CSCI 3202: Intro to Artificial Intelligence

Lecture 3: Agents & States

Rachel Cox

Department of Computer Science

Welcome!

Happy Friday!

We start at 10:10 am.

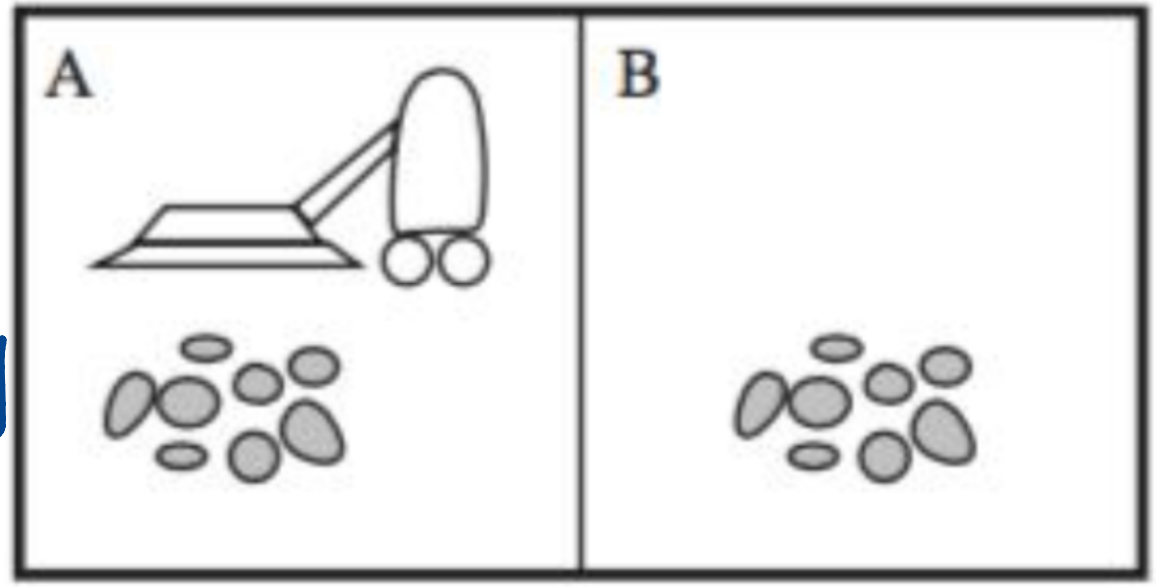
States

Sample/state space: the set of all possible world configurations, or states.

- aka states-of-the-world (SOW)

Example: Robot vacuum. How many possible (world) states are there?

- 2 locations for the vacuum
- combinations of clean or dirty
  $C(2,2) + C(2,1) + C(2,0)$
  $1 + 2 + 1$
  $= 4$



