北京航空航天大學
BEIHANG UNIVERSITY

Data Management and Artificial Intelligence

Lecture 9

Sebastian Wandelt (小塞)

Beihang University

Outline

- Review
- More XML/Webservices
- AI (finally…☺)
  - What is Artificial Intelligence?
  - Goal-based agents
  - Representing states and operators
  - Example problems
  - State-space search algorithms

Review

Lab class task

Please recall the notion of functional dependencies from the lecture. One way to formalize the entailment of a set of dependencies were the use of Armstrong's axioms:

1. Reflexivity: If $Y \subseteq X$, then $F| = X \rightarrow Y$.
2. Augmentation: If $F| = X \rightarrow Y$, then $F| = X, W \rightarrow Y, W$.
3. Transitivity: If $F| = X \rightarrow Y$ and $F| = Y \rightarrow Z$, then $F| = X \rightarrow Z$.

Note that the variables X, Y, Z above are possible sets of attributes, while W is a single attribute. For this task, please implement the above three axioms and finally compute the closure of an input set F_input, i.e., F_plus. In addition, print the overall procedure to obtain F_plus, which is essentially a sequence of axiom applications.

For an input set $F\_input = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, GD \rightarrow H\}$, the desired closure F_plus has a size of $1,815$. A possible procedure to obtain F_plus can be $Ref - Aug - Tran - Aug - Tran - Aug - Tran - Aug - Aug$.

More XML/Webservices

Use case 1: Word documents

- How does Word (and Excel, Powerpoint) actually store their data?
  - Let's try to find out …

Use case 2: Wechat

- How does Wechat actually exchange messages?
  - Any ideas/suggestions?

- How does Wechat actually exchange messages?
- The (simplest) text message looks like this:

```
<xml>
  <ToUserName>Alice</ToUserName>
 <FromUserName>Bob</FromUserName>
 <CreateTime>1348831860</CreateTime>
 <MsgType>text</MsgType>
 <Content>How are you doing?</Content>
 <MsgId>`1234567890123456`</MsgId>
</xml>
```

Use case 3: Openstreetmap

What is OpenStreetMap?

- Maps can be used by anyone without copyright restrictions
- Editable by anyone
  - Anyone can contribute new data to the map or even edit the existing data
- A free and open data source
  - The raw map data is available for anyone to use in whatever way they like

Why?

- Existing Data not Free & Open
  - Limits your rights to use the data as you wish
- Data as a Static Snapshot
  - Commercial data may be years old
- Existing data not editable
  - If you discover an error in an existing map, you cannot instantly edit it

Where on the web?

- Main site

  http://www.openstreetmap.org

- Wiki (the first point of call for all aspects of OSM-related information)…

  http://wiki.openstreetmap.org/wiki/Main_Page

Concepts at the core of OSM

- Nodes
  - Single coordinates
- Ways
  - Collection of nodes
- Relations
  - Collections of ways and nodes
- Super-Relations
  - Collections of relations, ways and nodes

Example

https://www.openstreetmap.org/way/30783956

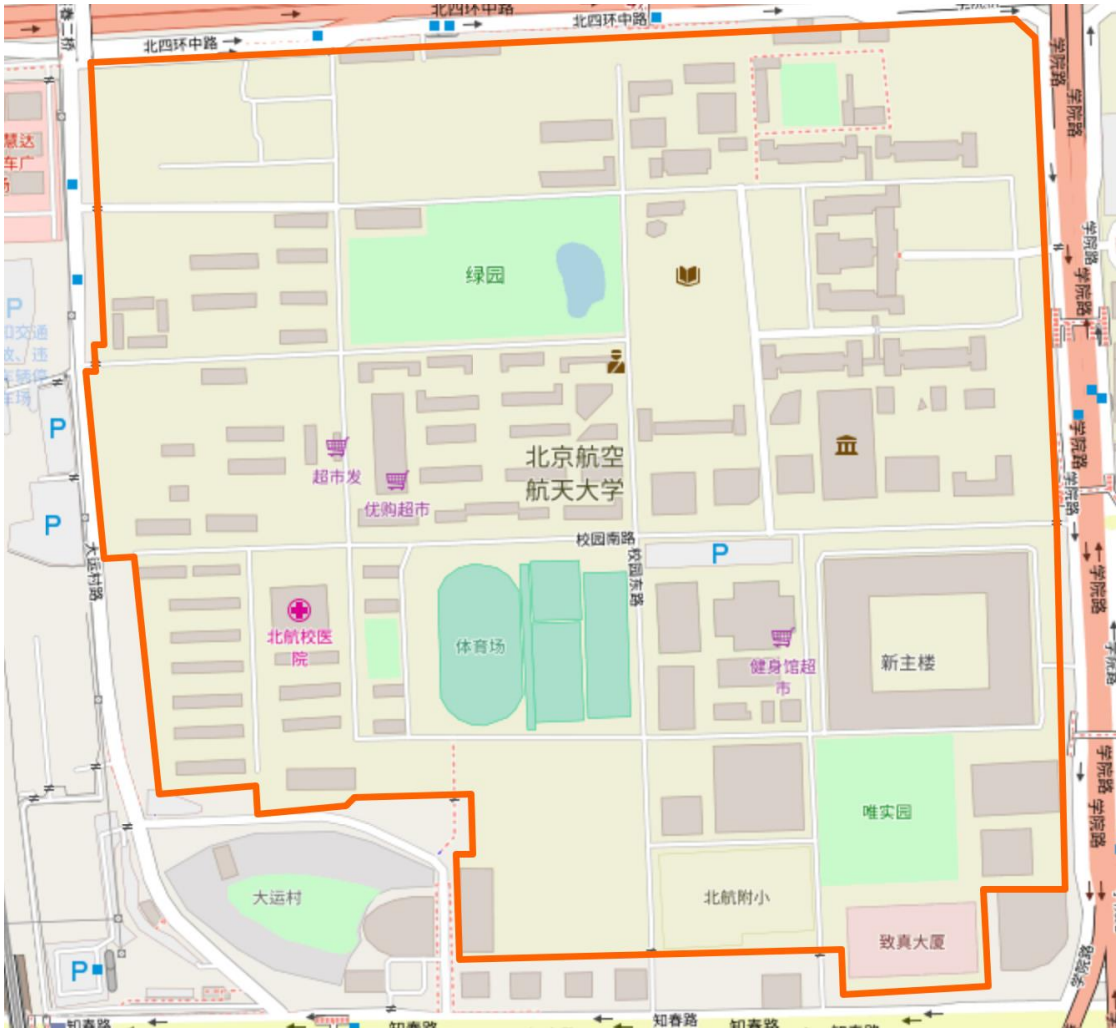
Way: Beihang University (30783956)

building attributes

Edited 17 days ago by Dingyuan Wang
Version #18 · Changeset #57445769

Tags

addr:district	海淀区
addr:housenumber	37
addr:postcode	100083
addr:province	北京市
addr:street	学院路
alt_name	北航
alt_name:en	Beijing University of Aeronautics and Astronautics
amenity	university
name	北京航空航天大学
name:de	Beihang Universität
name:en	Beihang University



Example

https://www.opens[illegible]83956

Way: Beihang University (30783956)

building attributes

Edited 17 days ago by Dingyuan Wang
Version #18 · Changeset #57445769

Tags

addr:district	海淀区
addr:housenumber	37
addr:postcode	100083
addr:province	北京市
addr:street	学院路
alt_name	北航
alt_name:en	Beijing University of Aeronautics and Astronautics
amenity	university
name	北京航空航天大学
name:de	Beihang Universität
name:en	Beihang University

Nodes

340411265
北门 (1364164098) (part of way 42943244)
340411267
340411268
东南门 (1030342347) (part of way 校园南路 (88777868))
340411269 (part of ways 学院国际大厦 (298657918) and 学院国际大厦 (577673503))
1367466437
1367466436
5291004970
5291004969
340411271
1367466442
1367469160
1367466441
1367466444
1367466446
1367466445
340411273
340411274
340411275
1367466452
1195735454 (part of way 121991971)
340411327 (part of way 121991971)
1364164085
西门 (1195735438) (part of way 572267663)

The OpenStreetMap API

- API to allow developers to make use of OSM data
- REST compliant web service
- Allows creation, retrieval and modification of map data
- Available at http://www.openstreetmap.org/api/0.6/

- Example:
  - https://www.openstreetmap.org/way/30783956
  - https://www.openstreetmap.org/api/0.6/way/30783956

OpenStreetMap API - Examples

- Retrieve map data in a given bounding box (OSM XML format)
  - http://www.openstreetmap.org/api/0.6/map?bbox=-1.5,50.98,-1.47,51.0
- Retrieve a given way (here, way with ID 223)
  - http://www.openstreetmap.org/api/0.6/way/30783956
  - (GET request, retrieves OSM XML)

Openstreetmap: Example query

https://www.openstreetmap.org/api/0.6/map?bbox=116.3346,39.97536,116.34842,39.98402

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version="0.6" generator="osmconvert 0.8.4" timestamp="2018-04-02T02:00:01Z">
    <bounds minlat="39.97536" minlon="116.3346" maxlat="39.98402" maxlon="116.3484002"/>
    <node id="27547660" lat="39.9763171" lon="116.347935" version="1"/>
    <node id="27547665" lat="39.983967" lon="116.3474887" version="1"/>
    <node id="233763157" lat="39.9783582" lon="116.347577" version="1"/>
    <node id="233763204" lat="39.9776244" lon="116.3475274" version="1"/>
    <node id="233763298" lat="39.9762586" lon="116.3476711" version="1"/>
    <node id="233763343" lat="39.9759973" lon="116.3476312" version="1"/>
    <node id="233763440" lat="39.9756373" lon="116.3475168" version="1"/>
    <node id="239433355" lat="39.9771255" lon="116.3354118" version="1"/>
    <node id="239433909" lat="39.9772792" lon="116.3353537" version="1"/>
    <node id="239434094" lat="39.9777292" lon="116.3351658" version="1"/>
    <node id="239434252" lat="39.9785672" lon="116.334987" version="1">
        <tag k="created_by" v="JOSM"/>
    </node>
    <node id="239434358" lat="39.97967" lon="116.3348666" version="1"/>
    <node id="244468026" lat="39.9839679" lon="116.3472968" version="1"/>
    <node id="276320137" lat="39.9754108" lon="116.3375568" version="1">
        <tag k="name" v="永和大王"/>
        <tag k="amenity" v="fast_food"/>
        <tag k="cuisine" v="chinese"/>
    </node>
    <node id="276320141" lat="39.9754334" lon="116.3373829" version="1">
        <tag k="shop" v="convenience"/>
        <tag k="name:en" v="7-Eleven"/>
    </node>
```

Python exercise: Print location of South-West gate

