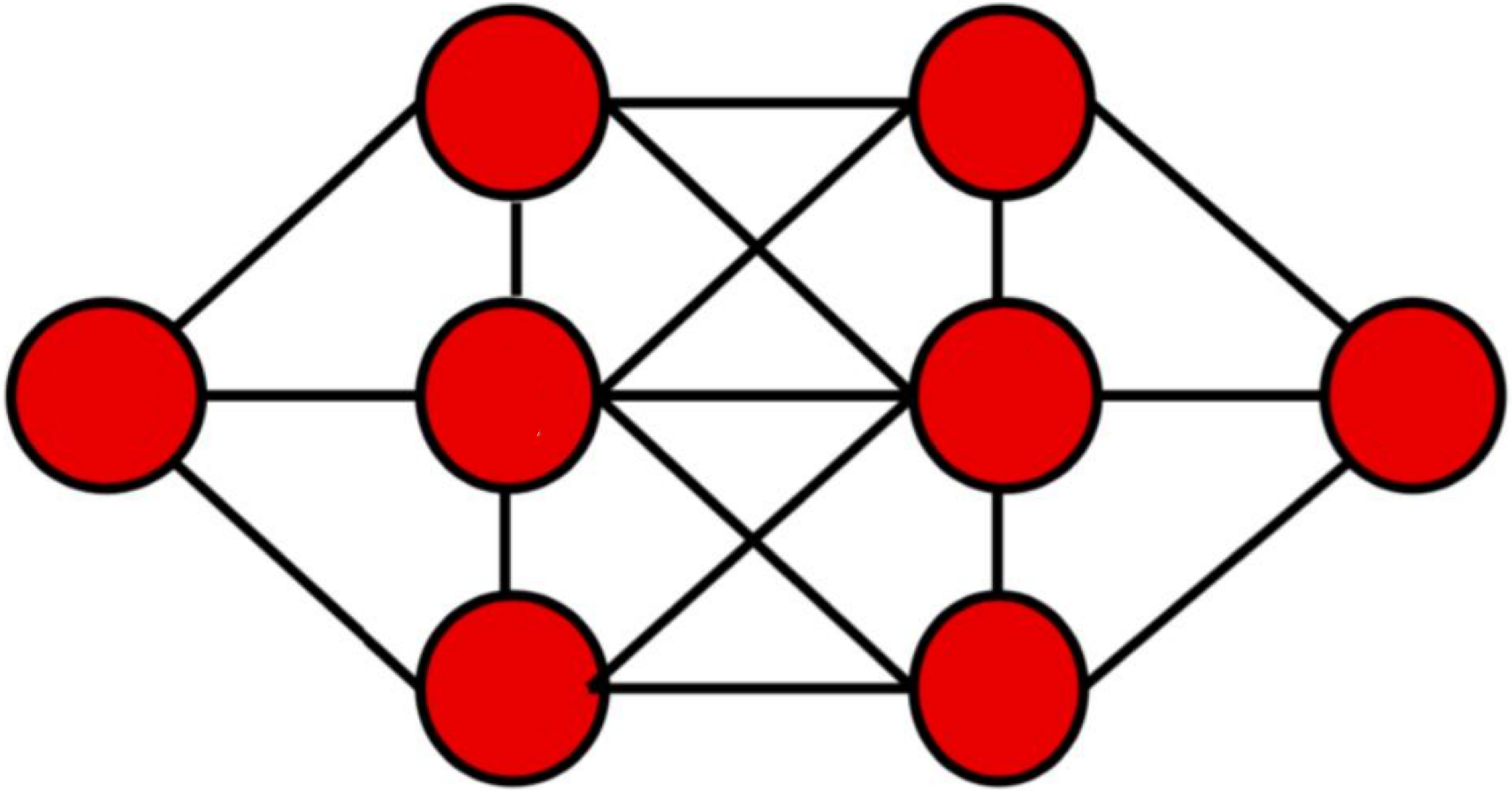
Introduction to Management Information Systems

Artificial Intelligence

3 lessons

1. AI background
2. How AI works
3. Intelligent Systems

add A to H but no two letters should be beside the nest alphabetical letter



Artificial Intelligence Background

Definitions

“Intelligence: The ability to learn and solve problems”

Webster’s Dictionary

“Artificial intelligence (AI) is the intelligence exhibited by machines or software’

Wikipedia

Definitions

"The study and design of intelligent agents, where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success."

Russel and Norvig AI book

"Just as the Industrial Revolution freed up a lot of humanity from physical drudgery, I think AI has the potential to free up humanity from a lot of the mental drudgery."

Andrew Ng

What is AI?

Thinking rationally

- mental process – use computational models
- use maths and logic
- Codify “right thinking” with logic

Acting rationally

- intelligent agents
- A rational agent is one that acts so as to achieve the best outcome, or when there is uncertainty, the best expected outcome.

What is AI?

Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

The textbook advocates "acting rationally“

What is AI?

