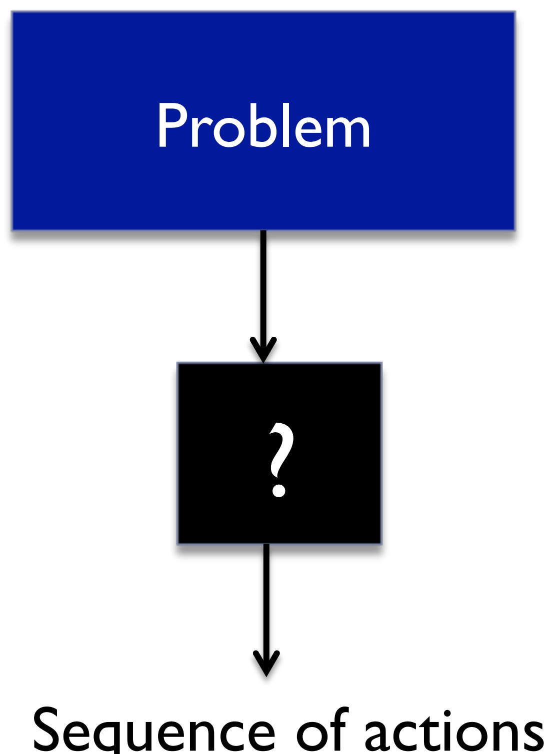
State space models: deterministic search

Devika Subramanian

A class of AI problems

- For many decision making problems, the answer is in the form of a sequence of steps or actions.



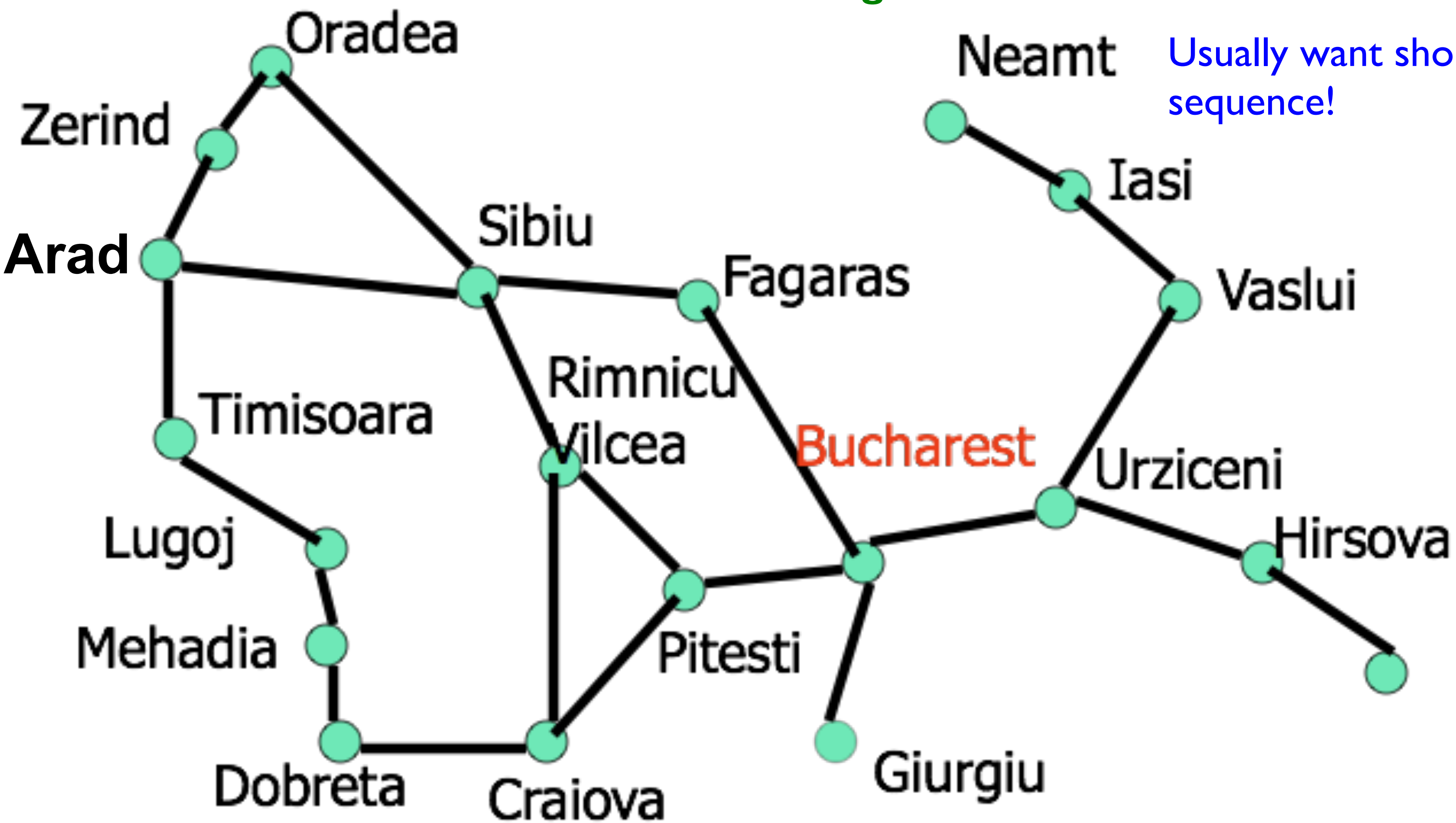
Typically we have constraints or preferences on that sequence of actions: shortest, fastest, or best according to some defined criteria