```
from lxml import etree
tree = etree.parse("map.osm")
t=tree.xpath("//node[tag[@v='北航西南门']]")
print(t[0].attrib)
```

```
{'id': '4903415115', 'visible': 'true', 'version': '1', 'changeset': '49358795',
'timestamp': '2017-06-08T08:15:19Z', 'user': 'JokerAustin', 'uid':
'6121636', 'lat': '39.9773714', 'lon': '116.3394841'}
```

OSM “planet” files

- The OSM server is intended for contributing and editing data
- Easily overloaded by excessive “read” access
- So OSM also makes its data available as downloadable “planet” files in OSM XML format
  - Something between 30 GB and 800 GB
- Application developers can use extracts of these “planet” files in offline software
- See wiki.openstreetmap.org/index.php/Planet.osm
- Country extracts available (e.g. at GeoFabrik, Germany), as whole planet rather difficult to manage
  - https://download.geofabrik.de/asia/china.html

What is Artificial Intelligence?

http://www-formal.stanford.edu/jmc/whatisai/

What is AI?

Q. What is artificial intelligence?

- A. It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Q. Yes, but what is intelligence?

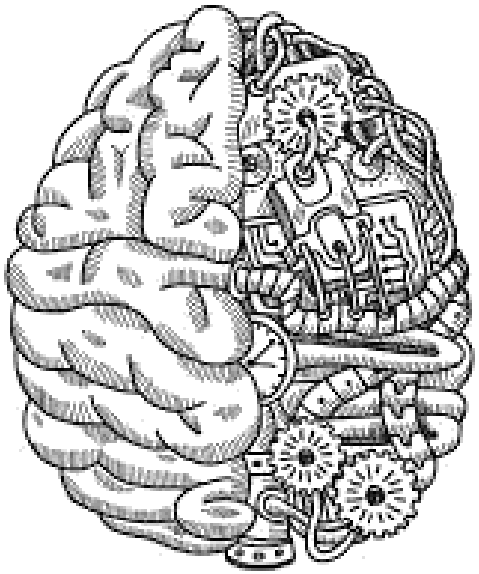
- A. Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

Q. Isn't there a solid definition of intelligence that doesn't depend on relating it to human intelligence?

- A. Not yet. The problem is that we cannot yet characterize in general what kinds of computational procedures we want to call intelligent. We understand some of the mechanisms of intelligence and not others.

Big questions in AI

- Can machines think?
- If so, how?
- If not, why not?
- What does this say about human beings?
- What does this say about the mind?



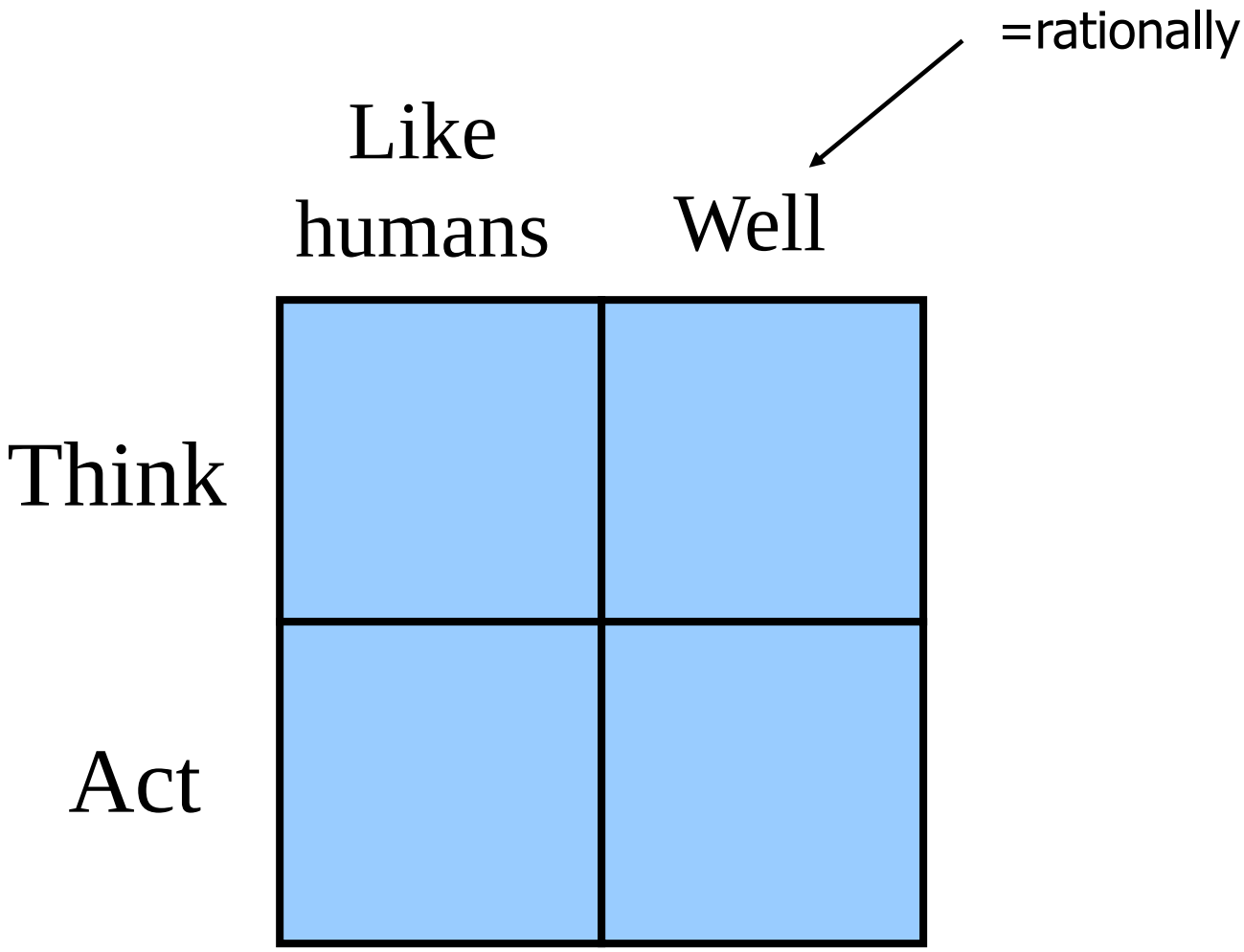
Why AI?

- **Engineering:** To get machines to do a wider variety of useful things
- **Cognitive Science:** As a way to understand how natural minds and mental phenomena work
- **Philosophy:** As a way to explore some basic and interesting (and important) philosophical questions

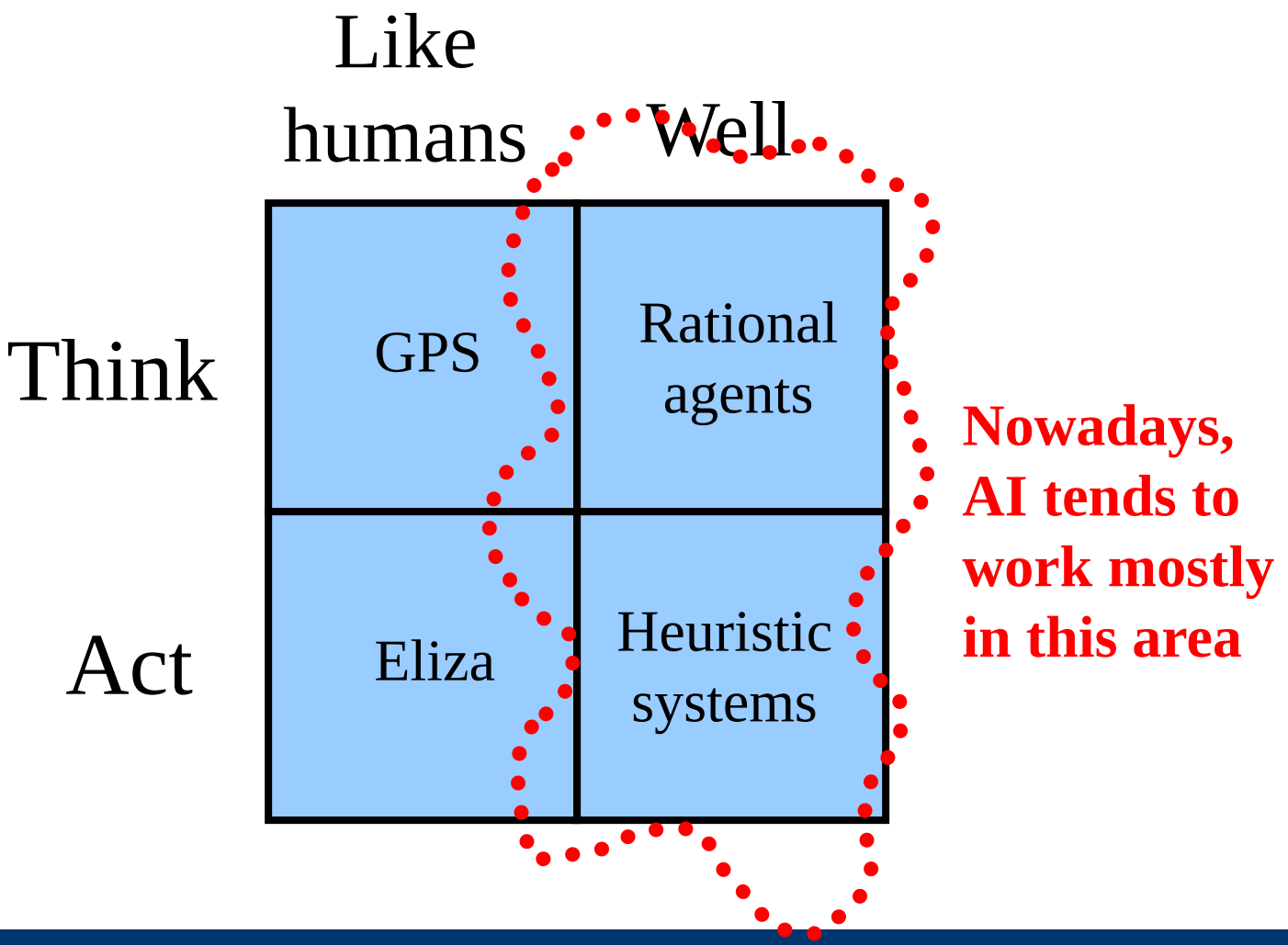
On the one hand I have a clear and distinct idea of myself, in so far as I am a thinking, non-extended thing; and on the other hand I have a distinct idea of body, in so far a this is simply an extended, non-thinking thing. And, accordingly, it is certain that I am really distinct from my body, and exist without it.

Rene Descartes

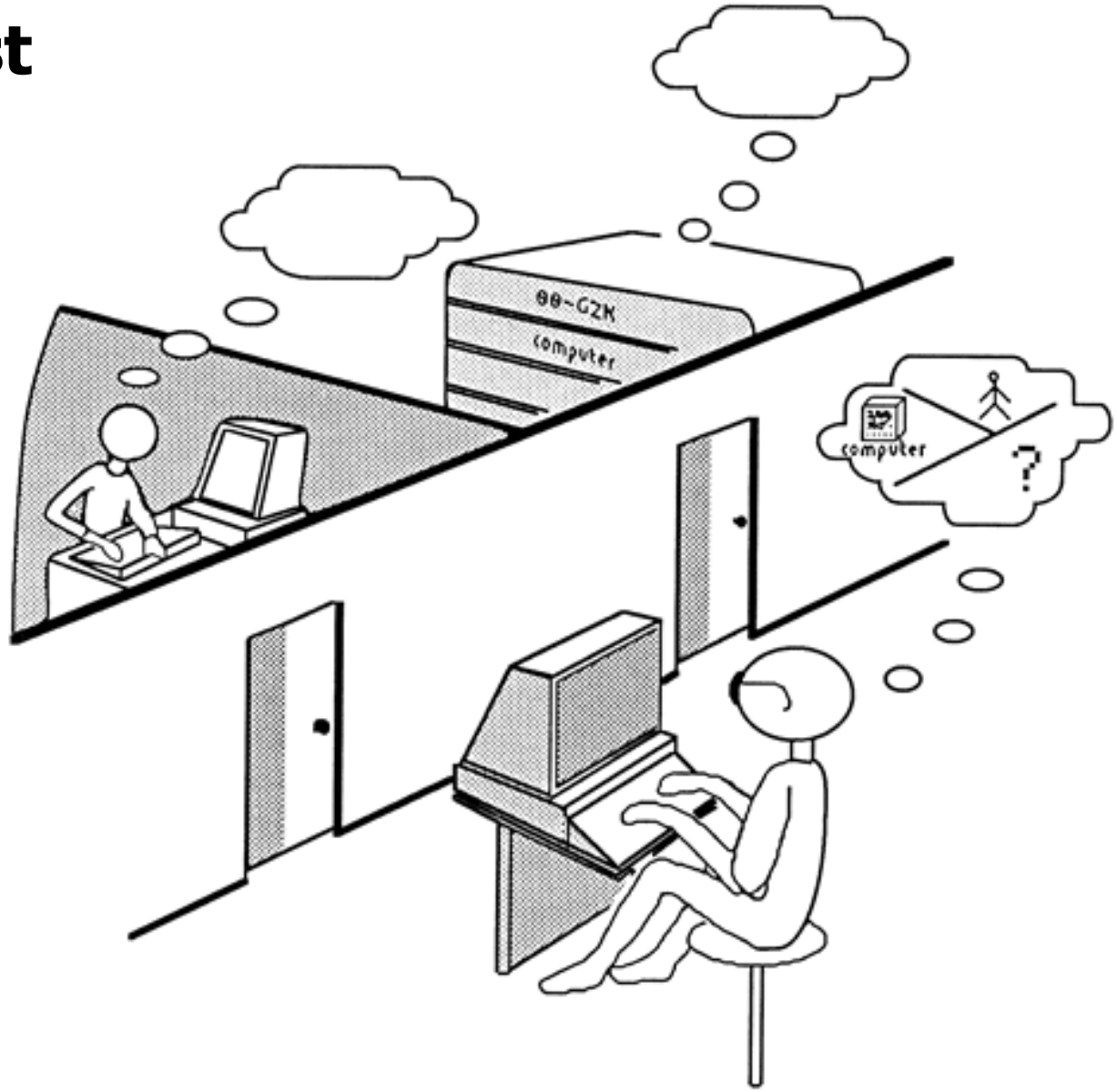
Possible approaches to AI



Possible approaches to AI



The Turing Test



Meet Eliza

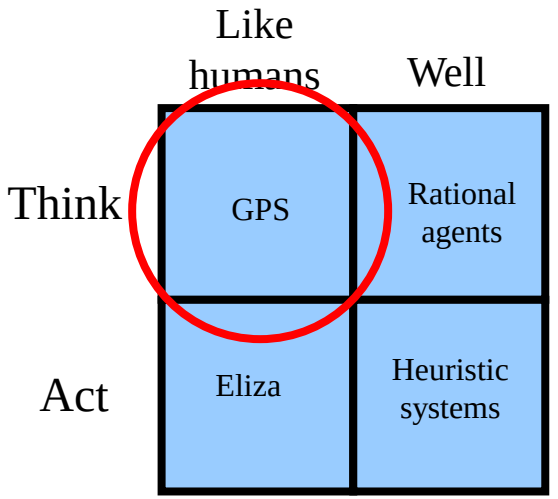
```
ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