Thinking humanly

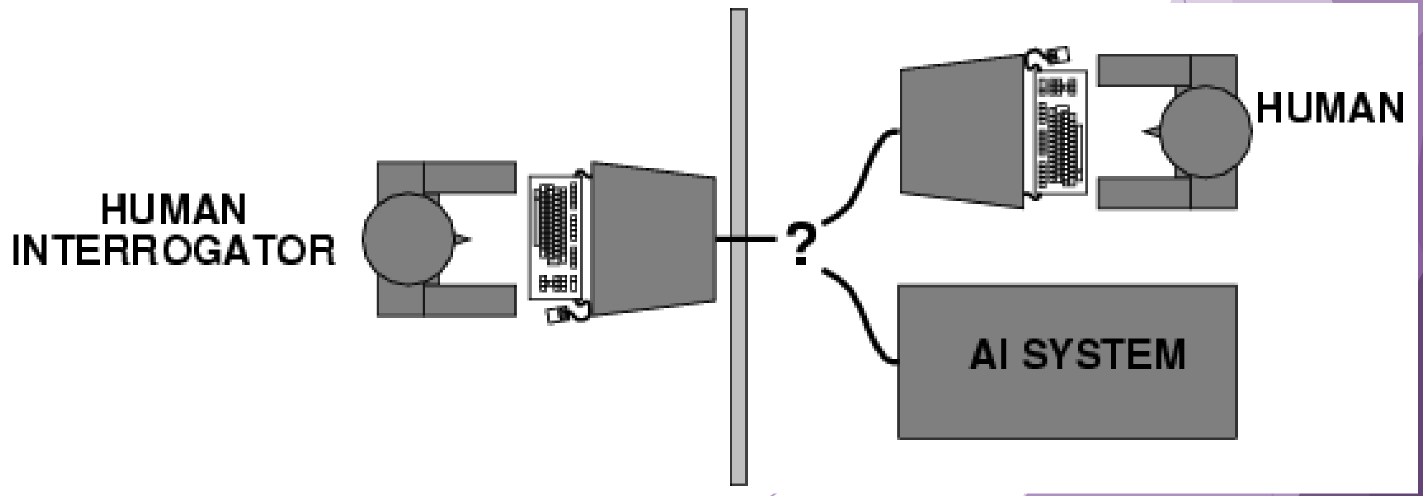
- computers to ‘think’, machines with minds
- cognitive science / approach
- how do humans think?

Acting humanly

- get computers to do things that humans are currently better at doing
- Flying – planes can, but do not mimic a bird
- Major components of AI: knowledge, reasoning, language, understanding, learningTuring test (1950)
- .

Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence":
- "Can machines think?" → "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game
- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in following 50 years



The Turing Test

(Can Machine think? A. M. Turing, 1950)

- Requires:
  - Natural language
  - Knowledge representation
  - Automated reasoning
  - Machine learning
  - (vision, robotics) for full test

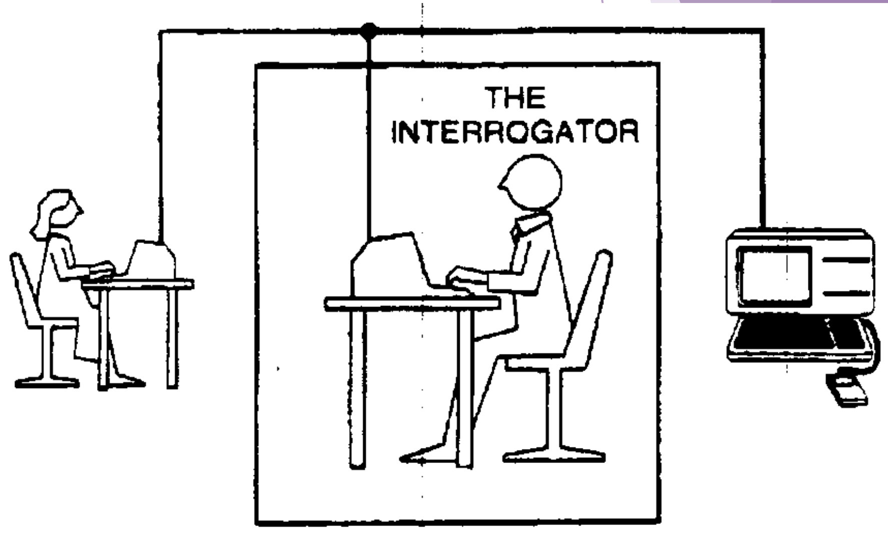


Figure 1.1 The Turing test.

Artificial Intelligence History (pre-LLMs)

History of AI

- **1940-1950**: Gestation of AI
  - McCulloch & Pitts: Boolean circuit to model of brain
  - Turing's Computing Machinery and Intelligence
    http://www.turingarchive.org/browse.php/B/9
- **1950-1970**: Early enthusiasm, great expectations
  - Early AI programs, Samuel's checkers program
  - Birth of AI @ Dartmouth meeting 1956.
  - Check out the MIT video "The thinking Machine" on youtube
    https://www.youtube.com/watch?v=aygSMgK3BEM
- **1970-1990**: Knowledge-based AI
  - Expert systems, AI becomes an industry
  - AI winter

History of AI

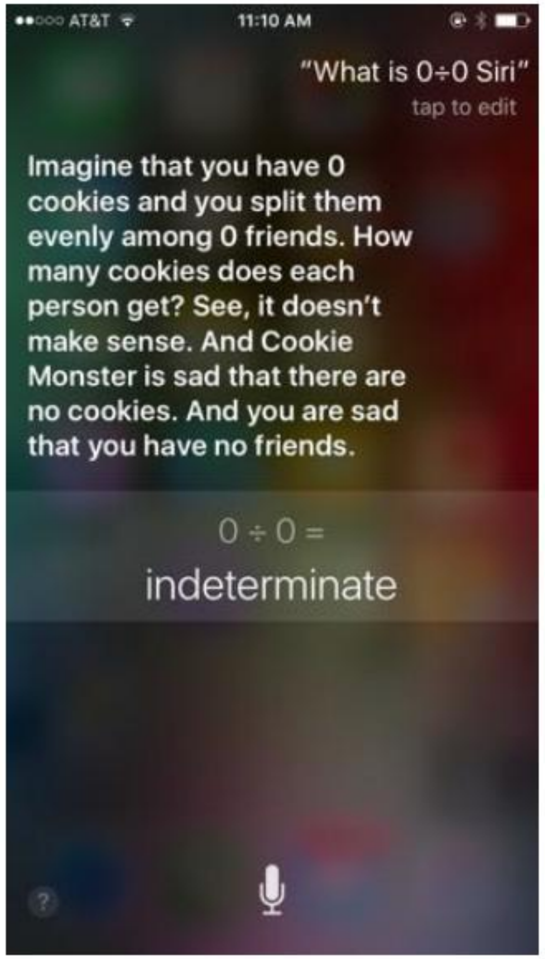
- **1990-present**: Scientific approaches
  - Neural Networks: le retour
  - The emergence of intelligent agents
  - AI becomes "scientific", use of probability to model uncertainty
  - AI Spring!
  - The availability of very large datasets.
    - Data will drive future discoveries and alleviate the complexity in AI.

