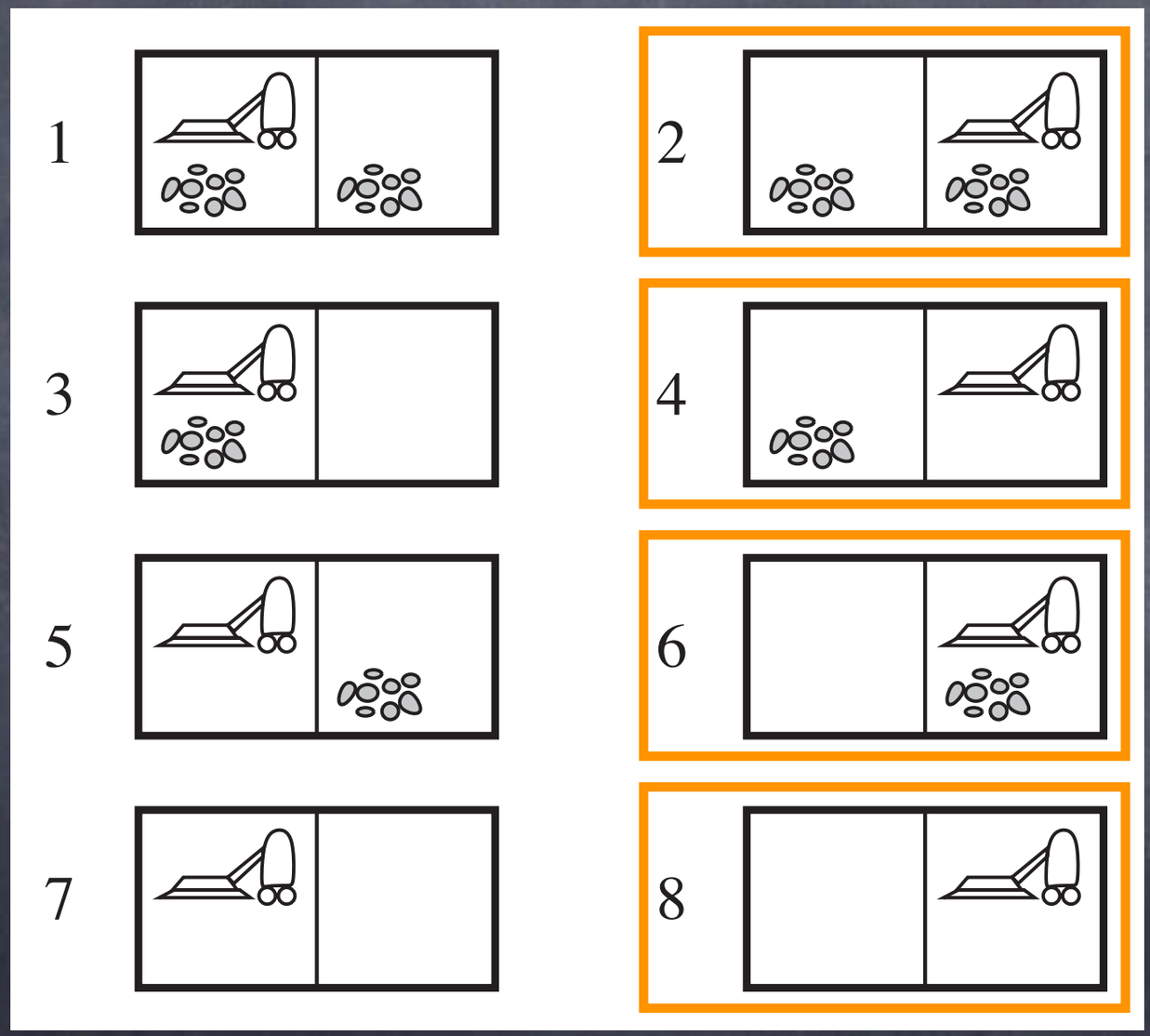
Right



Sensorless Vacuum World

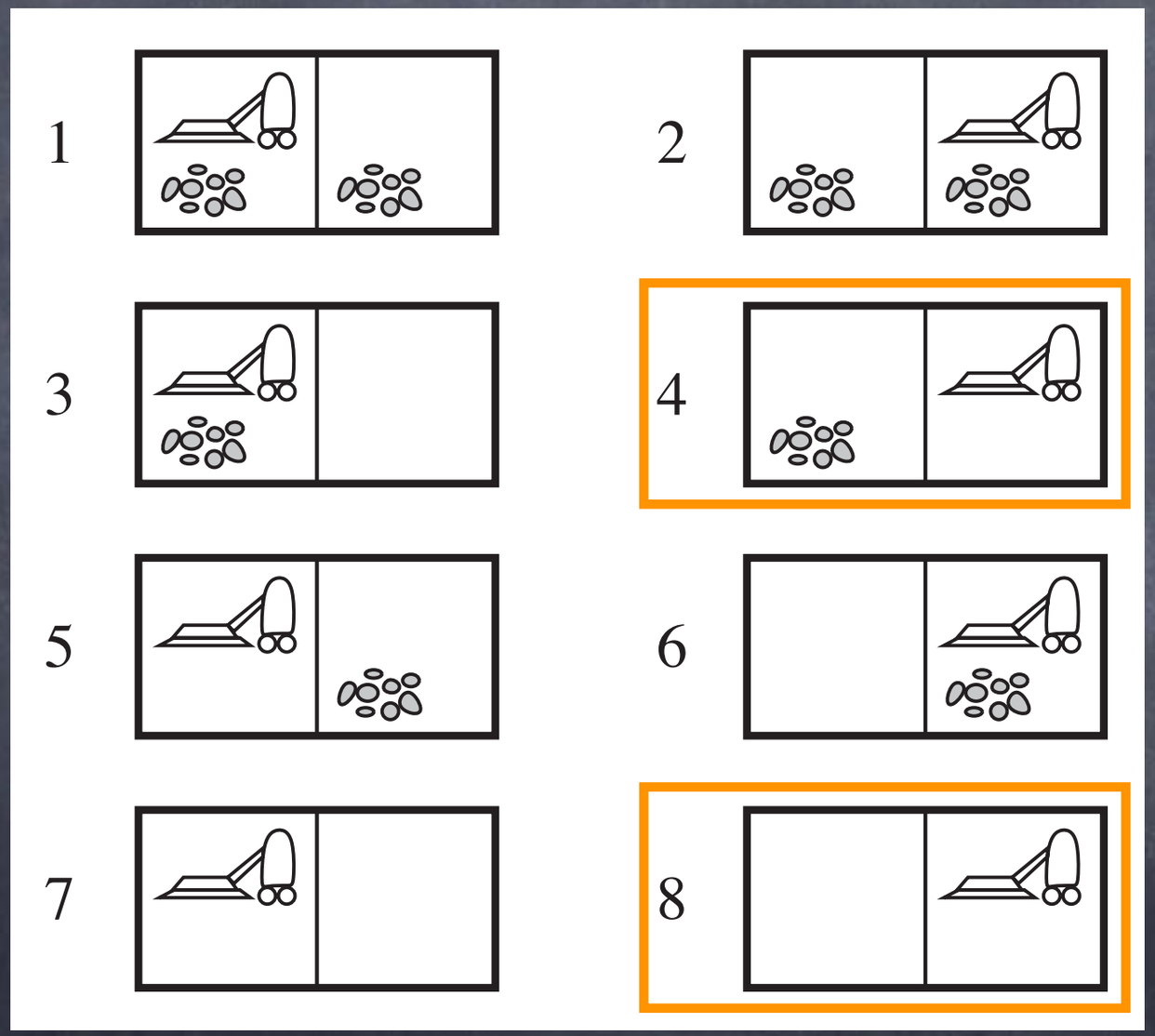
Right

Suck



Sensorless Vacuum World

Right
Suck

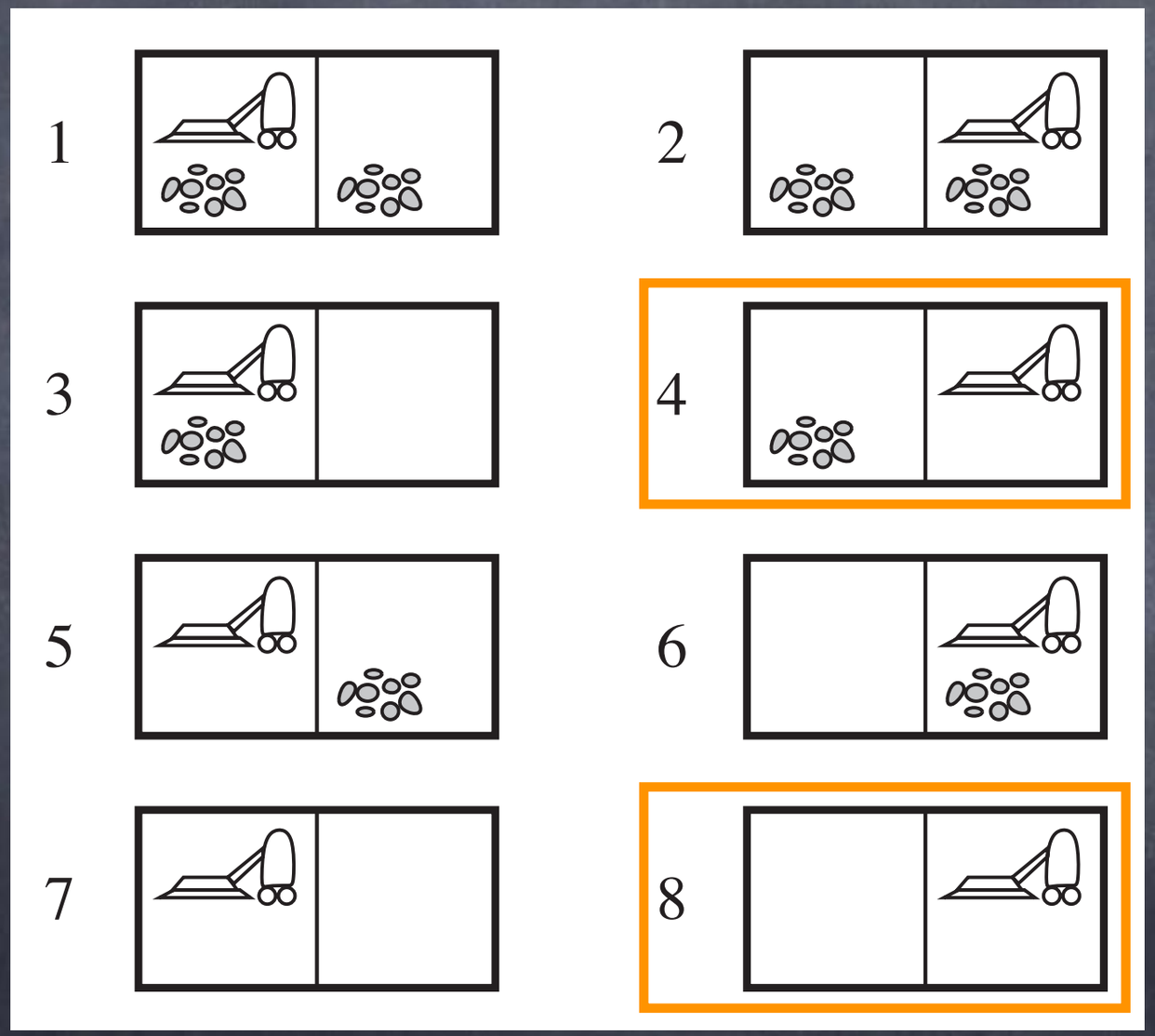


Sensorless Vacuum World

Right

Suck

Left