How does Eliza work?

```
[r'I feel (.*)',
[  "Good, tell me more about these feelings.",
  "Do you often feel %1?",
  "When do you usually feel %1?",
  "When you feel %1, what do you do?"]],

[r'I have (.*)',
[  "Why do you tell me that you've %1?",
  "Have you really %1?",
  "Now that you have %1, what will you do next?"]],

[r'I would (.*)',
[  "Could you explain why you would %1?",
  "Why would you %1?",
  "Who else knows that you would %1?"]],

[r'Is there (.*)',
[  "Do you think there is %1?",
  "It's likely that there is %1.",
  "Would you like there to be %1?"]],
```

I have **a problem** -> Why do you tell me that you've **a problem**?

Think like humans



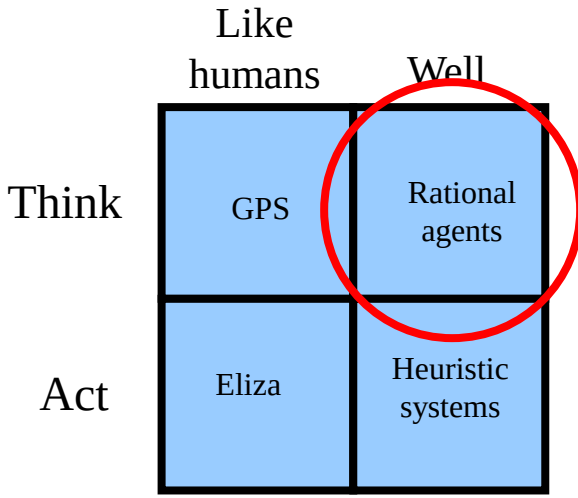
- Cognitive science approach
- Focus not just on behavior & I/O

  but also look at reasoning process

- Computational model should reflect “how” results were obtained
- Provides new language for expressing cognitive theories & new mechanisms for evaluating them
- GPS (**General Problem Solver**): Goal not just to produce humanlike behavior, but to produce a sequence of steps of reasoning process that was similar to those followed by a person

Think well

- Develop formal models of knowledge representation, reasoning, learning, memory, problem solving, that can be rendered in algorithms
- Often an emphasis on a systems that are provably correct, and guarantee finding an optimal solution



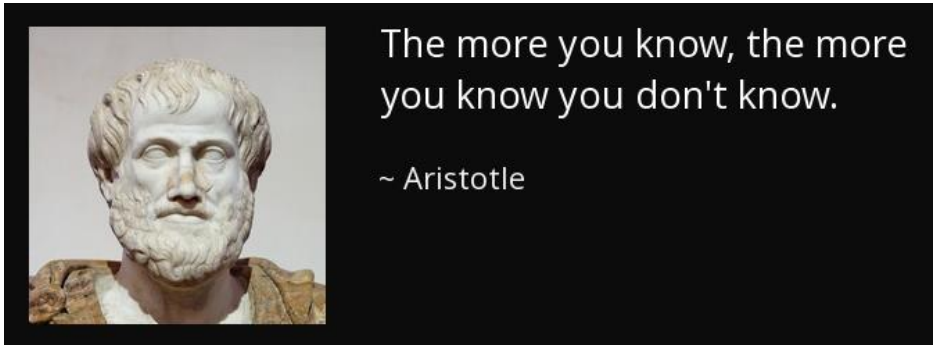
Socrates is a man

+

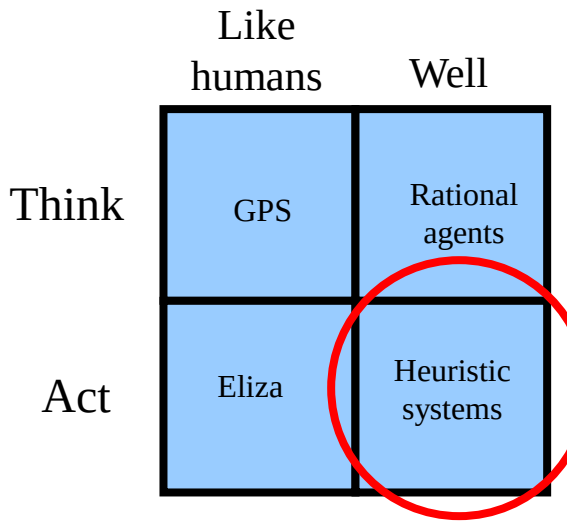
All men are mortal

⬇

Socrates is mortal



Act well



- For a given set of inputs, generate that's not necessarily correct but gets job done
- A heuristic (heuristic rule, heuristic method) is a rule of thumb, strategy, trick or simplification which drastically limits search for solutions in large problem spaces
- Heuristics don't guarantee optimal solutions or even any solution at all: **all that can be said for a useful heuristic is that it offers solutions which are good enough most of the time**

Goal-based agents

Disclaimer

- Large parts of this lecture and the next few weeks focuses on agents playing games
  - **"How does an agent solve a specific game until the end, obtaining the best possible score"?**
  - This is no limitation, since real world problems have similar structures
  - Plus, playing games is often more fun for students (☺)
- Now, we won't play fancy stuff, but usually games that require
  - LOGIC

  instead of
  - FAST REACTION TIME
  - A PRECISE HAND-EYE COORDINATION

Example: 8-Puzzle

- Given an initial configuration of 8 numbered tiles on a 3x3 board, move the tiles in such a way so as to produce a desired goal configuration of the tiles.

5	4	
6	1	8
7	3	2

Start State

1	2	3
8		4
7	6	5

Goal State

Solution?