Artificial Intelligence Application (pre-LLMs)

Applications of AI

Speech recognition

- Virtual assistants: Siri (Apple), Echo (Amazon), Google Now, Cortana (Microsoft).
- "They" helps get things done: send an email, make an appointment, find a restaurant, tell you the weather and more.
- Leverage deep neural networks to handle **speech recognition** and **natural language understanding**.


"What is 0÷0 Siri"
tap to edit

Imagine that you have 0 cookies and you split them evenly among 0 friends. How many cookies does each person get? See, it doesn't make sense. And Cookie Monster is sad that there are no cookies. And you are sad that you have no friends.

0 ÷ 0 =
indeterminate

Applications of AI

Handwriting recognition (check, zipcode)

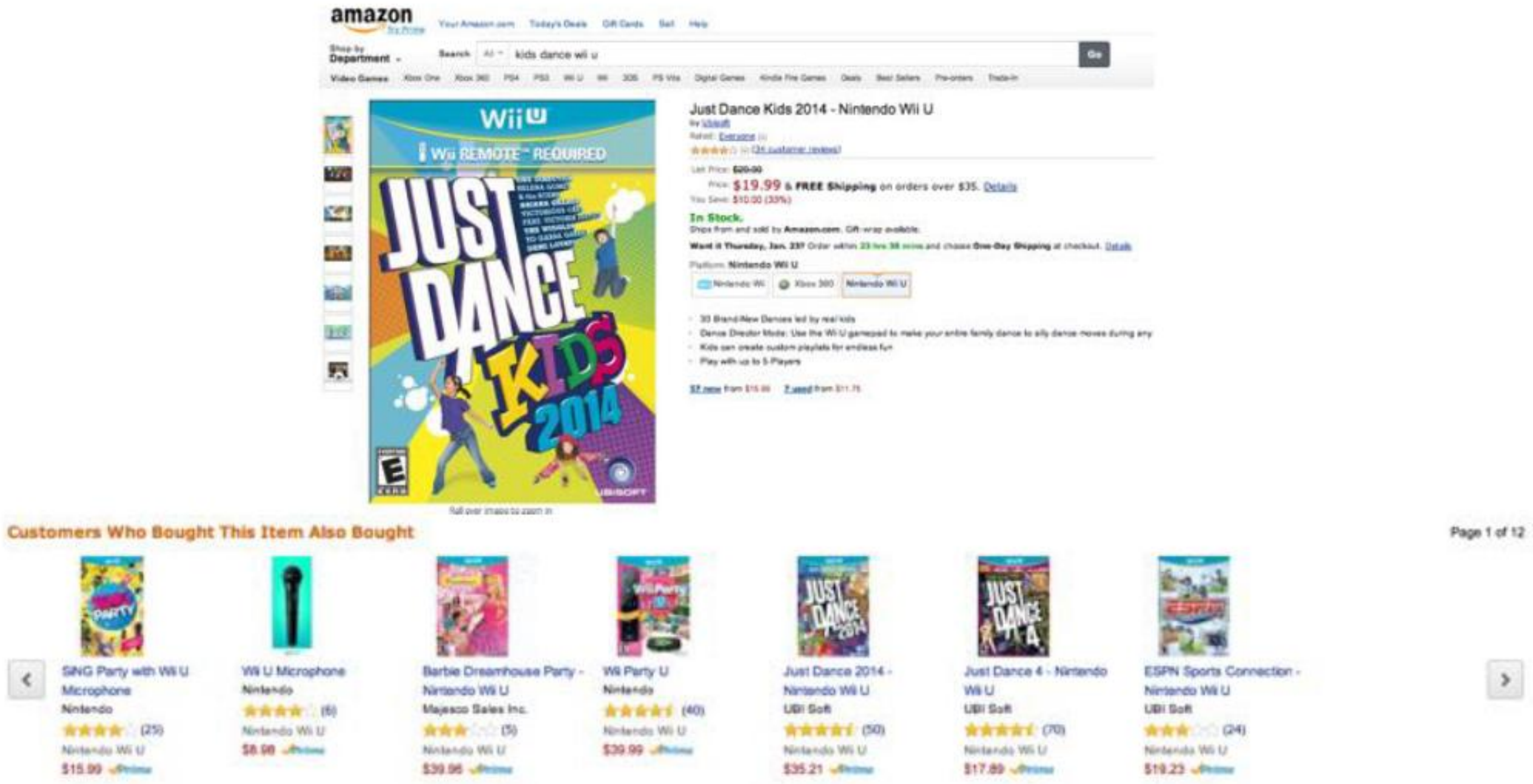
Applications of AI

Robotics: Awesome robots today! NAO, ASIMO, and more!