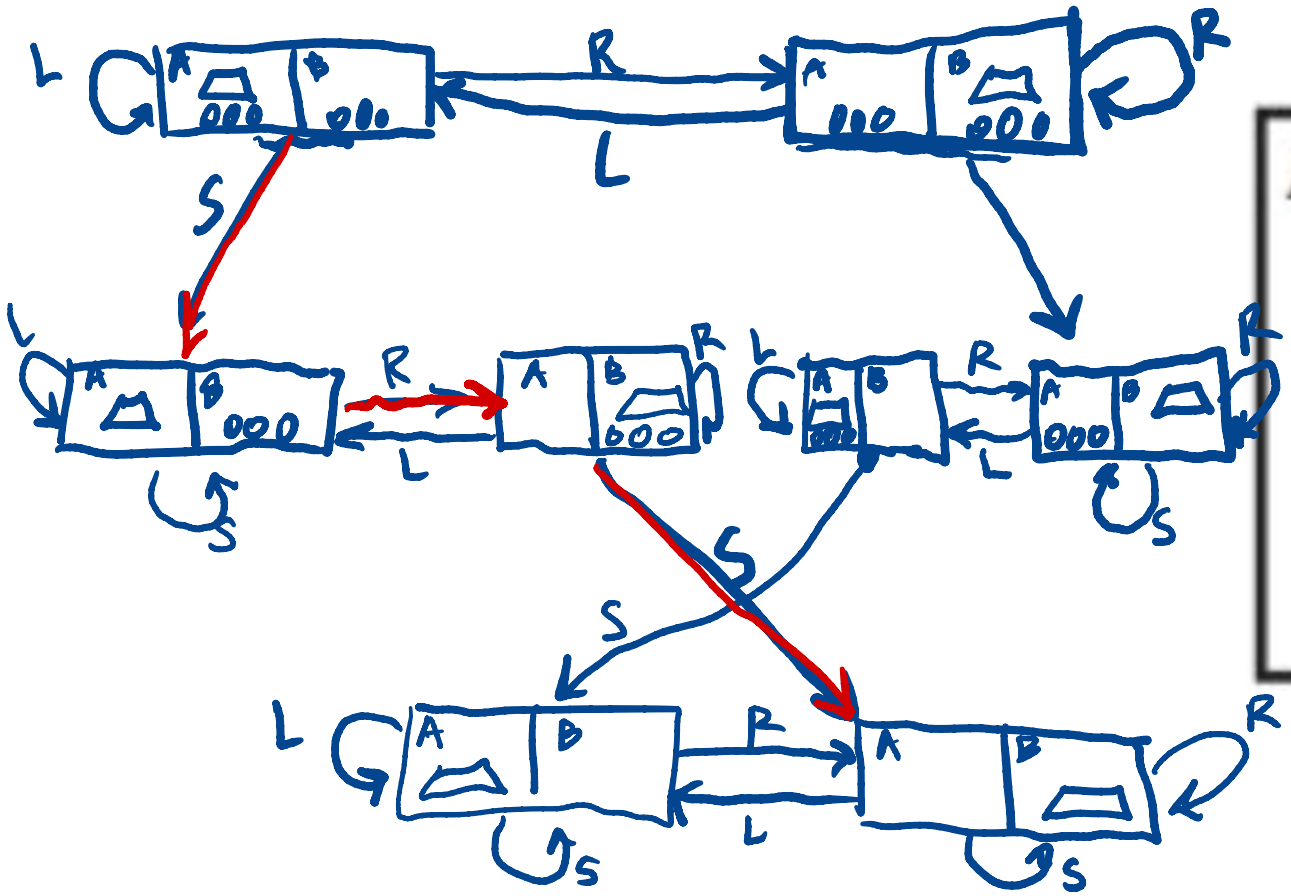
total world states $= 2 \times 4 = 8$

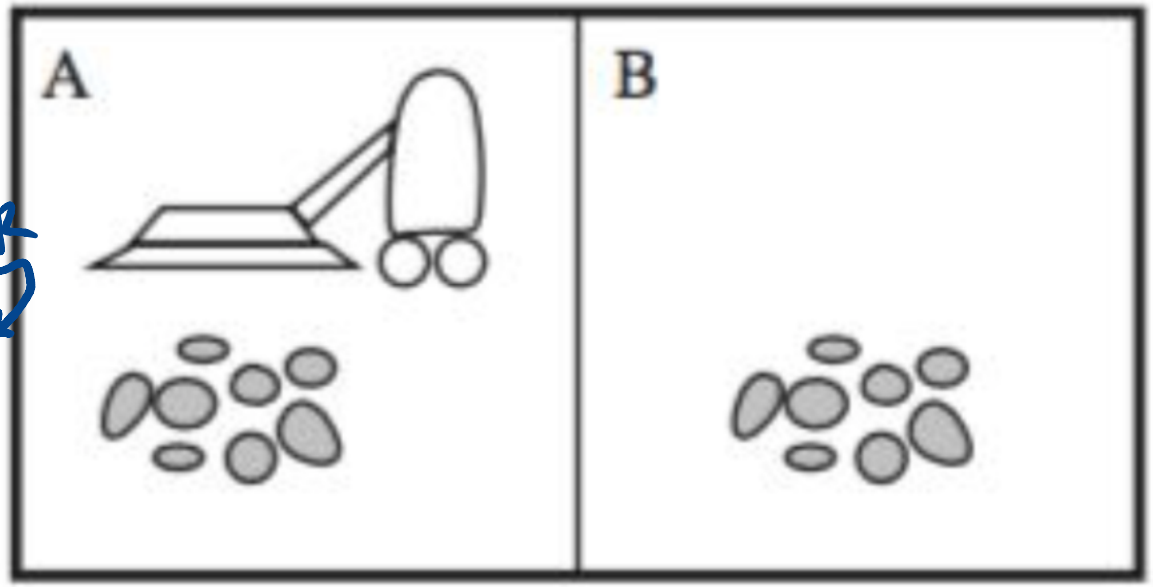
States

State space graph: a mathematical representation of the problem

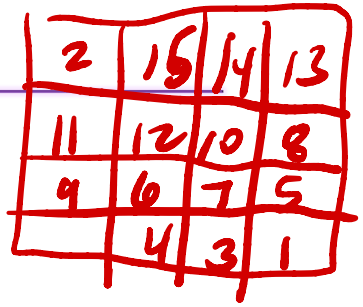
- Each state is a vertex on the graph
- Directed edges connect states by corresponding agent actions.

Roomba actions
left
Right
Suction





States



Example: Sliding blocks (Discrete counting problems refresher!)

What is the size of the state space?

16!

permutations!

one state of the world

States

Example: What is the size of the state space for this Pac-Man agent? White dots are consumable food, grid is 10x12.

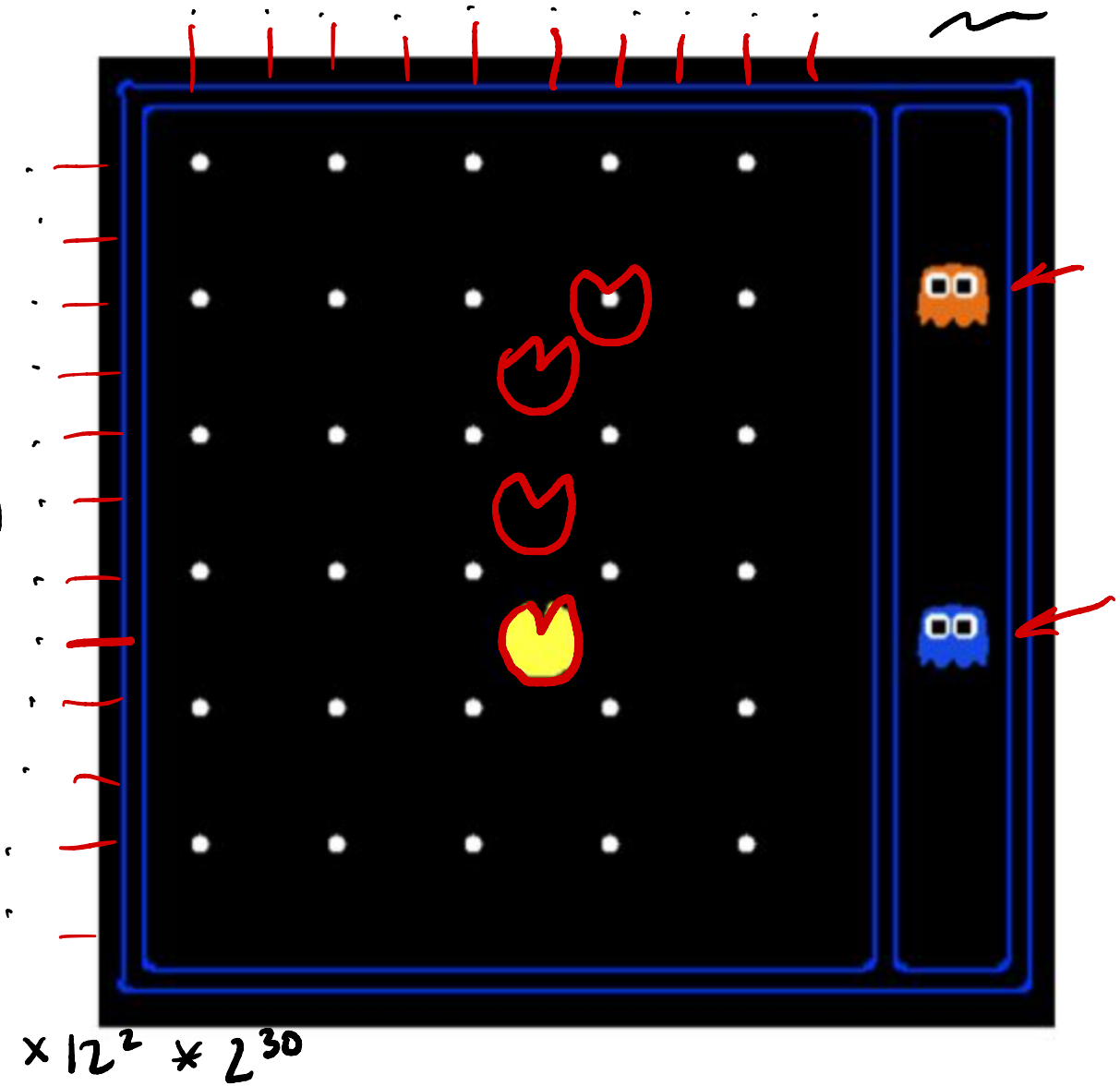
Number of spaces that Pacman can occupy = $12 * 10$