Example: 8-Puzzle

- Given an initial configuration of 8 numbered tiles on a 3x3 board, move the tiles in such a way so as to produce a desired goal configuration of the tiles.

5	4	
6	1	8
7	3	2

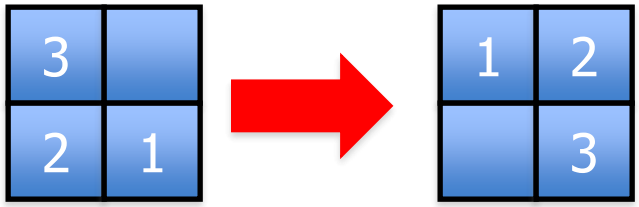
Start State

1	2	3
8		4
7	6	5

Goal State

Solution?

Simpler: 3-Puzzle

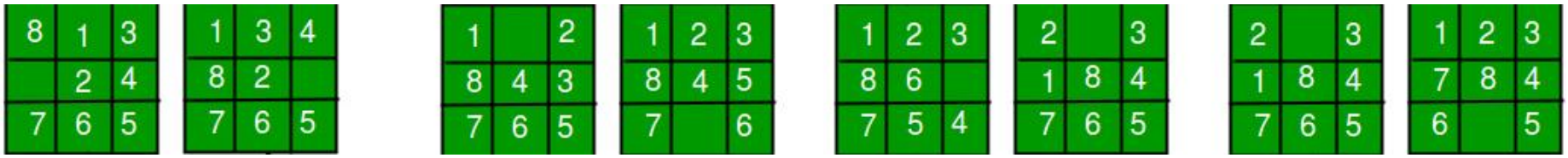


Solution?

Representing states and operators

Building goal-based agents

- To build a goal-based agent we need to answer the following questions:
  - How do we represent the state of the "world"?

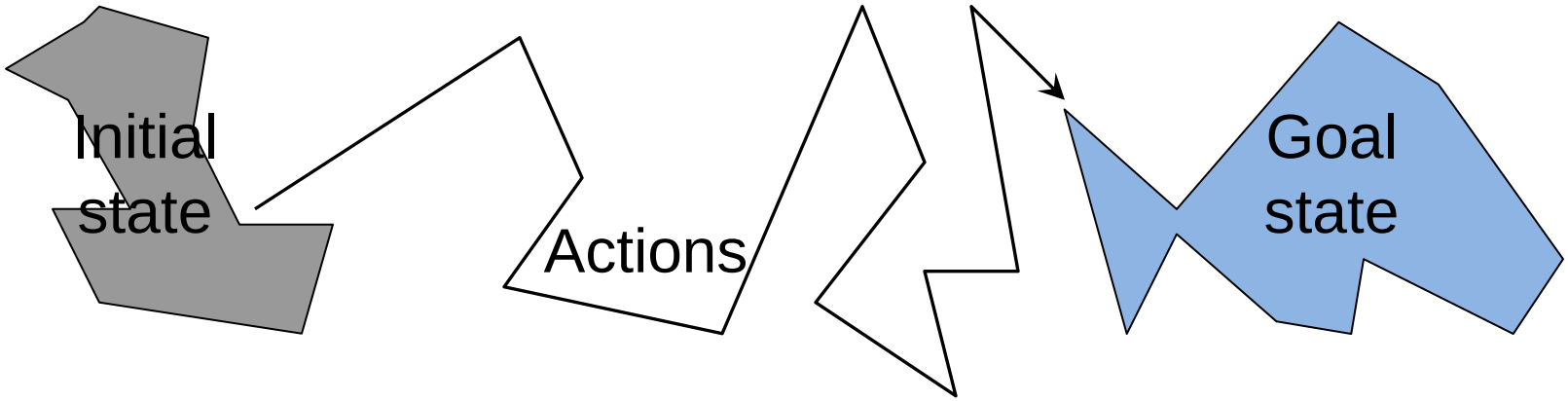


[8,1,3,*,2,4,7,6,5]

[1,2,3,8,6,*,7,5,4]

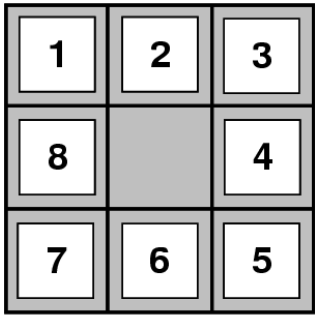
Building goal-based agents

- To build a goal-based agent we need to answer the following questions:
  - How do we represent the state of the "world"?
  - What is the goal to be achieved?
  - What are the actions?



What is the goal to be achieved?

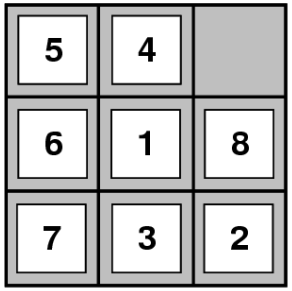
- Could describe a situation we want to achieve, a set of properties that we want to hold, etc.
- Requires defining a "goal test" so we know what it means to have achieved/satisfied our goal
- Psychologists and motivational speakers stress importance of establishing clear goals as a first step towards solving a problem
- We describe a goal of a game as states satisfying specific constraints



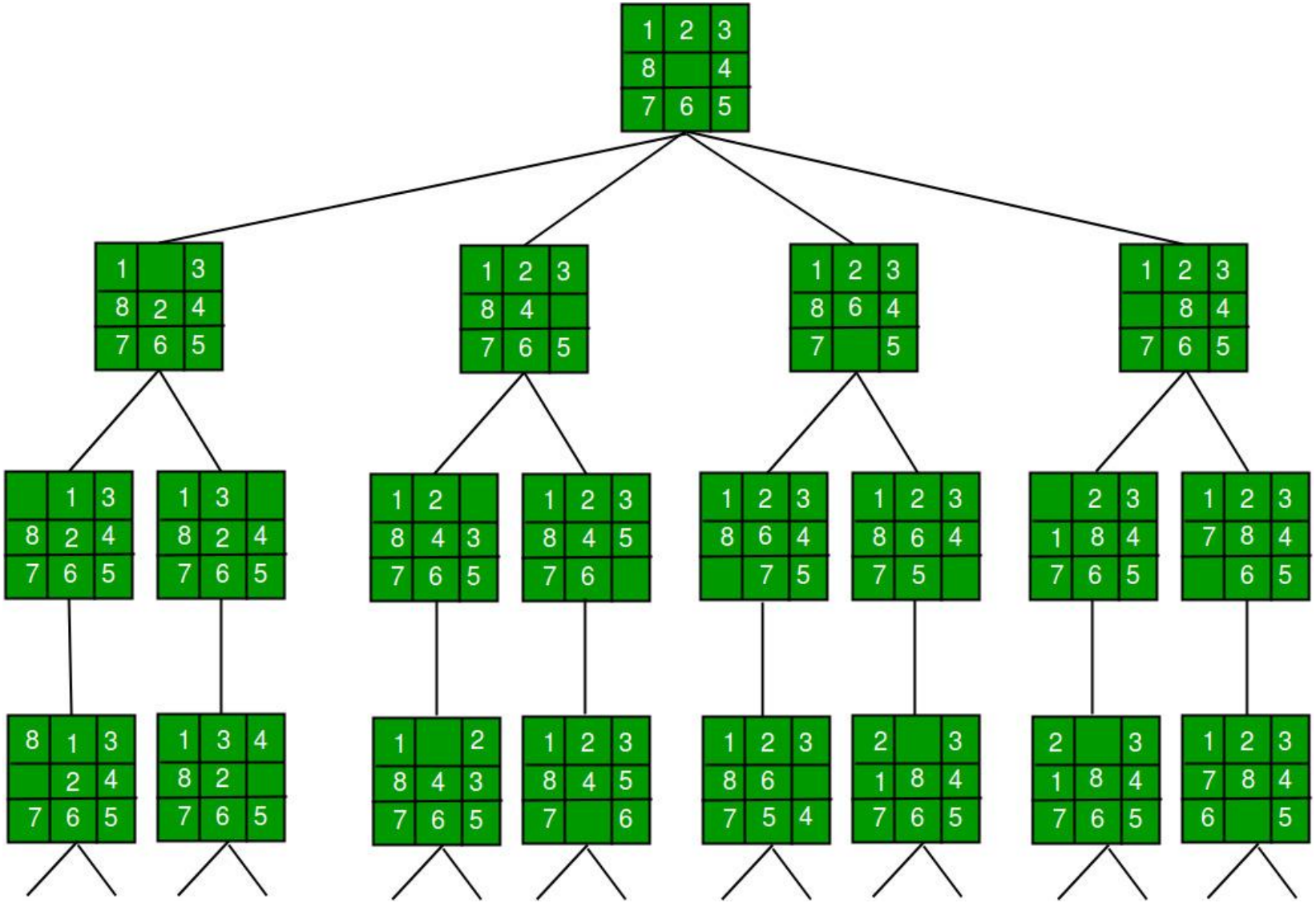
Goal State

What are the actions?

- Characterize primitive actions for making changes in the world to achieve a goal
- Deterministic world: no uncertainty in an action's effects. Given an action (a.k.a. operator) and description of the current world state, the action completely specifies
  - Whether action can be applied to the current world (i.e., is it applicable and legal?), and
  - What state will be after action is performed in the current world (i.e., no need for "history" information to compute the next state)



Building goal-based agents



Representing actions

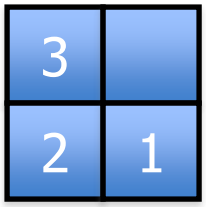
- Number of actions/operators depends on the representation used in describing a state
  - How to model actions in the 8-puzzle?

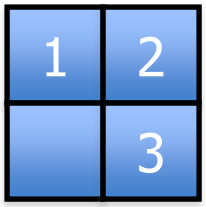
Representing actions

- Number of actions/operators depends on the representation used in describing a state
  a) Specify 4 possible moves for each of the 8 tiles, resulting in a total of 4*8=32 operators
  b) We could specify four moves for the "empty" square and we would only need 4 operators
- Representational shift can simplify a problem!

State spaces and their sizes

- State space S: Collection of a states in a game
- How large is |S| for 3-puzzle?



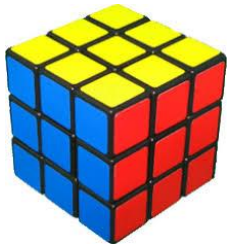


- How large is |S| for 8-puzzle?

|S| ~ Indication for difficulty

- The size of a few example problems:
  - Tic-Tac-Toe has about 3^9 states.