Credit: By Momotarou2012, via Wikimedia Commons.

Applications of AI

Recommendation systems (collaborative filtering)



Just Dance Kids 2014 - Nintendo Wii U

Price: $19.99 & FREE Shipping on orders over $35. Details

You Save: $10.00 (33%)

In Stock.

Ships from and sold by Amazon.com. Gift-wrap available.

Platform: Nintendo Wii U

- 33 Brand-New Dances led by real kids
- Dance Director Mode: Use the Wii U gamepad to make your entire family dance to silly dance moves during any
- Kids can create custom playlists for endless fun
- Play with up to 5 Players

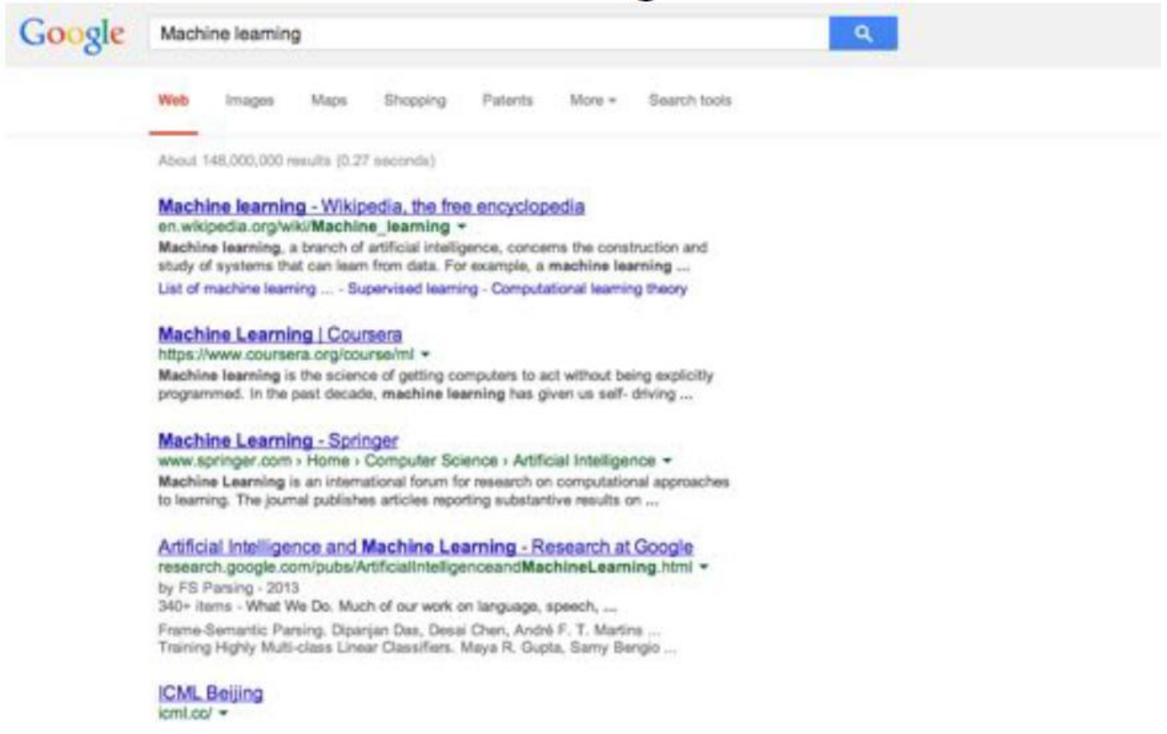
Customers Who Bought This Item Also Bought

Page 1 of 12

SiNG Party with Wii U Microphone	Wii U Microphone	Barbie Dreamhouse Party - Nintendo Wii U	Wii Party U	Just Dance 2014 - Nintendo Wii U	Just Dance 4 - Nintendo Wii U	ESPN Sports Connection - Nintendo Wii U
Nintendo	Nintendo	Majesco Sales Inc.	Nintendo	UBI Soft	UBI Soft	UBI Soft
(25)	(6)	(5)	(40)	(50)	(70)	(24)
Nintendo Wii U	Nintendo Wii U	Nintendo Wii U	Nintendo Wii U	Nintendo Wii U	Nintendo Wii U	Nintendo Wii U
$15.99	$8.98	$39.96	$39.99	$35.21	$17.89	$19.23

Applications of AI

Search engines



About 148,000,000 results (0.27 seconds)

Machine learning - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Machine_learning
Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. For example, a **machine learning** ...
List of machine learning ... - Supervised learning - Computational learning theory