Number of ways that Pacman can face = 4 (N, S, E, W)

Food pellets (there or not) = 2^{30}

Ghosts = 12^2

total number of world states = $120 \times 4 \times 12^2 * 2^{30}$

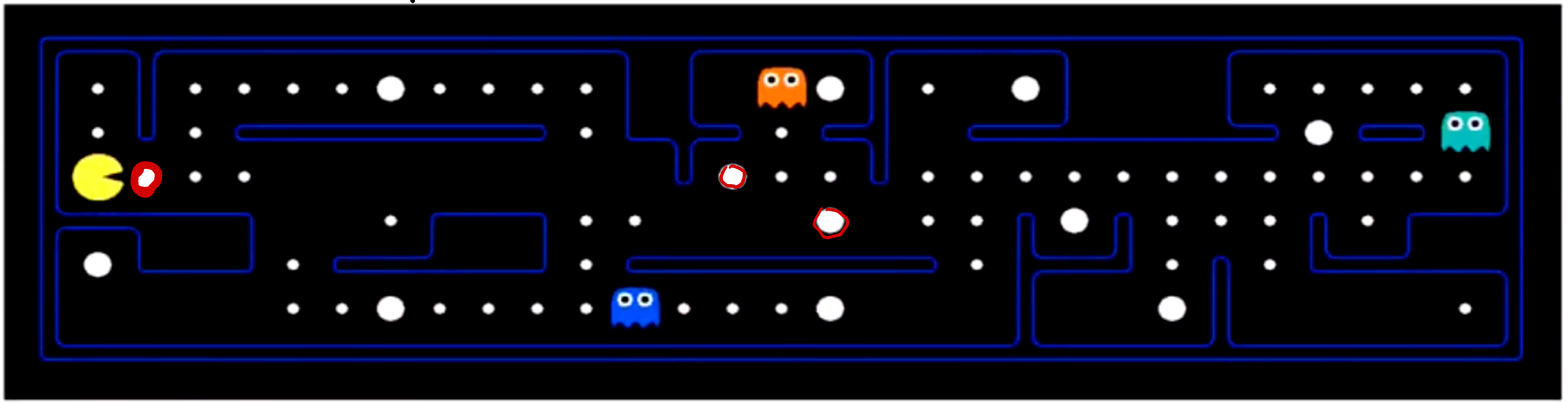


States

Example: Suppose your goal is to eat all of the food while keeping the ghosts "scared" constantly. What information would your state space need to include?