  - Checkers has about 10^{40} states.



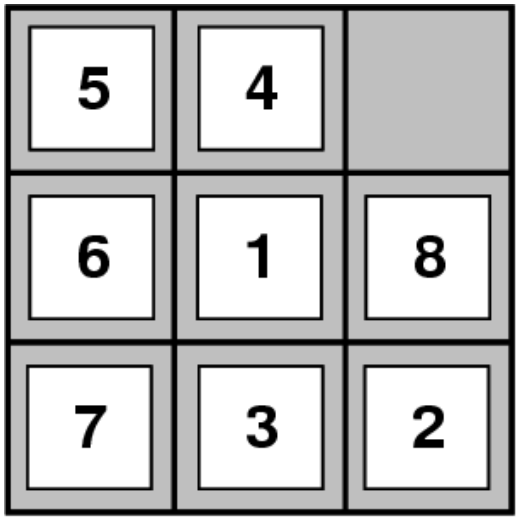
  - Rubik's Cube has about 10^{19} states.

  - Chess has about 10^{120} states in a typical game.

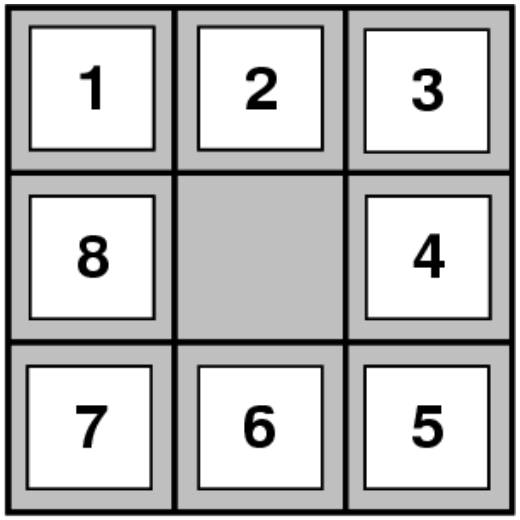
Example problems

8-Puzzle

- Given an initial configuration of 8 numbered tiles on a 3x3 board, move the tiles in such a way so as to produce a desired goal configuration of the tiles.



Start State



Goal State

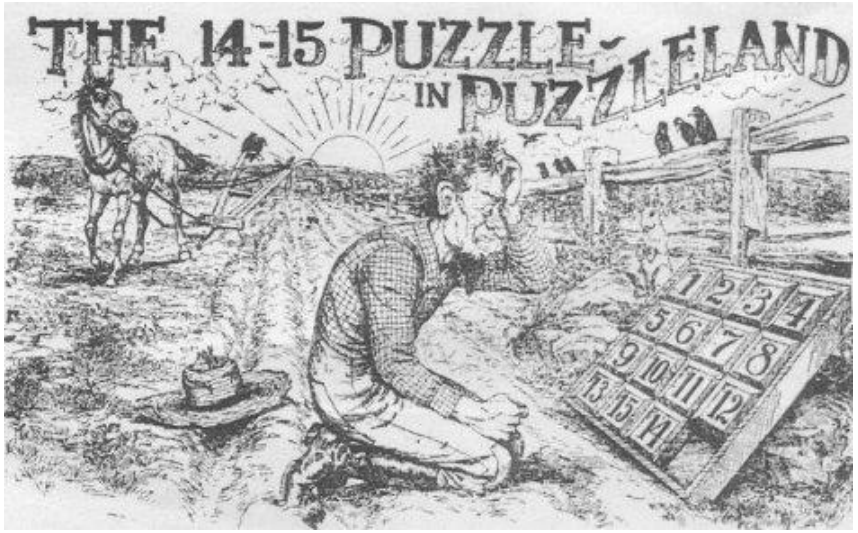
What are the states, goal test, actions?

8 puzzle

- **One state**: 3x3 array of the tiles on the board
- **Actions**: Move Blank Square Up, Down, Left, Right
  - More efficient operator encoding than one with four possible moves for each of eight distinct tiles
- **Initial State**: A particular board configuration
- **Goal**: A particular configuration of the board

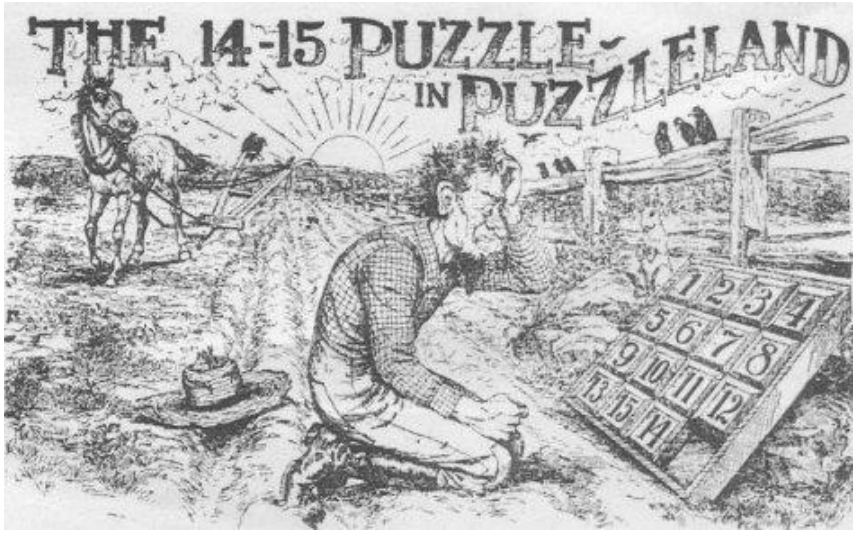
15 puzzle

- Popularized, but not invented by, Sam Loyd
- In the late 1800s he offered $1000 to all who could find a solution
- He sold many puzzles



15 puzzle

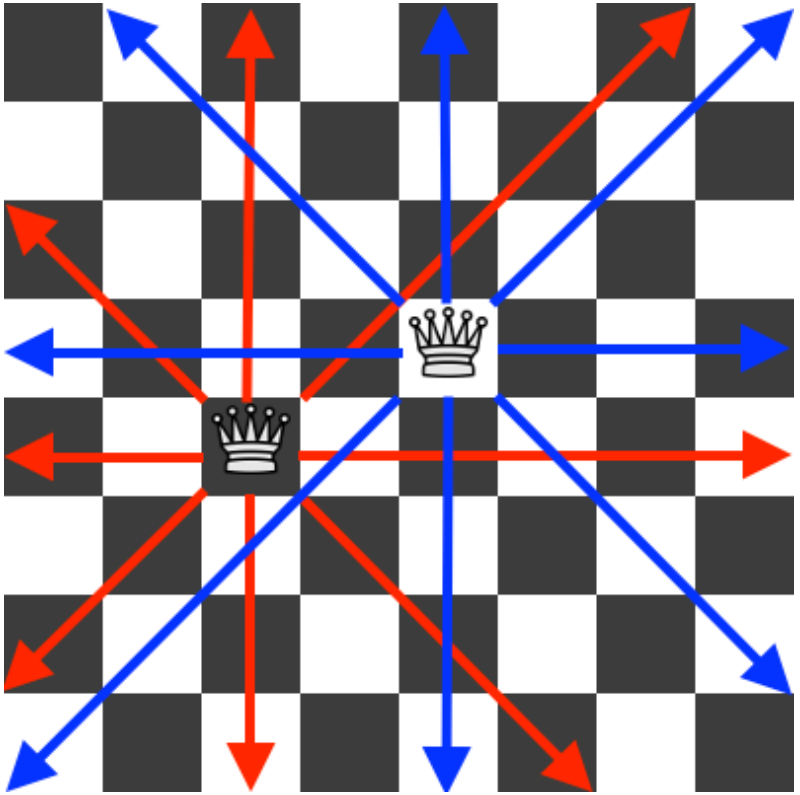
- Popularized, but not invented by, Sam Loyd
- In the late 1800s he offered $1000 to all who could find a solution
- He sold many puzzles
- The states form two disjoint spaces
- There was no path to the solution from his initial state!



8-Queens Puzzle

- Place eight queens on a chessboard such that no queen attacks any other queen

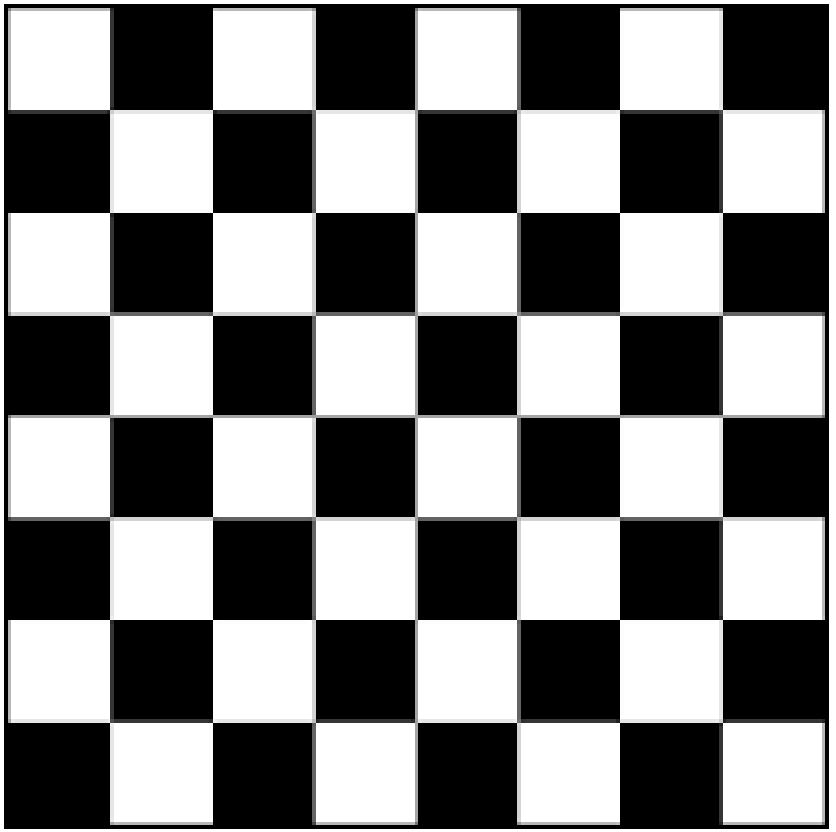
What is the state space?
What are the actions?



What are the states, goal test, actions?

