Announcements

Project 0: Python Tutorial.

Due on Monday 1/27. 11:59 pm.

Announcements

Project 0: Python Tutorial.
Due on Monday 1/27. 11:59 pm.

Homework 0: Math Diagnostic.
Due on Wednesday 1/29. 11:59 pm.

Announcements

Project 0: Python Tutorial.
Due on Monday 1/27. 11:59 pm.

Homework 0: Math Diagnostic.
Due on Wednesday 1/29. 11:59 pm.

Project 1: Search.
Out: longer than average. Get started.
Due Friday. 2/7. 11:59 pm.

Announcements

Project 0: Python Tutorial.
Due on Monday 1/27. 11:59 pm.

Homework 0: Math Diagnostic.
Due on Wednesday 1/29. 11:59 pm.

Project 1: Search.
Out: longer than average. Get started.
Due Friday. 2/7. 11:59 pm.

Sections.
Start next week!

Announcements

Project 0: Python Tutorial.
Due on Monday 1/27. 11:59 pm.

Homework 0: Math Diagnostic.
Due on Wednesday 1/29. 11:59 pm.

Project 1: Search.
Out: longer than average. Get started.
Due Friday. 2/7. 11:59 pm.

Sections.
Start next week!

Office Hours: still working on rooms.
Department/university issues still being dealt with.

Announcements

Project 0: Python Tutorial.
Due on Monday 1/27. 11:59 pm.

Homework 0: Math Diagnostic.
Due on Wednesday 1/29. 11:59 pm.

Project 1: Search.
Out: longer than average. Get started.
Due Friday. 2/7. 11:59 pm.

Sections.
Start next week!

Office Hours: still working on rooms.
Department/university issues still being dealt with.

Check: signed up for Piazza and Gradescope?

Announcements

Project 0: Python Tutorial.
Due on Monday 1/27. 11:59 pm.

Homework 0: Math Diagnostic.
Due on Wednesday 1/29. 11:59 pm.

Project 1: Search.
Out: longer than average. Get started.
Due Friday. 2/7. 11:59 pm.

Sections.
Start next week!

Office Hours: still working on rooms.
Department/university issues still being dealt with.

Check: signed up for Piazza and Gradescope?

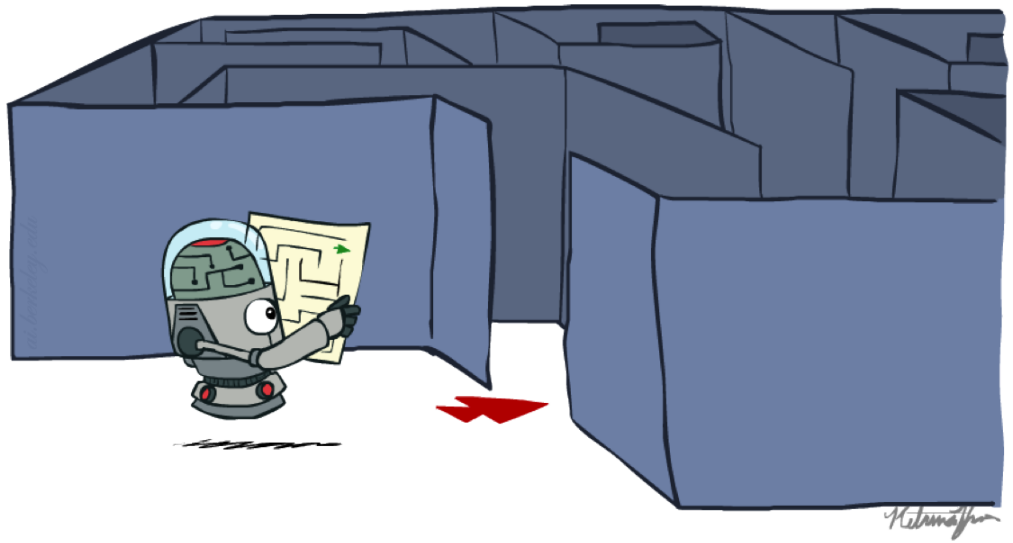
Pinned Post: AI in the news!

Lecture Attendance Link

Lecture Attendence Link: http://bit.ly/2GEMokS

CS 188: Artificial Intelligence

Search.



Today.

Agents that Plan Ahead

Today.

Agents that Plan Ahead

Search Problems.

Today.

- Agents that Plan Ahead
- Search Problems.
  - Model world with state space.

Today.

- Agents that Plan Ahead
- Search Problems.
  - Model world with state space.
  - Setting up state spaces.

Today.

- Agents that Plan Ahead
- Search Problems.
  - Model world with state space.
  - Setting up state spaces.
- Maybe today.

Today.

- Agents that Plan Ahead
- Search Problems.
  - Model world with state space.
  - Setting up state spaces.
- Maybe today.
- Uninformed Search Methods:

Today.

- Agents that Plan Ahead
- Search Problems.
  - Model world with state space.
  - Setting up state spaces.
- Maybe today.
- Uninformed Search Methods:
  - Depth-First Search

Today.

- Agents that Plan Ahead
- Search Problems.
  - Model world with state space.
  - Setting up state spaces.
- Maybe today.
- Uninformed Search Methods:
  - Depth-First Search
  - Breadth-First Search

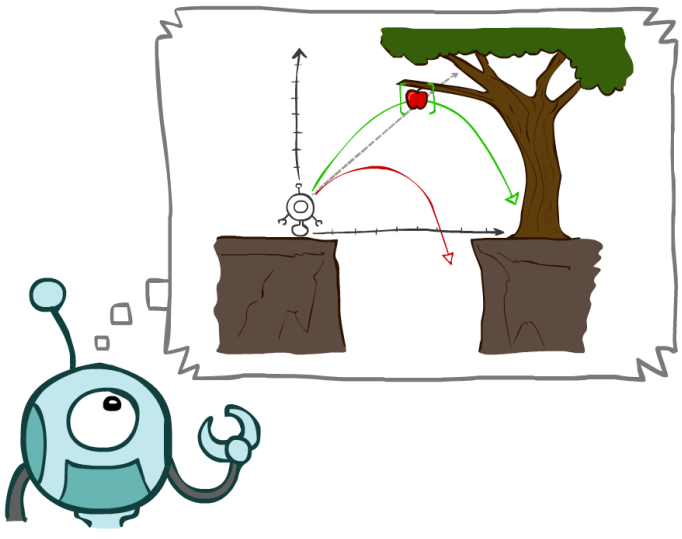
Today.

Agents that Plan Ahead

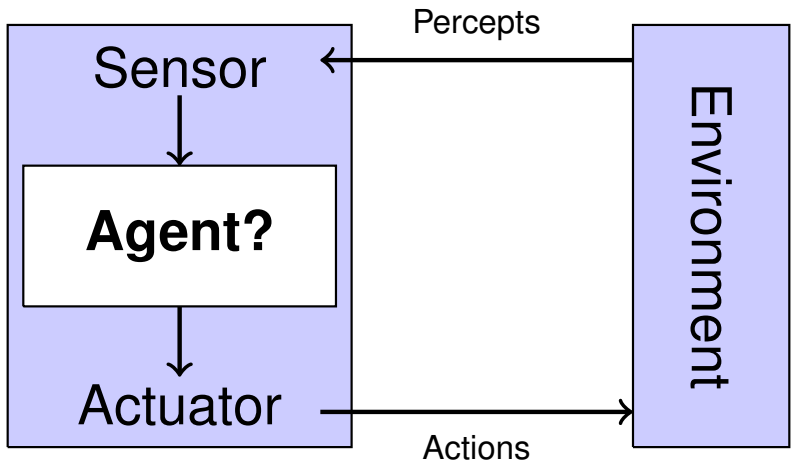
Search Problems.
- Model world with state space.
- Setting up state spaces.

Maybe today.

Uninformed Search Methods:
- Depth-First Search
- Breadth-First Search
- Uniform-Cost Search

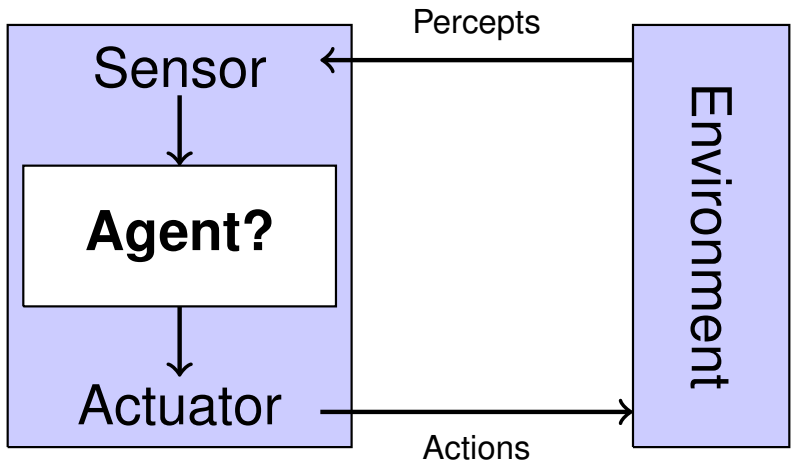


Frame of AI



An agent percieves its environment with sensors and acts on environment using actuators.

Frame of AI

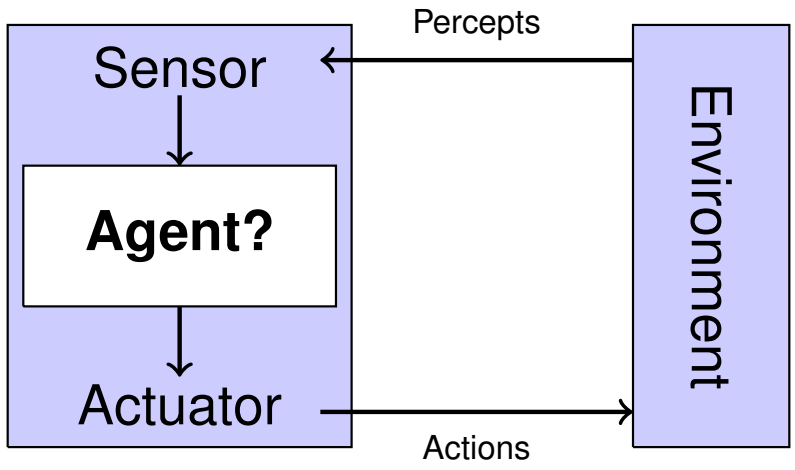


An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors:

Frame of AI

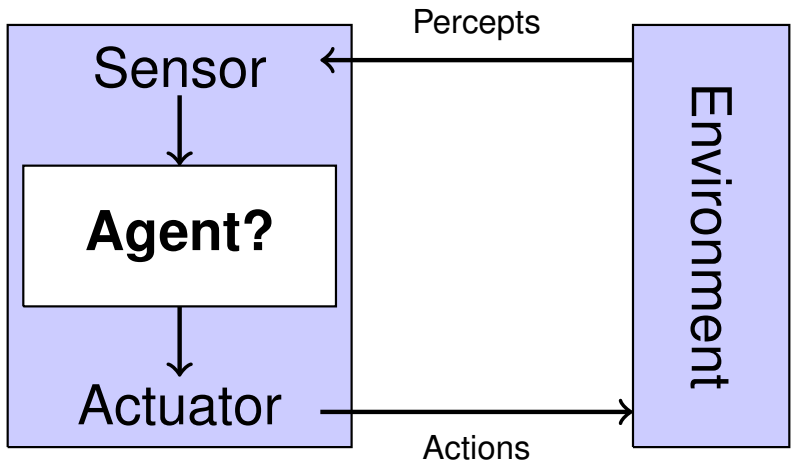


An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera,

Frame of AI

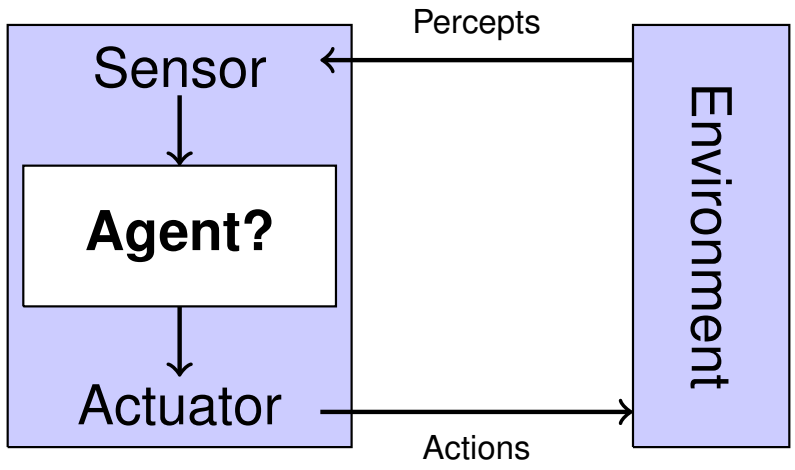


An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera, lidar,

Frame of AI

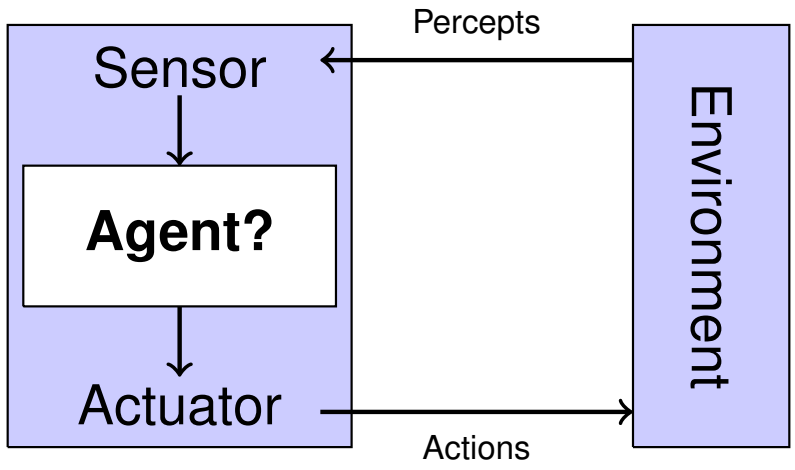


An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera, lidar, speed gauge,

Frame of AI



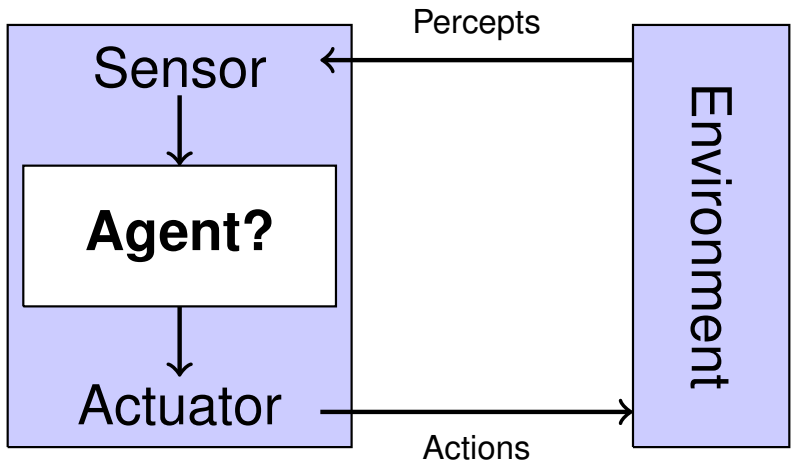
An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera, lidar, speed gauge, ..

Actuators:

Frame of AI



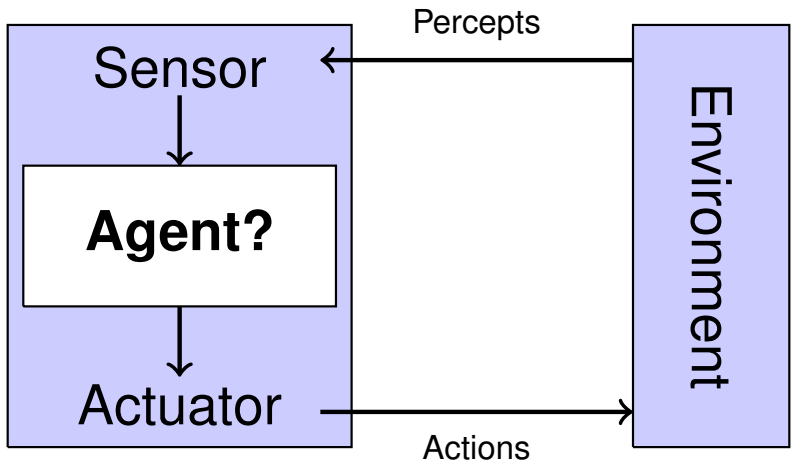
An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal,

Frame of AI



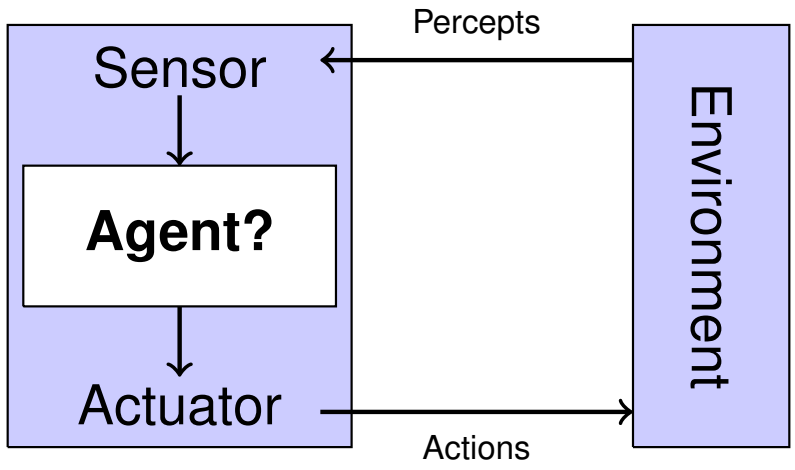
An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal, , steering wheel, brake petal,

Frame of AI



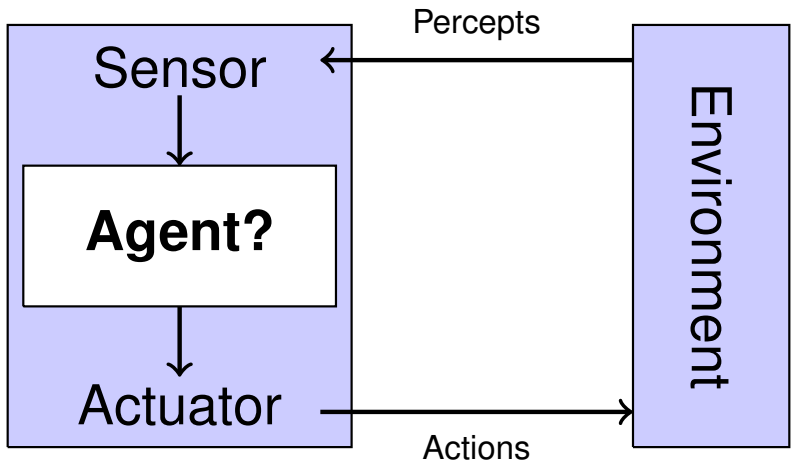
An agent percieves its environment with sensors and acts on environment using actuators.

Car:

Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal, , steering wheel, brake petal, ...

Frame of AI



An agent percieves its environment with sensors and acts on environment using actuators.

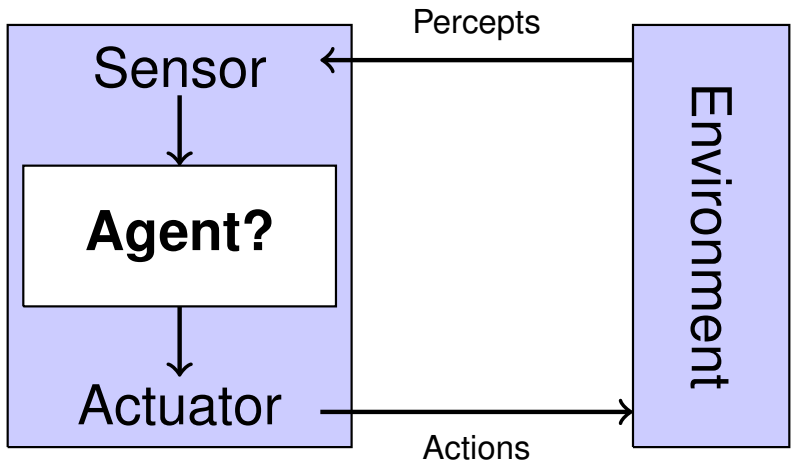
Car:

Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal, , steering wheel, brake petal, ...

Website,

Frame of AI



An agent percieves its environment with sensors and acts on environment using actuators.

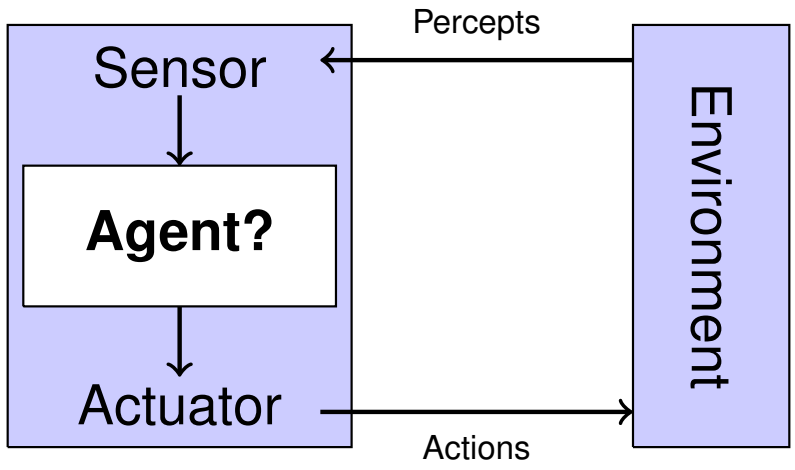
Car:

Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal, , steering wheel, brake petal, ...

Website, Program,

Frame of AI



An agent percieves its environment with sensors and acts on environment using actuators.

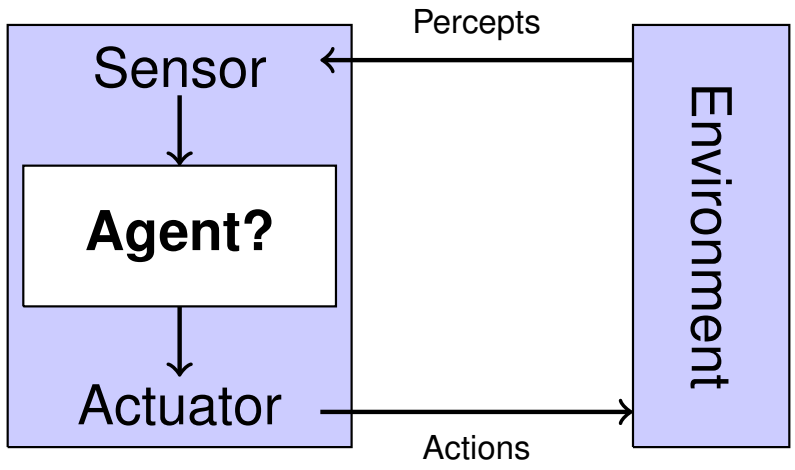
Car:

Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal, , steering wheel, brake petal, ...

Website, Program, ...

Frame of AI



An agent percieves its environment with sensors and acts on environment using actuators.

Car:

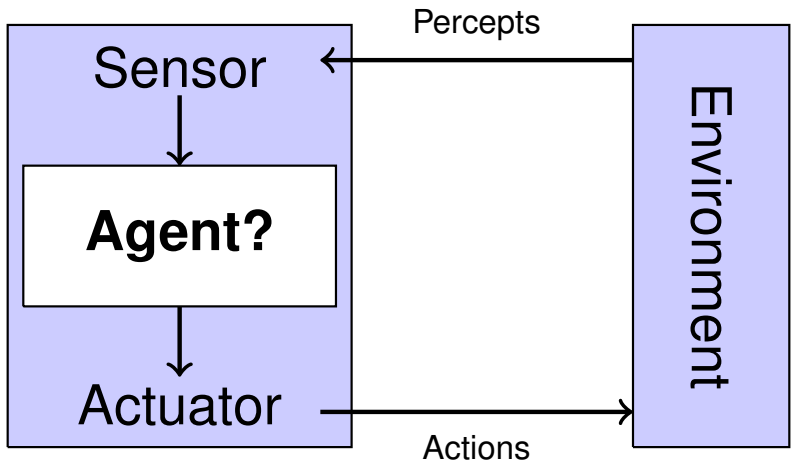
Sensors: camera, lidar, speed gauge, ..

Actuators: gas petal, , steering wheel, brake petal, ...

Website, Program, ...

Input/Output:

Frame of AI



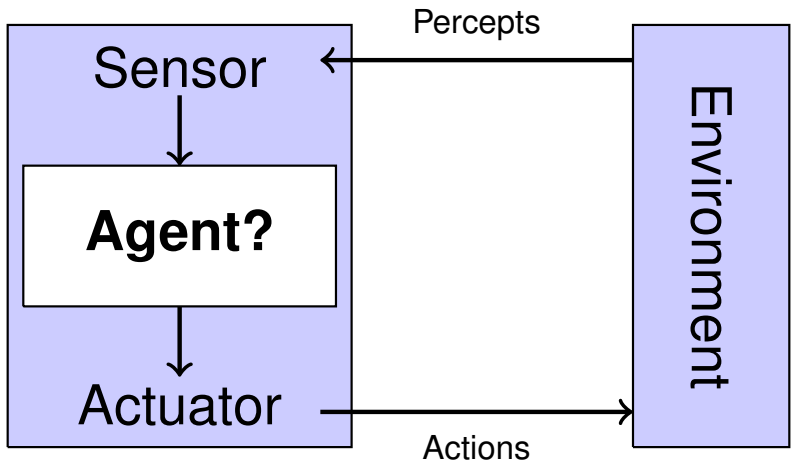
An agent percieves its environment with sensors and acts on environment using actuators.

Car:
Sensors: camera, lidar, speed gauge, ..
Actuators: gas petal, , steering wheel, brake petal, ...

Website, Program, ...

Input/Output: Outputs affect environment, which affects input

Frame of AI



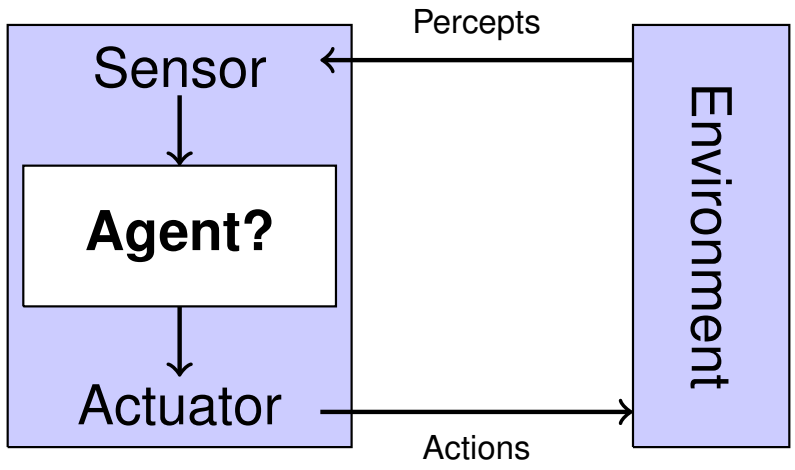
An agent percieves its environment with sensors and acts on environment using actuators.