time remaining for scared ghosts
distance to big pellet

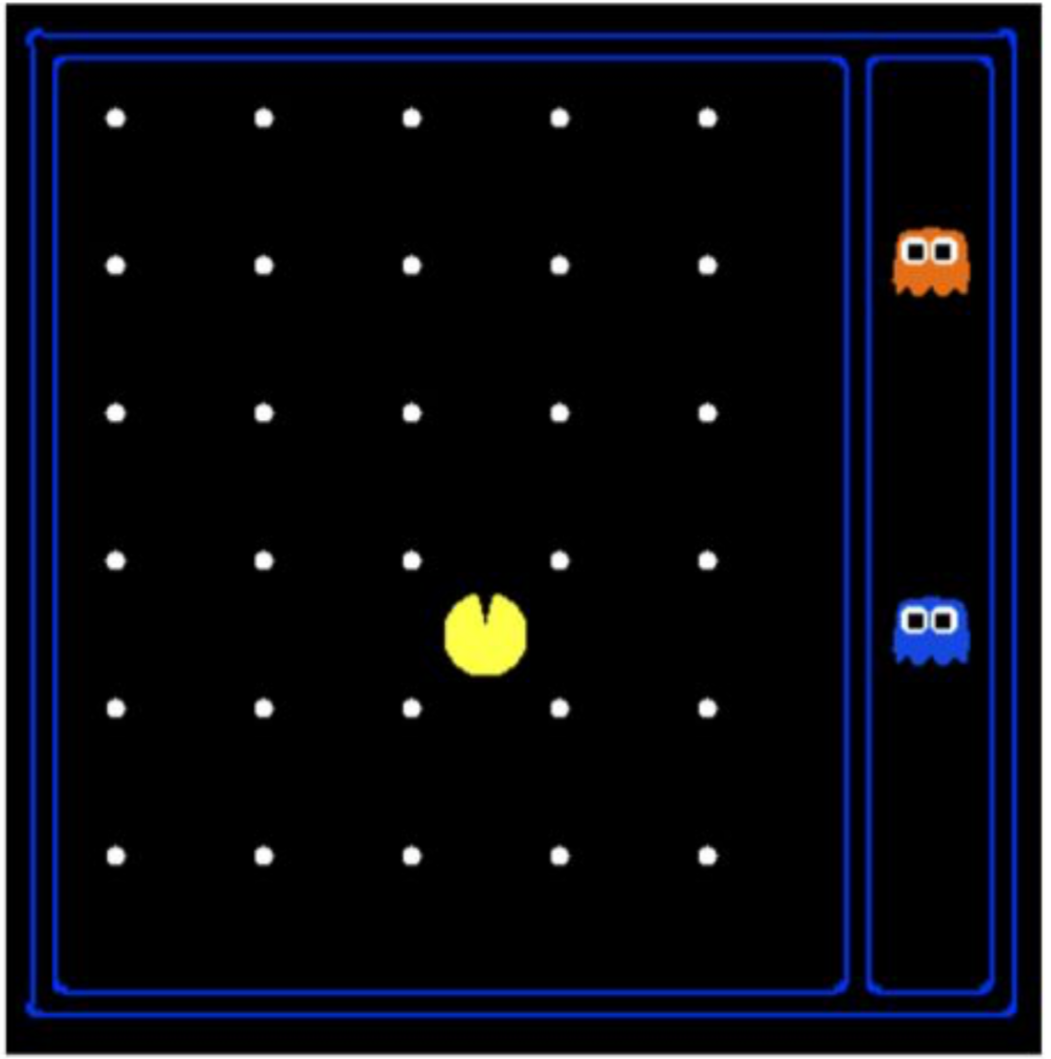
dot boolean
location of pacman



Search

A **search problem** consists of:

1. State space
2. Transition model
3. Actions
4. Initial state
5. Goal test



Search

1. **State space**
   a. What are all the possible ways the world could look?
   b. Forms a directed graph.
   c. A path in the state space is a sequence of states, connected by actions.
   d. A path cost function assigns a numeric cost (might be defined as in utility) to each path.
   e. Sum of the step costs (typically)

2. **Transition model**

   a. function that returns state_new that results from doing an action to state_old
   b. “successor”: any state reachable from a given state by a single action.

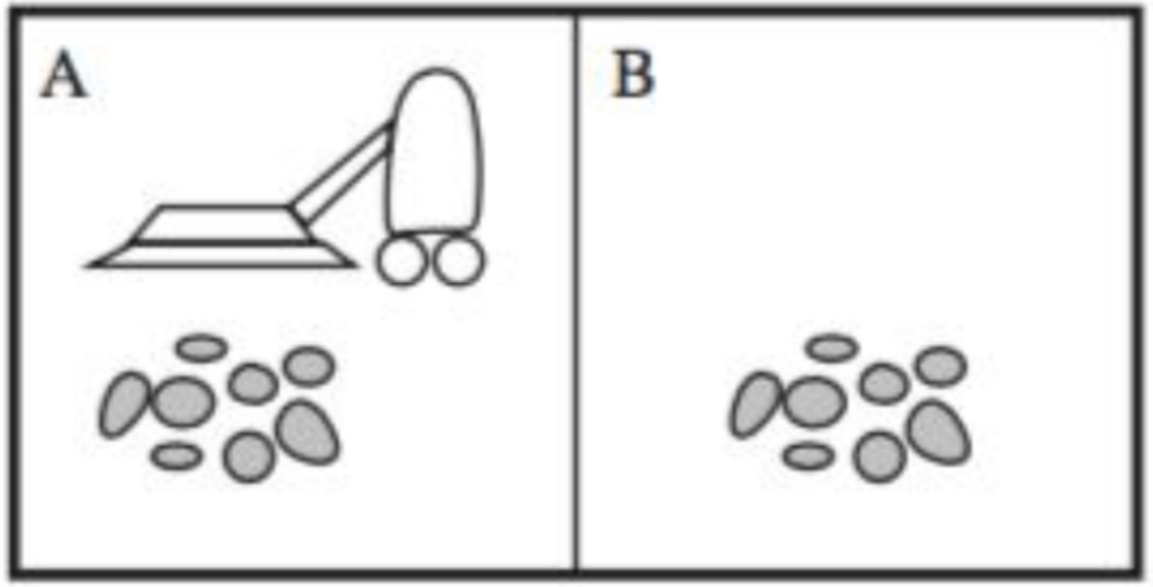
3. **Actions**
   a. What can the agent do? (operations on the environment)

Search

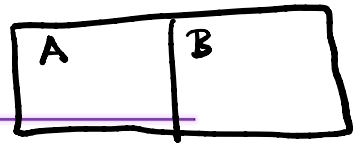
4. Initial state
   a. e.g. $[A, 'dirty']$ for the vacuum

5. Goal test
   a. Determines whether a given state is the goal state.

['A', [D, D]]