8-Queens Puzzle

- Place eight queens on a chessboard such that no queen attacks any other queen



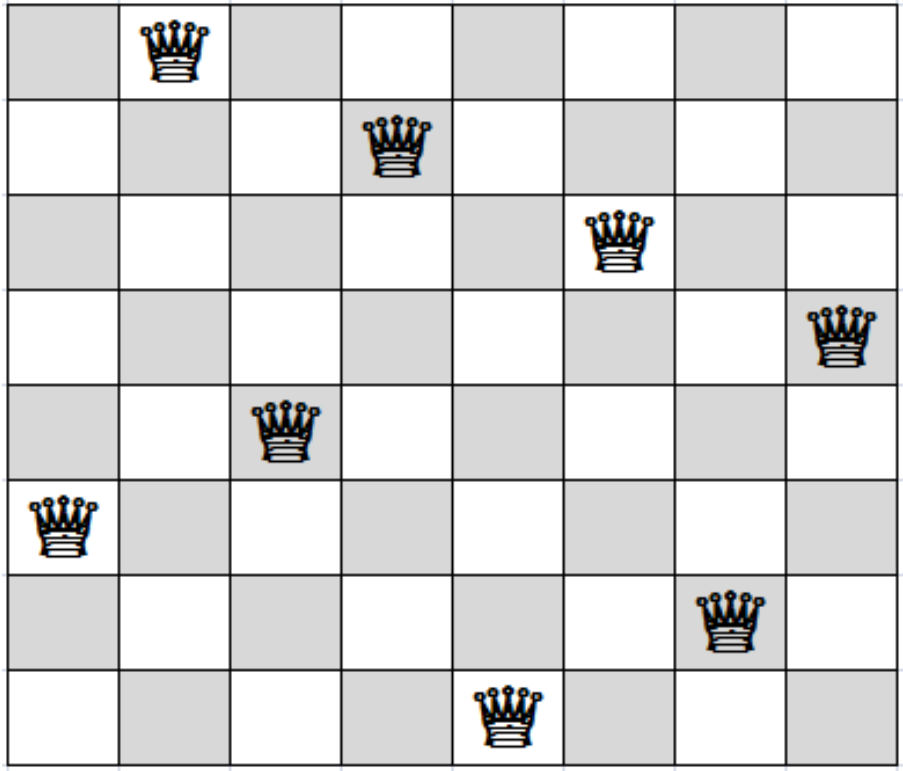
Can you find a solution?

What are the states, goal test, actions?

8-Queens Puzzle

- Place eight queens on a chessboard such that no queen attacks any other queen

Solution:



What are the states, goal test, actions?

Missionaries and Cannibals

- There are 3 missionaries, 3 cannibals, and 1 boat that can carry up to two people on one side of a river
  - Goal: Move all the missionaries and cannibals across the river
  - Constraint: Missionaries can't be outnumbered by cannibals on either side of river, or else the missionaries are killed

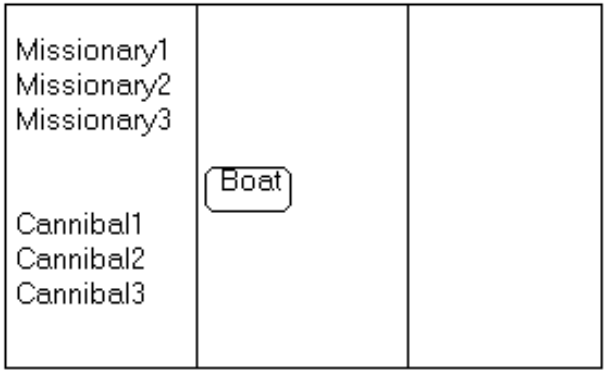
Solution?

Missionary1 Missionary2 Missionary3 Cannibal1 Cannibal2 Cannibal3	Boat	

Missionaries and Cannibals

- There are 3 missionaries, 3 cannibals, and 1 boat that can carry up to two people on one side of a river
  - Goal: Move all the missionaries and cannibals across the river
  - Constraint: Missionaries can't be outnumbered by cannibals on either side of river, or else the missionaries are killed
  - State: configuration of missionaries and cannibals and boat on each side of river
  - Actions: Move boat containing some set of occupants across the river (in either direction) to the other side

Solution?



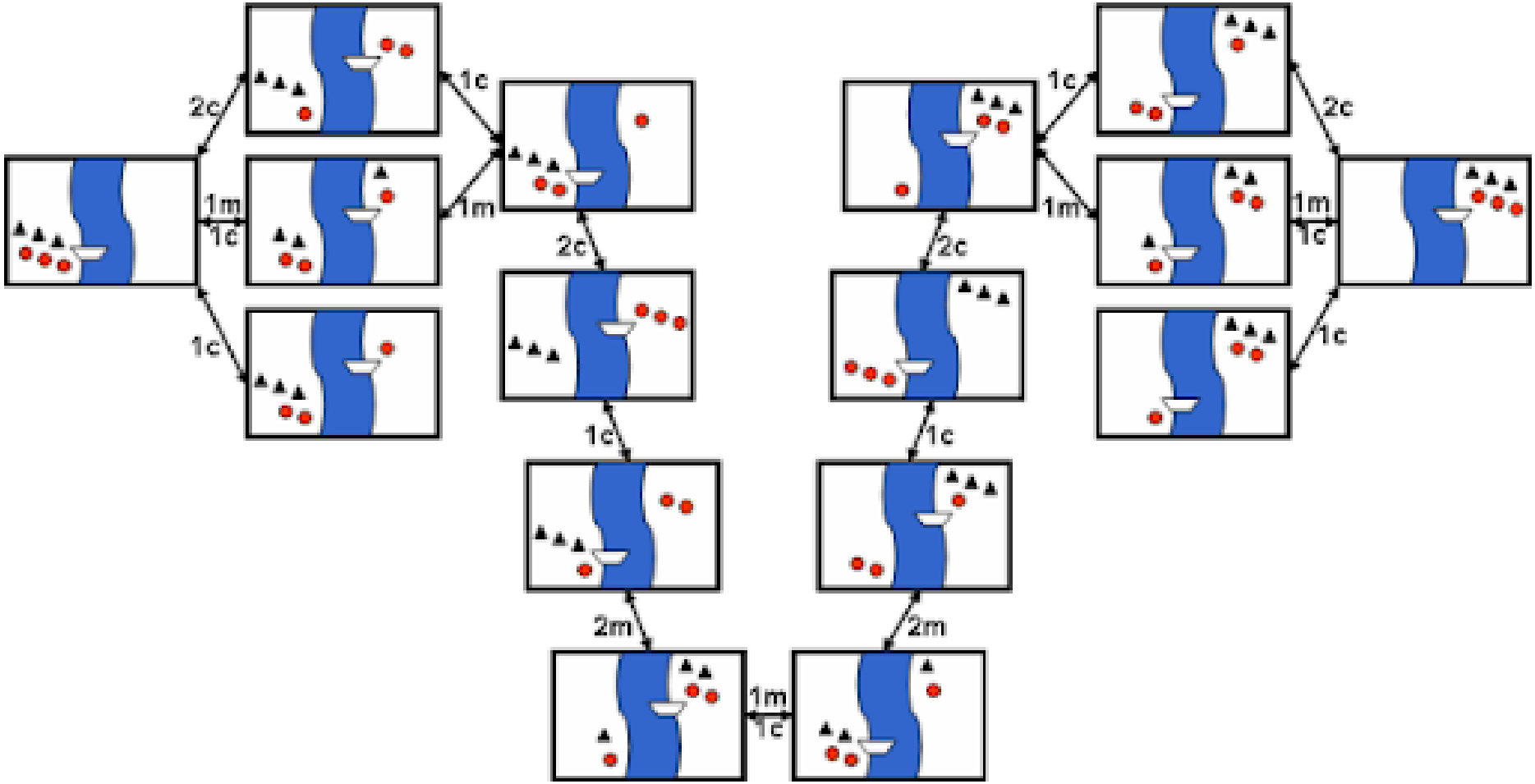
Missionaries and Cannibals Solution

	Near side		***Far side***
0 Initial setup:	MMMCCC	B	-
1 Two cannibals cross over:	MMMC	B	CC
2 One comes back:	MMMCC	B	C
3 Two cannibals go over again:	MMM	B	CCC
4 One comes back:	MMMC	B	CC
5 Two missionaries cross:	MC	B	MMCC
6 A missionary & cannibal return:	MMCC	B	MC
7 Two missionaries cross again:	CC	B	MMMC
8 A cannibal returns:	CCC	B	MMM
9 Two cannibals cross:	C	B	MMMCC
10 One returns:	CC	B	MMMC
11 And brings over the third:	-	B	MMMCCC

Missionaries and Cannibals Solution

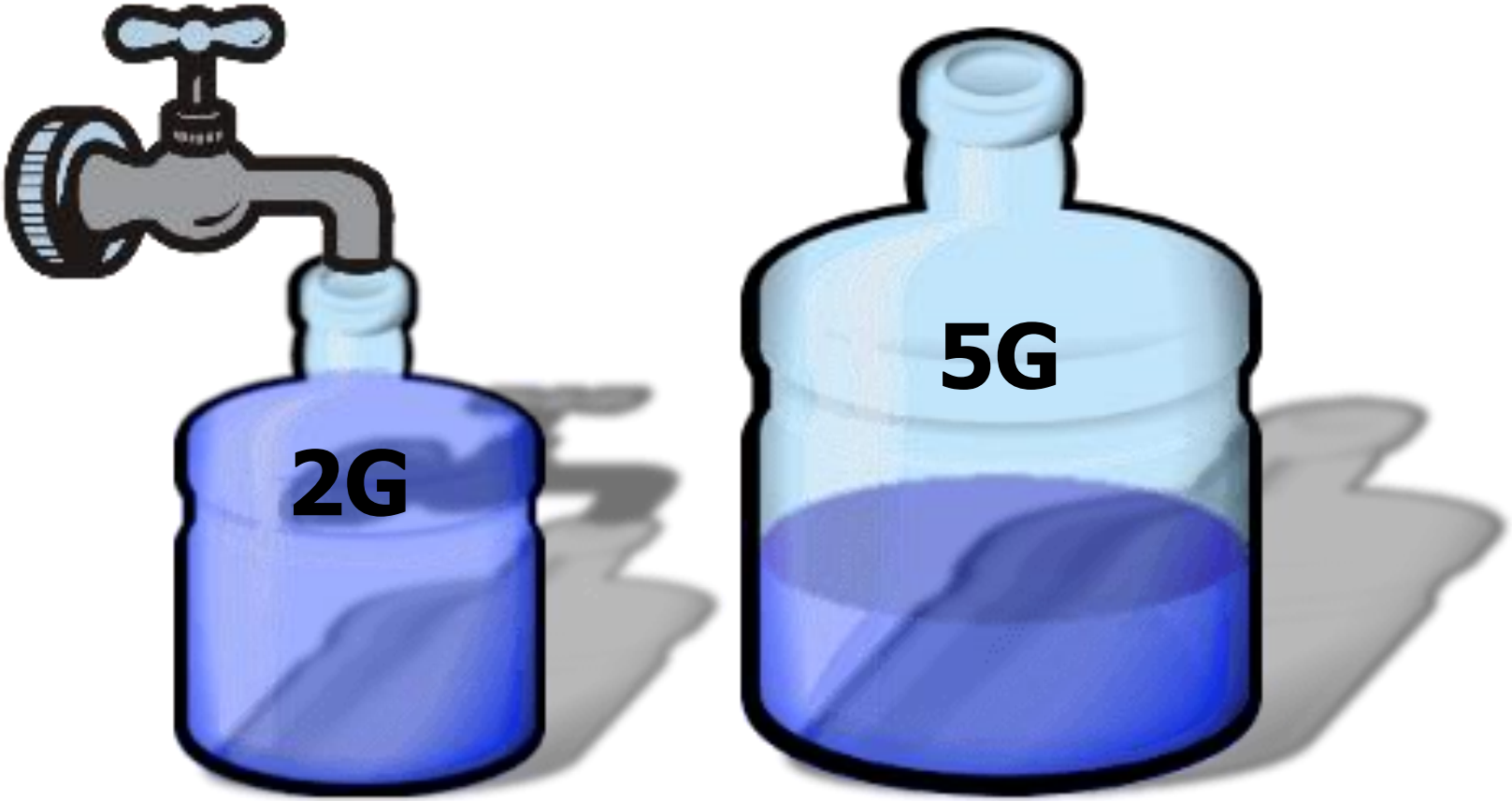
https://www.novelgames.com/en/missionaries/