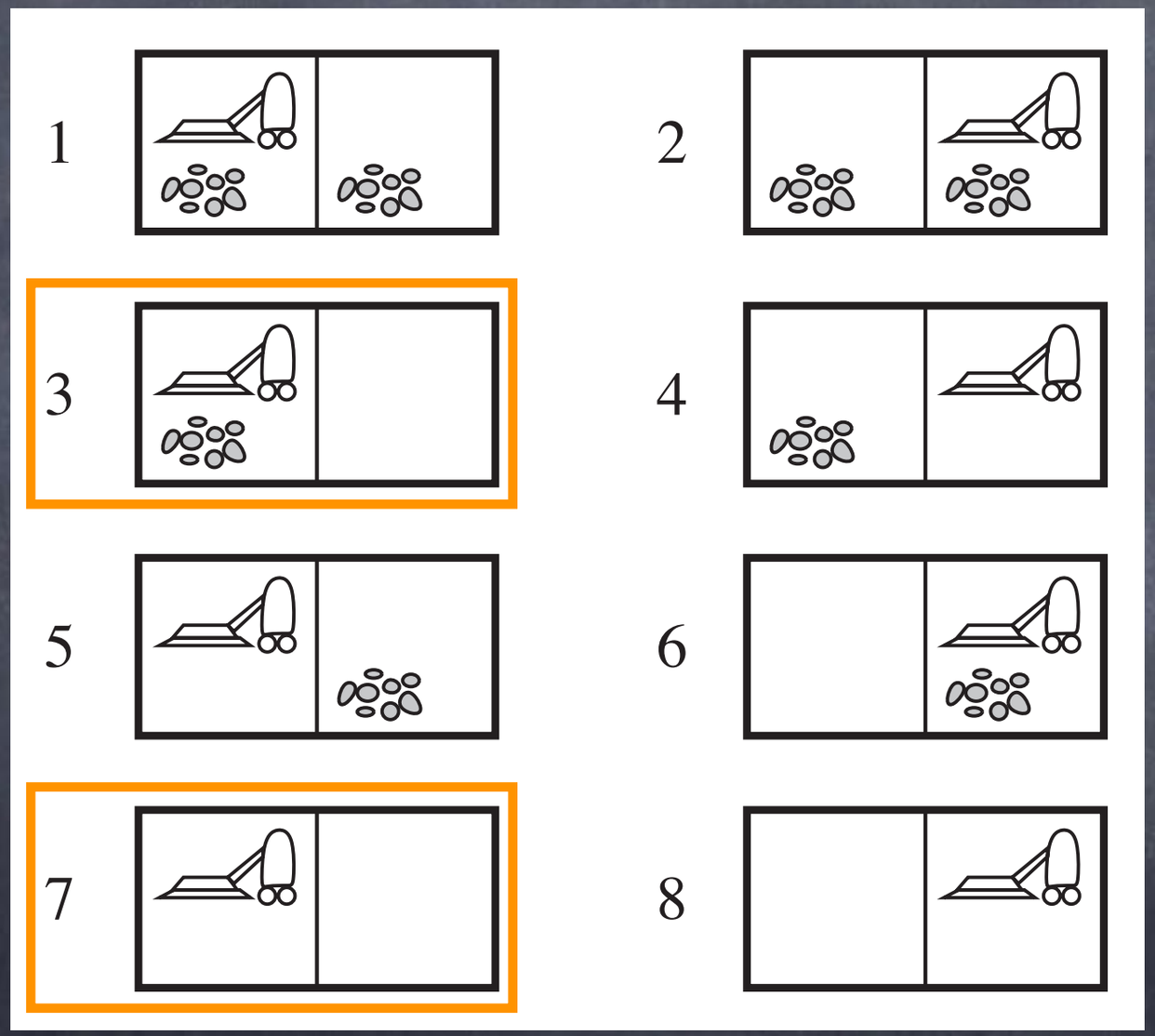


Sensorless Vacuum World

Right

Suck

Left



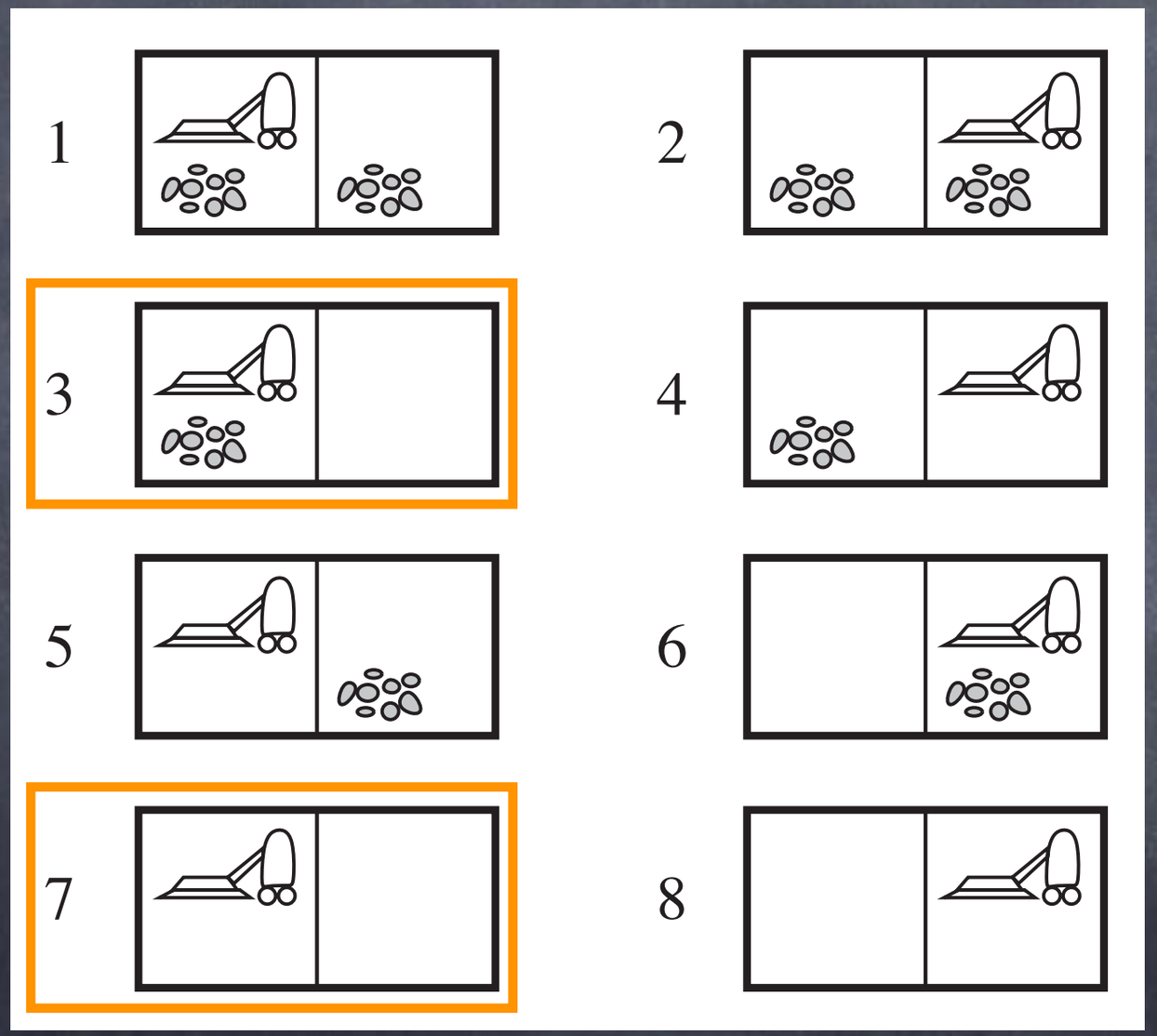
Sensorless Vacuum World

Right

Suck

Left

Suck



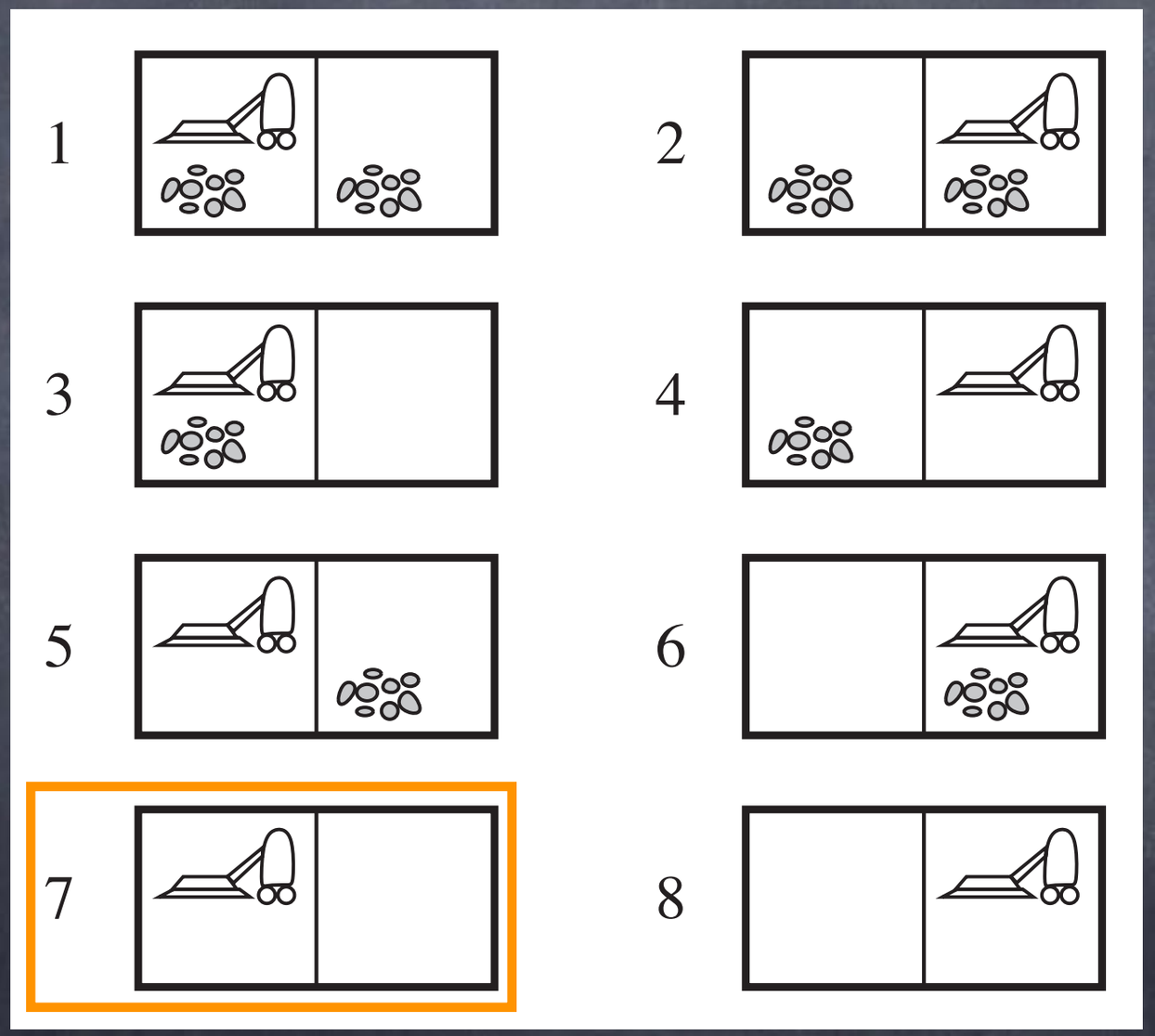
Sensorless Vacuum World

Right