Search

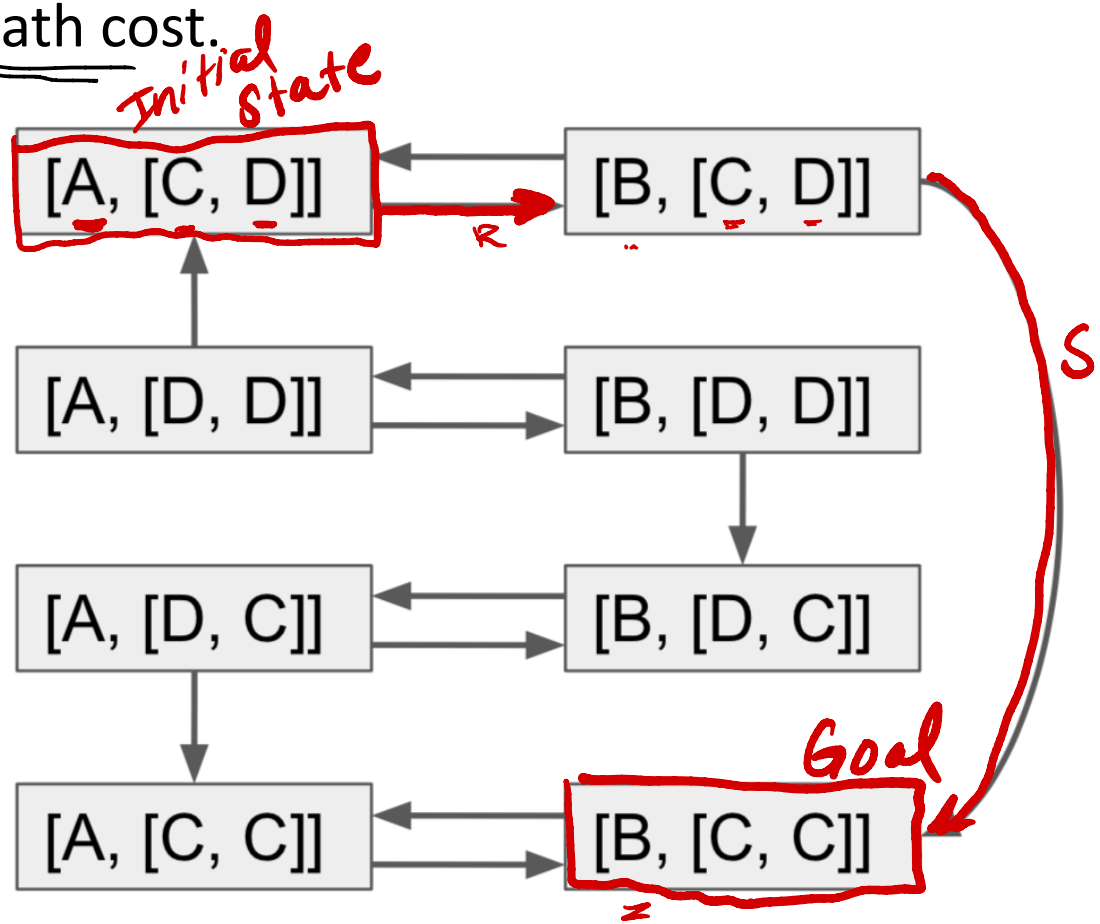


- A **solution** to a problem (search) is a sequence of actions that leads from the initial state to the goal state.

- An **optimal solution** is a solution with the lowest path cost.

Intelligent agents are supposed to maximize their performance measure.

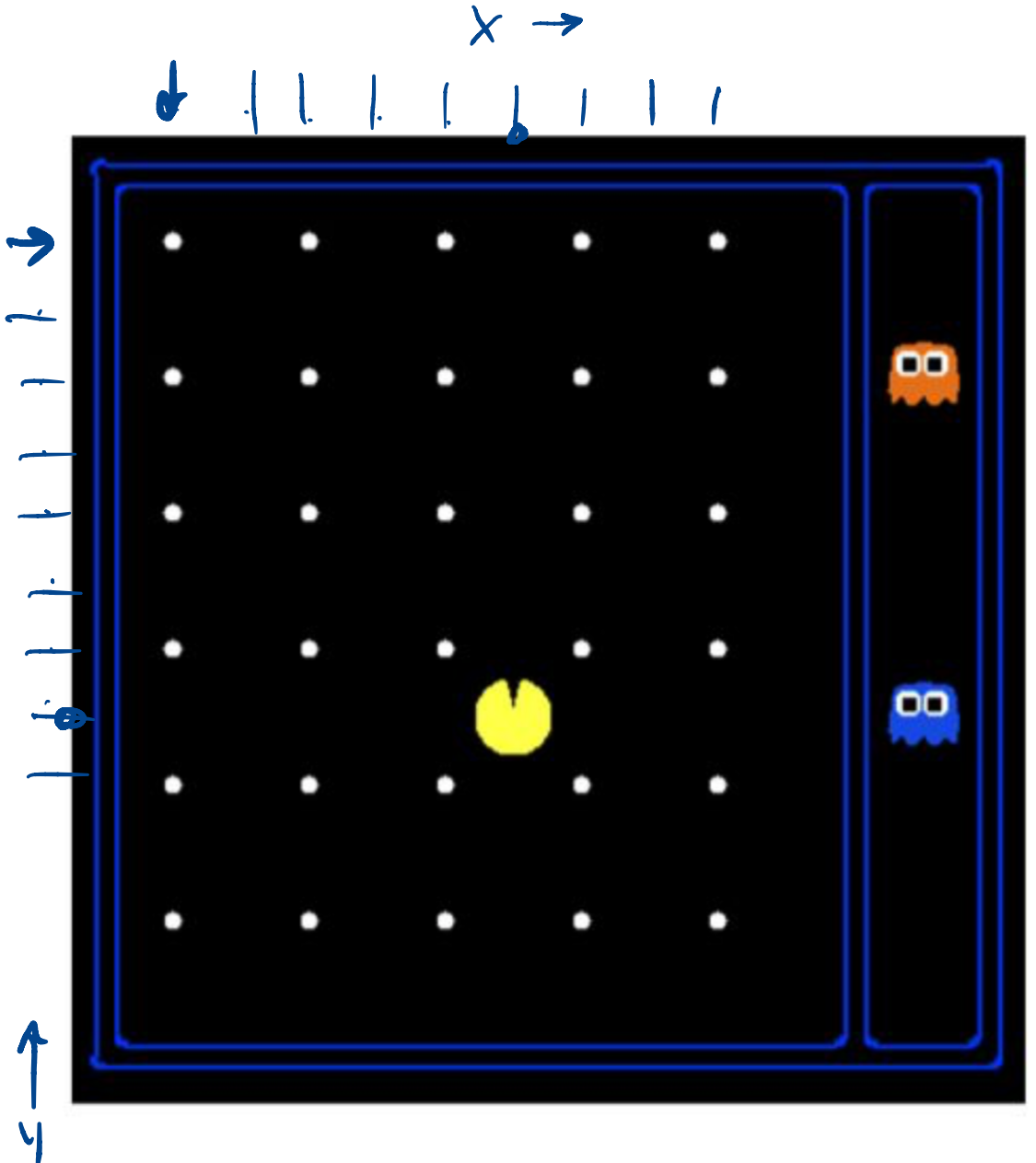
Goals - helps organize behavior



Search

Example: Pac-Man

1. State space — location/direction Pacman, food, ghosts
2. Transition model
   - locations of Pacman + ghosts
   - dot booleans
3. Actions — Left, right, up, down
4. Initial state — (6, 8)
5. Goal test — All food boolean dots = 0