Route finding

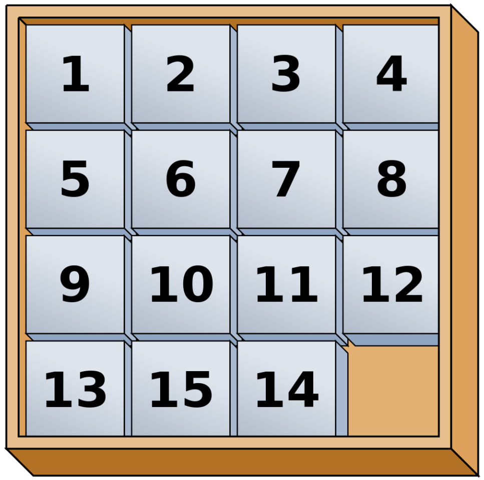
Problem: How do I get from Arad to Bucharest?

Answer: Arad → Sibiu,
Sibiu → Fagaras,
Fagaras → Bucharest

Usually want shortest sequence!



More puzzles



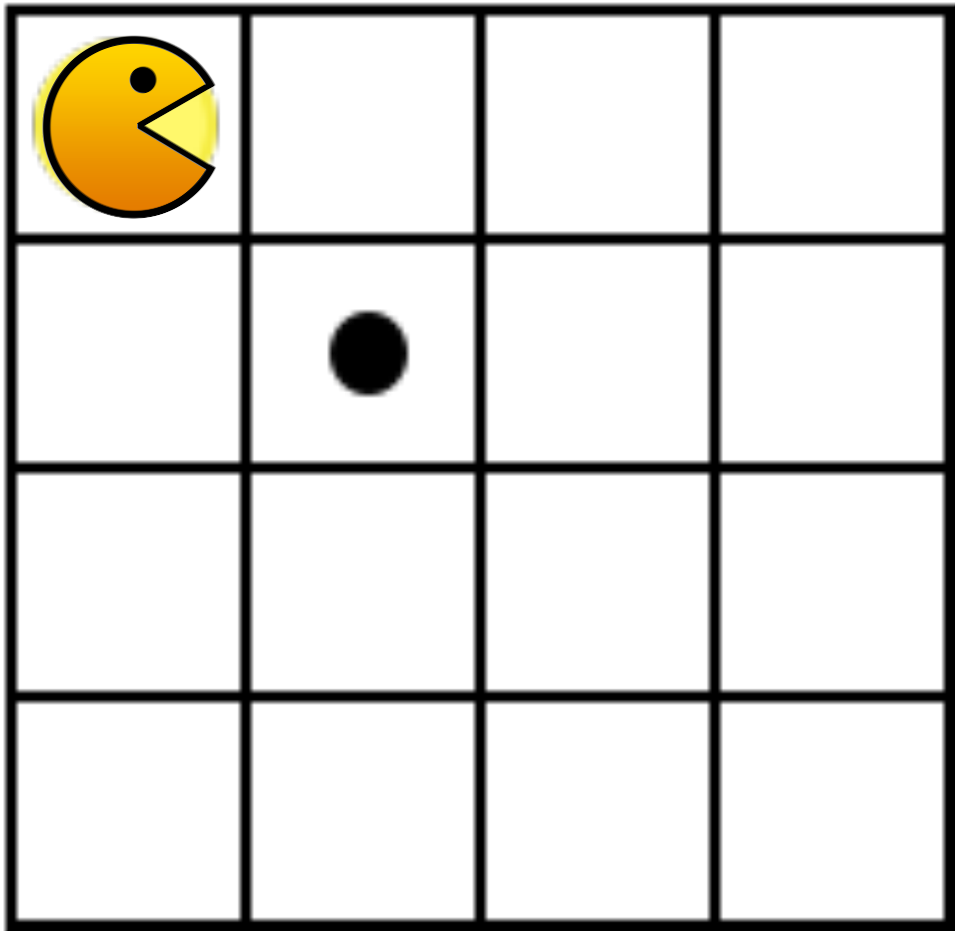
Problem: how can we get the board/cube into a desired configuration?