Car:
Sensors: camera, lidar, speed gauge, ..
Actuators: gas petal, , steering wheel, brake petal, ...

Website, Program, ...

Input/Output: Outputs affect environment, which affects input , which outputs,

Frame of AI



An agent percieves its environment with sensors and acts on environment using actuators.

Car:
Sensors: camera, lidar, speed gauge, ..
Actuators: gas petal, , steering wheel, brake petal, ...

Website, Program, ...

Input/Output: Outputs affect environment, which affects input , which outputs, . . .

Rationality

A rational agent chooses actions that maximize expected utility.

Rationality

A rational agent chooses actions that maximize expected utility.

Today: agents that have a goal, and a cost.
E.g., reach goal with lowest cost.

Rationality

A rational agent chooses actions that maximize expected utility.

Today: agents that have a goal, and a cost.
E.g., reach goal with lowest cost.

Later: agents have numerical utilities, rewards, etc.
E.g., takes action that maximizes total reward over time.
(Reward: largest total profit. or expected total profit.)

Agent Design

The environment largely determines the agent design.

Agent Design

The environment largely determines the agent design.

Fully/partialy observable $\rightarrow$ agent request memory (internal state)

Agent Design

The environment largely determines the agent design.

- Fully/partialy observable $\rightarrow$ agent request memory (internal state)
- Discrete/ continuous $\rightarrow$ agent can/can't enumerate all states

Agent Design

The environment largely determines the agent design.

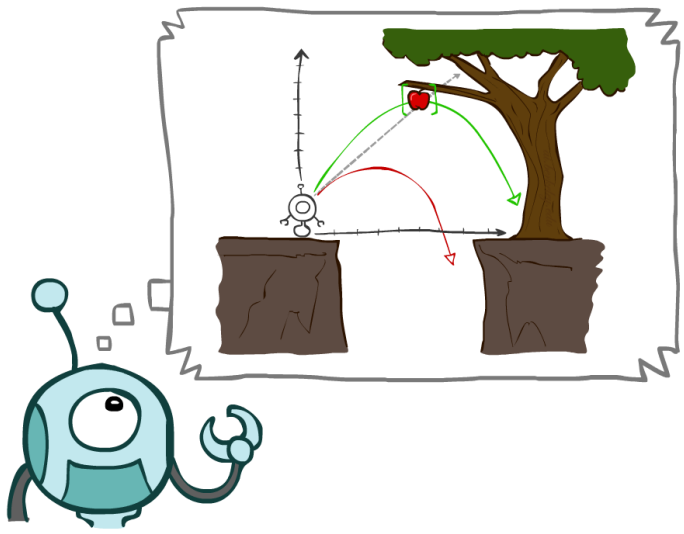
- Fully/partialy observable $\rightarrow$ agent request memory (internal state)
- Discrete/ continuous $\rightarrow$ agent can/can't enumerate all states
- Stochastic/deterministic $\rightarrow$ agent deals with contingencies

Agent Design

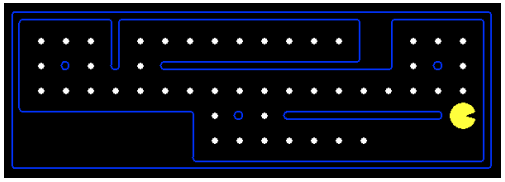
The environment largely determines the agent design.

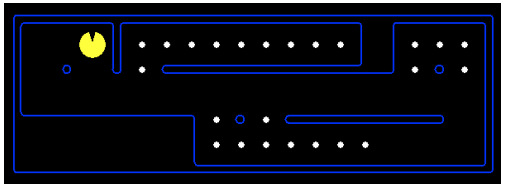
- Fully/partialy observable $\rightarrow$ agent request memory (internal state)
- Discrete/ continuous $\rightarrow$ agent can/can't enumerate all states
- Stochastic/deterministic $\rightarrow$ agent deals with contingencies
- Single-agent/multi-agent $\rightarrow$ agent may need to behave randomly.

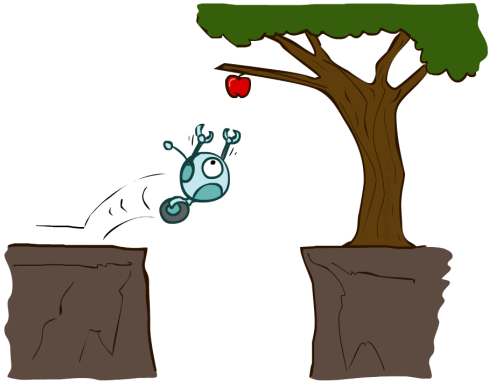
Agents that Plan



Reflex Agents



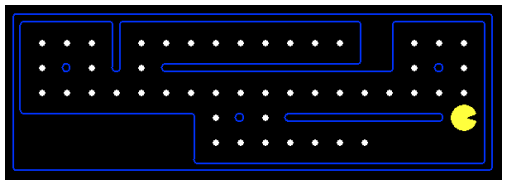


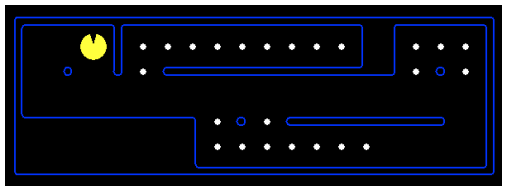


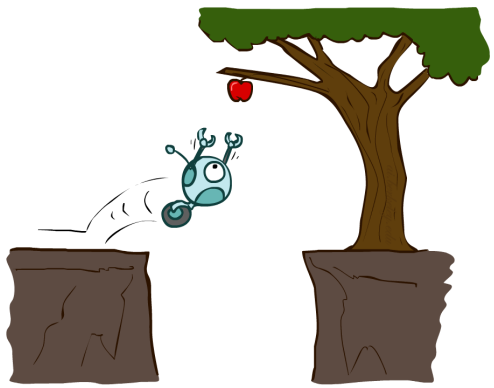
Reflex agents:

- Choose action based on current percept (and maybe memory)

Reflex Agents



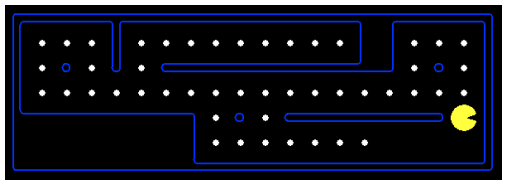


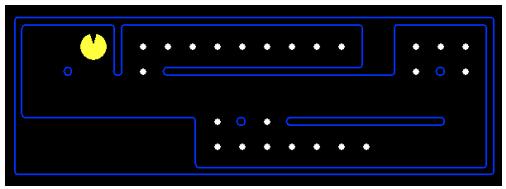


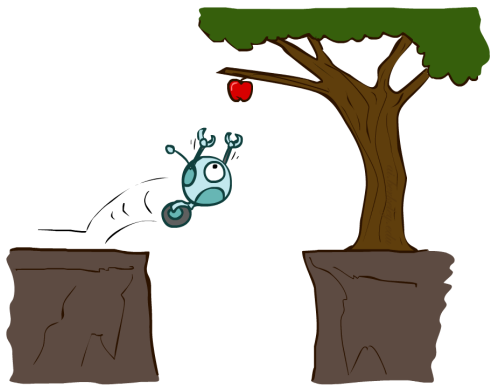
Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state

Reflex Agents



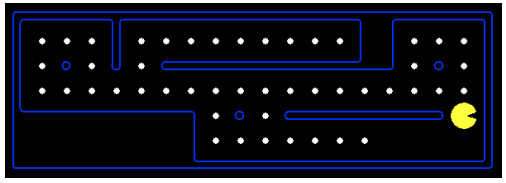


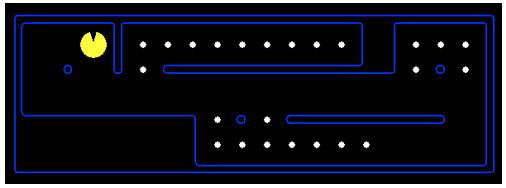


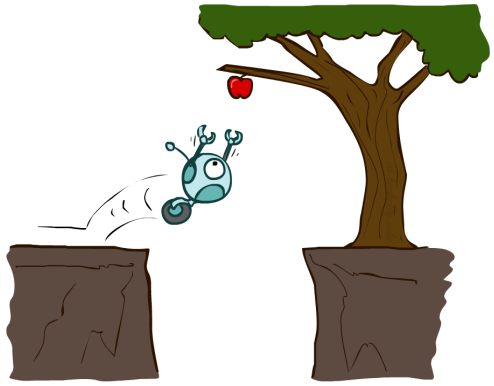
Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.

Reflex Agents



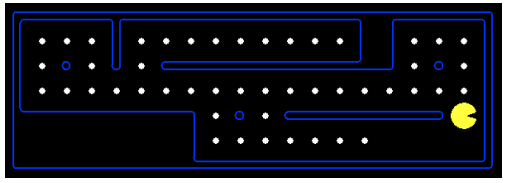


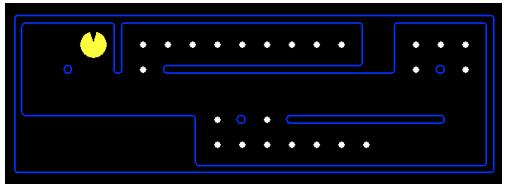


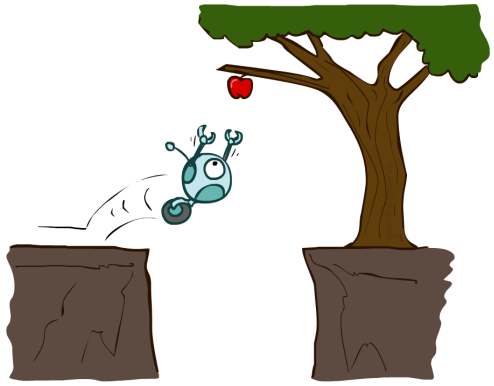
Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Reflex Agents



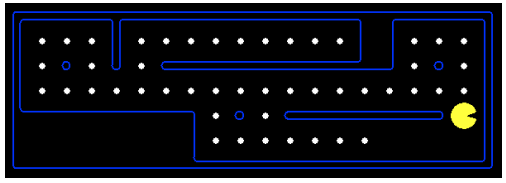


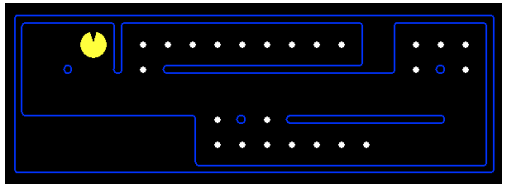


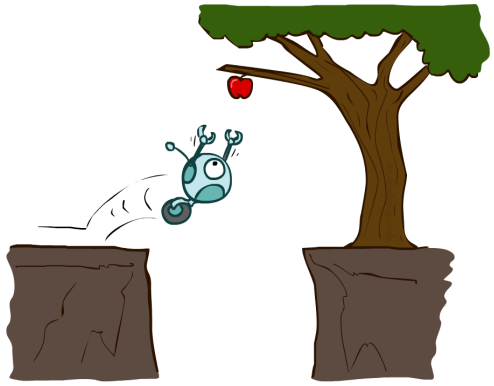
Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Reflex Agents





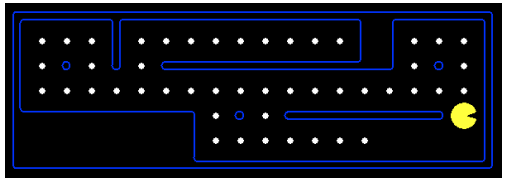


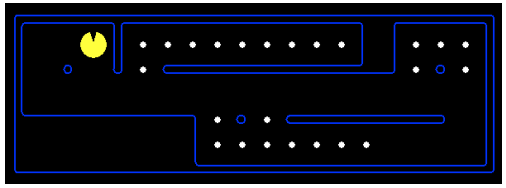
Reflex agents:

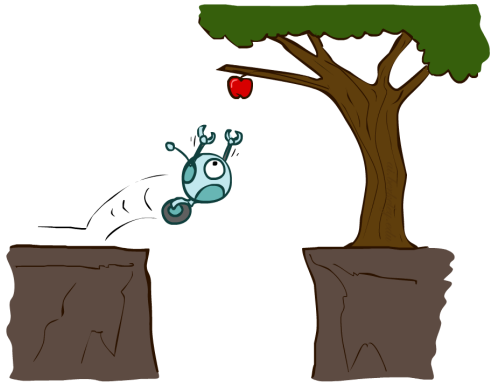
- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Can a reflex agent be rational?

Reflex Agents







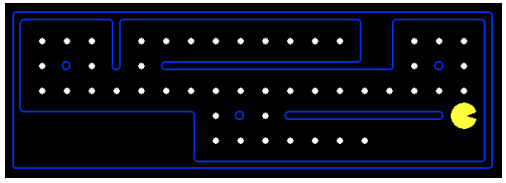
Reflex agents:

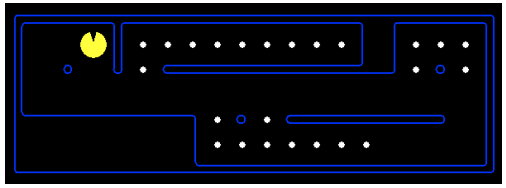
- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

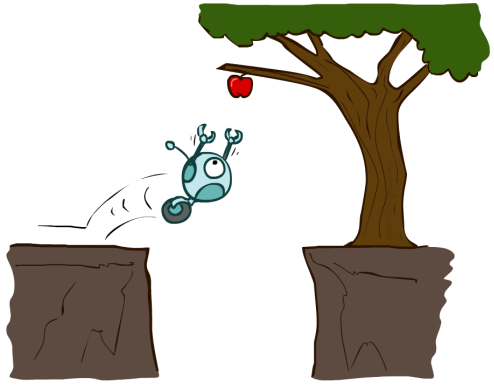
Can a reflex agent be rational?

Examples:

Reflex Agents







Reflex agents:

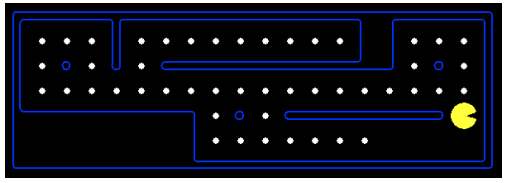
- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

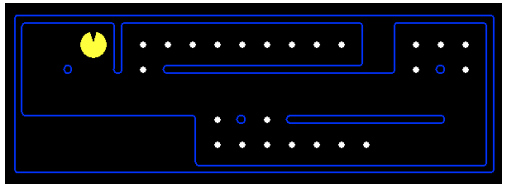
Can a reflex agent be rational?

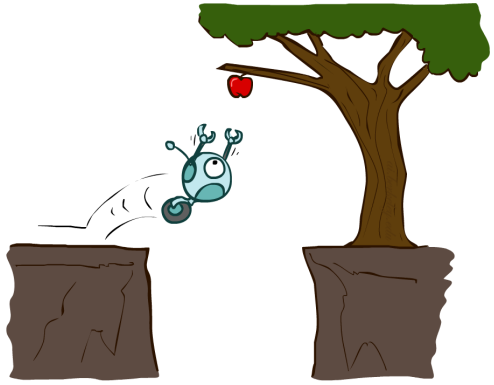
Examples:

Stove:

Reflex Agents







Reflex agents:

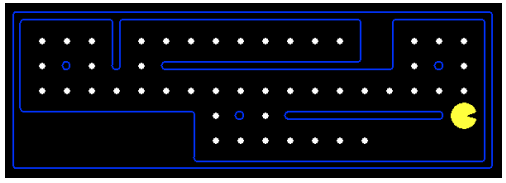
- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

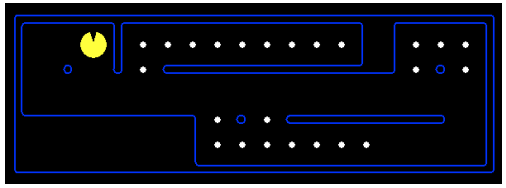
Can a reflex agent be rational?

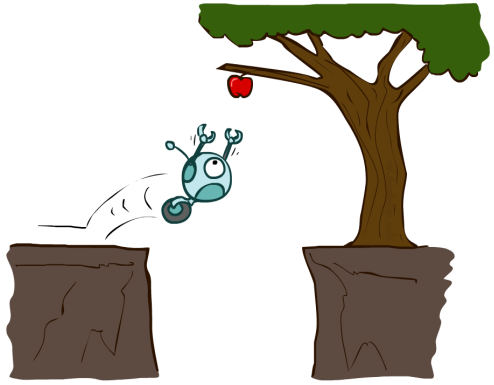
Examples:

Stove: hot,

Reflex Agents







Reflex agents:

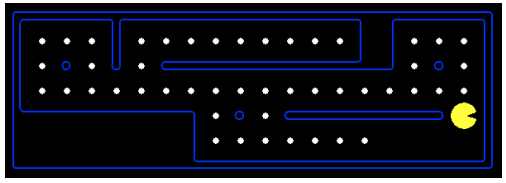
- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

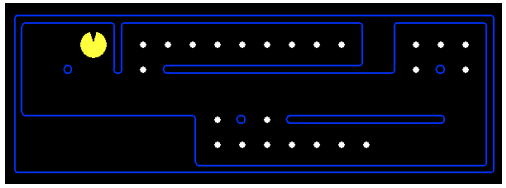
Can a reflex agent be rational?

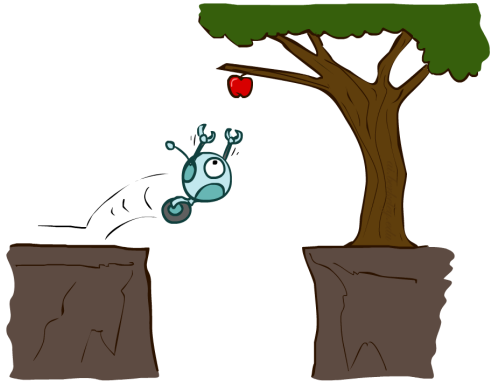
Examples:

Stove: hot, ouch!

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

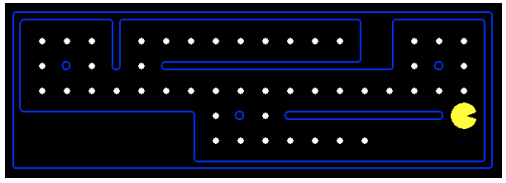
Can a reflex agent be rational?

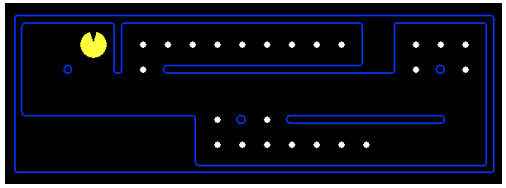
Examples:

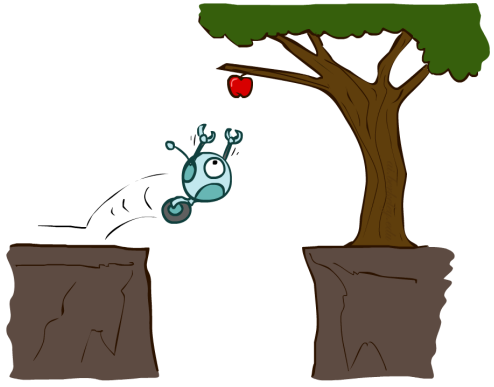
Stove: hot, ouch!

Car:

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

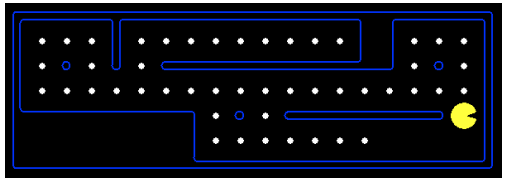
Can a reflex agent be rational?

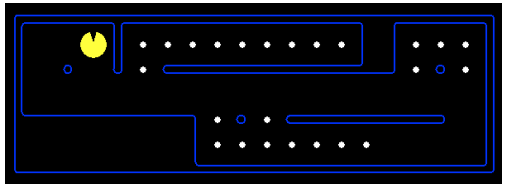
Examples:

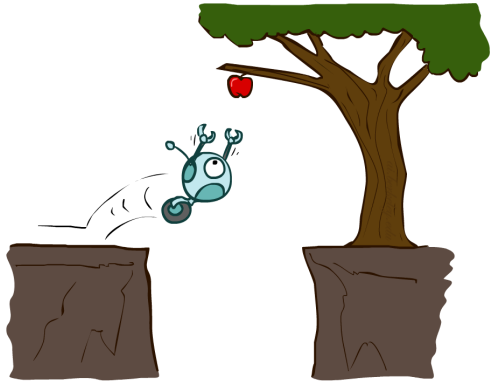
Stove: hot, ouch!

Car: deer,

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

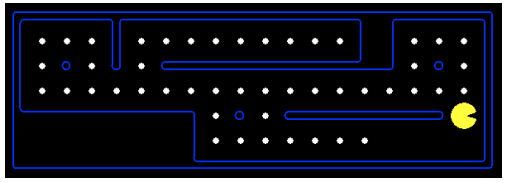
Can a reflex agent be rational?

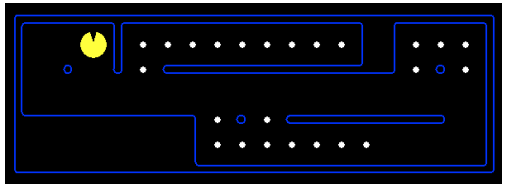
Examples:

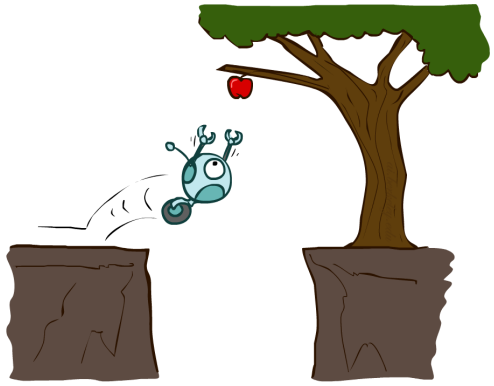
Stove: hot, ouch!

Car: deer, brake!

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Can a reflex agent be rational?

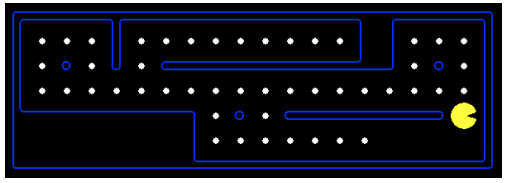
Examples:

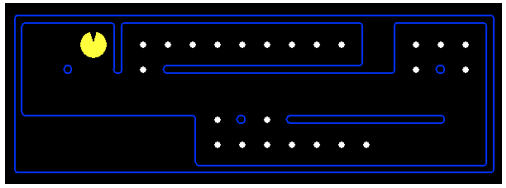
Stove: hot, ouch!

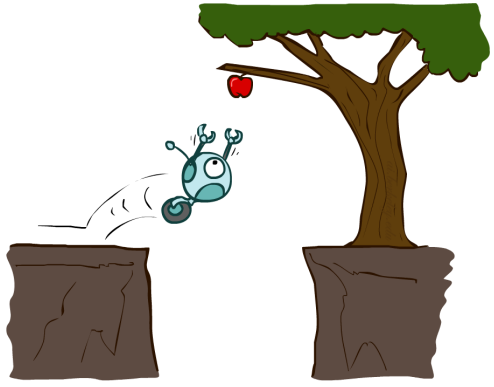
Car: deer, brake!

Go to Tahoe:

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Can a reflex agent be rational?

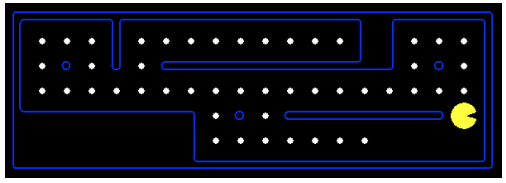
Examples:

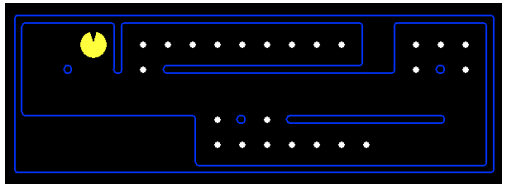
Stove: hot, ouch!

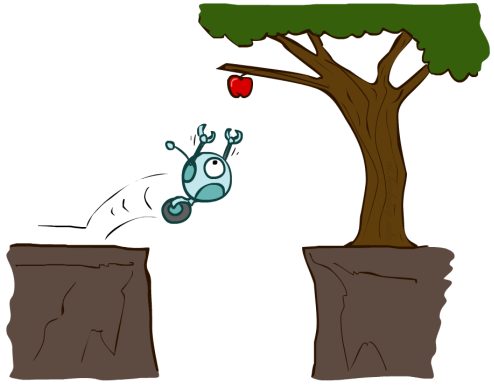
Car: deer, brake!

Go to Tahoe: skid in snow,

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Can a reflex agent be rational?

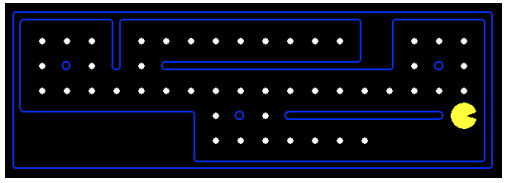
Examples:

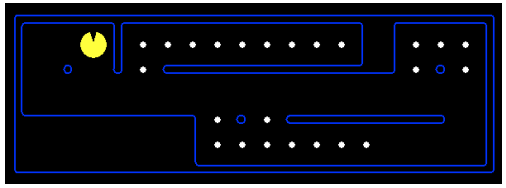
Stove: hot, ouch!

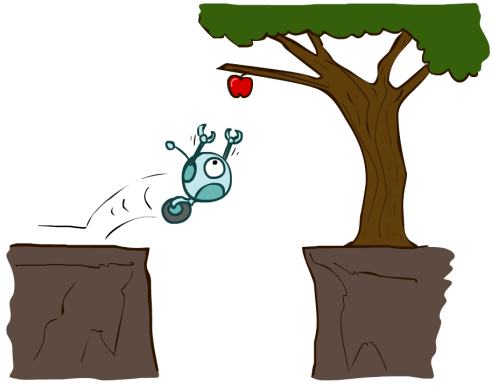
Car: deer, brake!

Go to Tahoe: skid in snow, ???

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Can a reflex agent be rational?

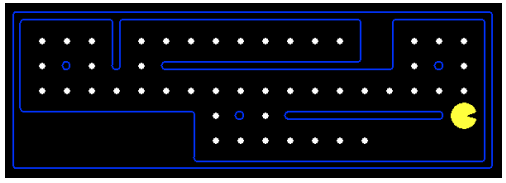
Examples:

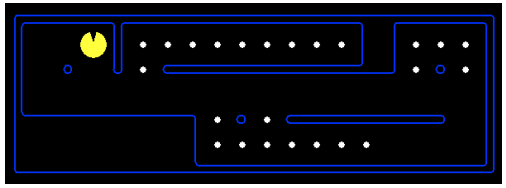
Stove: hot, ouch!