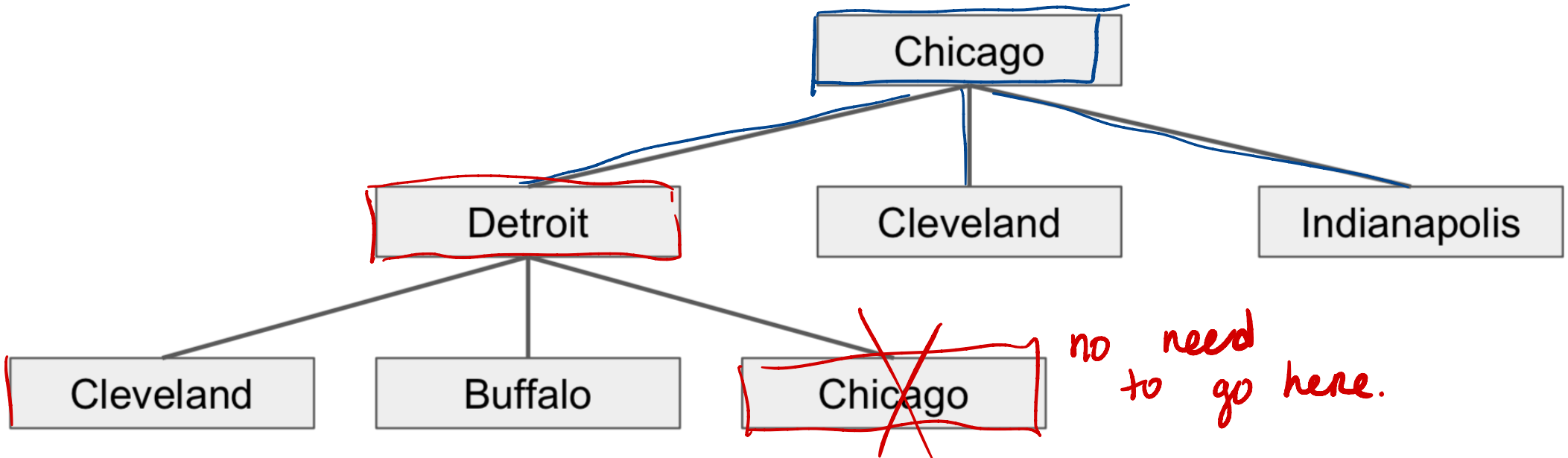
Search

Example: Traveling in the US northeast

1. State space - cities (nodes)
2. Transition model - roads
3. Actions - driving
4. Initial state 'Detroit'
5. Goal test current state == 'New York' ?

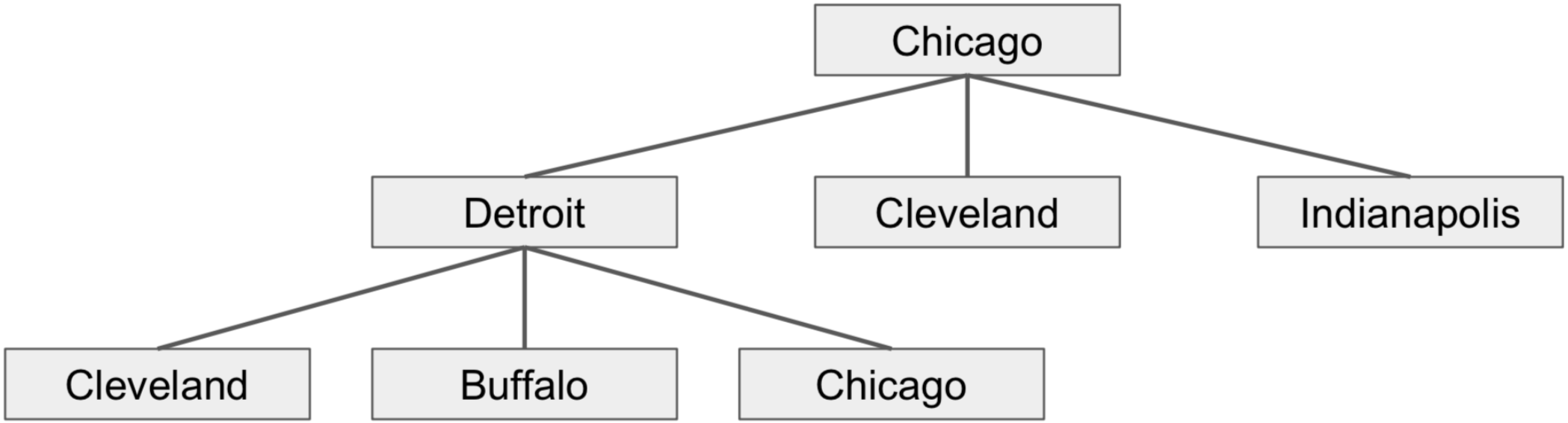
Search Trees

- "What if" tree of plans and outcomes.
- Initial state is root node.
- Children correspond to successor states.
- For most problems, we can't/don't want to build the whole tree.