Answer: here is the sequence of moves…

Usually want shortest sequence!

Pacman movement planning

Problem: how can we get pacman to the food dot?



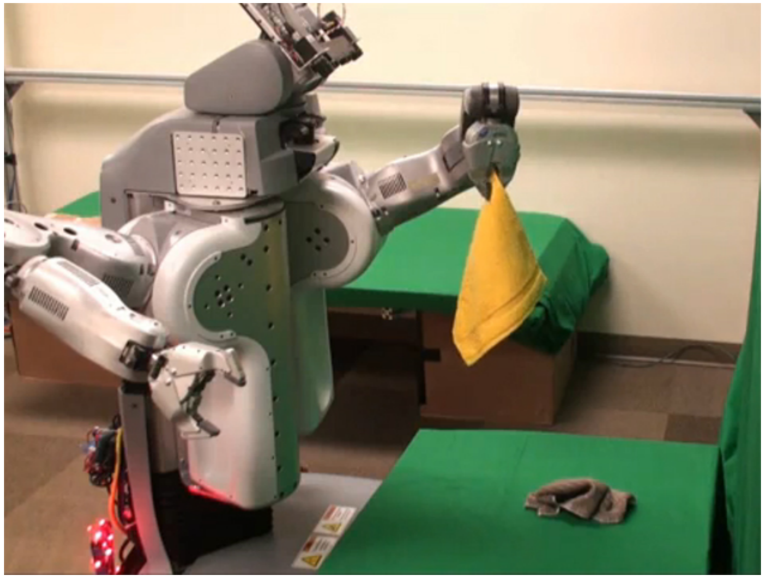
Answer: here is a sequence of moves. east, south

Usually want shortest sequence!

A broad range of logistics problems fit into this genre.

Towel folding

- Problem: how can the robot fold this towel?



Usually want shortest sequence or the safest sequence!

- Answer: follow this sequence of moves: rotate-joint-23-by-15-degrees, move-arm-by-4-cm,….

Machine translation

- Problem: translate the German phrase *eine kleine Nachtmusik* into English

- Answer: replace each German word by the corresponding English word. *eine* → a, *kleine* → little, *NachtMusik* → night music

Usually want the most fluent sequence!

Course planning

- Problem: how do I get a BS in CS at Rice?