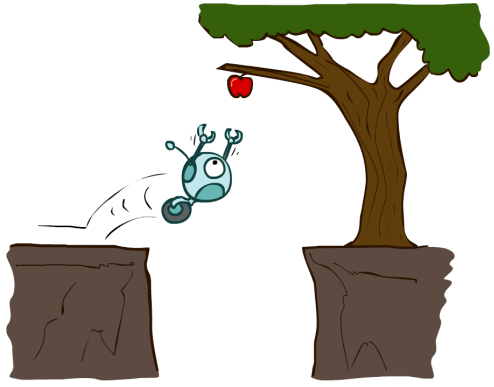
Car: deer, brake!

Go to Tahoe: skid in snow, ???

Reflex Agents







Reflex agents:

- Choose action based on current percept (and maybe memory)
- May have memory or a model of the world's current state
- Do not consider the future consequences of their actions.
- Consider how the world IS.

Can a reflex agent be rational?

Examples:

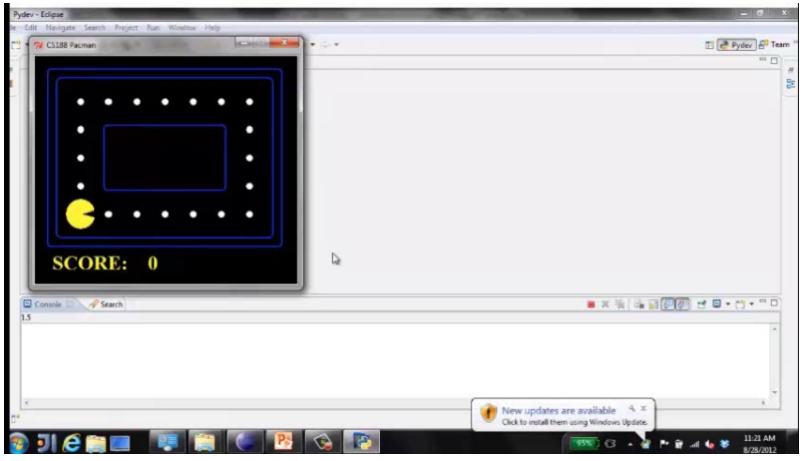
Stove: hot, ouch!

Car: deer, brake!

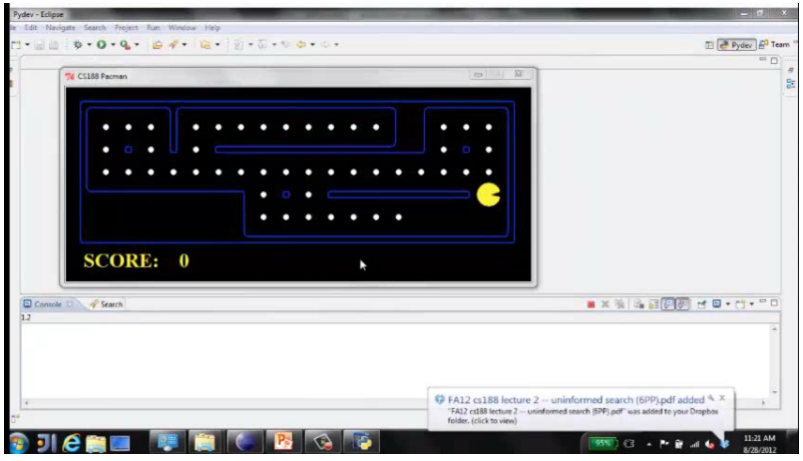
Go to Tahoe: skid in snow, ???

[Demo: reflex (L2D1 and L2D2)]

Video of Demo Reflex Optimal

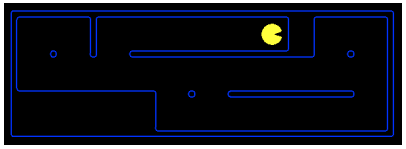


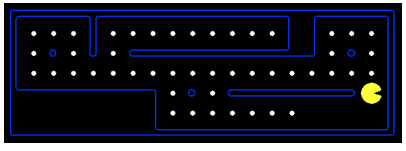
Video of Demo Reflex Odd

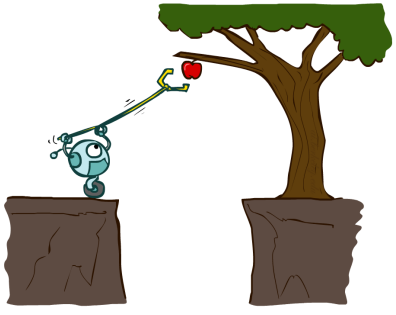


Planning Agents

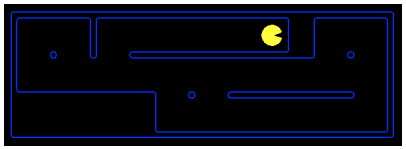
Planning agents:

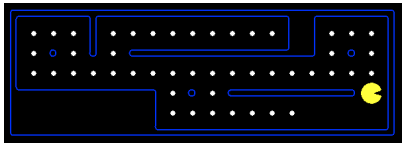


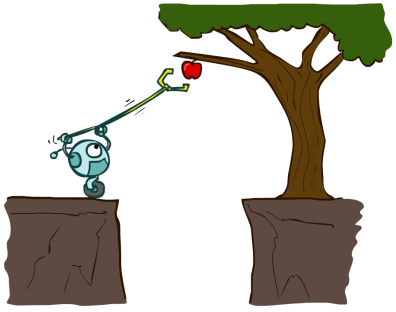




Planning Agents



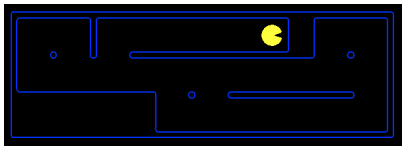


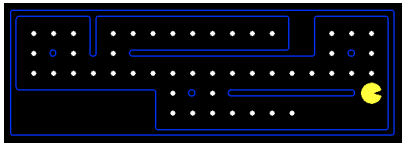


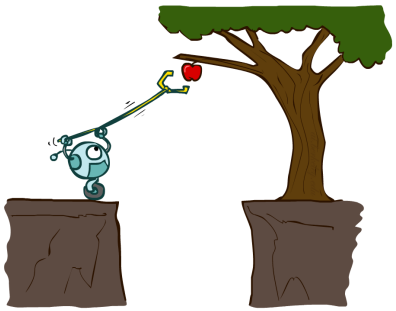
Planning agents:

Ask “what if?”

Planning Agents





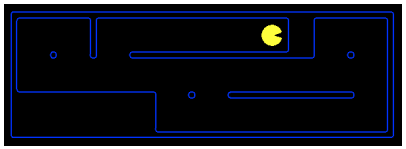


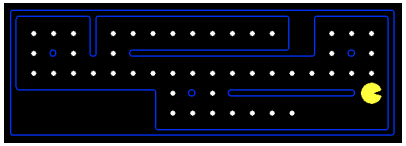
Planning agents:

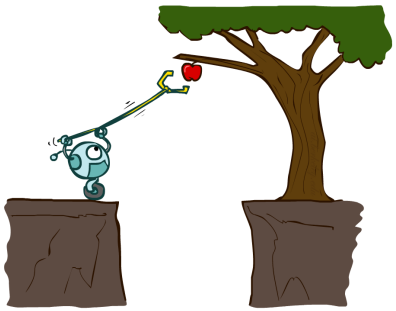
Ask “what if?”

Decisions based on (hypothesized) consequences of actions.

Planning Agents







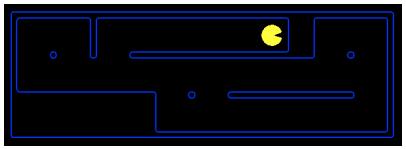
Planning agents:

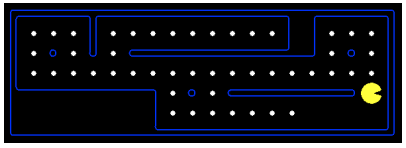
Ask “what if?”

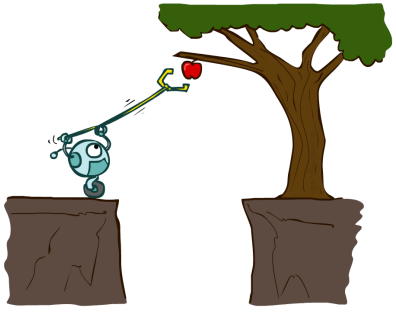
Decisions based on (hypothesized) consequences of actions.

Must have a model of how the world evolves in response to actions.

Planning Agents







Planning agents:

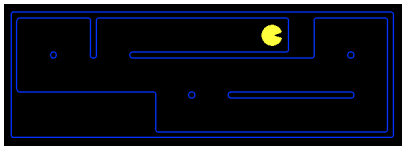
Ask “what if?”

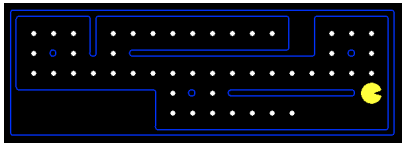
Decisions based on (hypothesized) consequences of actions.

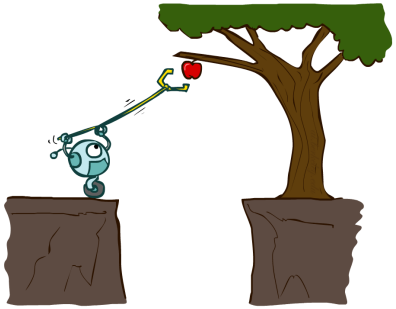
Must have a model of how the world evolves in response to actions.

Must formulate a goal (test).

Planning Agents







Planning agents:

Ask “what if?”

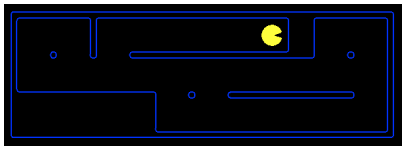
Decisions based on (hypothesized) consequences of actions.

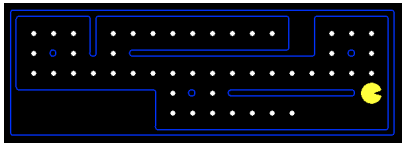
Must have a model of how the world evolves in response to actions.

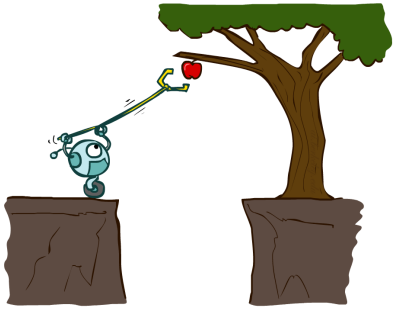
Must formulate a goal (test).

Consider how the world WOULD BE.

Planning Agents







Planning agents:

Ask “what if?”

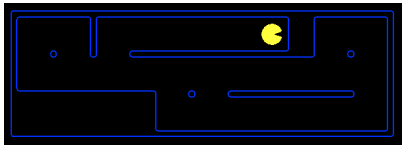
Decisions based on (hypothesized) consequences of actions.

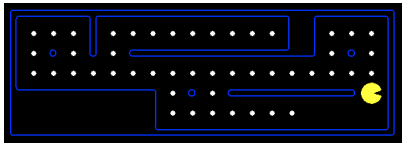
Must have a model of how the world evolves in response to actions.

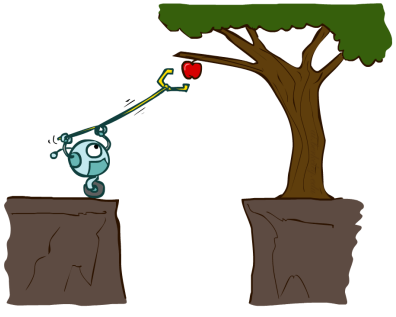
Must formulate a goal (test).

Consider how the world WOULD BE.

Planning Agents







Planning agents:

Ask “what if?”

Decisions based on (hypothesized) consequences of actions.

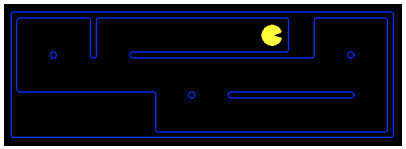
Must have a model of how the world evolves in response to actions.

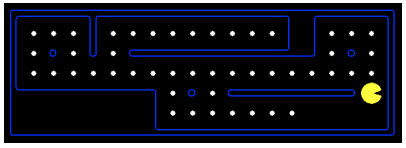
Must formulate a goal (test).

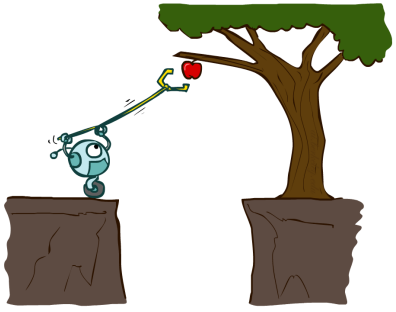
Consider how the world WOULD BE.

Optimal or not optimal.

Planning Agents





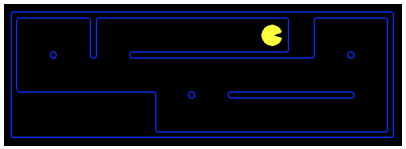


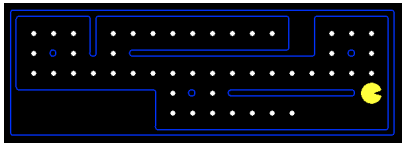
Planning agents:

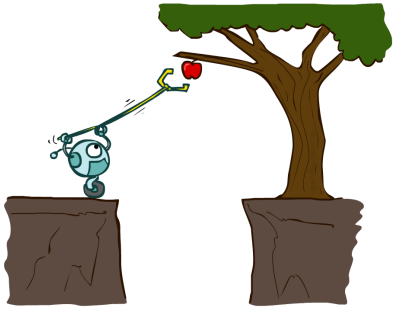
Ask “what if?”
Decisions based on (hypothesized) consequences of actions.
Must have a model of how the world evolves in response to actions.
Must formulate a goal (test).
Consider how the world WOULD BE.

Optimal or not optimal.
Complete or not.

Planning Agents





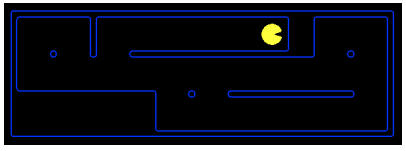


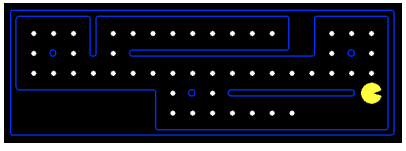
Planning agents:

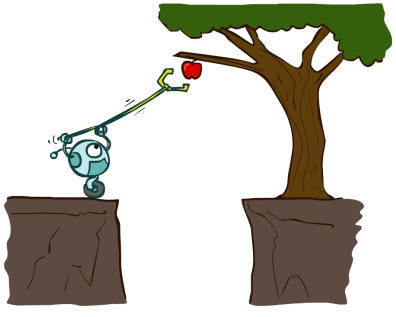
Ask “what if?”
Decisions based on (hypothesized) consequences of actions.
Must have a model of how the world evolves in response to actions.
Must formulate a goal (test).
Consider how the world WOULD BE.

Optimal or not optimal.
Complete or not.
Planning vs. replanning

Planning Agents







Planning agents:

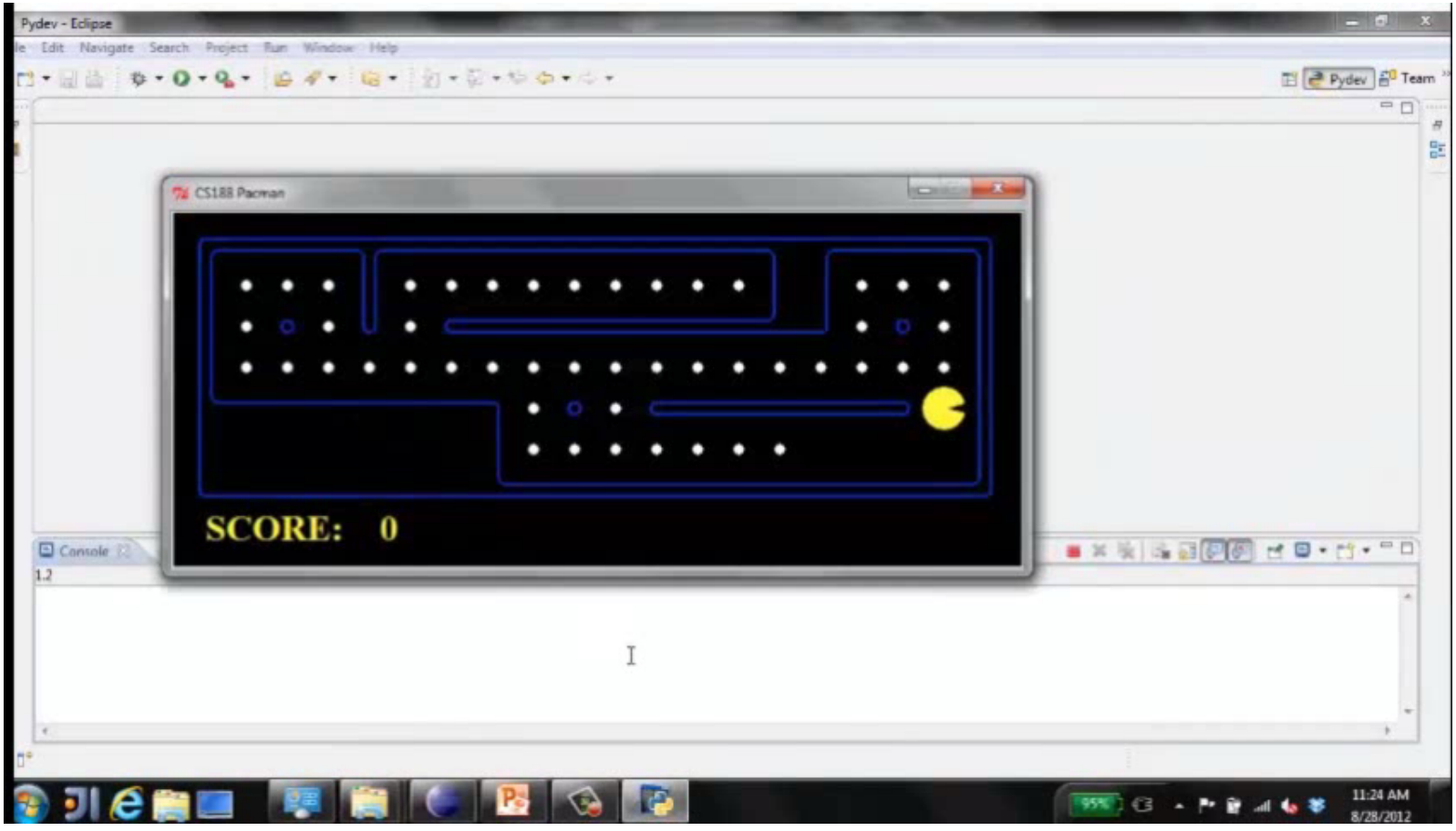
Ask “what if?”
Decisions based on (hypothesized) consequences of actions.
Must have a model of how the world evolves in response to actions.
Must formulate a goal (test).
Consider how the world WOULD BE.

Optimal or not optimal.
Complete or not.
Planning vs. replanning

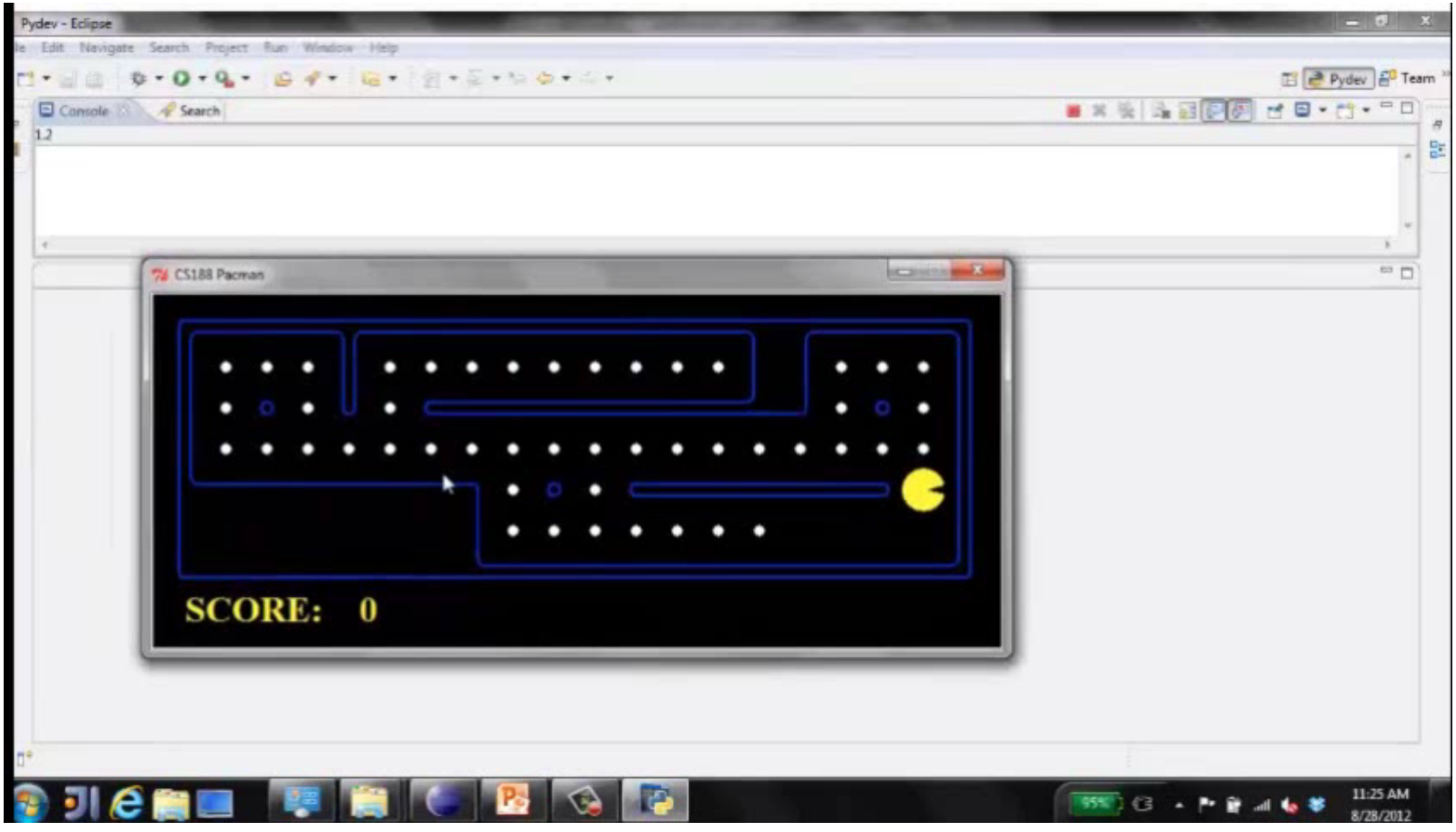
[Demo: nearest dot re-planning (L2D3), mastermind (L2D4)]

Video of Demo

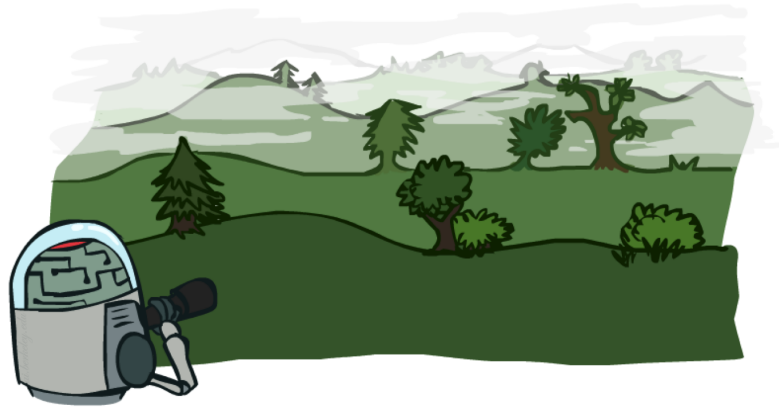
Replanning



Video of Demo Mastermind



Search Problems



Search Problems

A search problem consists of:

Search Problems

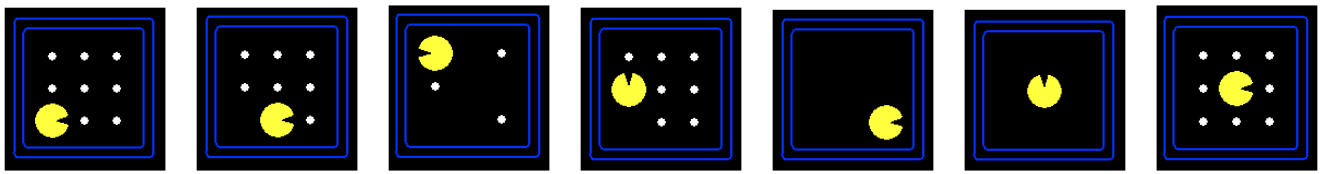
A search problem consists of:

A state space

Search Problems

A search problem consists of:

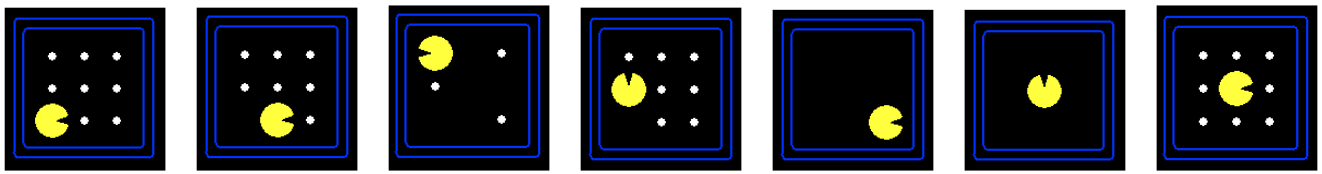
A state space



Search Problems

A search problem consists of:

A state space

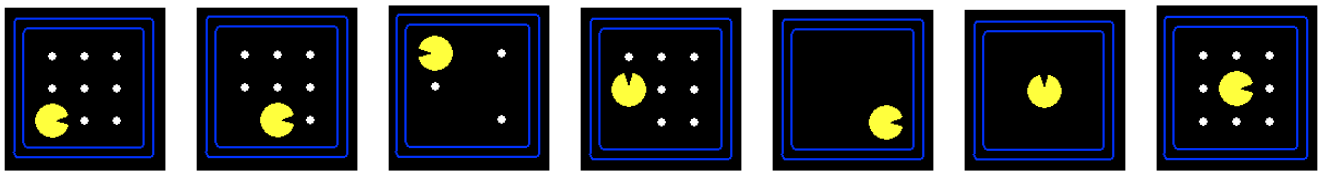


A successor function
(with actions, costs)