Search Trees

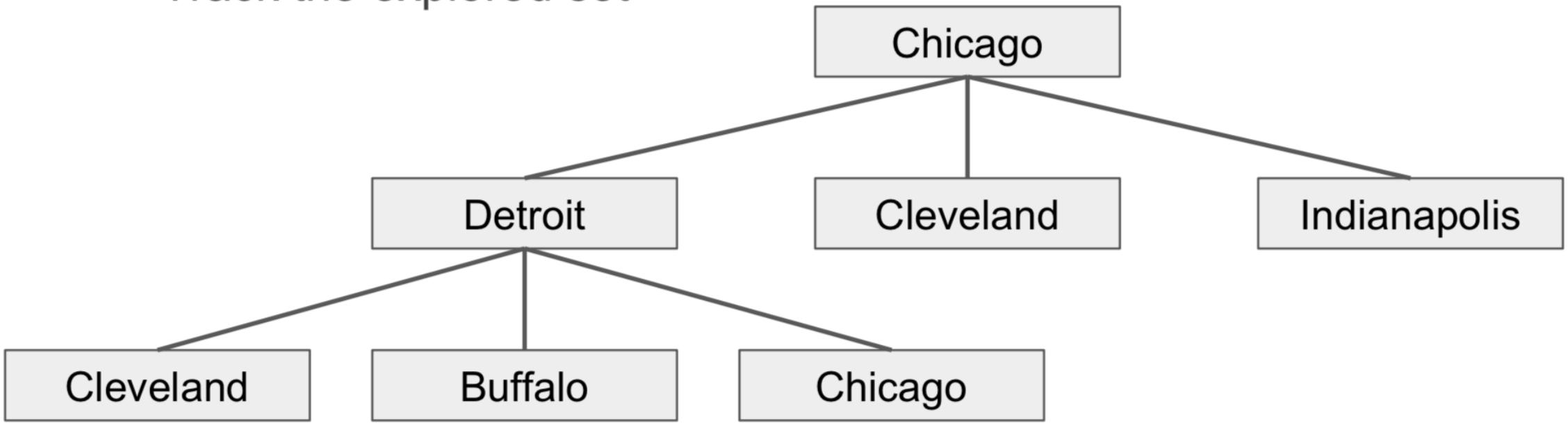
- **Expanding** the current state -- where to explore more?
- **Generating** new sets of states
  - **Leaves** on the **frontier**; the **internal vertices** are the **explored set**.
- How we choose which leaves to expand further depends on our search strategy



Search Trees

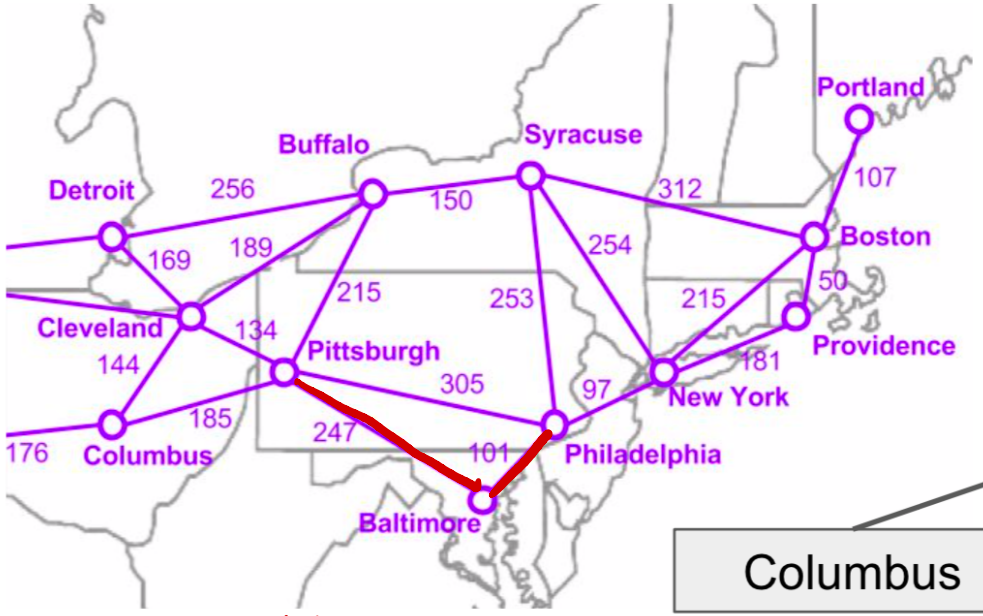
- **Loopy paths/redundant states** -- don't explore the same leaf twice!