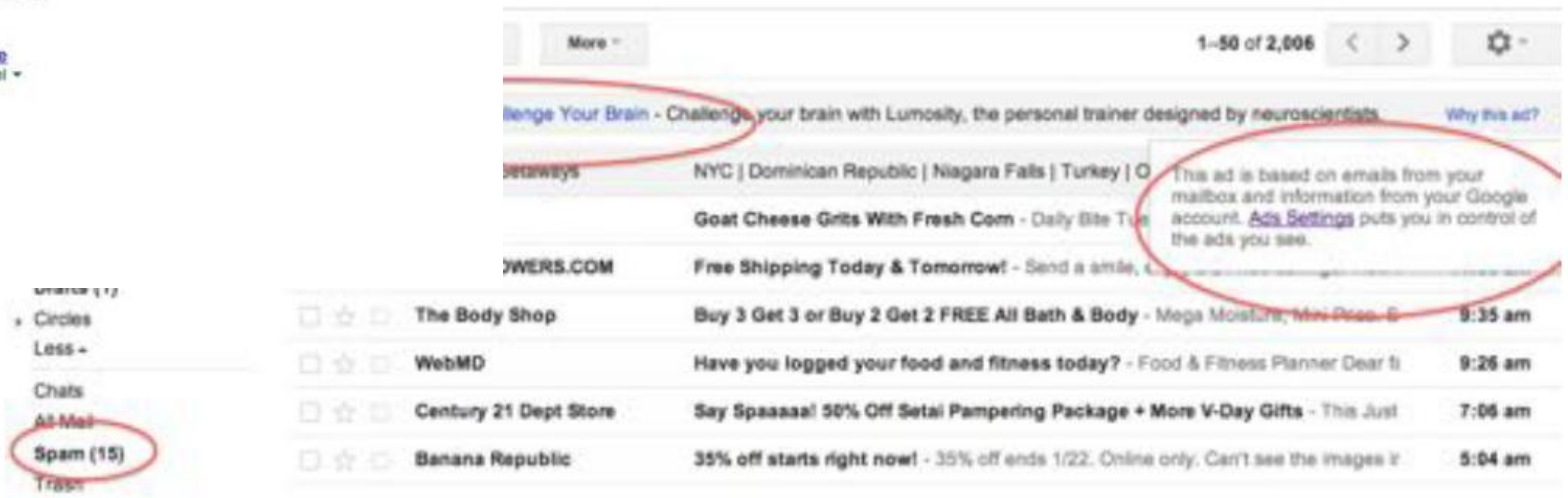
Machine Learning | Coursera
https://www.coursera.org/course/ml
Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, **machine learning** has given us self- driving ...

Machine Learning - Springer
www.springer.com › Home › Computer Science › Artificial Intelligence
Machine Learning is an international forum for research on computational approaches to learning. The journal publishes articles reporting substantive results on ...

Artificial Intelligence and Machine Learning - Research at Google
research.google.com/pubs/ArtificialIntelligenceandMachineLearning.html
by FS Parsing - 2013
340+ items - What We Do. Much of our work on language, speech, ...
Frame-Semantic Parsing. Dipanjan Das, Desai Chen, André F. T. Martins ...
Training Highly Multi-class Linear Classifiers. Maya R. Gupta, Samy Bengio ...

ICML Beijing
icml.cc/

Email



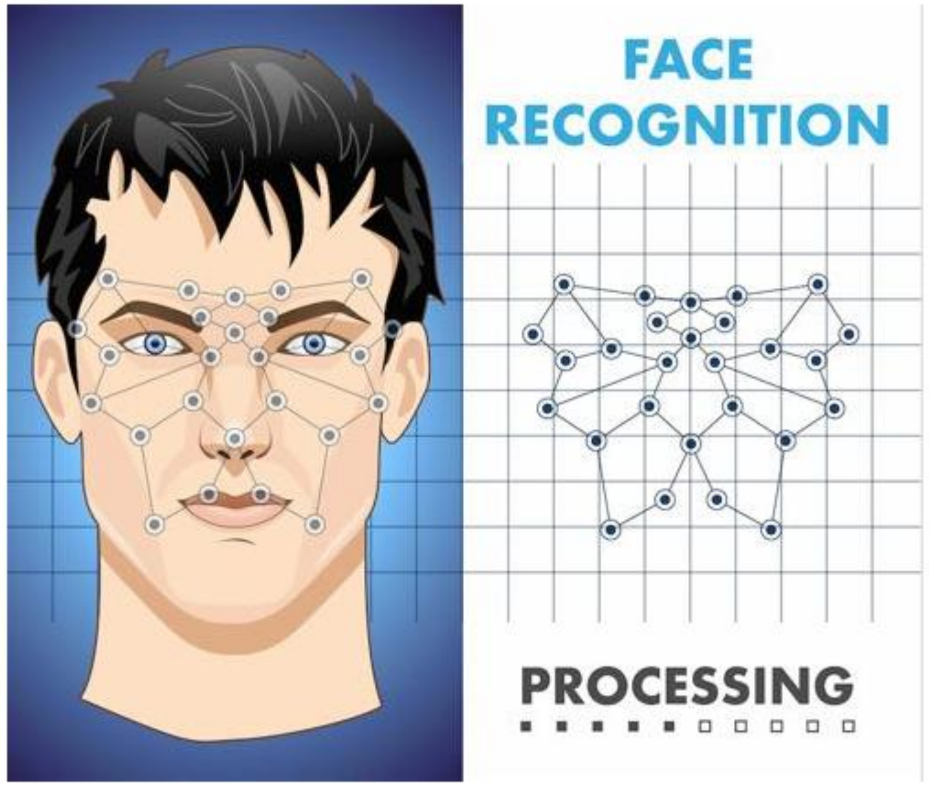
llenge Your Brain - Challenge your brain with Lumosity, the personal trainer designed by neuroscientists   Why this ad?