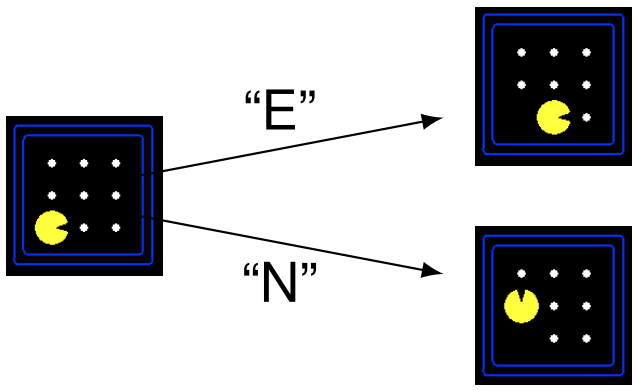
Search Problems

A search problem consists of:

A state space



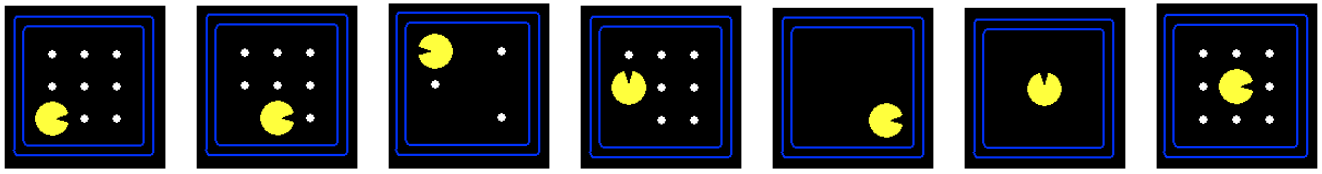
A successor function
(with actions, costs)



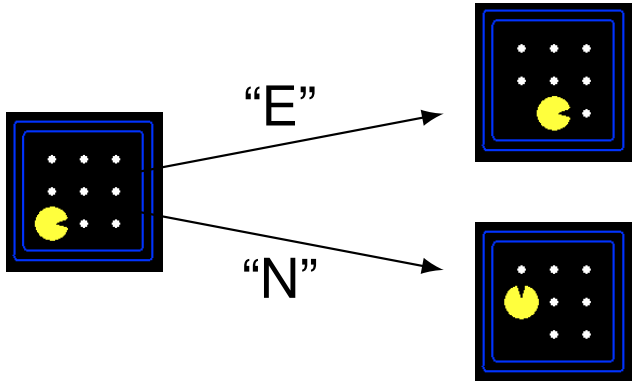
Search Problems

A search problem consists of:

A state space



A successor function
(with actions, costs)

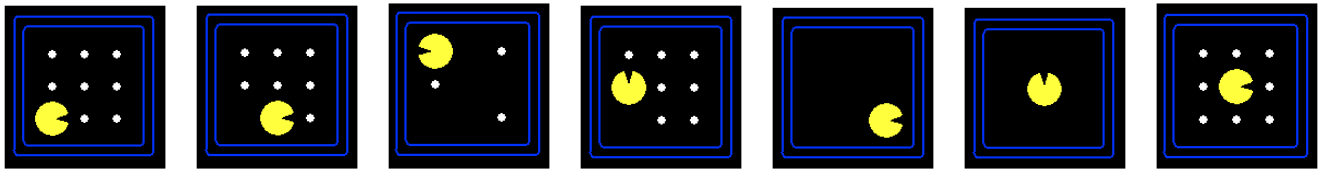


A start state and a goal test

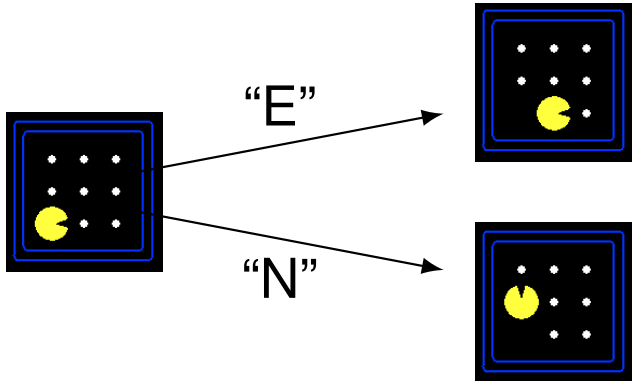
Search Problems

A search problem consists of:

A state space



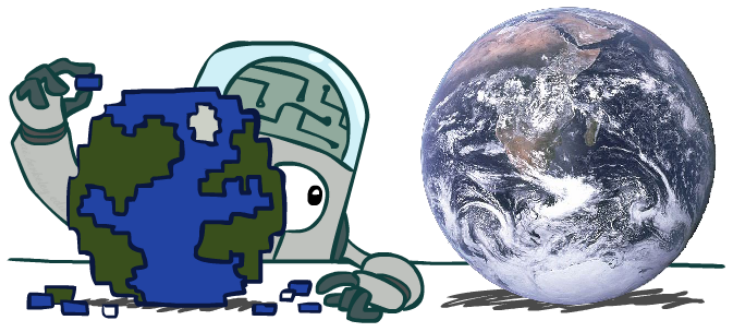
A successor function
(with actions, costs)



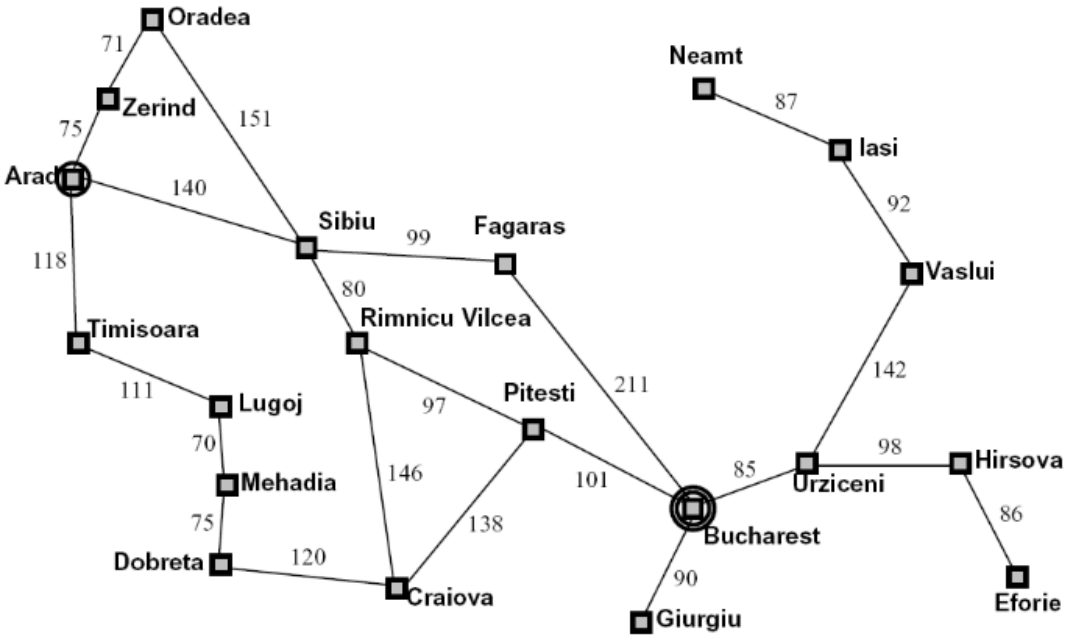
A start state and a goal test

A solution is a sequence of actions (a plan) which transforms the start state to a goal state

Search Problems Are Models

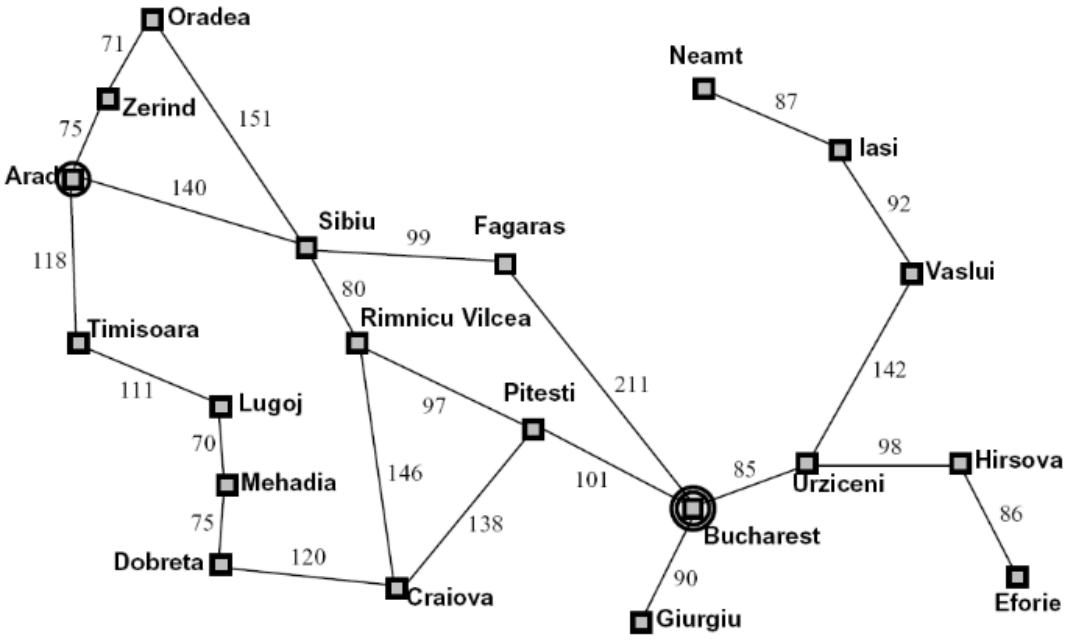


Example: Traveling in Romania



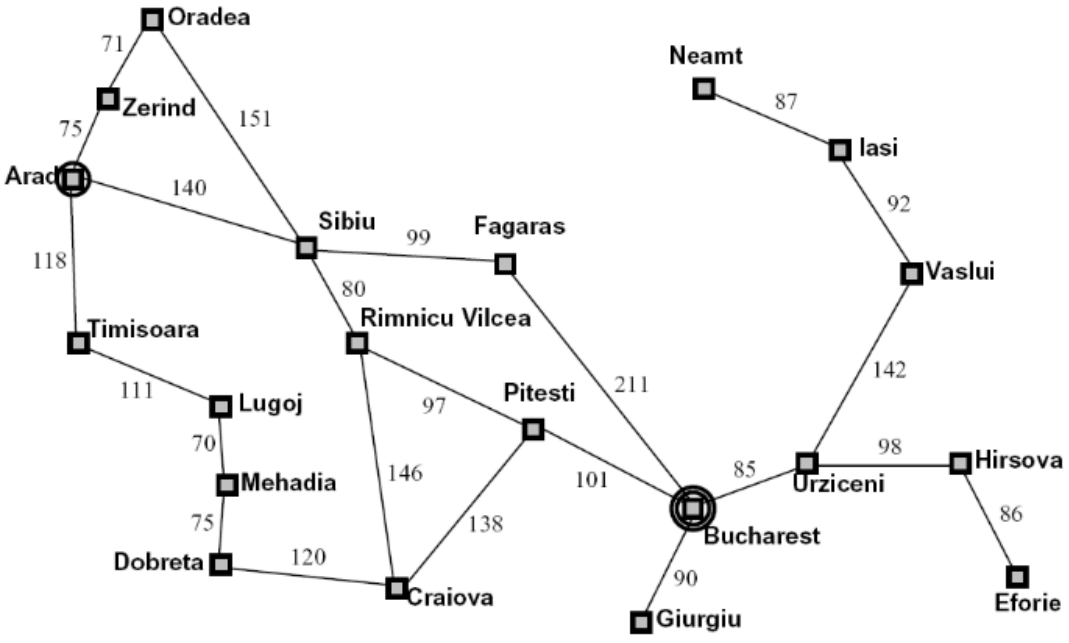
State space:

Example: Traveling in Romania



State space:
Cities

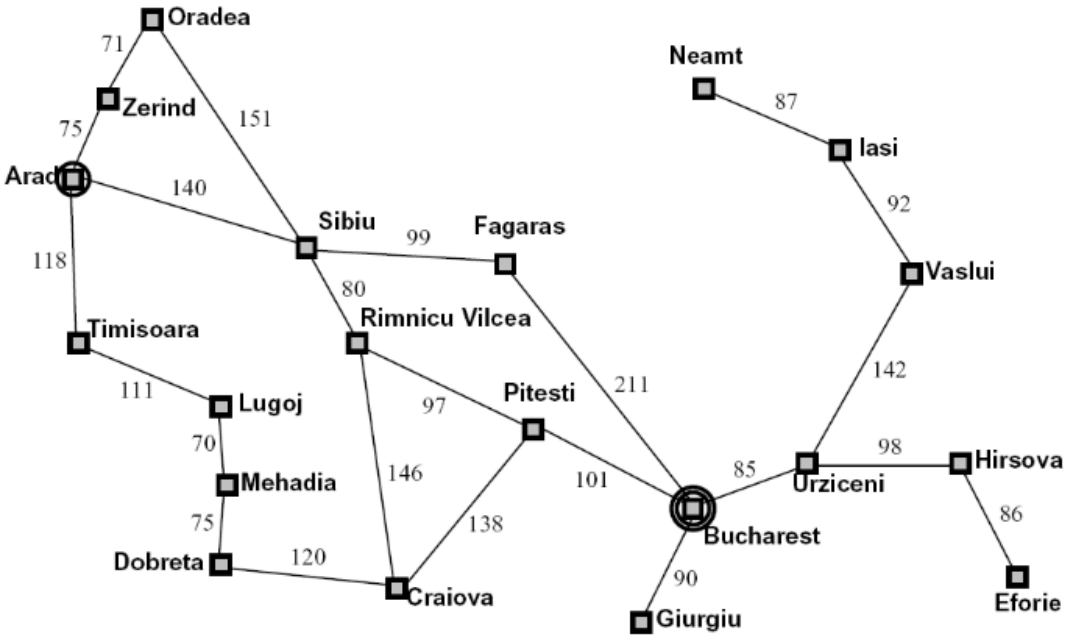
Example: Traveling in Romania



State space:
Cities

Successor function:

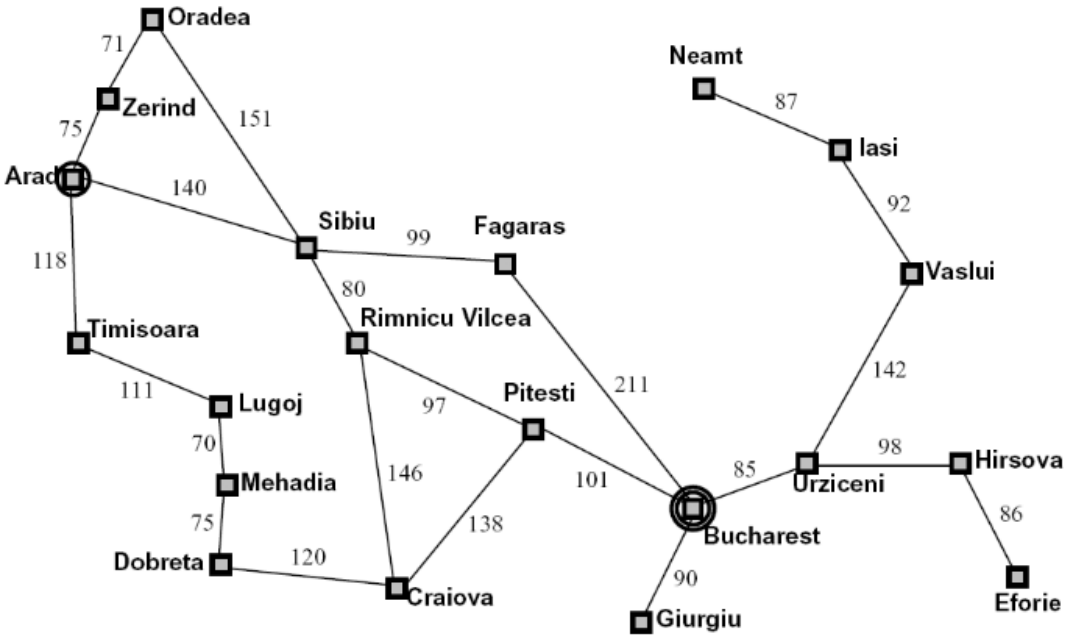
Example: Traveling in Romania



State space:
Cities

Successor function:
Roads: Neighboring city with cost = distance

Example: Traveling in Romania

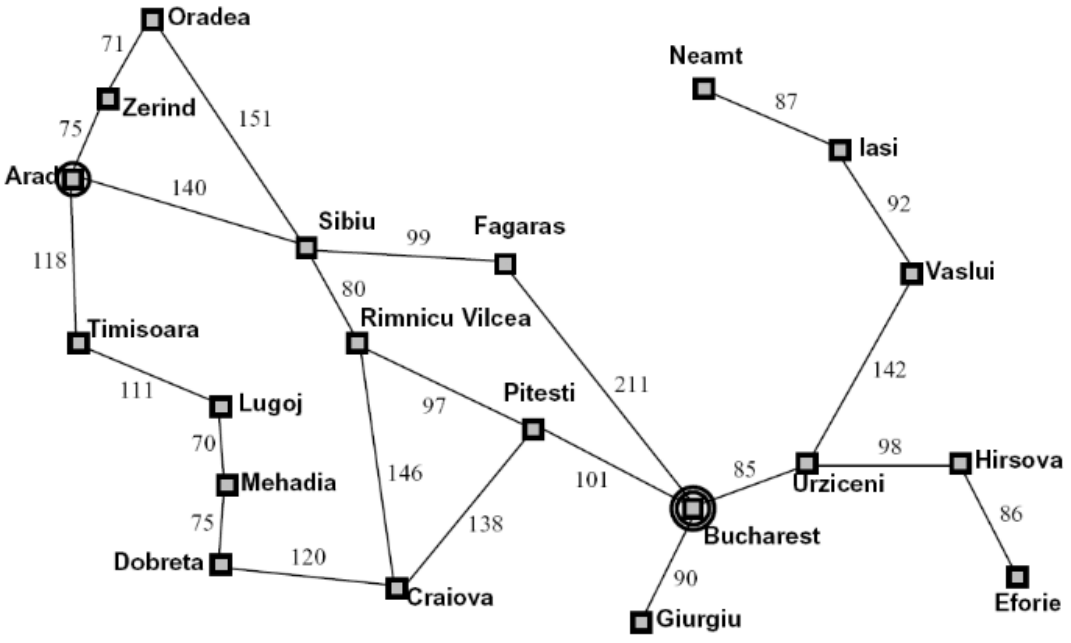


State space:
Cities

Successor function:
Roads: Neighboring city with cost = distance

Start state:

Example: Traveling in Romania

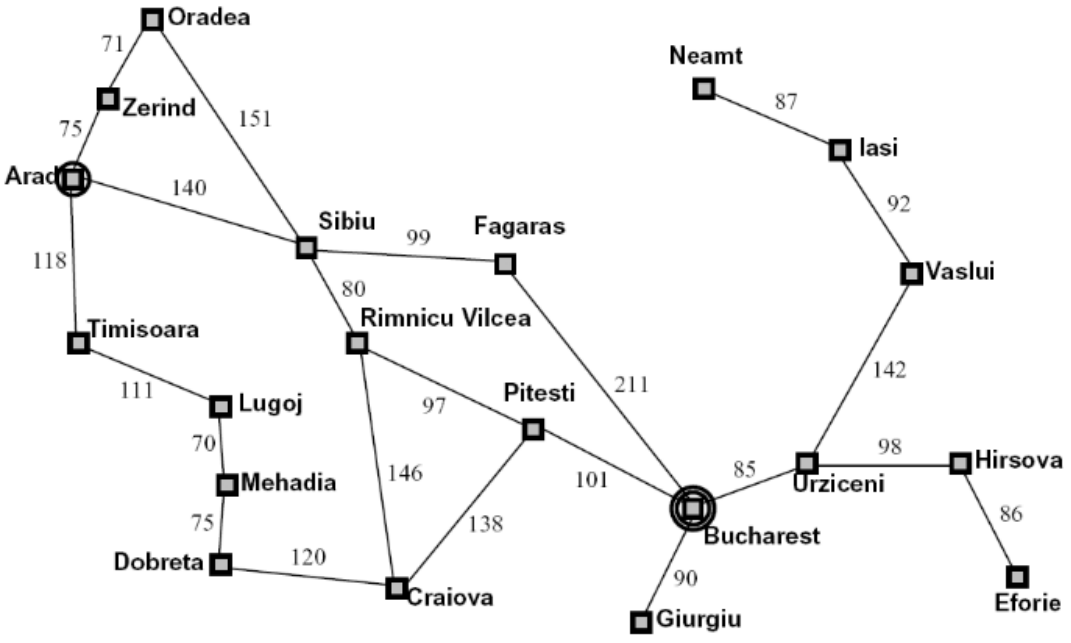


State space:
Cities

Successor function:
Roads: Neighboring city with cost = distance

Start state:
Arad

Example: Traveling in Romania



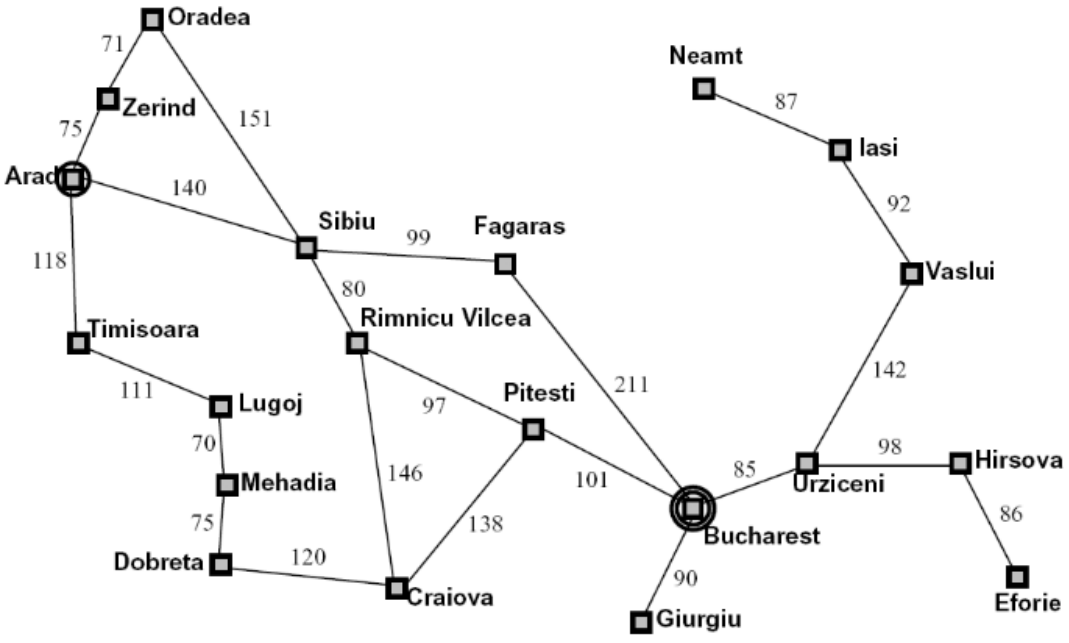
State space:
Cities

Successor function:
Roads: Neighboring city with cost = distance

Start state:
Arad

Goal test:

Example: Traveling in Romania



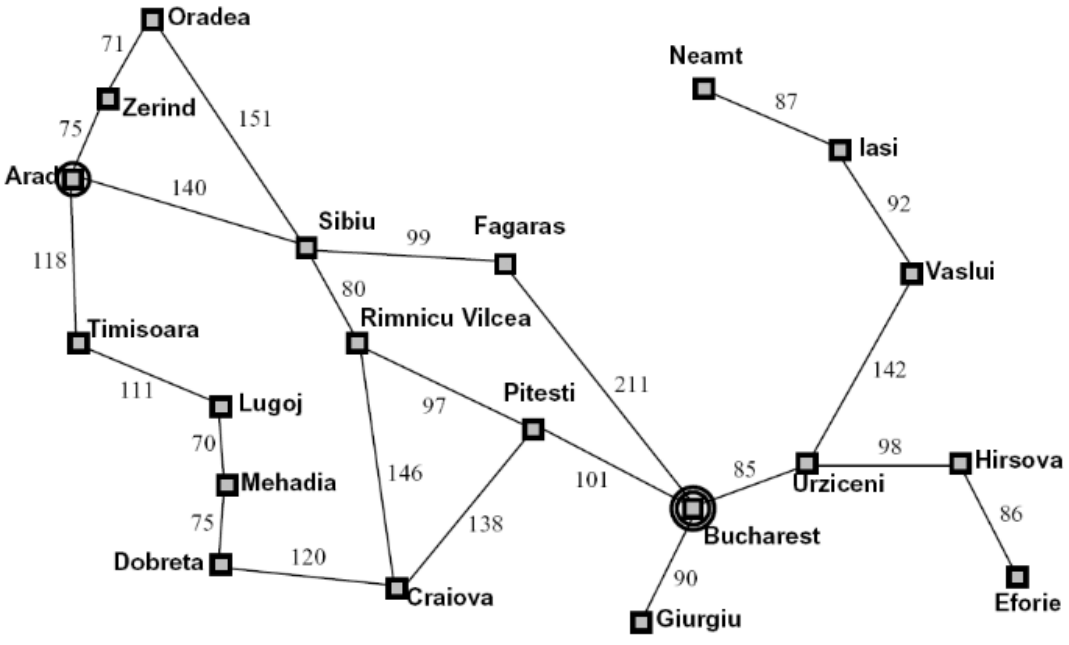
State space:
Cities

Successor function:
Roads: Neighboring city with cost = distance

Start state:
Arad

Goal test:
Is state == Bucharest?

Example: Traveling in Romania



State space:
Cities

Successor function:
Roads: Neighboring city with cost = distance

Start state:
Arad

Goal test:
Is state == Bucharest?

Solution?

What's in a State Space?

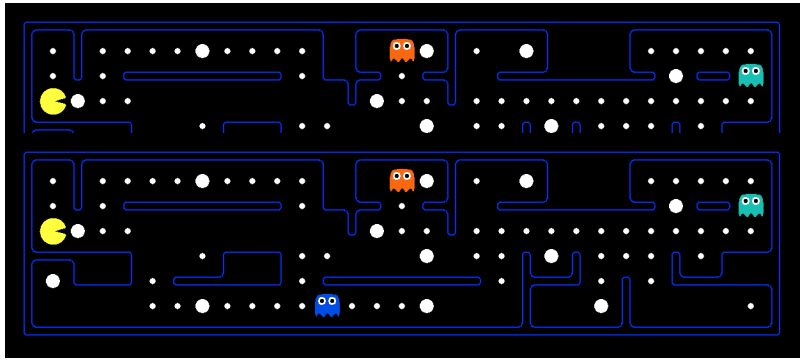
The world state includes every last detail of the environment.

What's in a State Space?

The world state includes every last detail of the environment.

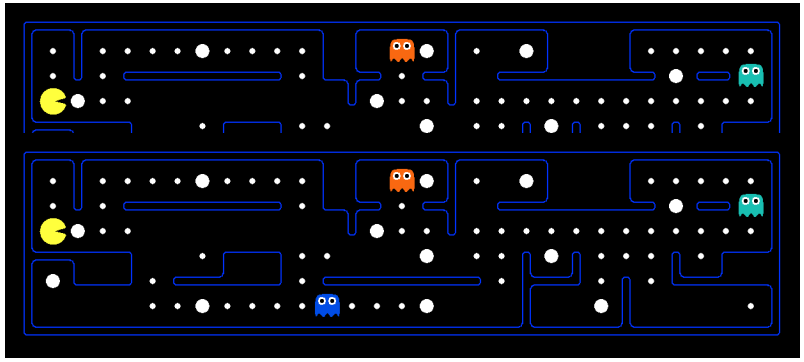
What's in a State Space?