  → Track the explored set

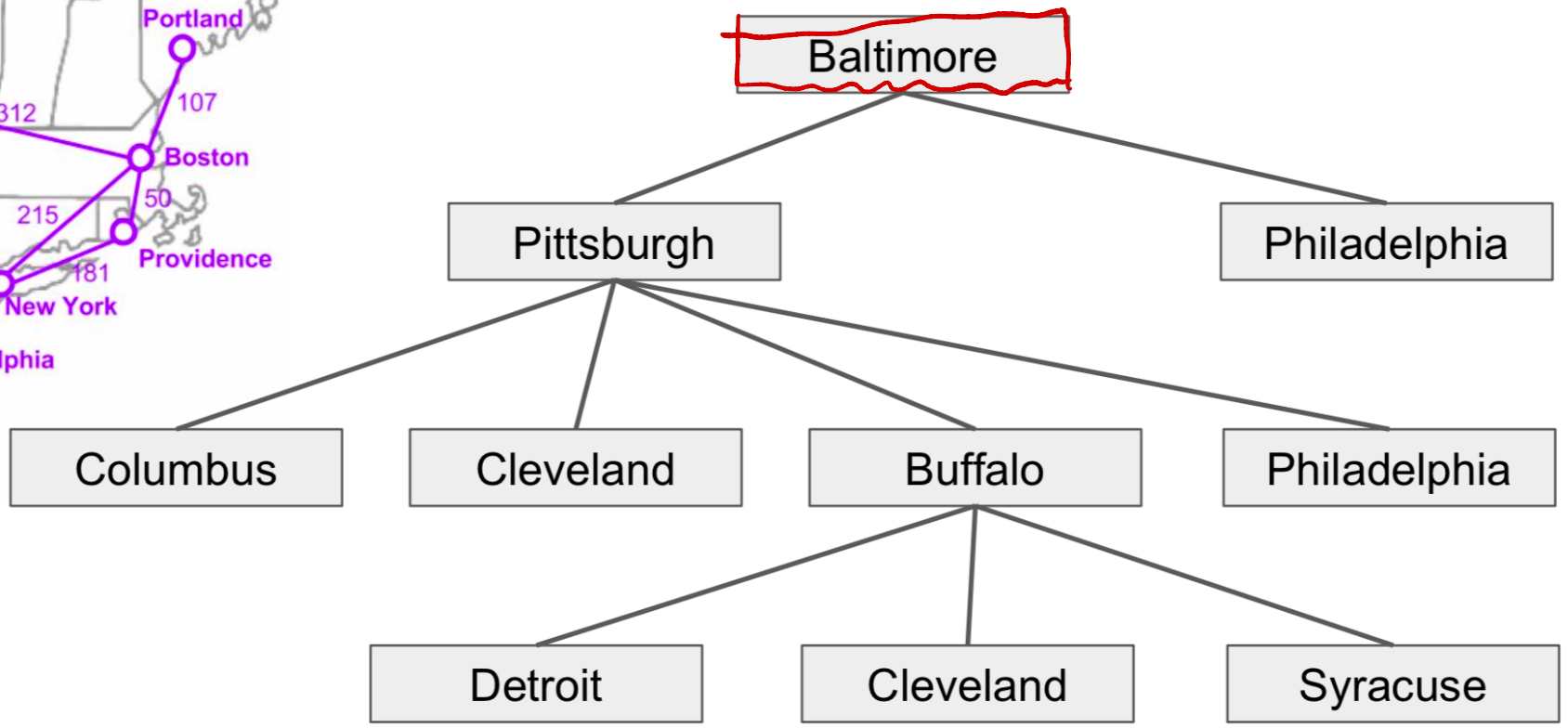


Search Trees vs Graphs

Each node on the search tree is a path on the problem graph

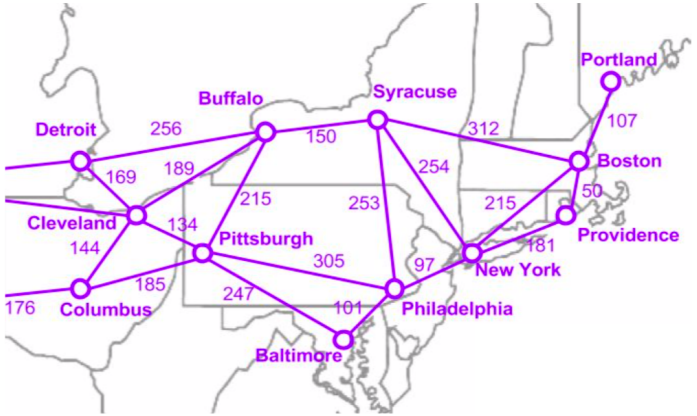


State space graph

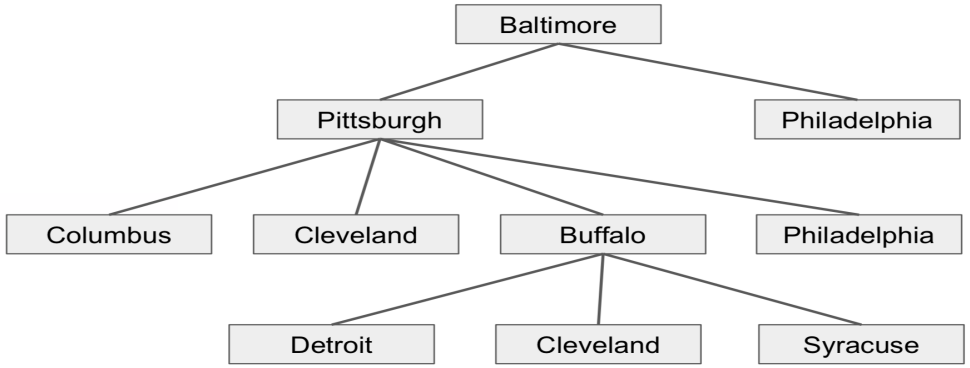


Search Trees vs Graphs

graph



search tree

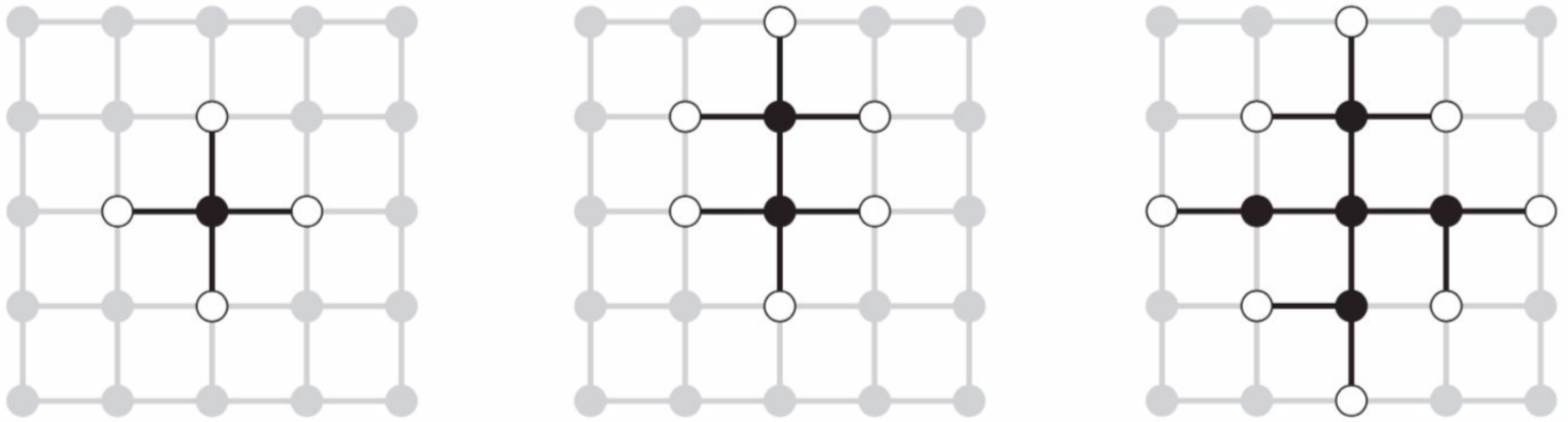


- Mathematical representation of a search problem
- Nodes are abstracted world configurations
- Arcs represent successors (action results)
- Goal test is a set of goal node(s)
- Each state only occurs once

- Nodes show states but correspond to paths
- Children correspond to successors
- Leaves are the “frontier” aka “fringe”
- Interior nodes are the ”explored” states
- Root node is the initial state
- Construct on demand, construct as little as possible

Search on a rectangular grid (2D)

- Just a (probably) more complicated graph to search than the US roadways one
- Each state has 4 successors
  → Search tree of depth d has 4^d leaves…
  → but many are already explored
- Useful for games (later!)



(a) (b) (c)

Next Time

Coding Examples!