This ad is based on emails from your mailbox and information from your Google account. Ads Settings puts you in control of the ads you see.

etaways	NYC | Dominican Republic | Niagara Falls | Turkey | O	
	Goat Cheese Grits With Fresh Corn - Daily Bite T	
WERS.COM	**Free Shipping Today & Tomorrow!** - Send a smile,	
The Body Shop	**Buy 3 Get 3 or Buy 2 Get 2 FREE All Bath & Body** - Mega Moisture, Mini Price. S	9:35 am
WebMD	**Have you logged your food and fitness today?** - Food & Fitness Planner Dear fi	9:26 am
Century 21 Dept Store	**Say Spaaaaa! 50% Off Setai Pampering Package + More V-Day Gifts** - This Just	7:06 am
Banana Republic	**35% off starts right now!** - 35% off ends 1/22. Online only. Can't see the images ir	5:04 am

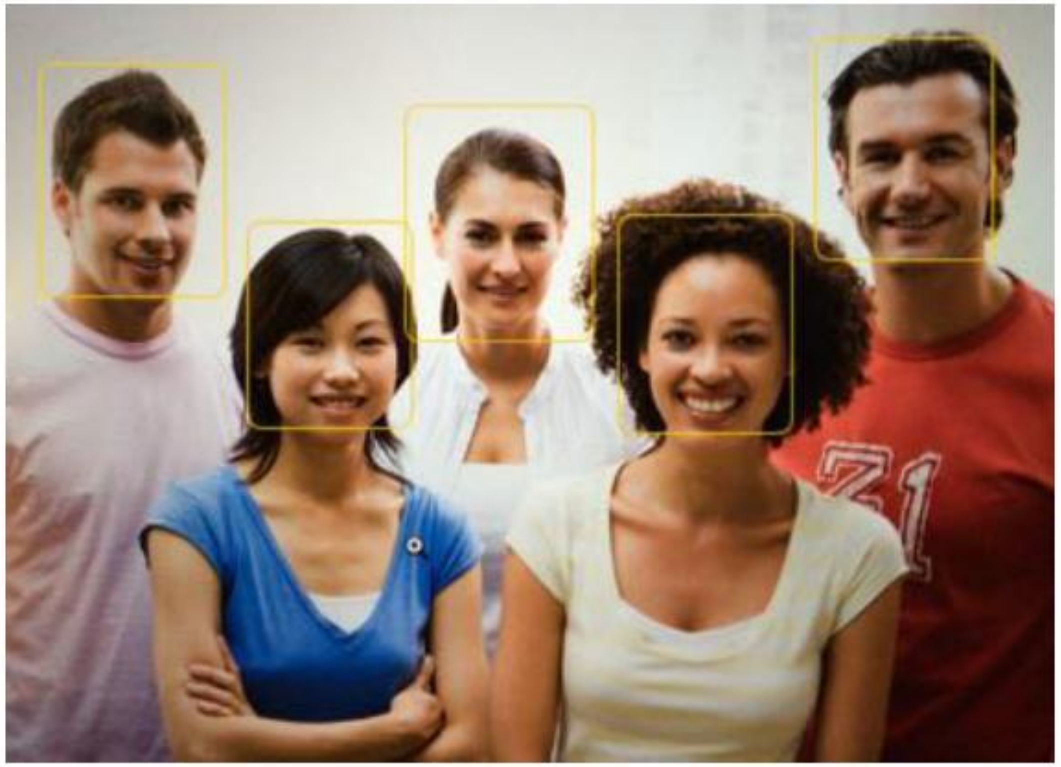
Applications of AI

Face recognition



Applications of AI

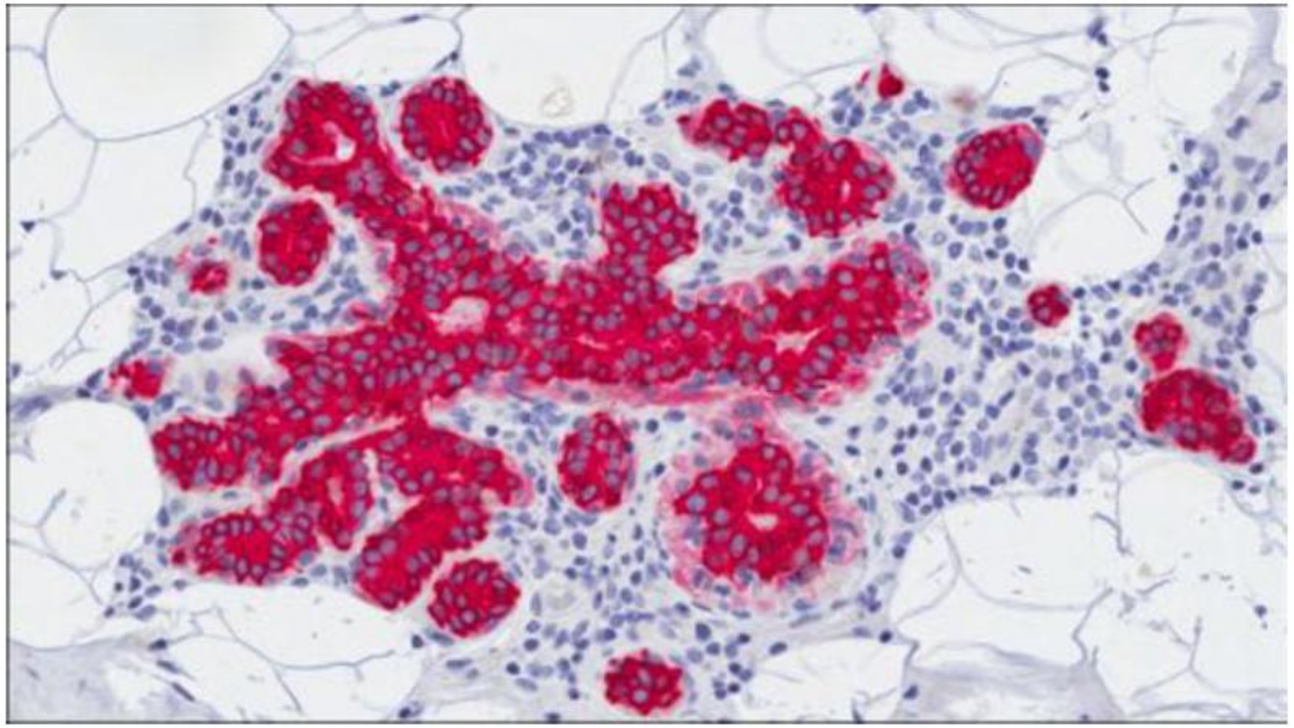
Face detection



Viola-Jones method.