The world state includes every last detail of the environment.



What's in a State Space?

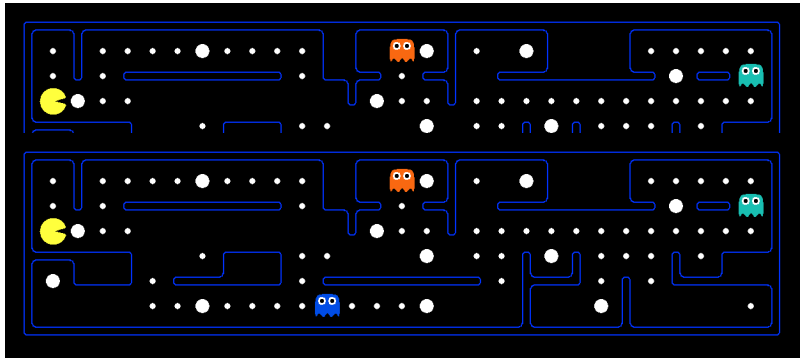
The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

What's in a State Space?

The world state includes every last detail of the environment.

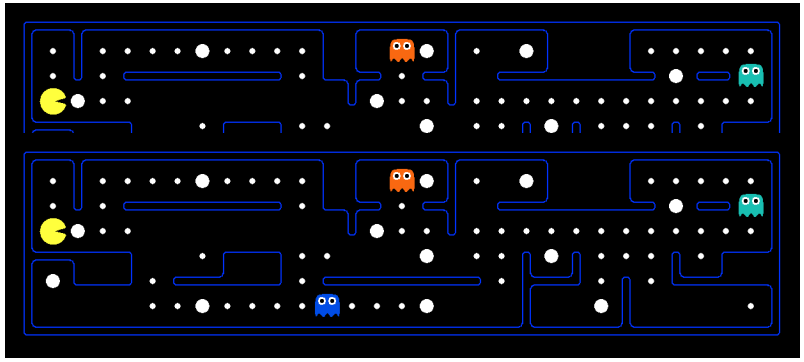


Search state keeps only details needed for planning (abstraction).

Problem: Pathing

What's in a State Space?

The world state includes every last detail of the environment.



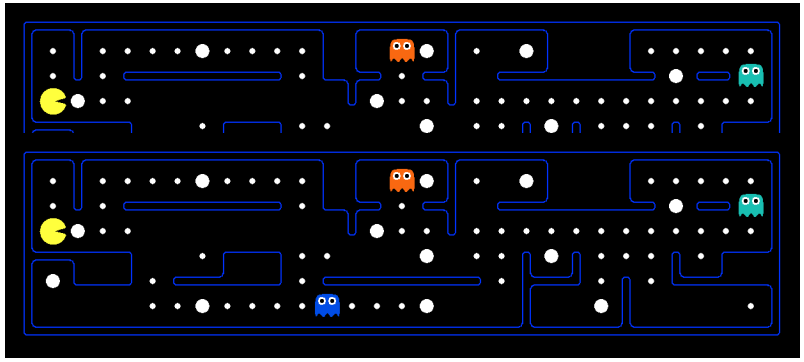
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States:

What's in a State Space?

The world state includes every last detail of the environment.



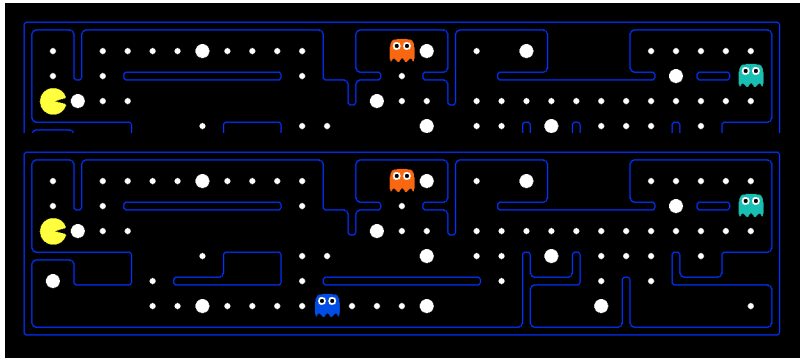
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location

What's in a State Space?

The world state includes every last detail of the environment.



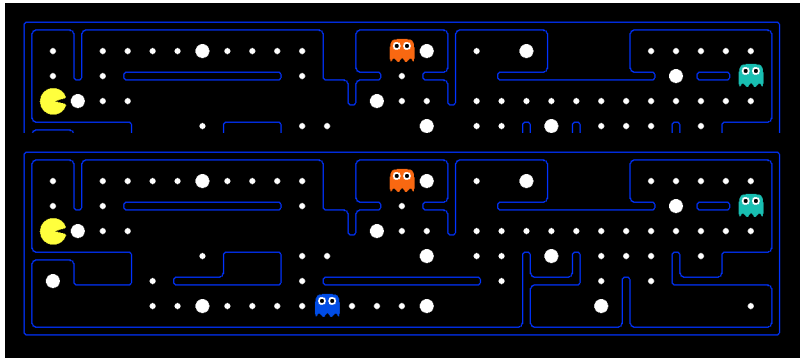
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location
Actions:

What's in a State Space?

The world state includes every last detail of the environment.



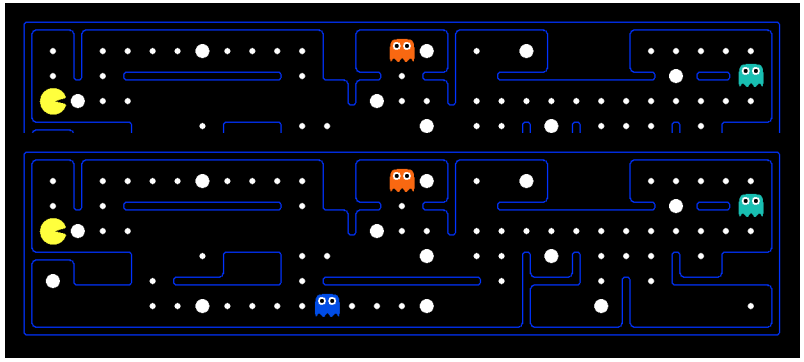
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location
Actions: NSEW

What's in a State Space?

The world state includes every last detail of the environment.



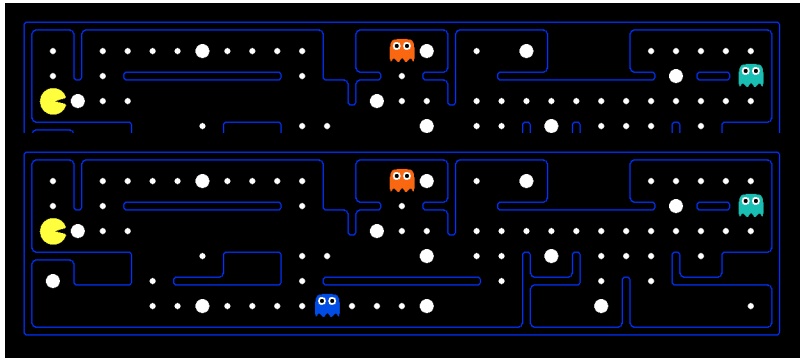
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location
Actions: NSEW
Successor:

What's in a State Space?

The world state includes every last detail of the environment.



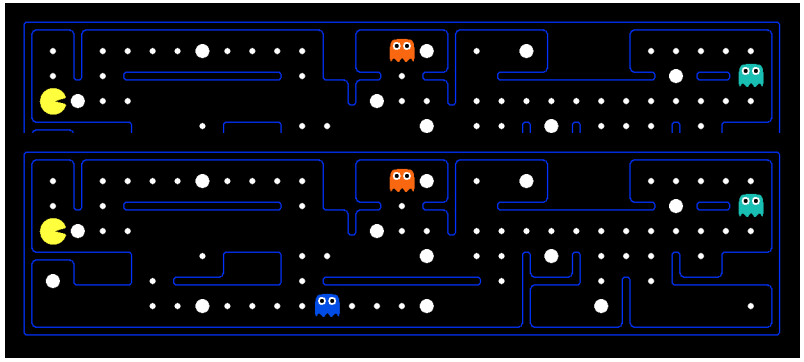
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location
Actions: NSEW
Successor: update location only

What's in a State Space?

The world state includes every last detail of the environment.



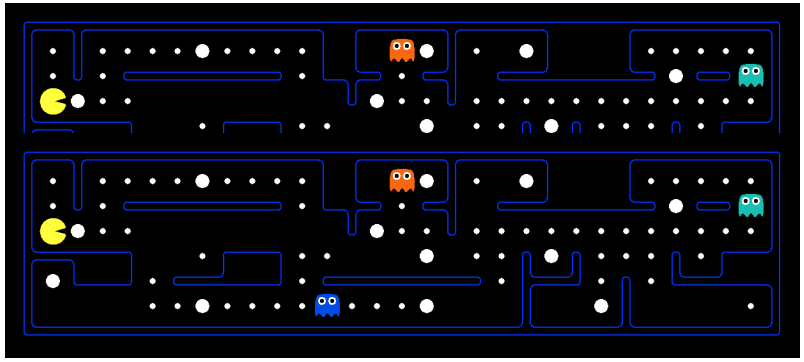
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test:

What's in a State Space?

The world state includes every last detail of the environment.



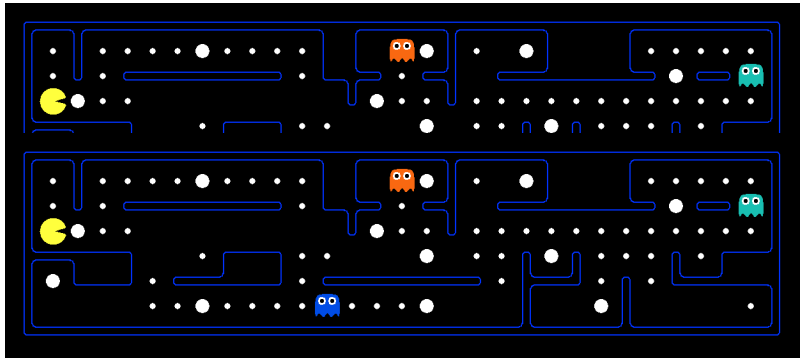
Search state keeps only details needed for planning (abstraction).

Problem: Pathing

States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

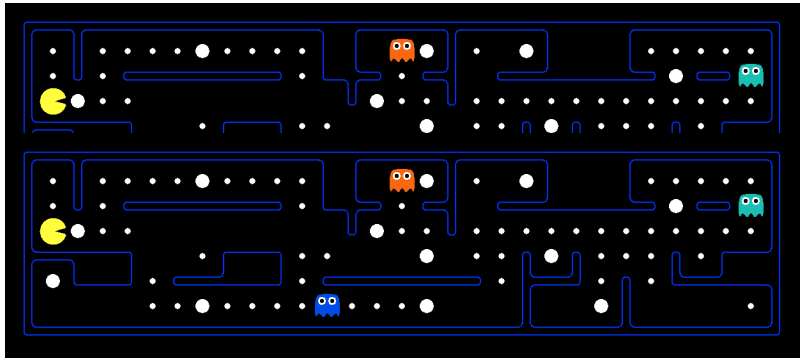
Problem: Pathing

States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

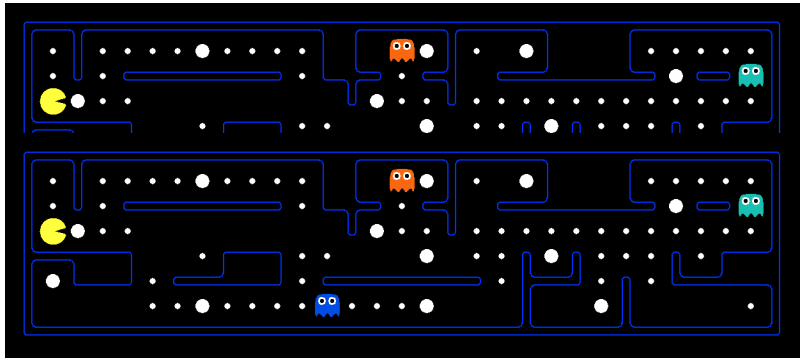
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States:

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

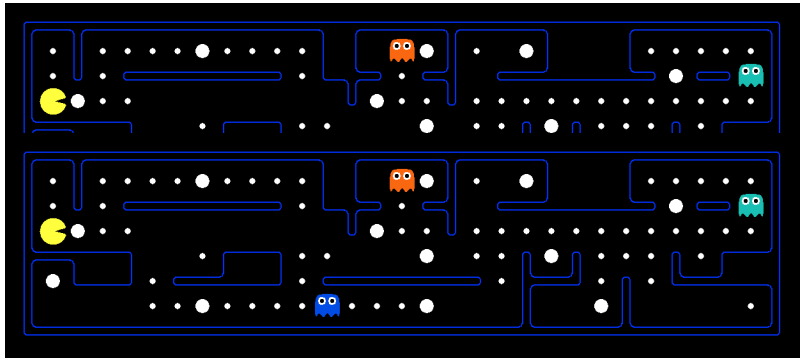
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

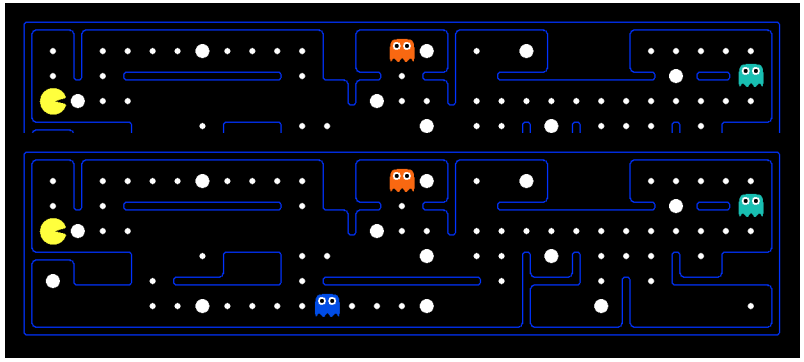
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$
Actions:

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

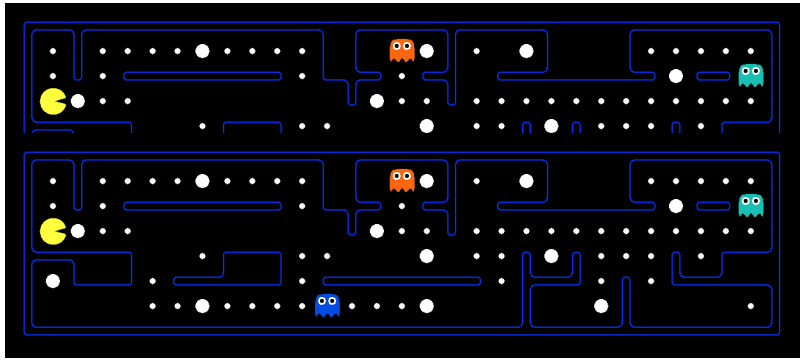
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$
Actions: NSEW

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

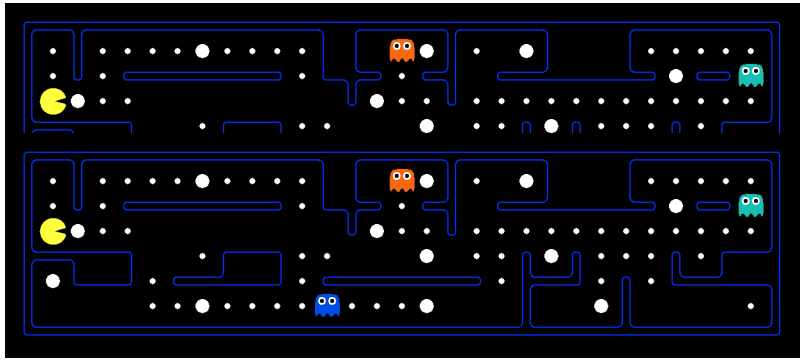
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$
Actions: NSEW
Successor:

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

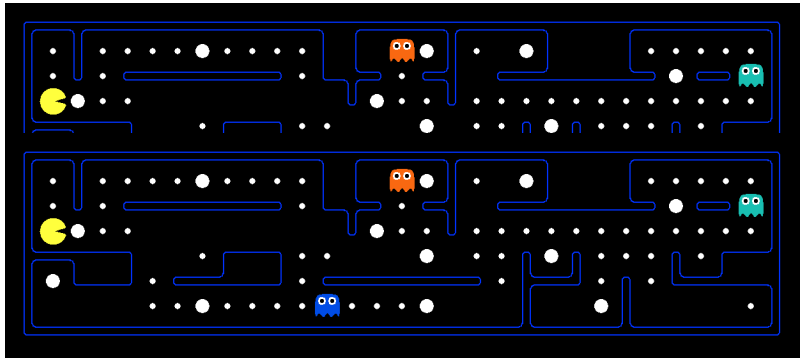
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$
Actions: NSEW
Successor: update location and possibly a dot boolean

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

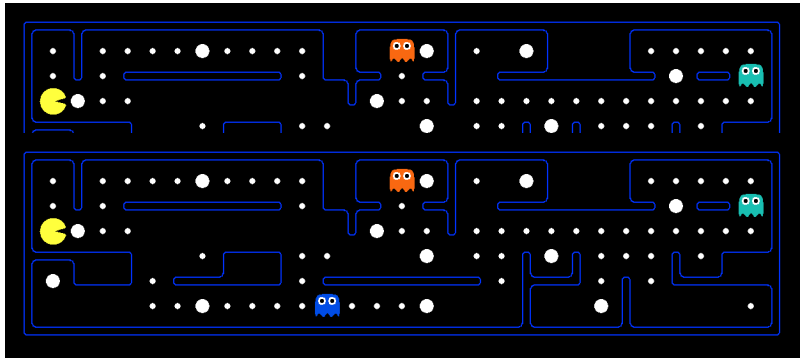
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$
Actions: NSEW
Successor: update location and possibly a dot boolean
Goal test:

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

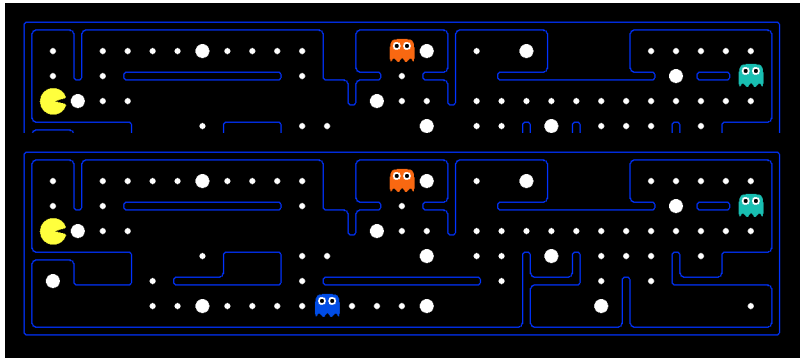
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

States: $\{(x, y)$, dot booleans$\}$
Actions: NSEW
Successor: update location and possibly a dot boolean
Goal test: dots all false

What's in a State Space?

The world state includes every last detail of the environment.



Search state keeps only details needed for planning (abstraction).

Problem: Pathing

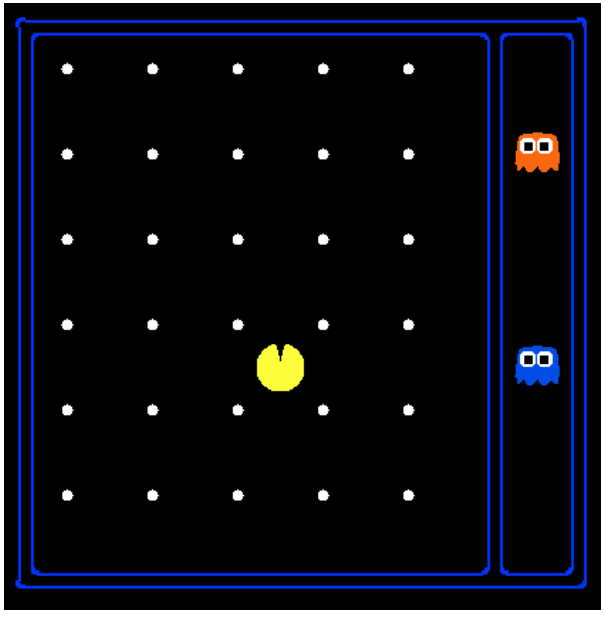
States: (x, y) location
Actions: NSEW
Successor: update location only
Goal test: is $(x, y) =$END?

Problem: Eat-All-Dots

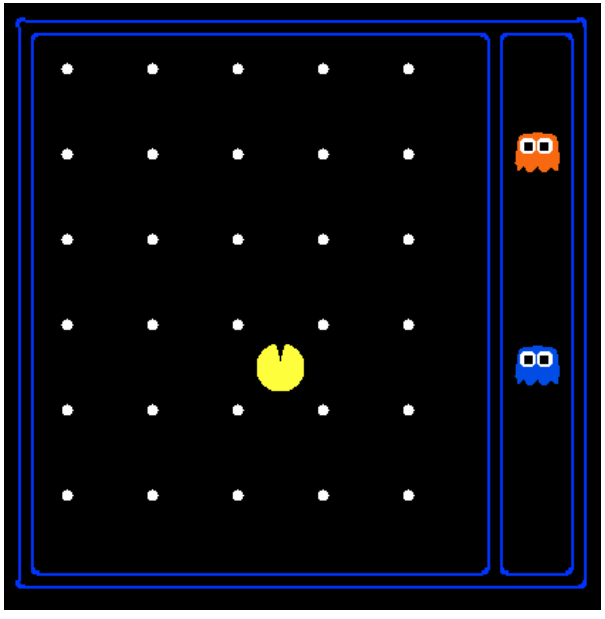
States: $\{(x, y)$, dot booleans$\}$
Actions: NSEW
Successor: update location and possibly a dot boolean
Goal test: dots all false

State Space Sizes?

World state:



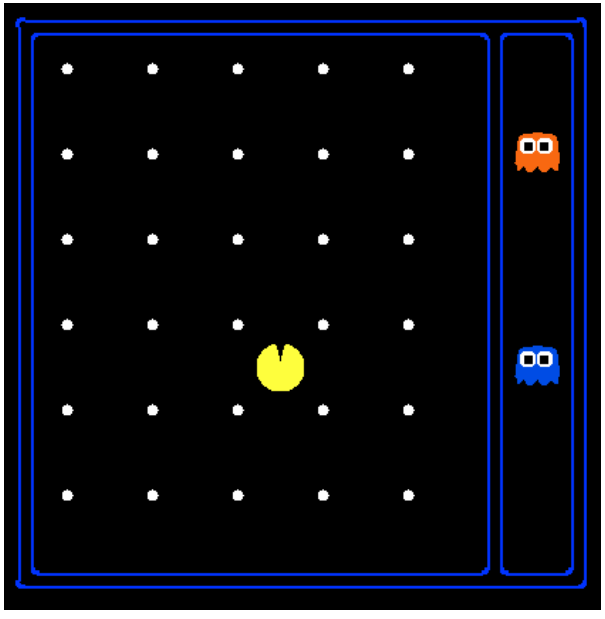
State Space Sizes?



World state:

Agent positions:

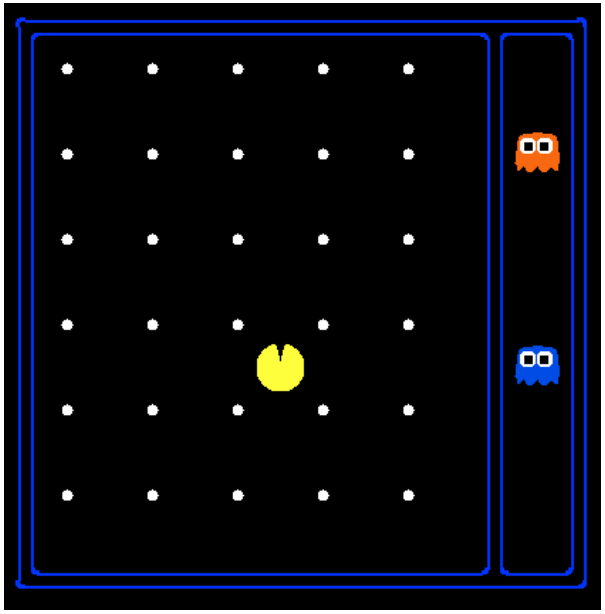
State Space Sizes?



World state:

Agent positions: 120

State Space Sizes?

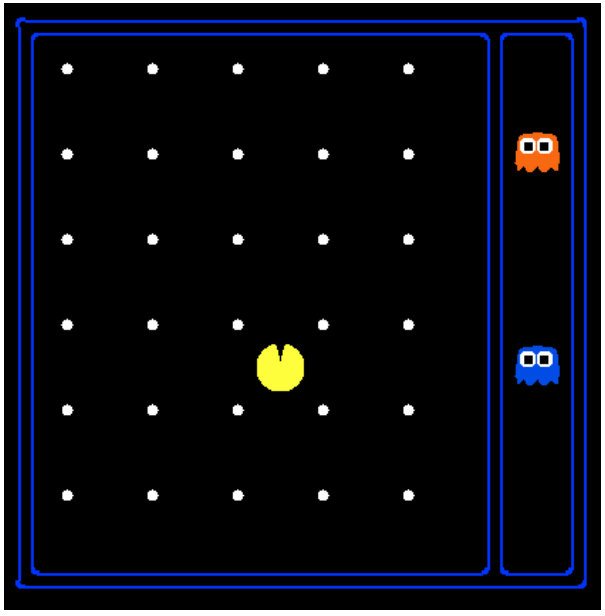


World state:

Agent positions: 120

Food count:

State Space Sizes?

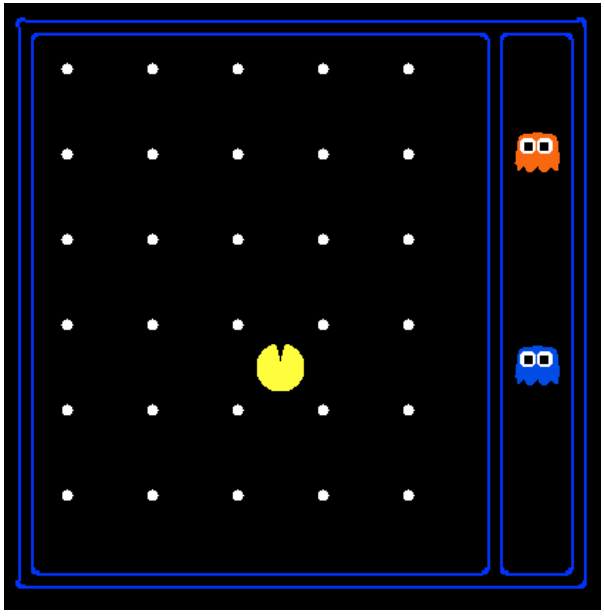


World state:

Agent positions: 120

Food count: 30

State Space Sizes?