Missionaries and Cannibals Solution(s)



Water Jug Problem

Given a full 5 gallon jug and a full 2 gallon jug, goal is to fill the 2 gallon jug with exactly one gallon



Water Jug Problem

Given a full 5 gallon jug and an empty 2 gallon jug, goal is to fill the 2 gallon jug with exactly one gallon

- State = (x,y), where x is water in the 5G jug and y is water in the 2G gallon jug
- Initial State = (5,2)
- Goal State = (*,1), where * means any amount

Action table

Action	Cond.	Transition	Effect
Empty5		(x,y)→(0,y)	Empty 5G jug
Empty2		(x,y)→(x,0)	Empty 2G jug
2to5	$x \leq 3$	(x,2)→(x+2,0)	Pour 2G into 5G
5to2	$x \geq 2$	(x,0)→(x-2,2)	Pour 5G into 2G
5to2part	$y < 2$	(1,y)→(0,y+1)	Pour partial 5G into 2G

Knowledge representation issues

- What's in a state?
  - Is boat color relevant to solving the M&C problem? Is sunspot activity relevant to predicting the stock market? This a hard problem that's usually left to a person's decision.
- The right level of abstraction to describe the world
  - Too fine-grained and we'll "miss the forest for the trees." Too coarse-grained and we'll miss critical details for solving the problem.

State-space search algorithms

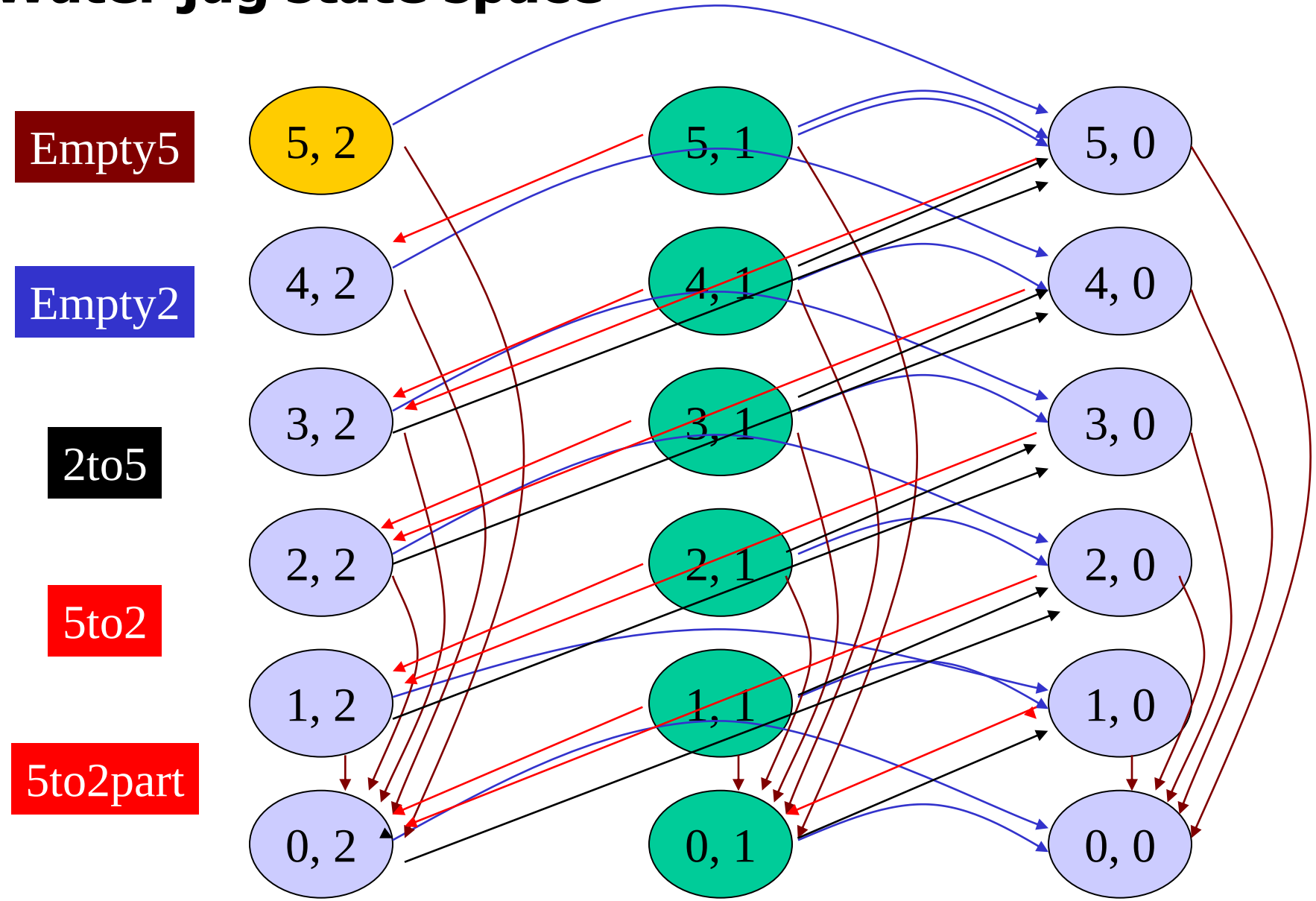
Formalizing search in a state space

- A state space is a graph (V, E) where V is a set of nodes and E is a set of links, and each link is directed from a node to another node
- Each node is a data structure with a state description and other info, such as the node's parent, the name of the action that generated it from that parent, and other bookkeeping data
- Each link is an instance of one of the actions. When the action is applied to the state at its source node, then the resulting state is the link's destination node

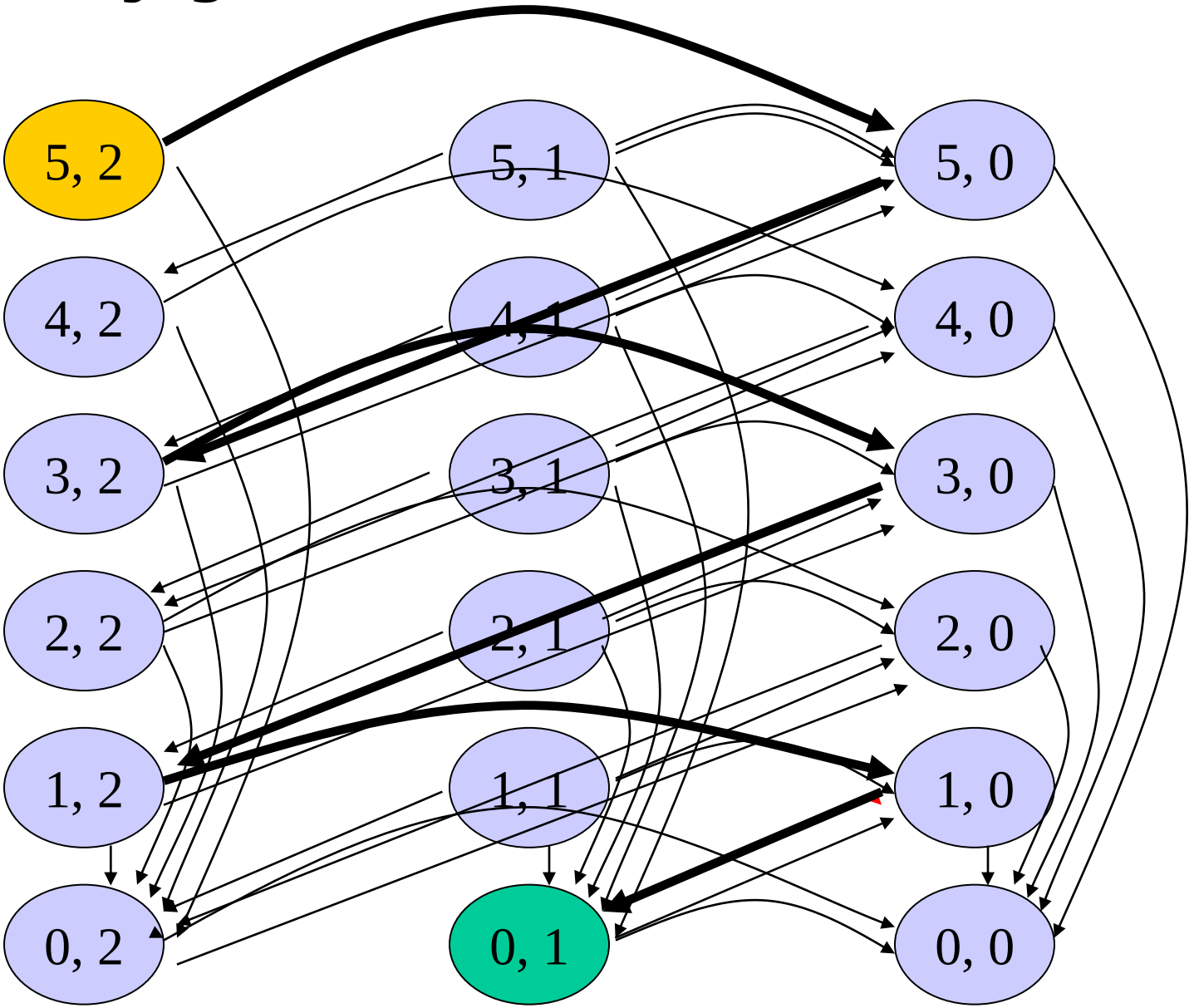
Formalizing search in a state space

- Each link has fixed, positive cost associated with it corresponding to the action cost
- Each node has a set of successor nodes corresponding to all of the legal actions that can be applied at the node's state
  - Expanding a node = generating its successor nodes and adding them and their associated links to the graph
- One or more nodes are marked as start nodes
- A goal test predicate is applied to a state to determine if its associated node is a goal node