Year	Autumn Term	Credits	Winter Term	Credits	Spring Term	Credits
1st year	Math *	5	Math	5	Math	5
	Chem 142	5	Chem 152	5	Chem 162	5
	English Composition	5	VLPA / I&S	5	Ocean 200	3
	Ocean 100	1			Ocean 201	2
	1st Year Credits	**16**		**15**		**15**
2nd Year	Ocean 210	3	Biol 180	5	Biol 200	5
	Physics *	5	Physics	5	Physics	5
	ESS 210, 211 or 212	5	VLPA / I&S	5	Ocean 220	3/5
	Electives	2				
	2nd Year Credits	**15**		**15**		**13/15**
3rd Year	Ocean 410	4	Ocean 400	4	Ocean 401	3
	Ocean 430	4	Ocean 420	4	U / D Science	5
	U / D Science	5	VLPA / I&S	5	VLPA / I&S	5
	Electives	2	Electives	2	Electives	2
	3rd Year Credits	**15**		**15**		**15**
4th Year	Ocean 4xx	3	Ocean 443	3	Ocean 444	5
	U / D Science	5	U / D Science	5	VLPA / I&S	5
	VLPA / I&S	5	Electives	6	Electives	5
	Electives	2				
	4th Year Credits	**15**		**14**		**15**

- Answer: take comp140, then comp182, then comp215,…

What is common among all these examples?

- Answer/output is a sequence of atomic actions.
- There is a set of atomic actions (discrete or continuous) that is available and we need to string elements of that set together to construct a solution sequence.
- We have functions that can calculate the quality of a sequence, and our objective is to find the best sequence according to that objective function.
- Generally, there are O(n!) action sequences of length n; so we need clever ways to look for solution sequences of high quality.

Newell/Simon assertion

- Any AI problem can be framed as a search for a sequence of moves or actions in some space!
- 1975 Turing Award lecture
  - http://doi.library.cmu.edu/10.1184/pmc/newell/box00048/fld04143/bdl0001/doc0001

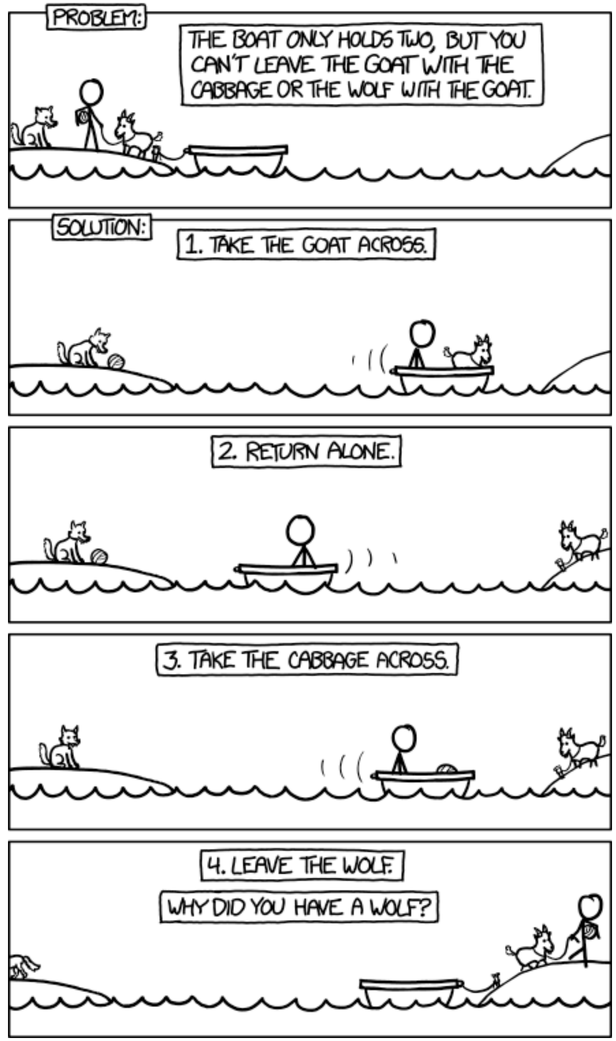
A puzzle

- Problem: A farmer wants to get his cabbage, goat, wolf across a river. He has a boat that only holds two. He cannot leave cabbage and goat alone or the goat and wolf alone. How many river crossings does he need?