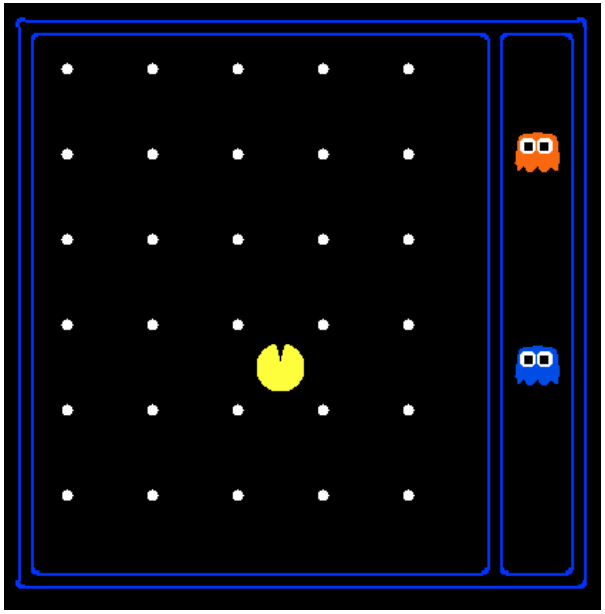
World state:

Agent positions: 120

Food count: 30

Ghost positions:

State Space Sizes?



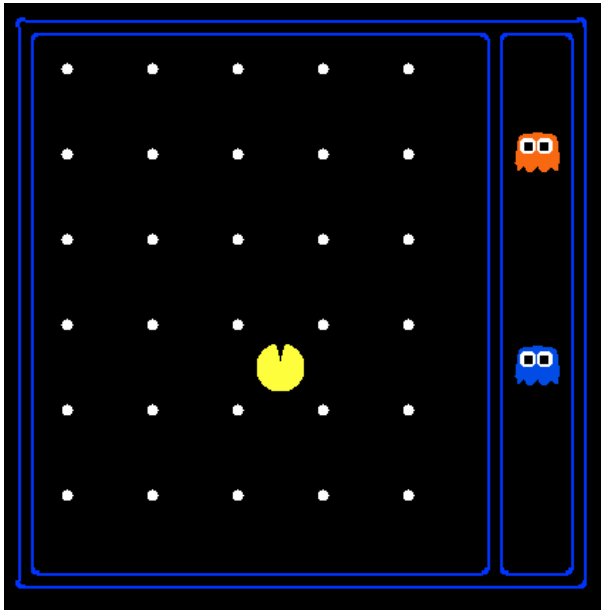
World state:

Agent positions: 120

Food count: 30

Ghost positions: 12

State Space Sizes?



World state:

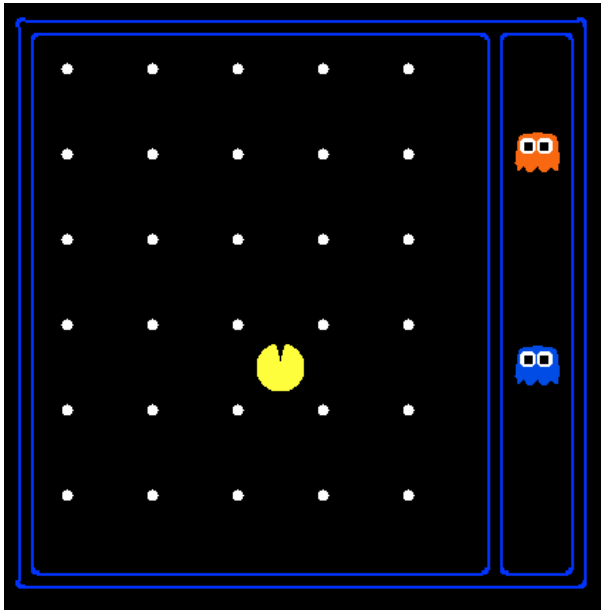
Agent positions: 120

Food count: 30

Ghost positions: 12

Agent facing:

State Space Sizes?



World state:

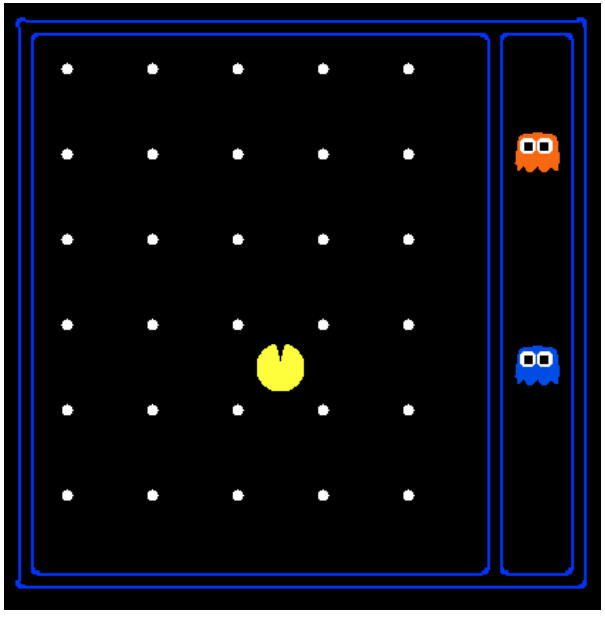
Agent positions: 120

Food count: 30

Ghost positions: 12

Agent facing: NSEW

State Space Sizes?



World state:

Agent positions: 120

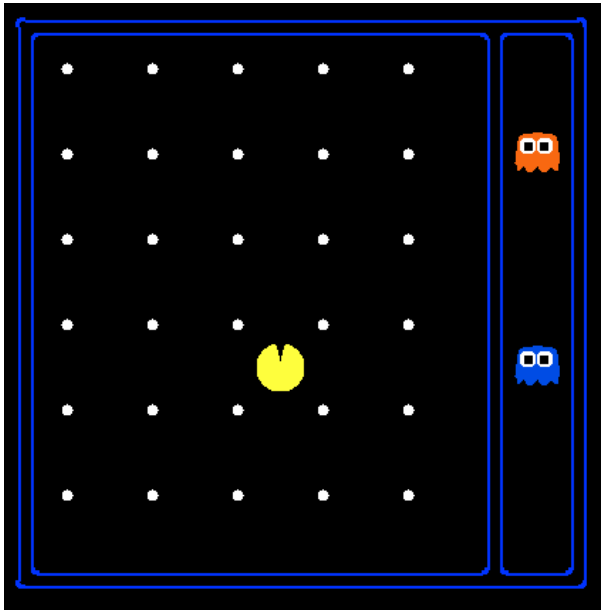
Food count: 30

Ghost positions: 12

Agent facing: NSEW

How many World states?

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

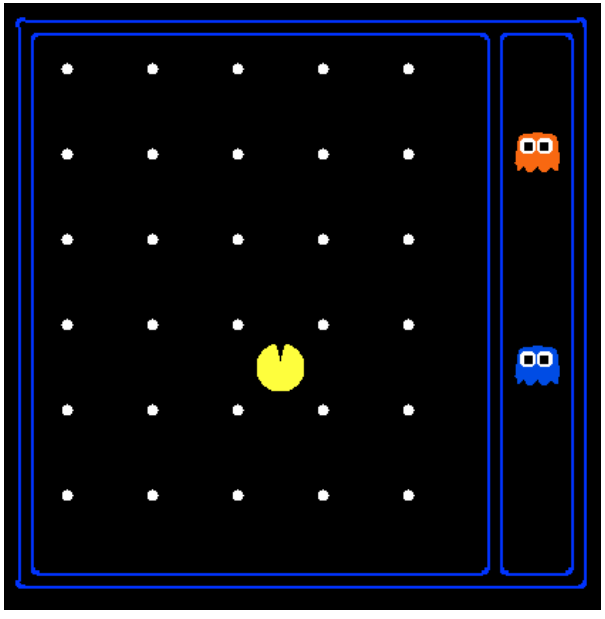
Ghost positions: 12

Agent facing: NSEW

How many World states?

120

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

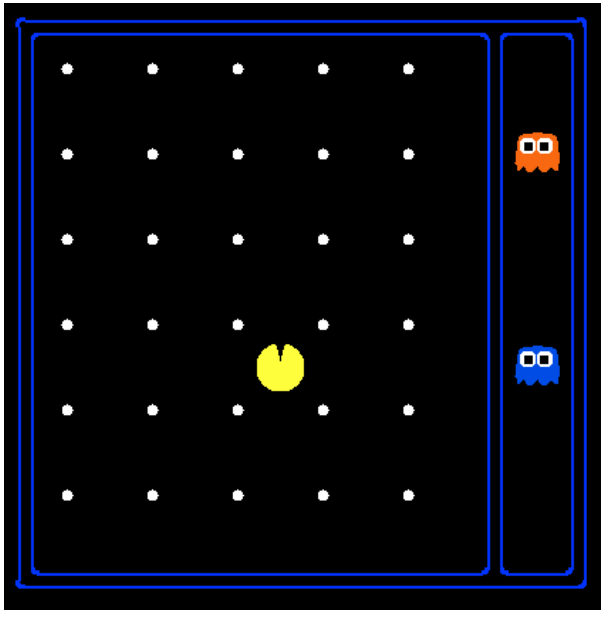
Ghost positions: 12

Agent facing: NSEW

How many World states?

$$120 \times (2^{30}) \times (12^2) \times 4$$

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

Ghost positions: 12

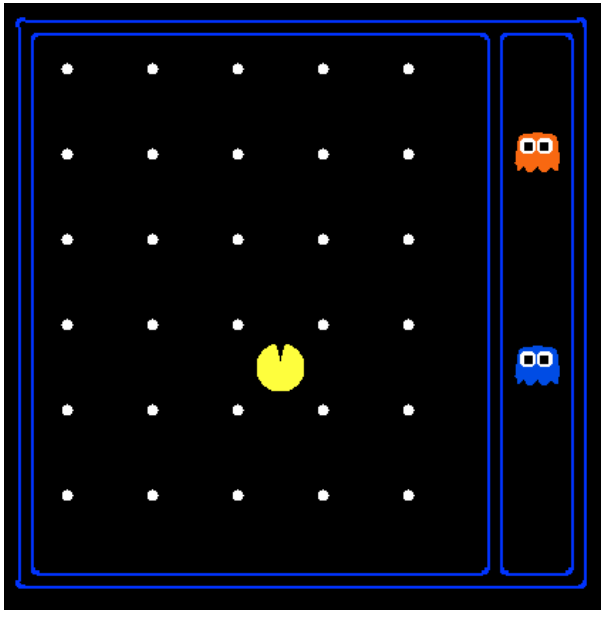
Agent facing: NSEW

How many World states?

$$120 \times (2^{30}) \times (12^2) \times 4$$

States for pathing?

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

Ghost positions: 12

Agent facing: NSEW

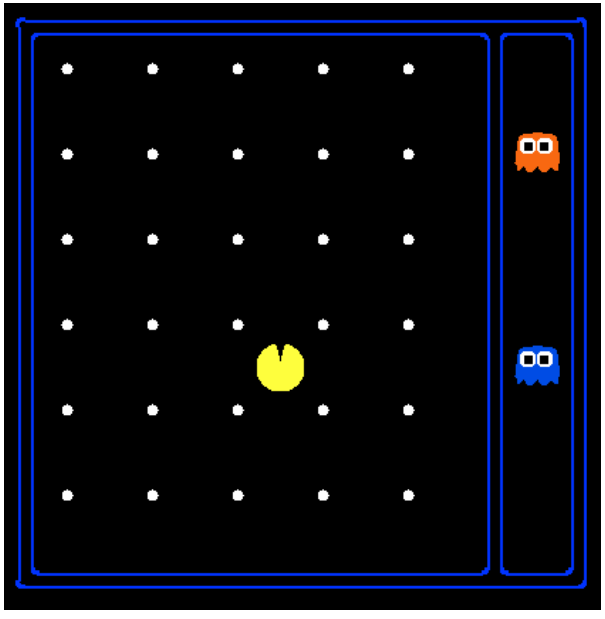
How many World states?

$120 \times (2^{30}) \times (12^2) \times 4$

States for pathing?

120

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

Ghost positions: 12

Agent facing: NSEW

How many World states?

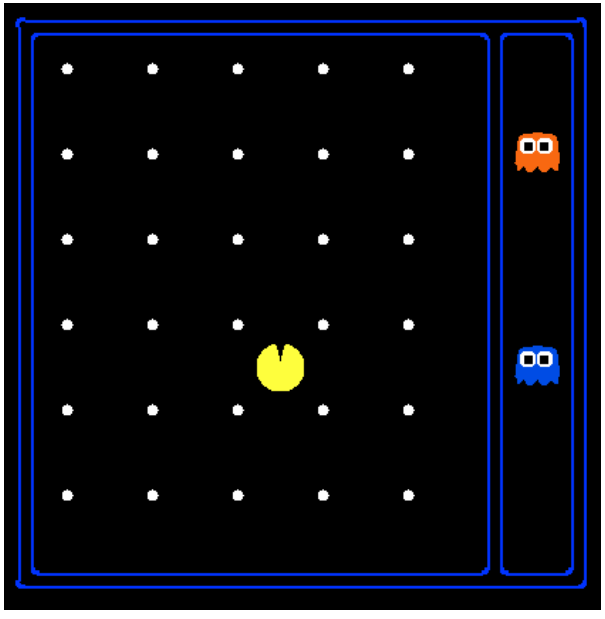
$120 \times (2^{30}) \times (12^2) \times 4$

States for pathing?

120

States for eat-all-dots?

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

Ghost positions: 12

Agent facing: NSEW

How many World states?

$120 \times (2^{30}) \times (12^2) \times 4$

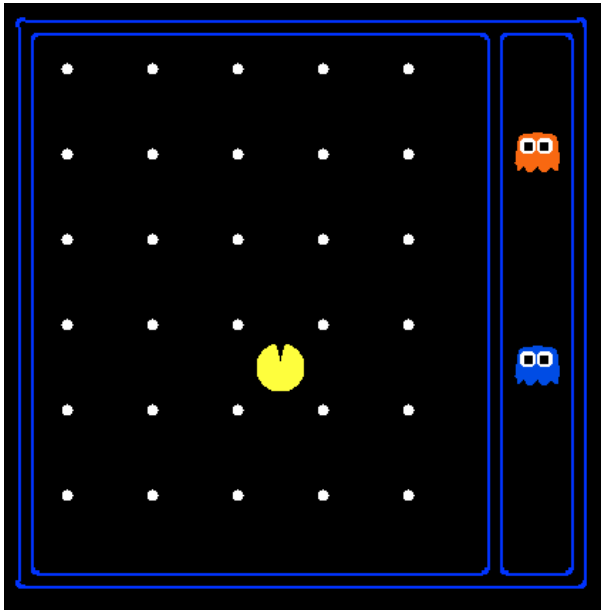
States for pathing?

120

States for eat-all-dots?

120

State Space Sizes?



World state:

Agent positions: 120

Food count: 30

Ghost positions: 12

Agent facing: NSEW

How many World states?

$120 \times (2^{30}) \times (12^2) \times 4$

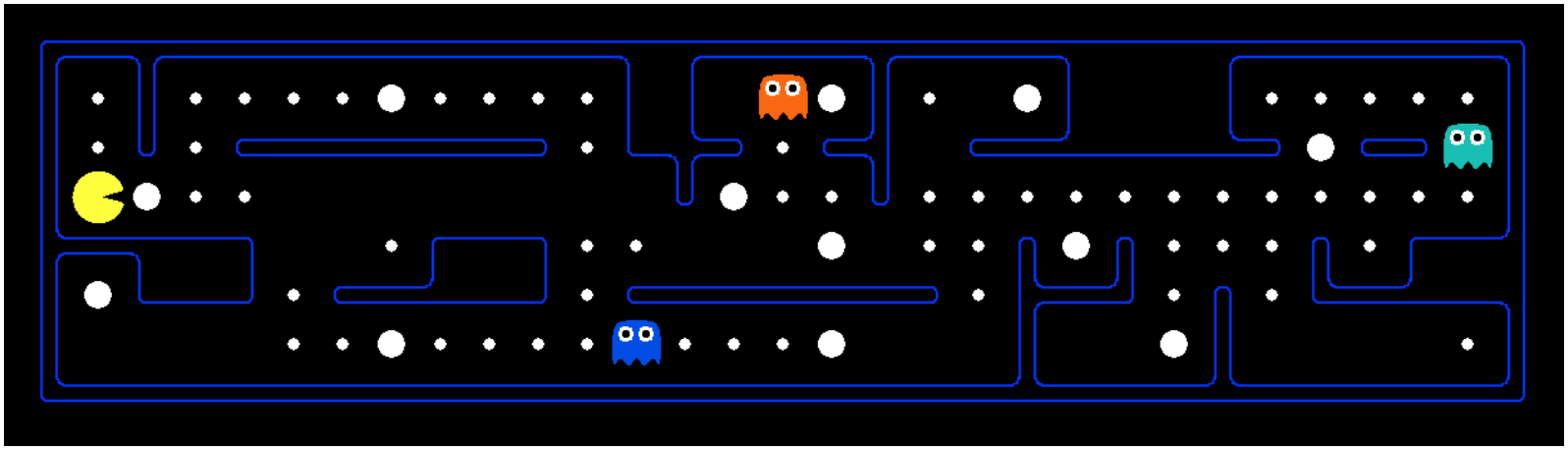
States for pathing?

120

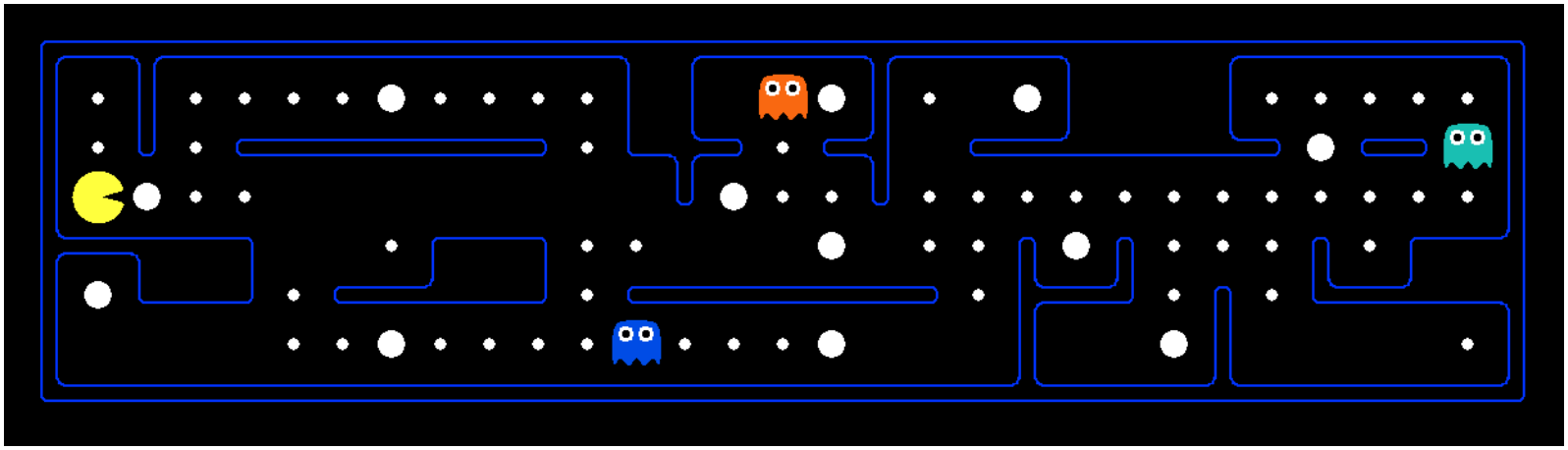
States for eat-all-dots?

$120 \times (2^{30})$

Problem: Safe Passage

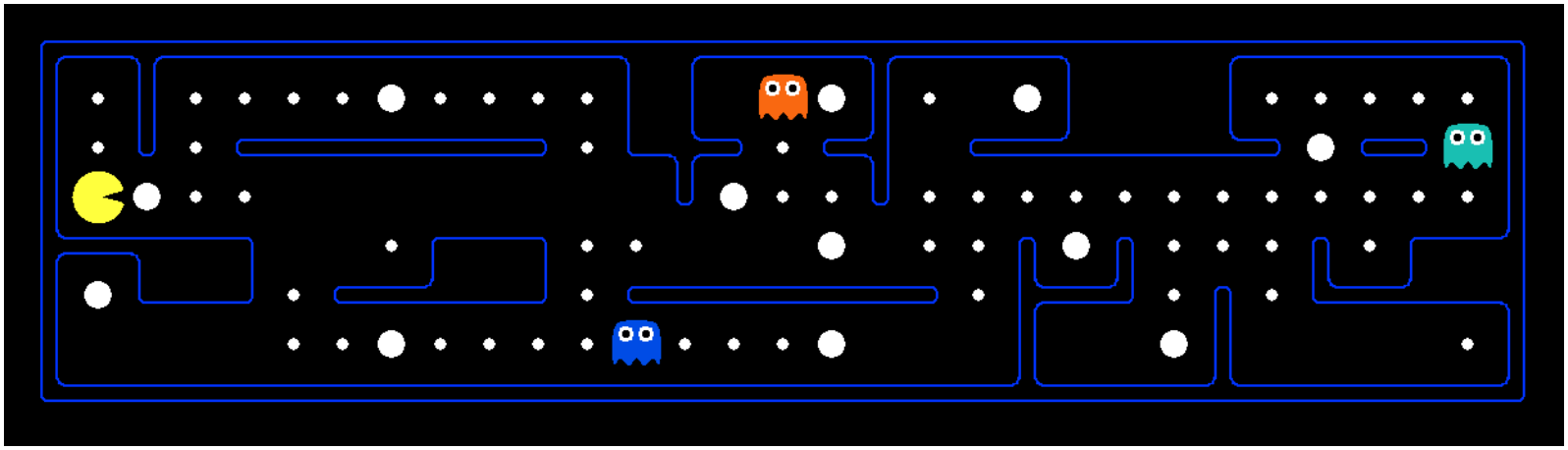


Problem: Safe Passage



Problem: eat all dots while keeping the ghosts perma-scared?

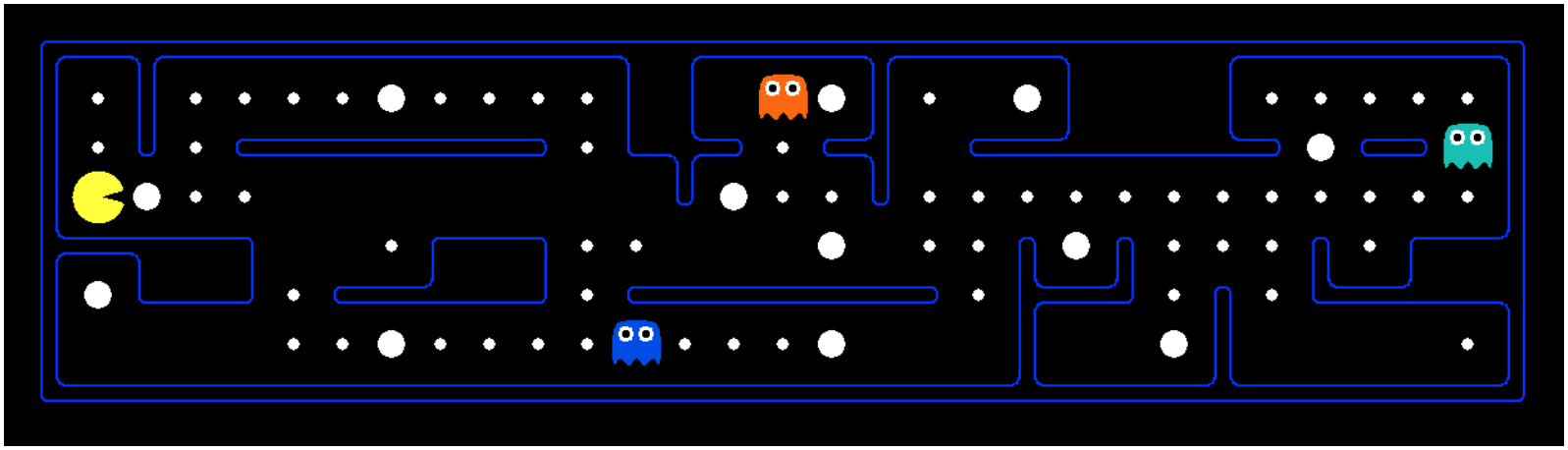
Problem: Safe Passage



Problem: eat all dots while keeping the ghosts perma-scared?

What does the state space have to specify?

Problem: Safe Passage

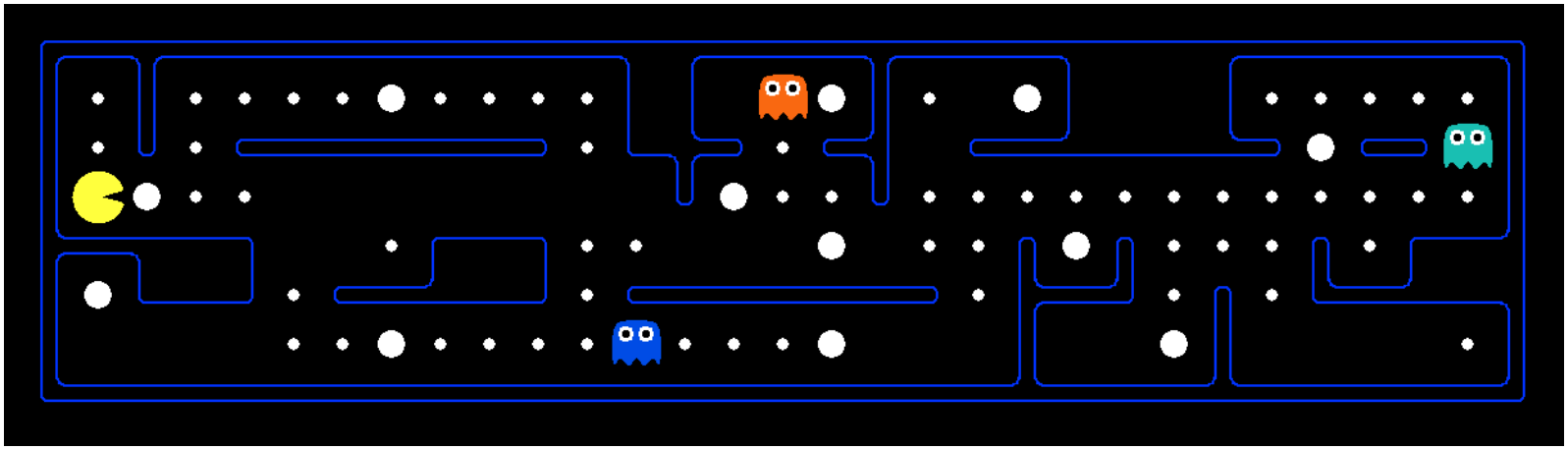


Problem: eat all dots while keeping the ghosts perma-scared?