Applications of AI

Detection of breast cancer in mammography images



Applications of AI

Chess (1997): Kasparov vs. IBM Deep Blue

(Left) Copyright 2007, S.M.S.I., Inc. - Owen Williams, The Kasparov Agency, via Wikimedia Commons (Right) By James the photographer, via Wikimedia Commons

Powerful search algorithms!

Applications of AI

Jeopardy! (2011): Humans vs. IBM Watson

By Rosemaryetoufee (Own work), via Wikimedia Commons

Natural Language Understanding and information extraction!

Applications of AI

Go (2016): Lee Sedol versus Google AlphaGo

AlphaGo

(Left) By LG Electronics, via Wikimedia Commons (Right) By Google DeepMind, via Wikimedia Commons

Deep Learning, reinforcement learning, and search algorithms!

Applications of AI

Autonomous driving

By User Spaceape on en.wikipedia, via Wikimedia Commons

- DARPA Grand Challenge
  - 2005: 132 miles
  - 2007: Urban challenge
  - 2009: Google self-driving car

State of the Art

- Speech recognition
- Autonomous planning and scheduling
- Financial forecasting
- Game playing, video games
- Spam fighting
- Logistics planning
- Robotics (household, surgery, navigation)
- Machine translation
- Information extraction
- VLSI layout
- Automatic assembly
- Sentiment analysis
- Fraud detection
- Recommendation systems
- Web search engines
- Autonomous cars
- Energy optimization
- Question answering systems
- Social network analysis
- Medical diagnosis, imaging
- Route finding
- Traveling salesperson
- Protein design
- Document summarization
- Transportation/scheduling
- Computer animation

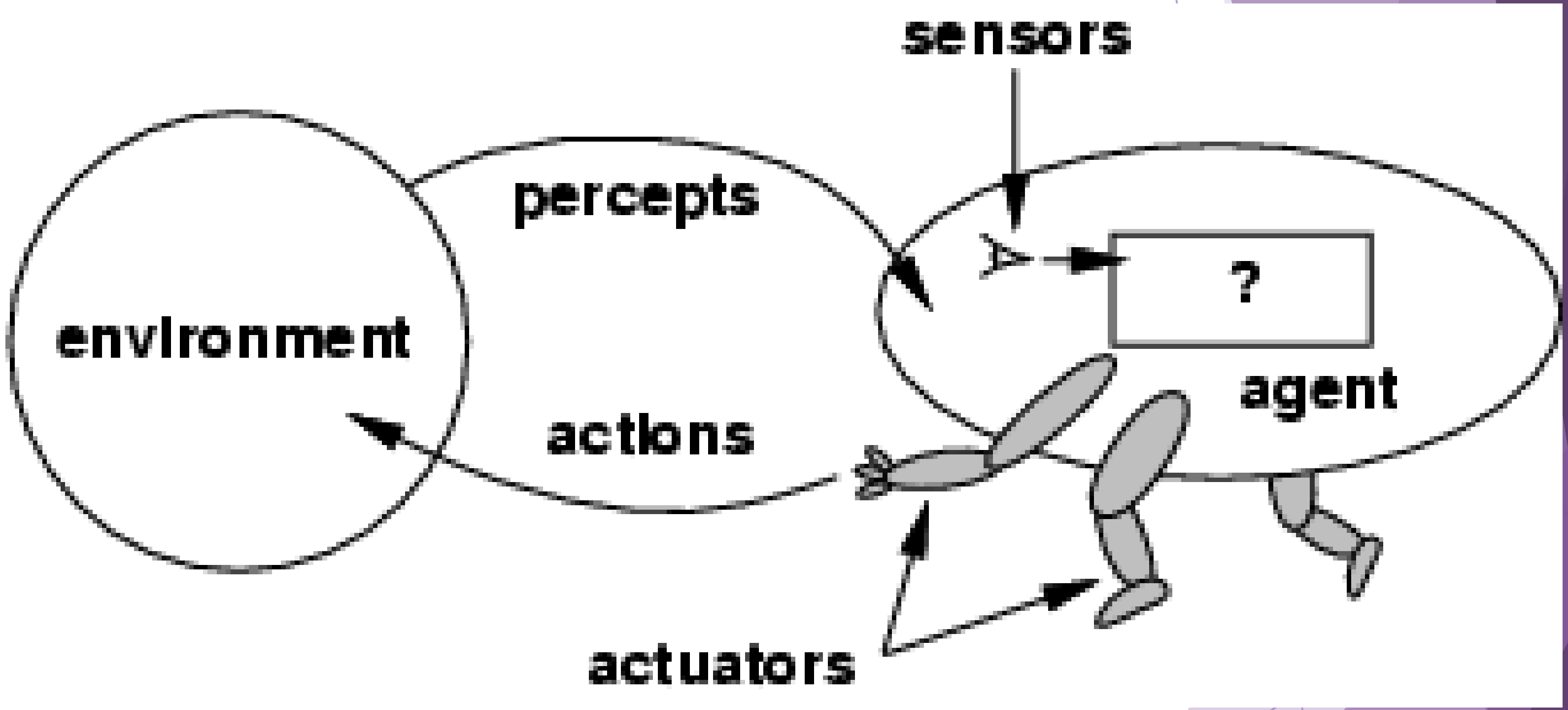
What is Artificial Intelligence and what are agents?