  http://coolmath-games.com/Logic-wolfsheepcabbage/index.html

- Answer: a sequence of crossing actions (length of that sequence)

xkcd's take on puzzle



Modeling the problem

Problem

State-space model

Sequence of actions

Start state

Sequence of actions

Goal state

Rolling out a what-if tree

FGWC||

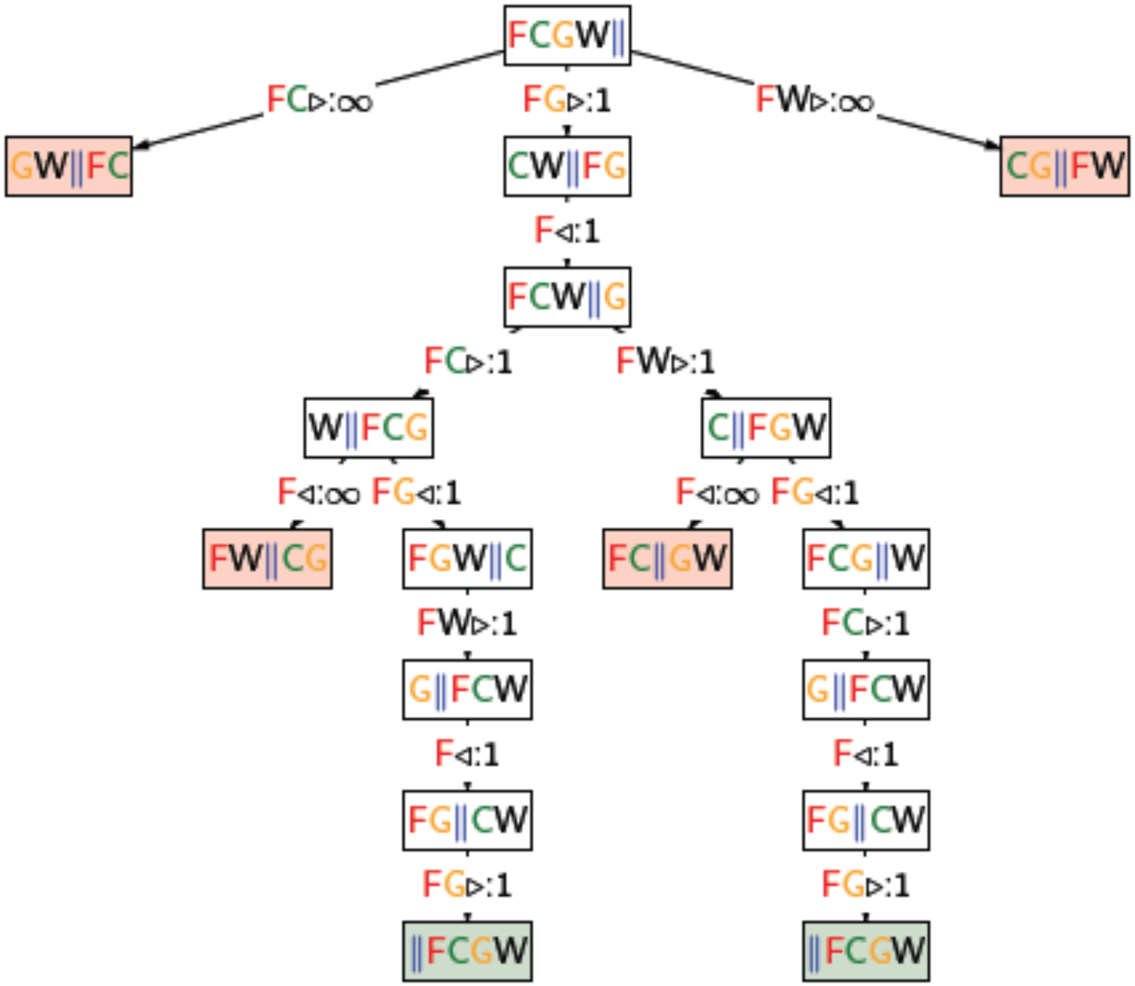
FC→ FG→ FW→

GW||FC CW||FG CG||FW

F←

FCW||G

The full state space search



Slide from P. Liang

The state space model

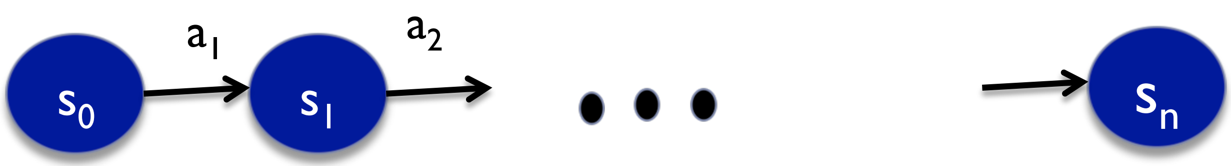
- Ingredients
  - A discrete set S of states
  - A discrete set A of actions
- Actions(s): a set of actions that are available in state s
- Successor(s,a): state which results from doing action a in state s
- Cost(s,a): cost of doing action a in state s
- s_{start}: the initial state
- isGoal(s): is state s a goal state? A solution to a problem modeled in the state space framework is a sequence of actions in A that maps the initial state to a state that satisfies the goal test.

Solving a state space problem