What does the state space have to specify?

agent position

Problem: Safe Passage

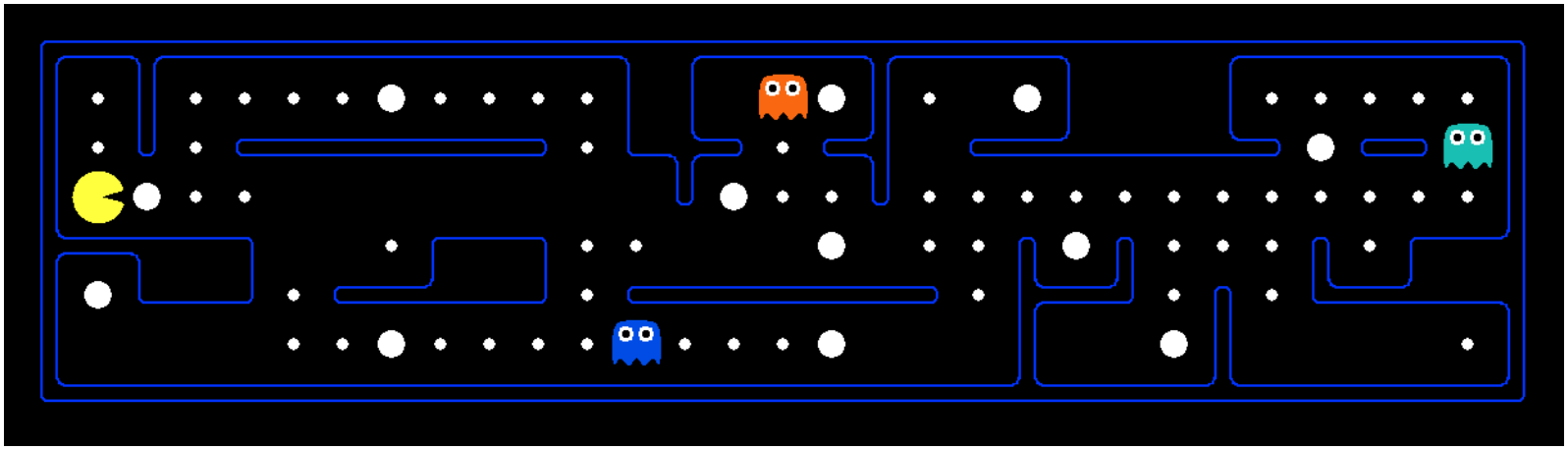


Problem: eat all dots while keeping the ghosts perma-scared?

What does the state space have to specify?

agent position
dot booleans

Problem: Safe Passage

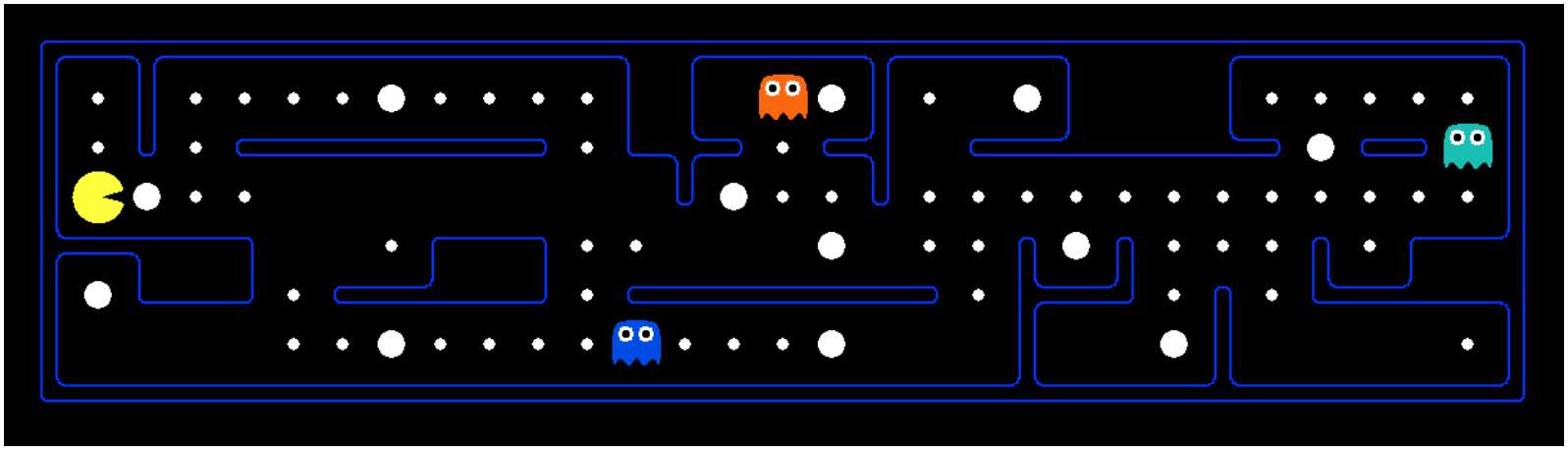


Problem: eat all dots while keeping the ghosts perma-scared?

What does the state space have to specify?

agent position
dot booleans
power pellet booleans

Problem: Safe Passage



Problem: eat all dots while keeping the ghosts perma-scared?

What does the state space have to specify?

agent position
dot booleans
power pellet booleans
remaining scared time

Agent Design

The environment largely determines the agent design.

Agent Design

The environment largely determines the agent design.

Fully/partialy observable $\rightarrow$ agent request memory (internal state)

Agent Design

The environment largely determines the agent design.

- Fully/partialy observable $\rightarrow$ agent request memory (internal state)
- Discrete/ continuous $\rightarrow$ agent can/can't enumerate all states

Agent Design

The environment largely determines the agent design.

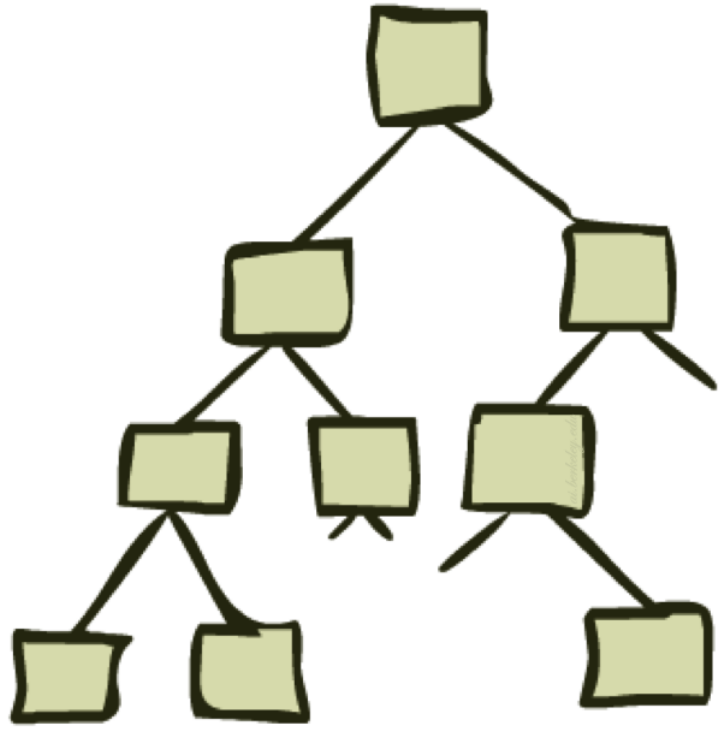
- Fully/partialy observable $\rightarrow$ agent request memory (internal state)
- Discrete/ continuous $\rightarrow$ agent can/can't enumerate all states
- Stochastic/deterministic $\rightarrow$ agent deals with contingencies

Agent Design

The environment largely determines the agent design.

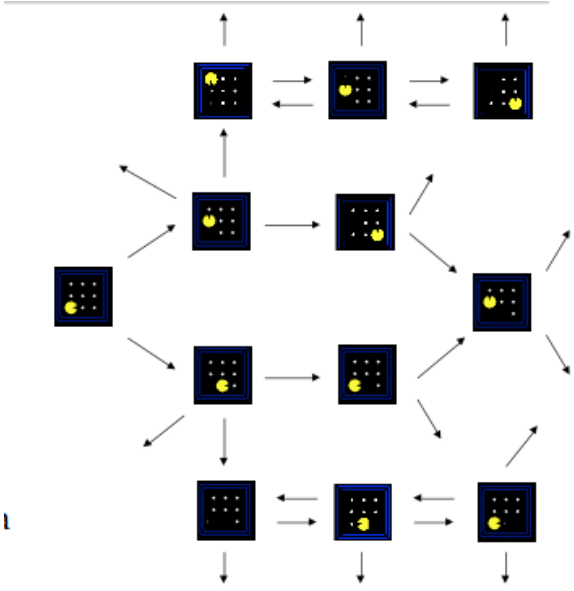
- Fully/partialy observable $\rightarrow$ agent request memory (internal state)
- Discrete/ continuous $\rightarrow$ agent can/can't enumerate all states
- Stochastic/deterministic $\rightarrow$ agent deals with contingencies
- Single-agent/multi-agent $\rightarrow$ agent may need to behave randomly.

State Space Graphs and Search Trees

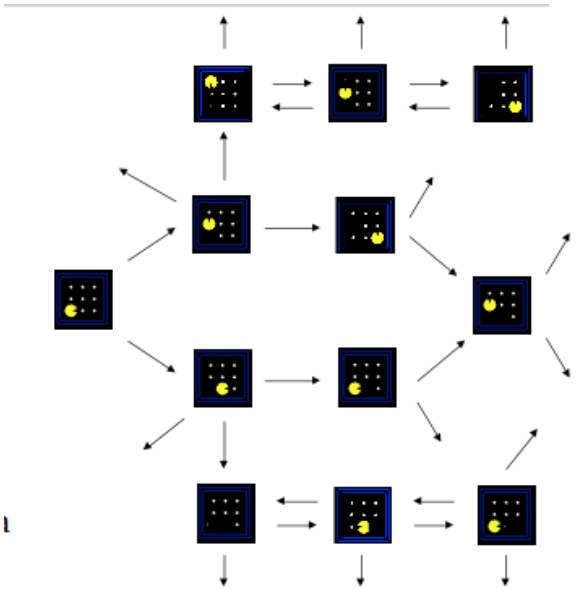


State Space Graphs

State space graph: A mathematical representation of a search problem



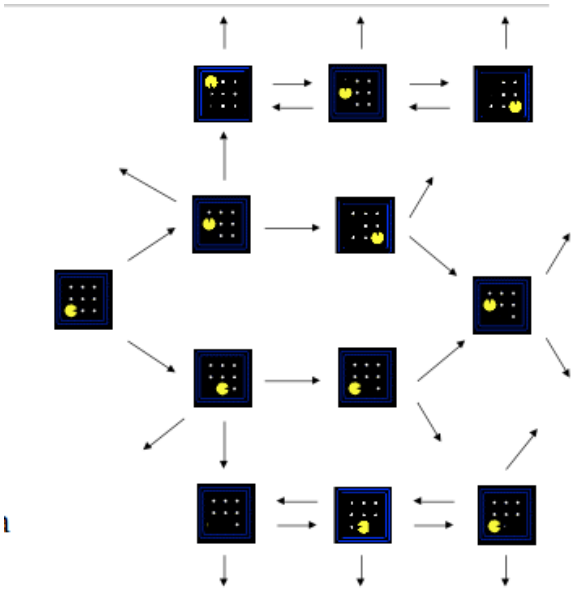
State Space Graphs



State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.

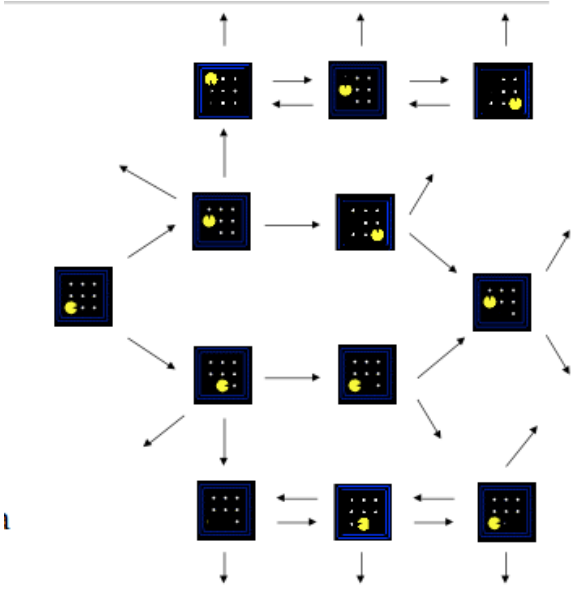
State Space Graphs



State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.
- Arcs represent successors (action results).

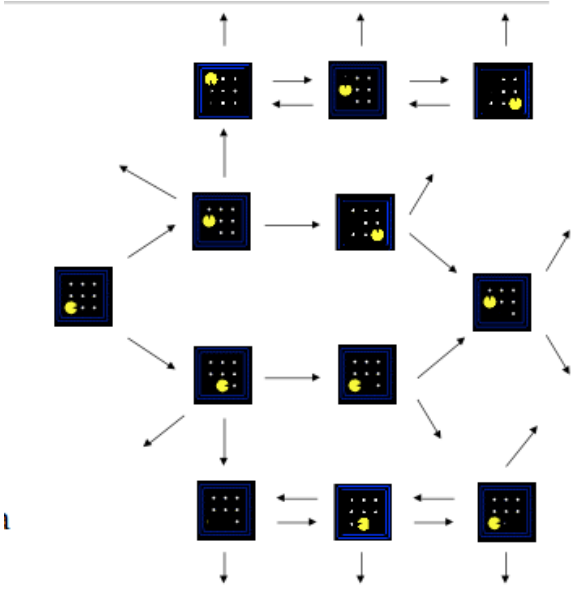
State Space Graphs



State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.
- Arcs represent successors (action results).
- The goal test is a set of goal nodes (maybe only one).

State Space Graphs

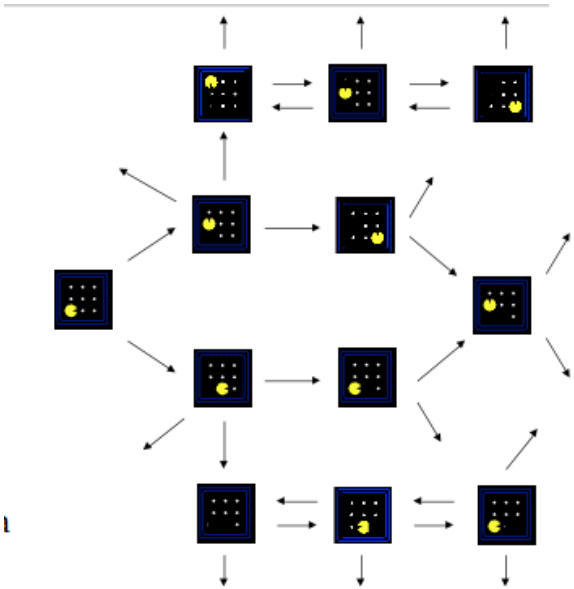


State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.
- Arcs represent successors (action results).
- The goal test is a set of goal nodes (maybe only one).

In a state space graph, each state occurs only once!

State Space Graphs



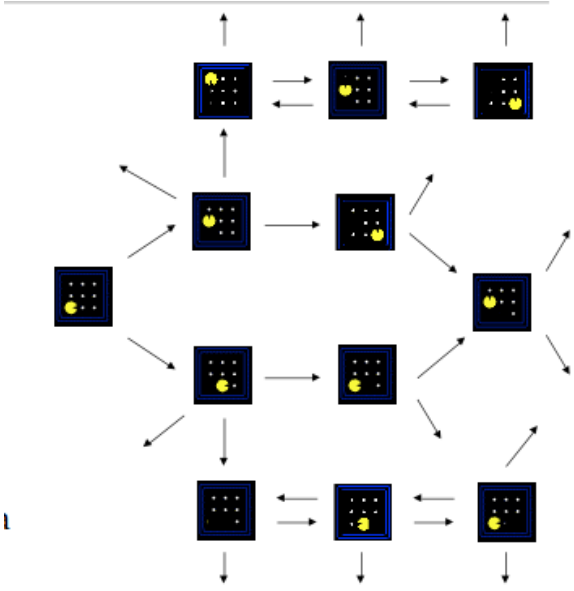
State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.
- Arcs represent successors (action results).
- The goal test is a set of goal nodes (maybe only one).

In a state space graph, each state occurs only once!

We can rarely build this full graph in memory (it's too big),

State Space Graphs



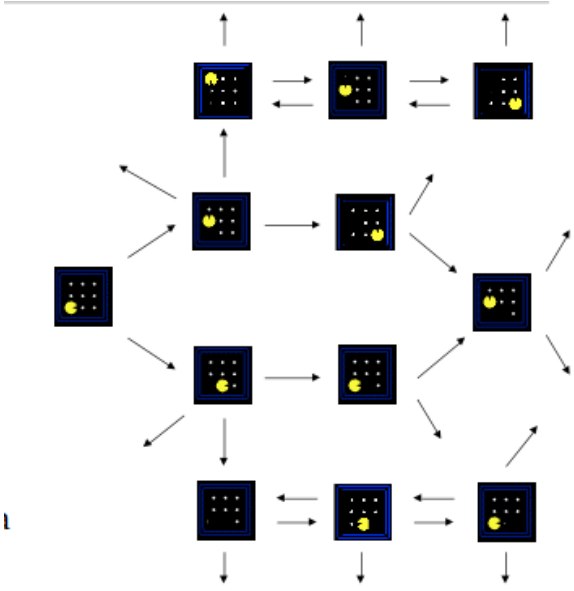
State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.
- Arcs represent successors (action results).
- The goal test is a set of goal nodes (maybe only one).

In a state space graph, each state occurs only once!

We can rarely build this full graph in memory (it's too big), but it's a useful idea.

State Space Graphs



State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations.
- Arcs represent successors (action results).
- The goal test is a set of goal nodes (maybe only one).

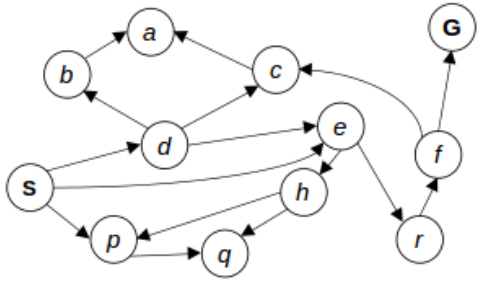
In a state space graph, each state occurs only once!

We can rarely build this full graph in memory (it's too big), but it's a useful idea.

E.g., replanning.

State Space Graphs

State space graph: A mathematical representation of a search problem

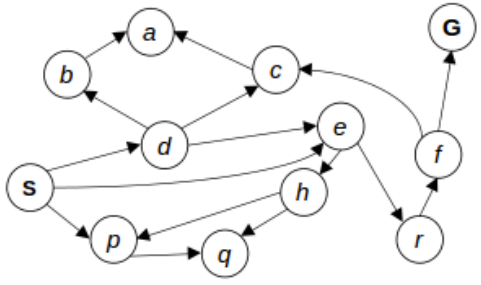


Tiny search graph for a tiny search problem

State Space Graphs

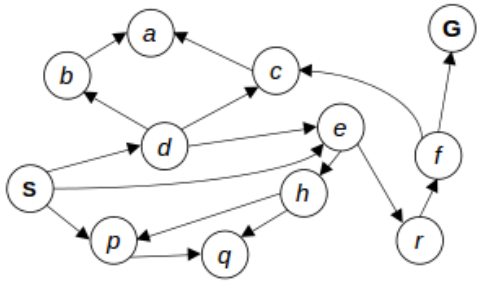
State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations



Tiny search graph for a tiny search problem

State Space Graphs

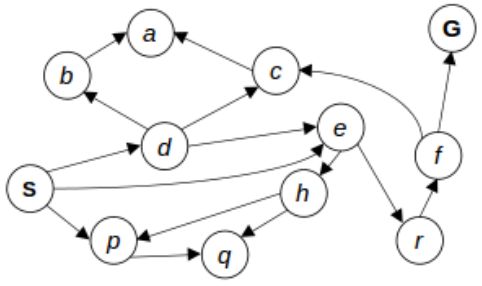


Tiny search graph for a tiny search problem

State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations
- Arcs represent successors (action results)

State Space Graphs

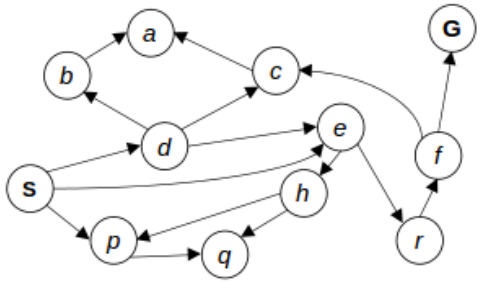


Tiny search graph for a tiny search problem

State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations
- Arcs represent successors (action results)
- The goal test is a set of goal nodes (maybe only one)

State Space Graphs



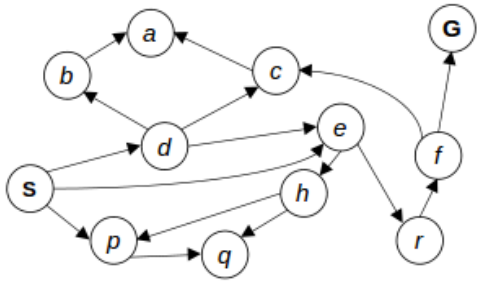
Tiny search graph for a tiny search problem

State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations
- Arcs represent successors (action results)
- The goal test is a set of goal nodes (maybe only one)

In a state space graph, each state occurs only once!

State Space Graphs



Tiny search graph for a tiny search problem

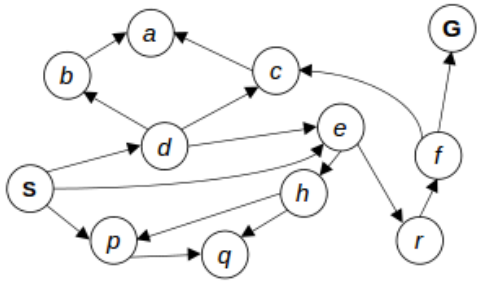
State space graph: A mathematical representation of a search problem

- Nodes are (abstracted) world configurations
- Arcs represent successors (action results)
- The goal test is a set of goal nodes (maybe only one)

In a state space graph, each state occurs only once!

We can rarely build this full graph in memory (it's too big),

State Space Graphs



Tiny search graph for a tiny search problem

State space graph: A mathematical representation of a search problem

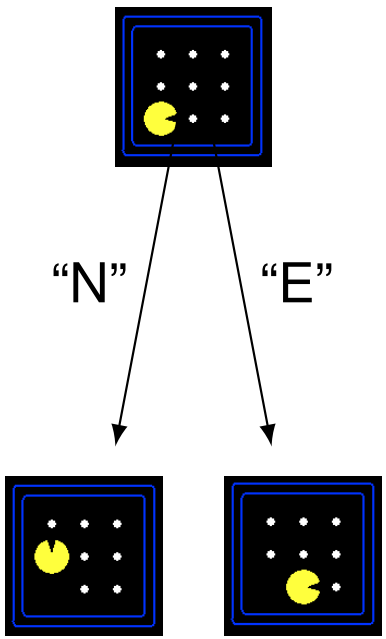
- Nodes are (abstracted) world configurations
- Arcs represent successors (action results)
- The goal test is a set of goal nodes (maybe only one)

In a state space graph, each state occurs only once!

We can rarely build this full graph in memory (it's too big), but it's a useful idea.

Search Trees

This is now / start

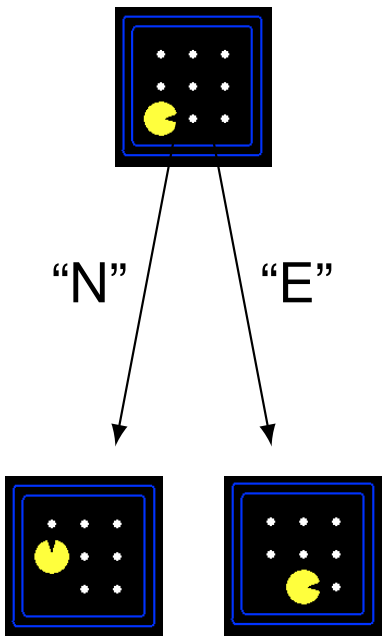


Possible futures.

A search tree:

Search Trees

This is now / start



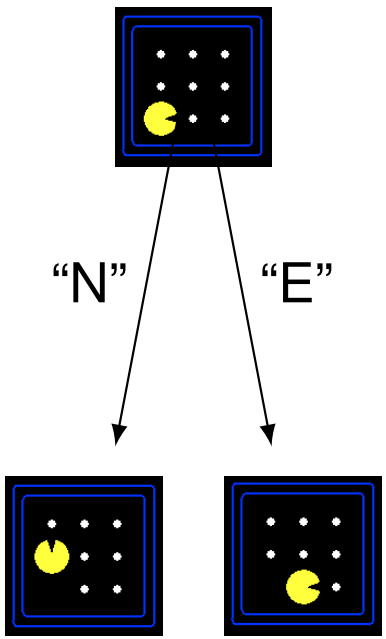
Possible futures.

A search tree:

A “what if” tree of plans and their outcomes

Search Trees

This is now / start



Possible futures.

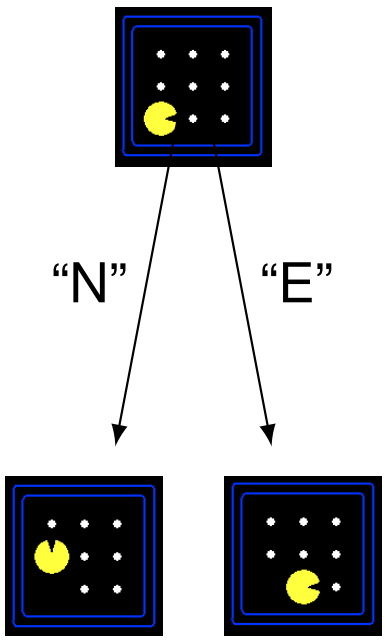
A search tree:

A “what if” tree of plans and their outcomes

The start state is the root node

Search Trees

This is now / start



Possible futures.

A search tree:

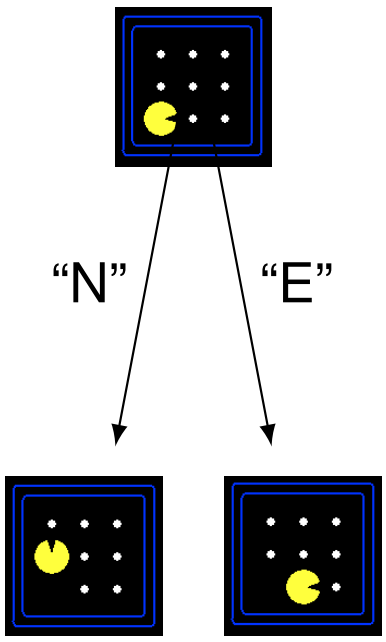
A “what if” tree of plans and their outcomes

The start state is the root node

Children correspond to successors

Search Trees

This is now / start



Possible futures.