Water jug state space



One water jug solution



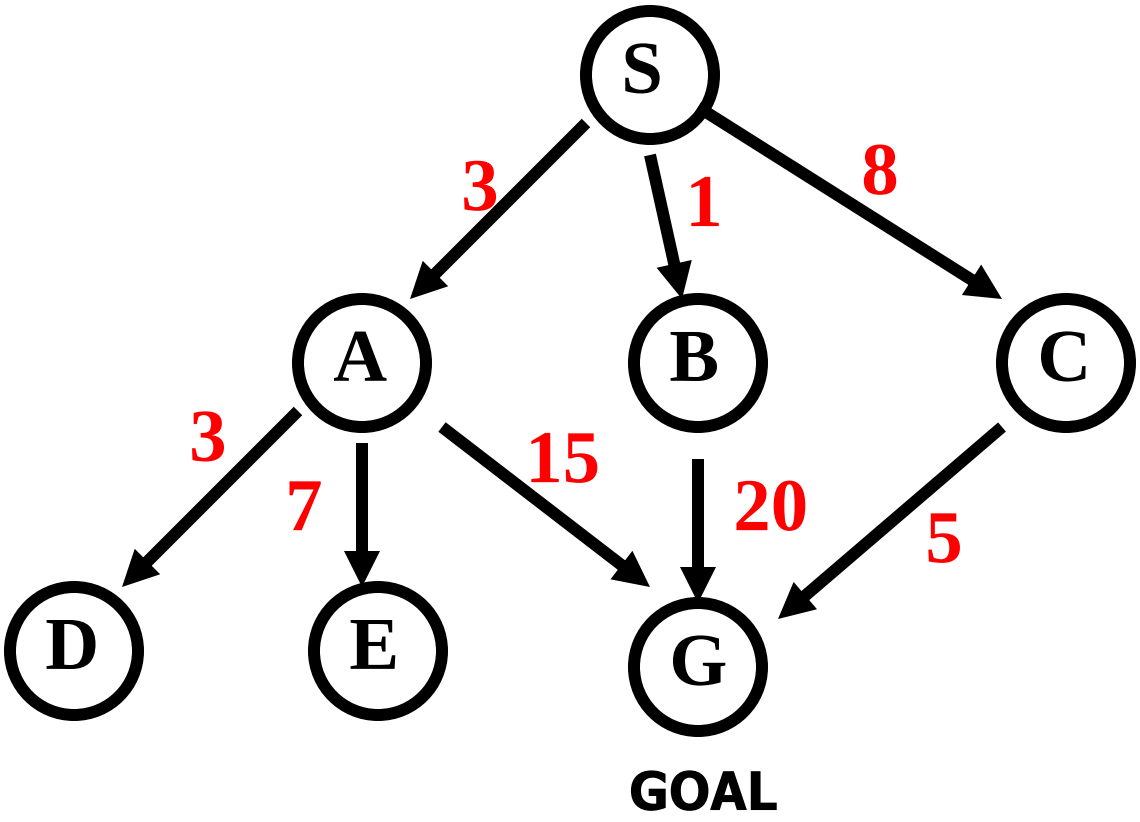
Formalizing search

- A solution is a sequence of actions that is associated with a path in a state space from a start node to a goal node.
- The cost of a solution is the sum of the arc costs on the solution path.
  - If all arcs have the same (unit) cost, then the solution cost is just the length of the solution (number of steps / state transitions)
- How to find such a solution?
  - Not so easy, because we cannot materialize the whole graph!
  - Most games have a too large state space

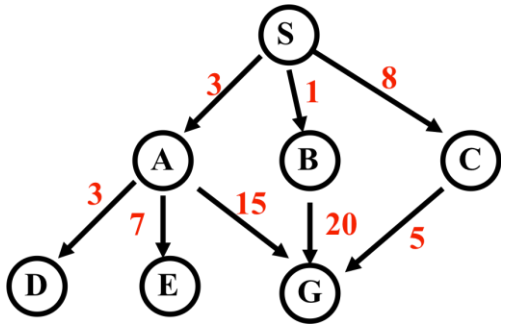
Formalizing search

- State-space search is searching through a state space for a solution by making explicit a **sufficient portion** of an implicit state-space graph to find a goal node
  - Initially V={S}, where S is the start node, E={}
  - On expanding S, its successor nodes are generated and added to V and associated arcs added to E
  - Process continues until a goal node is found
- Nodes represent a partial solution path (+ cost of partial solution path) from S to the node

Example of uninformed search strategies



Breadth-First Search

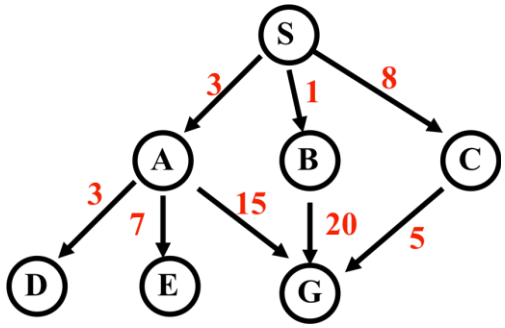


Expanded node	Nodes list
	{ S^0 }
S^0	{ A^3 B^1 C^8 }
A^3	{ B^1 C^8 D^6 E^{10} G^{18} }
B^1	{ C^8 D^6 E^{10} G^{18} G^{21} }
C^8	{ D^6 E^{10} G^{18} G^{21} G^{13} }
D^6	{ E^{10} G^{18} G^{21} G^{13} }
E^{10}	{ G^{18} G^{21} G^{13} }
G^{18}	{ G^{21} G^{13} }

Solution path found is S A G , cost 18

Number of nodes expanded (including goal node) = 7

Depth-First Search

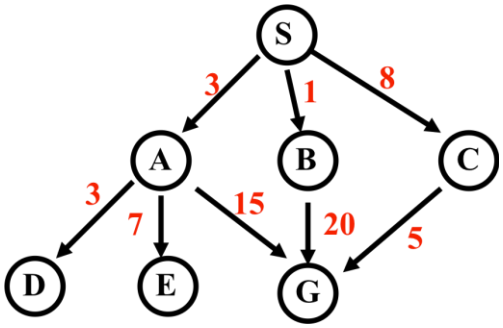


Expanded node	Nodes list
	{ S^0 }
S^0	{ A^3 B^1 C^8 }
A^3	{ D^6 E^{10} G^{18} B^1 C^8 }
D^6	{ E^{10} G^{18} B^1 C^8 }
E^{10}	{ G^{18} B^1 C^8 }
G^{18}	{ B^1 C^8 }

Solution path found is S A G, cost 18

Number of nodes expanded (including goal node) = 5

Uniform-Cost Search

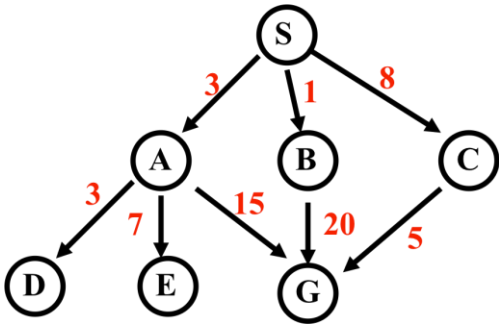


Expanded node	Nodes list
	{ S^0 }
S^0	{ B^1 A^3 C^8 }
B^1	{ A^3 C^8 G^{21} }
A^3	{ D^6 C^8 E^{10} G^{18} G^{21} }
D^6	{ C^8 E^{10} G^{18} G^{21} }
C^8	{ E^{10} G^{13} G^{18} G^{21} }
E^{10}	{ G^{13} G^{18} G^{21} }
G^{13}	{ G^{18} G^{21} }

Solution path found is S C G, cost 13

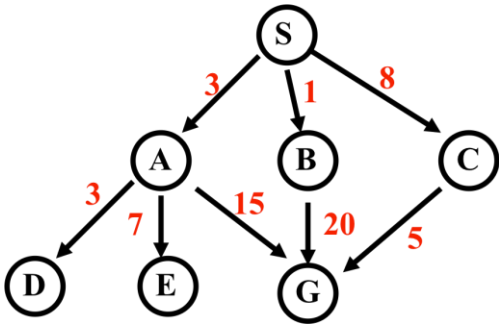
Number of nodes expanded (including goal node) = 7

Summary: How they perform



- Depth-First Search:
  - 4 Expanded nodes: S A D E G
  - Solution found: S A G (cost 18)
- Breadth-First Search:
  - 7 Expanded nodes: S A B C D E G
  - Solution found: S A G (cost 18)
- Uniform-Cost Search:
  - 7 Expanded nodes: S A D B C E G
  - Solution found: S C G (cost 13)
  - This is the only uninformed search that concerns costs.

Holy Grail Search



Expanded node	Nodes list
	{ S^0 }
S^0	{C^8 A^3 B^1 }
C^8	{ G^{13} A^3 B^1 }
G^{13}	{ A^3 B^1 }

Solution path found is S C G, cost 13 (optimal)

Number of nodes expanded (including goal node) = 3
(as few as possible!)

If only we knew where we were headed…

Summary

- We are dealing with rational agents
  - Describing a problem as a to-be-explored state space
  - Actions describe the moves in each state
  - For exploration, many search strategies exist
- We want to design agents that can reach the goal as fast as possible
  - Finding the best solution quickly for large problems is hard
  - That is the challenge for AI
- Most parts of the lecture deal with agents solving games
  - No real restriction (think about the hype on Alpha GO Zero)
  - The concept is very similar for other problem domains

Thank you very much!