- The cost of a sequence $a_1,\ldots,a_n$ to get from start state s_0 to a goal state is:

$$\text{scost}(a_1,\ldots,a_n) = \sum_{i=1}^{n} \text{cost}(s_{i-1}, a_i)$$

$$s_i = successor(s_{i-1}, a_i)$$

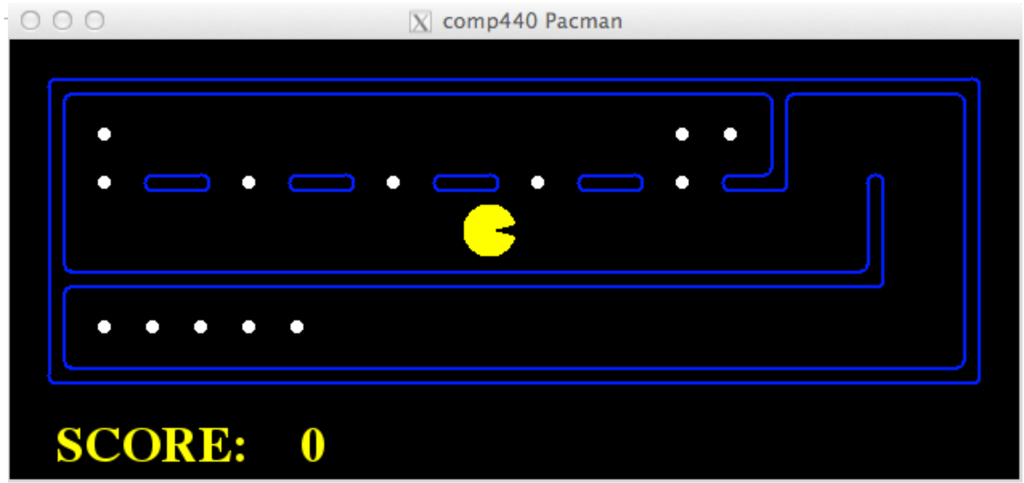


- Find a sequence of actions with minimal cost

State space models are abstractions

- States in the real world have a lot of complexity to them.
- Formulating a problem as a search problem requires identification of aspects of the world that are relevant to the problem.
- This is a very creative process and requires considerable human ingenuity and judgment!

Example of state space construction



- Problem: path finding
  - States: (x,y) location
  - Actions: NSEW
  - Successor: update location
  - Start state: start location
  - Goal test: (x,y)== end
- Problem: eat all dots as quickly as possible
  - States: {(x,y), boolean matrix for dots}
  - Actions: NSEW
  - Successor: update location and boolean matrix for dots
  - Start state: start location
  - Goal test: all dots are false