A search tree:

A “what if” tree of plans and their outcomes

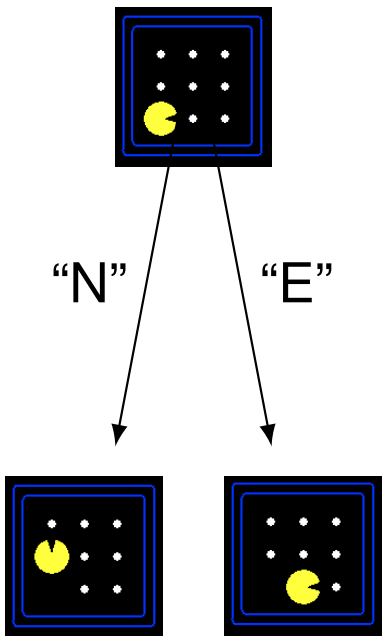
The start state is the root node

Children correspond to successors

Nodes show states, but correspond to PLANS that achieve those states

Search Trees

This is now / start



Possible futures.

A search tree:

A “what if” tree of plans and their outcomes

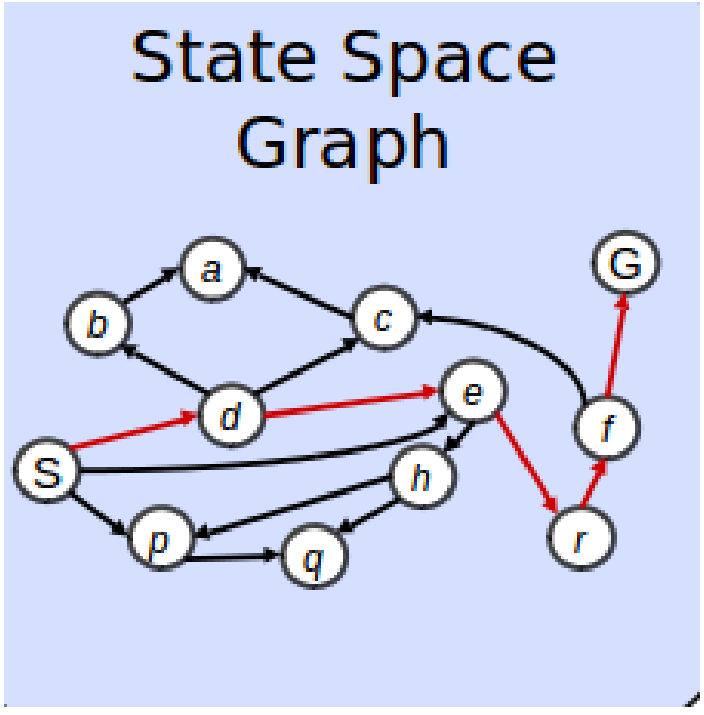
The start state is the root node

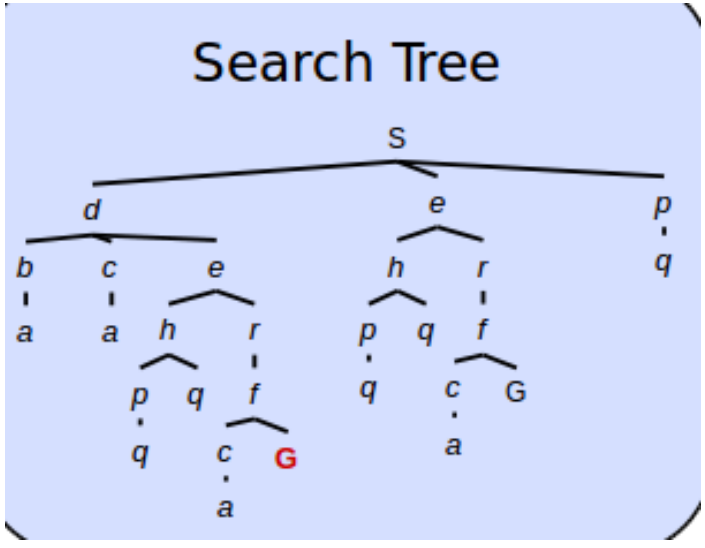
Children correspond to successors

Nodes show states, but correspond to PLANS that achieve those states

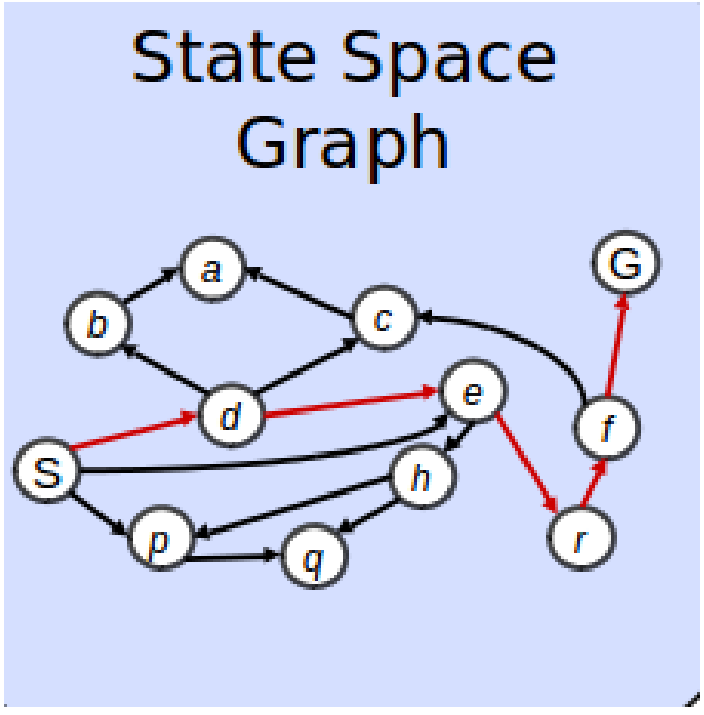
For most problems, we can never actually build the whole tree

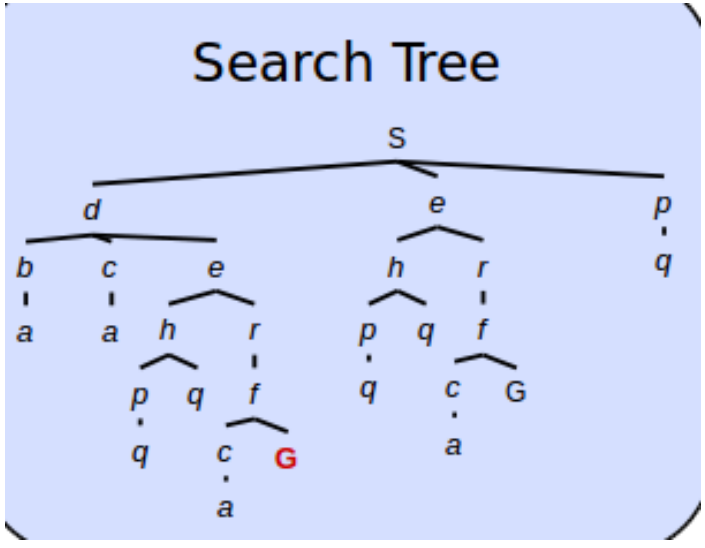
State Space Graphs vs. Search Trees





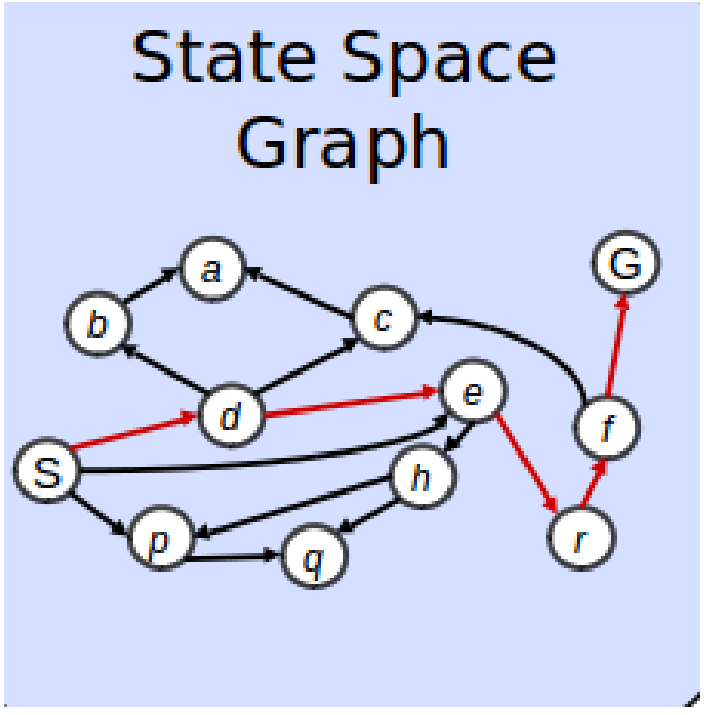
State Space Graphs vs. Search Trees

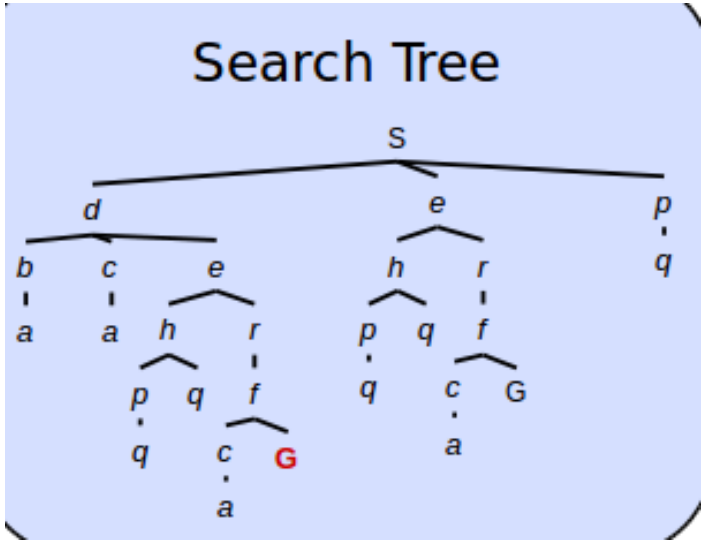




Each NODE in search tree is an entire PATH in state space graph.

State Space Graphs vs. Search Trees



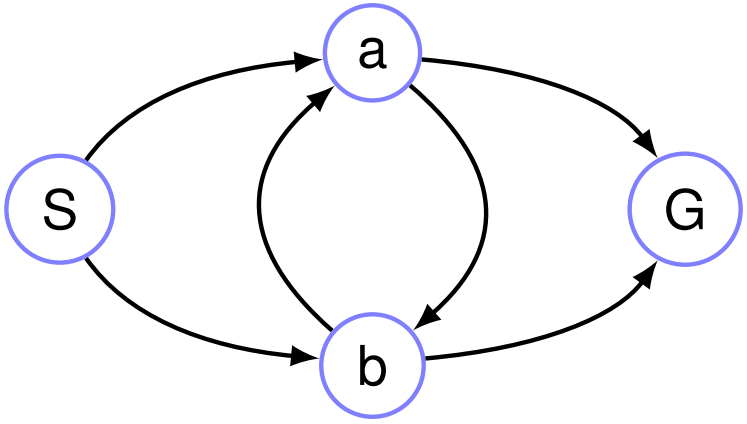


Each NODE in search tree is an entire PATH in state space graph.

We construct both on demand – and we construct as little as possible.

Quiz:State Space Graphs vs. Search Trees

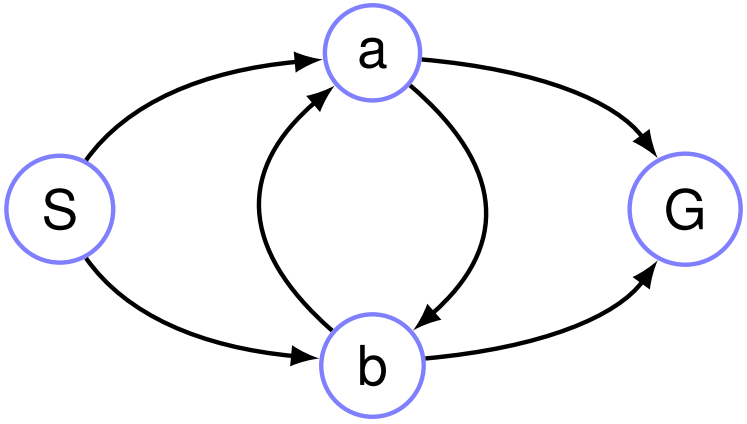
Consider this 4-state graph:



How large could search tree be?

Quiz:State Space Graphs vs. Search Trees

Consider this 4-state graph:

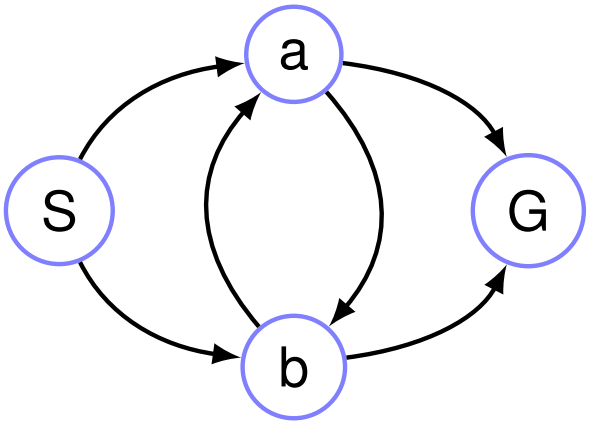


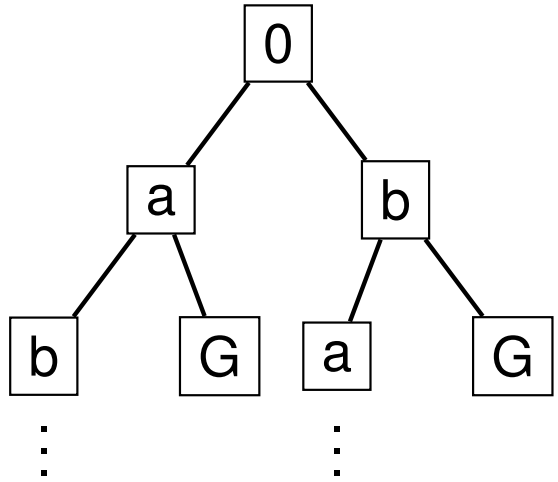
How large could search tree be?

∞

Quiz:State Space Graphs vs. Search Trees

Consider this 4-state graph:

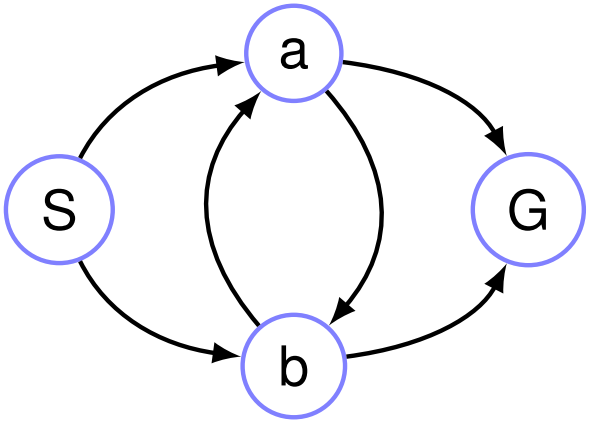


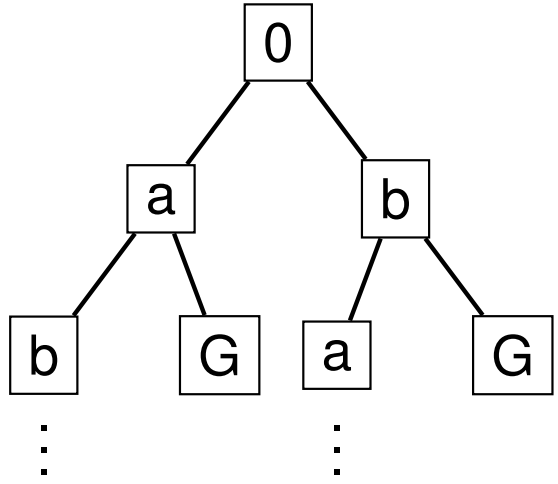


How large could search tree be?

Quiz:State Space Graphs vs. Search Trees

Consider this 4-state graph:

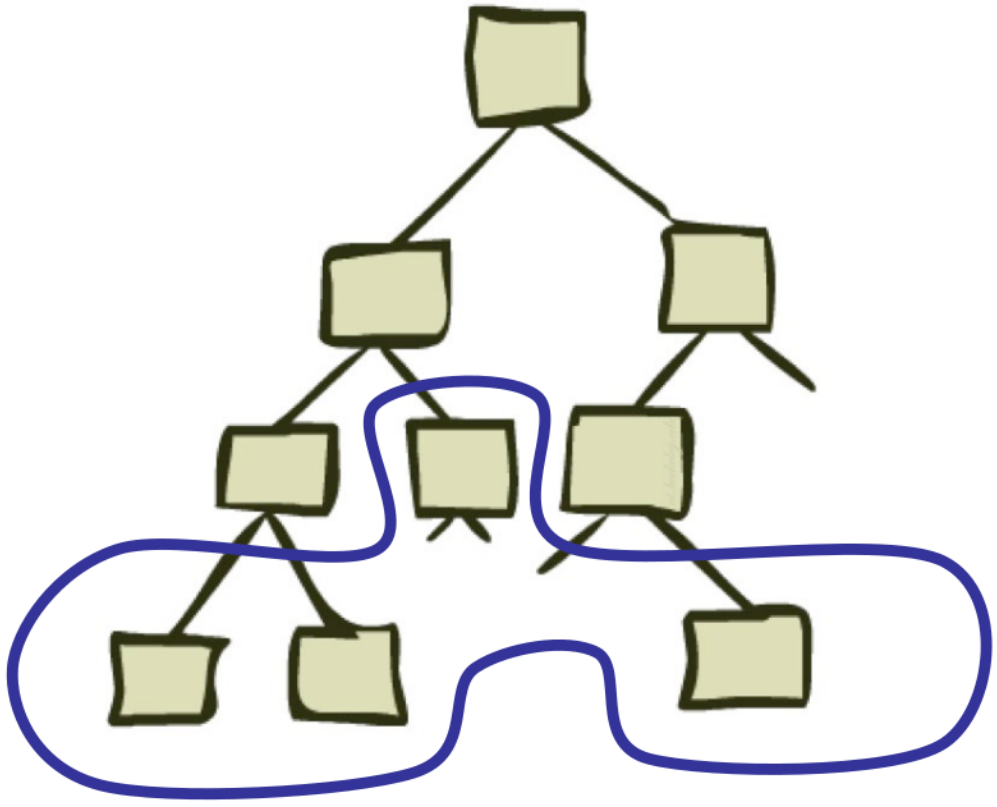




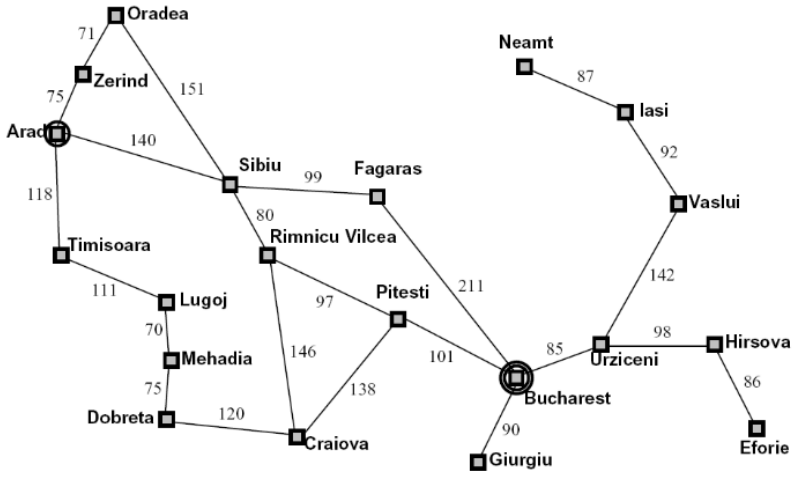
How large could search tree be?

∞

Tree Search



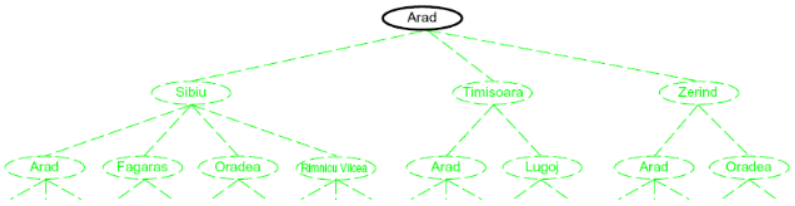
Search Example: Romania



Searching with a Search Tree

Search:

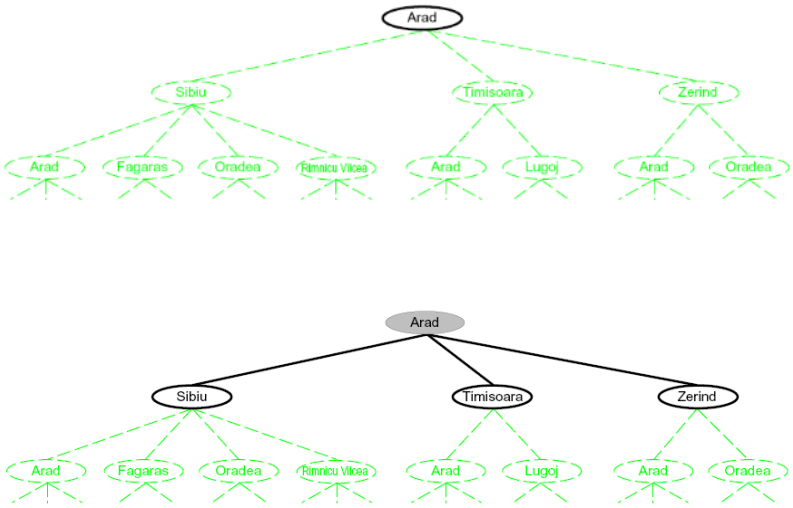
Searching with a Search Tree



Search:

(1) Expand out potential plans (tree nodes)

Searching with a Search Tree

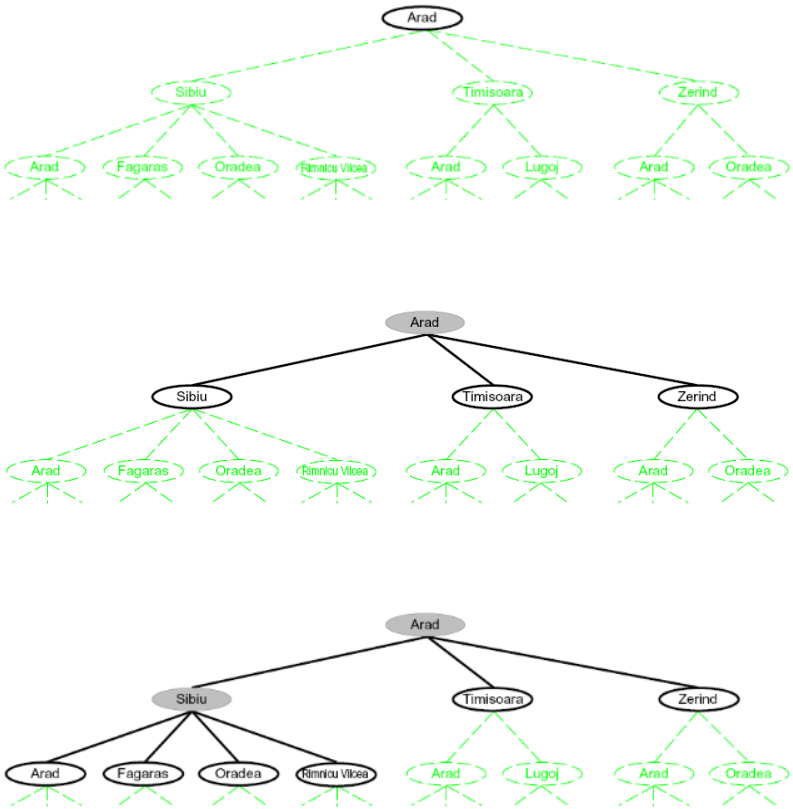


Search:

(1) Expand out potential plans (tree nodes)

(2) Maintain a fringe of partial plans under consideration.

Searching with a Search Tree

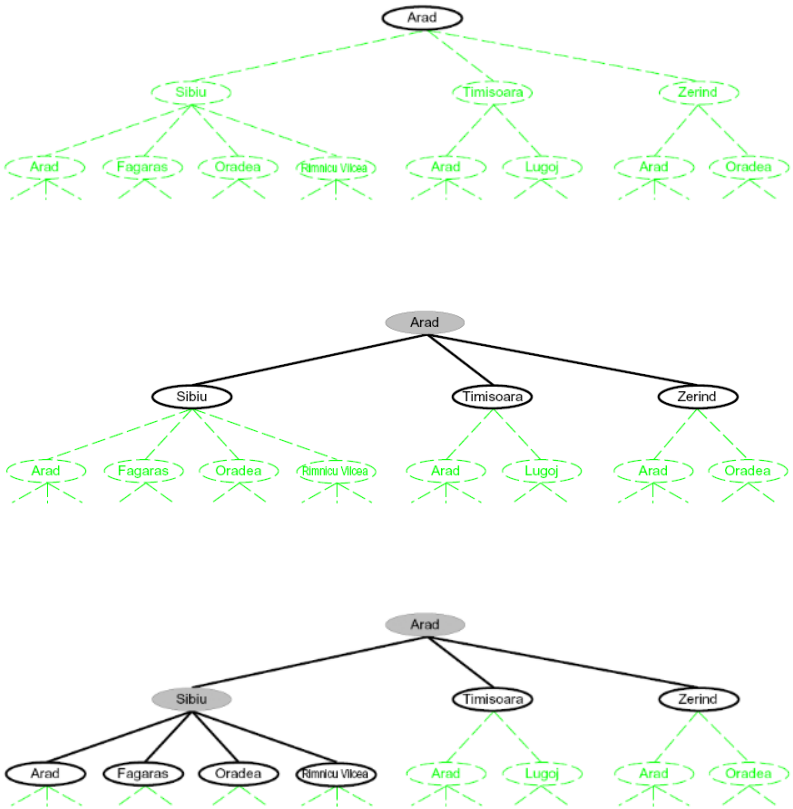


Search:

(1) Expand out potential plans (tree nodes)

(2) Maintain a fringe of partial plans under consideration.

Searching with a Search Tree



Search:

(1) Expand out potential plans (tree nodes)

(2) Maintain a fringe of partial plans under consideration.

(3) Try to expand as few tree nodes as possible

General Tree Search

Important ideas:

General Tree Search

Important ideas:

- Fringe

General Tree Search

Important ideas:

- Fringe
- Expansion

General Tree Search

Important ideas:

- Fringe
- Expansion
- Exploration strategy

General Tree Search

Important ideas:

- Fringe
- Expansion
- Exploration strategy

Main question: which fringe nodes to explore?

General Tree Search

Important ideas:

- Fringe
- Expansion
- Exploration strategy

Main question: which fringe nodes to explore?

```
function TREE-SEARCH( problem, strategy) returns a solution, or failure
    initialize the search tree using the initial state of problem
    loop do
        if there are no candidates for expansion then return failure
        choose a leaf node for expansion according to strategy
        if the node contains a goal state then return the corresponding solution
        else expand the node and add the resulting nodes to the search tree
    end
```