Suck

Left

Suck



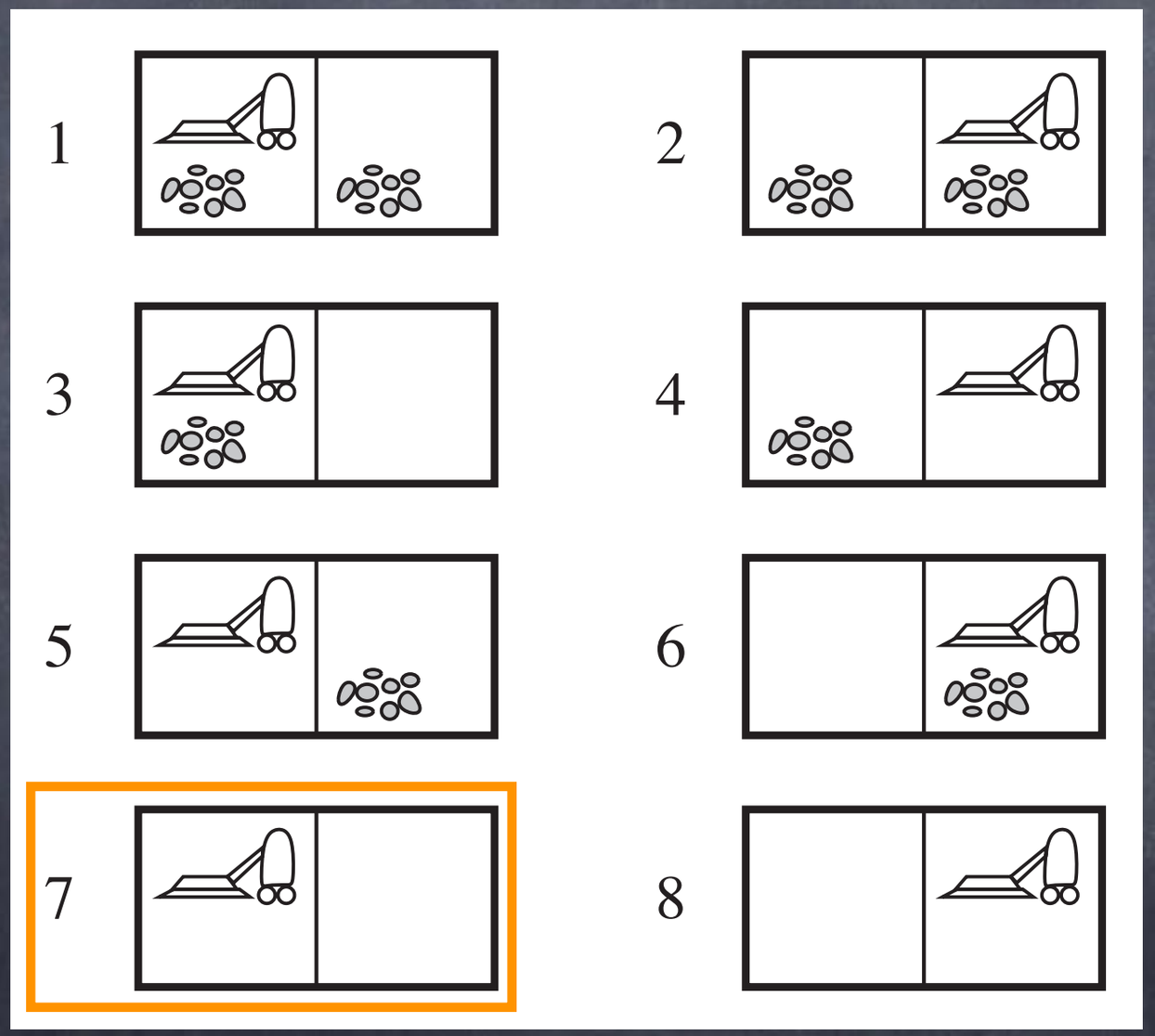
Sensorless Vacuum World

Right

Suck

Left

Suck



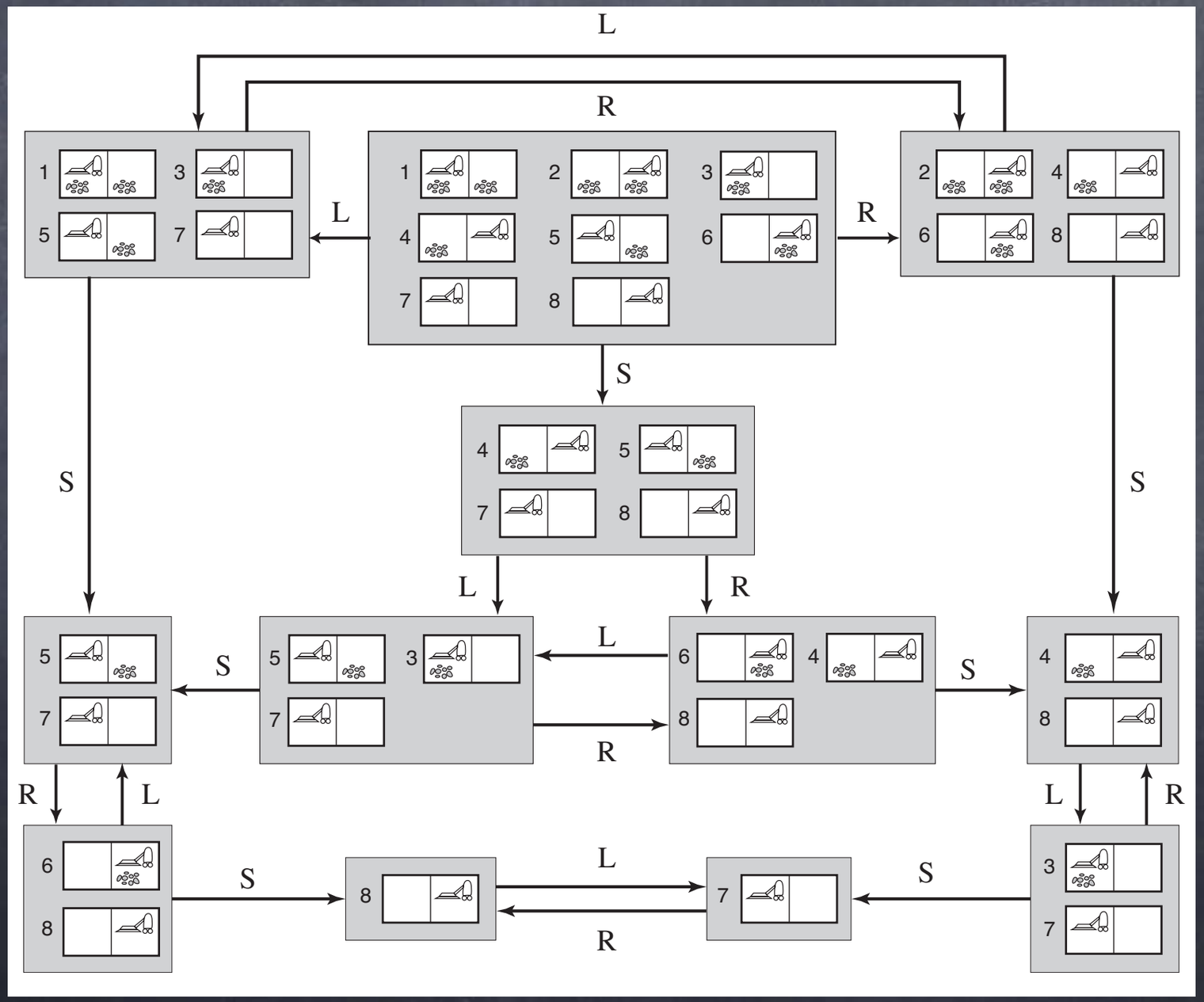
Belief State

- An agent's current beliefs about what physical (real-world) state it might be in
- Belief state is fully observable
  - The agent knows its own beliefs (for now)

Belief State-Space Search

- Belief states: Every possible set of physical states (N physical, 2^N belief)
- Initial state: Set of all states in problem
- Actions: Union or intersection of applicable actions in each state in the belief state
- Transition model: Union (liberal) or intersection (conservative) of result states (prediction)
- Goal test: Every state in belief state is a goal state
- Path cost: "Tricky"

Vacuum World Belief State Space



Searching with Partial Observations

- Search through space of belief states
- Sensorless problem-solving "completely impractical"
- Solutions:
  - Use a better representation of state
  - Incorporate observations that update the current belief state

Summary

- Local Search
  - Hill-climbing, Random Restarts, Simulated Annealing, Local Beam Search, Genetic Algorithms
- Search in nondeterministic domains
  - AND-OR trees, contingency plans
- Search in partially-observable domains
  - Belief states, belief-state-space search

Unit 1: Search

- Problem solving and state-space search
- Search strategies
- Adversarial search
- Local search
- Search in nondeterministic and partially observable domains

For next time:

Unit 1 Exam