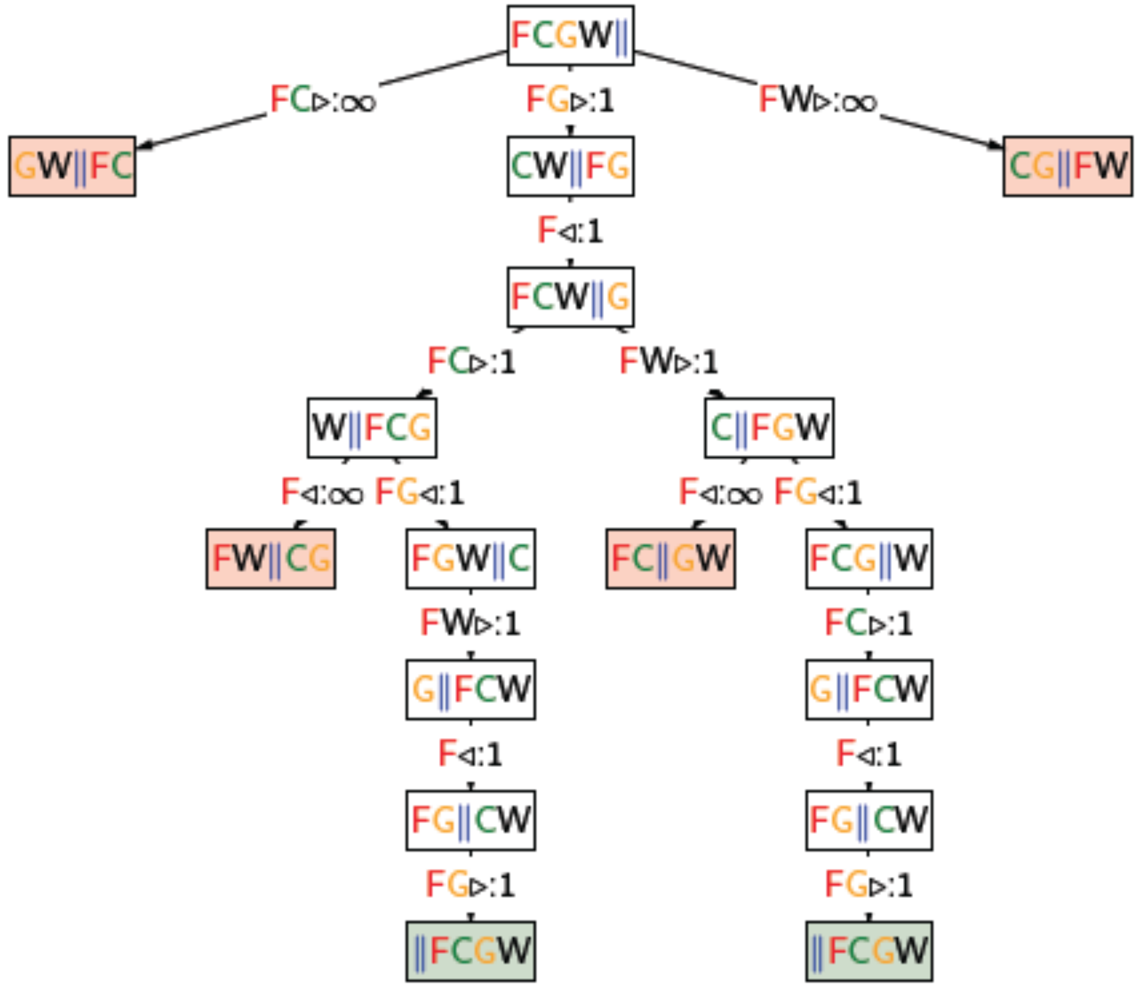
Estimating state space sizes

- Pacman board
  - Open locations: 100
  - Any open location could have a dot
  - Agent direction: NSEW (which way pacman is facing)
- How many states?
  - Full state space: $100 \times 2^{100} \times 4$
  - For path finding: ??
  - For eating all dots: ??

State space graphs

- A state space graph is a mathematical representation of a state space model of a problem.
  - Nodes of graph are abstract world states
  - Edges of graph represent actions that lead to successor states
  - Start state is a node in the state space
  - Goal test is a set of nodes satisfying the goal test
- In a search graph, each state occurs only once
- For non-toy problems, we do not build the state space graph in practice, but it is a useful theoretical concept.

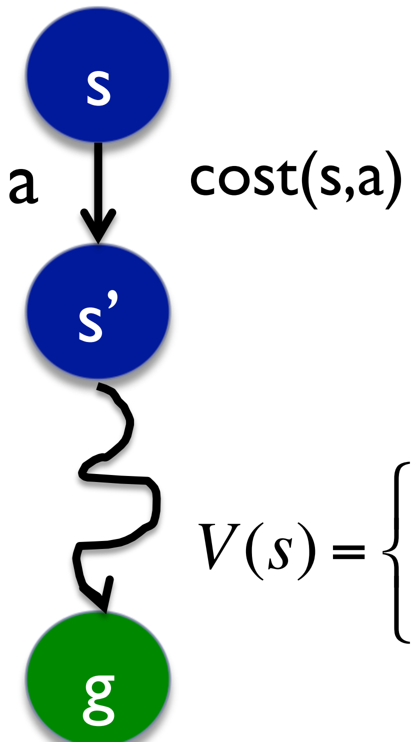
The state space graph for puzzle



Slide from P. Liang

Finding least cost solutions

- Let V(s) = least cost solution from state s (cost of best action sequence from state s to a goal state in S)



$$V(s) = \begin{cases} 0 \text{ if } s \text{ is a goal state} \\ \min_{a \in A} \text{cost}(s,a) + V(\mathit{successor}(s,a)) \text{ otherwise} \end{cases}$$

Dynamic programming

- Compute the recurrence on V(s)

$$V(s) = \begin{cases} 0 \text{ if } s \text{ is a goal state} \\ \min_{a \in A} \text{cost}(s,a) + V(successor(s,a)) \text{ otherwise} \end{cases}$$

- $V_0(s) = 0$ for all states s in S
- Repeat
  - for every non-goal state s in S:
    - $V_{t+1}(s) = \min_{a \text{ in } A} \text{cost}(s,a) + V_t(\text{successor}(s,a))$
- Until $V_t \approx V_{t+1}$

Bellman-Ford algorithm

Complexity of DP

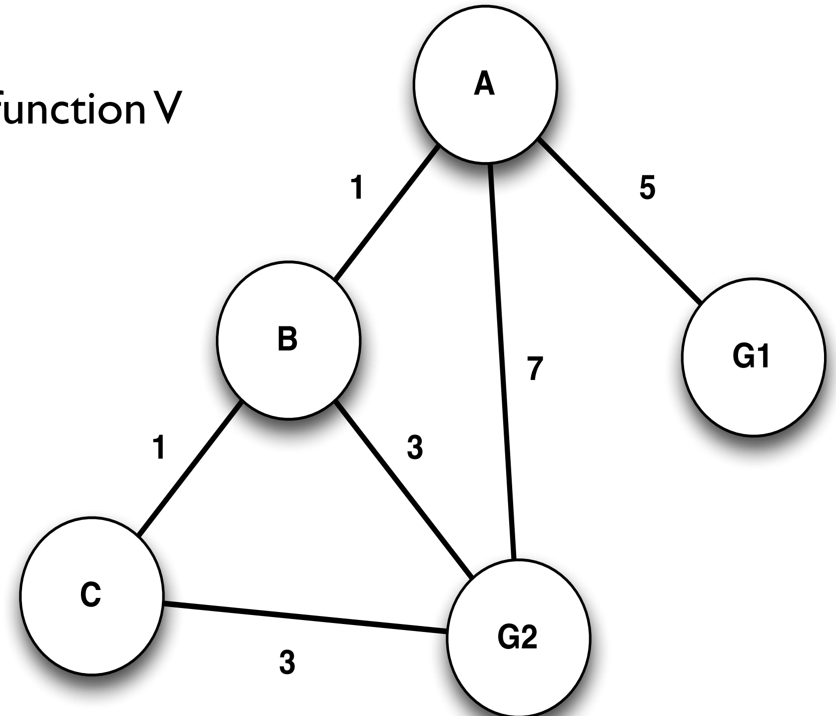
- For a graph with n nodes
  - Space complexity: $O(n)$
  - Time complexity: $O(n^2|A|)$

- Practical only for toy problems

DP example

State	V_0	V_1	V_2	V_3	V_4	V_5
A	0	1	2	3	4	4
B	0	1	2	3	3	3
C	0	1	2	3	3	3
G1	0	0	0	0	0	0
G2	0	0	0	0	0	0

Value function V



State	a_1	a_2	a_3	a_4	a_5
A	B	B	B	B	B
B	A, C	A, C	A, C, G2	G2	G2
C	B	B	B, G2	G2	G2

Best action

Best path from A: A→B→G2
Best path from B: B→G2
Best path from C: C→G2