Agents

- An agent is anything that can be viewed as
  - perceiving its environment through sensors and
  - acting upon that environment through actuators

Agents

- An agent
  1. perceives
  2. thinks
  3. acts



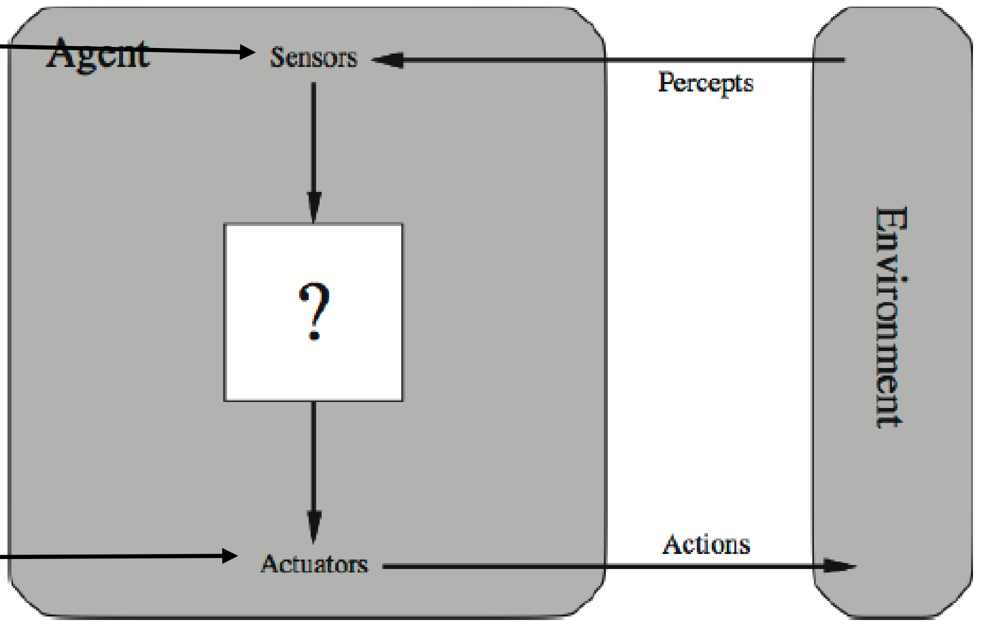
Agents

- A cycle or loop

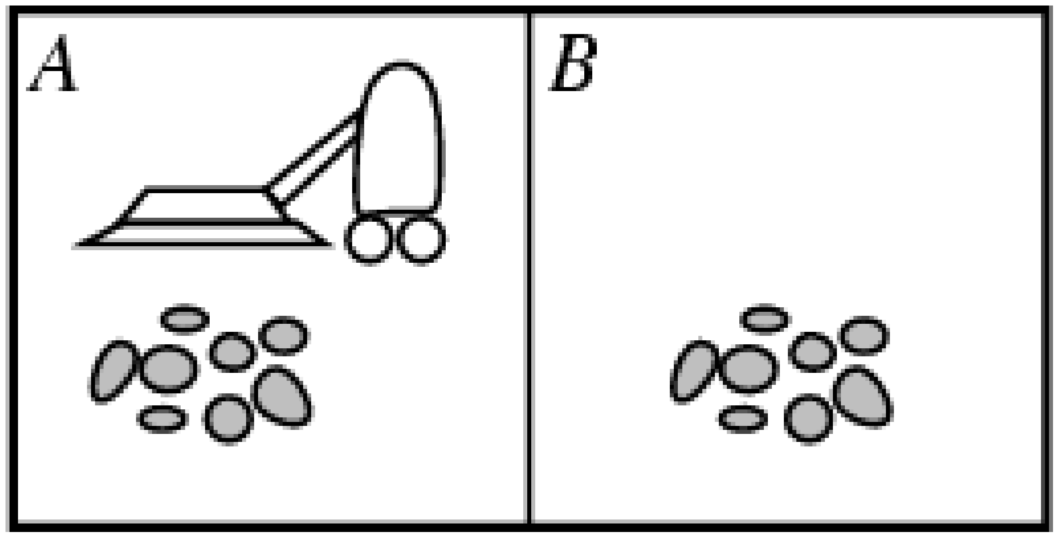
1. sense the world

2. thinking? deciding?

3. acting



Vacuum-cleaner world



- Percepts: location and contents, e.g., [A,Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*
- Agent function: mapping from percepts to actions.

PEAS

PEAS:

- Performance measure,
- Environment,
- Actuators,
- Sensors

PEAS

- automated taxi driver:
  - Performance measure:
    - Safety, fast, legal, comfortable trip, maximize profits
  - Environment:
    - Roads, other traffic, pedestrians, customers
  - Actuators:
    - Steering wheel, accelerator, brake, signal, horn
  - Sensors:
    - Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard, other

PEAS

Agent: automatic vacuum cleaner (Roomba)

- Performance measure: cleanness, distance, security, efficiency, battery
- Environment: room, objects in room
- Actuators: wheels, brushes, etc.
- Sensors: dirt, cliff, bump, infrared wall, and camera

http://www.irobot.co.uk/Home-Robots/Vacuuming

Environment types

Deterministic (vs. stochastic):

- Deterministic - The next state of the environment is completely determined by the current state and the action executed by the agent.
- Strategic - If the environment is deterministic except for the actions of other agents
- Stochastic - The next state of the environment is not determined solely by the current state and the action executed by the agent.
  - Non-deterministic
  - Probabilistic

Environment types

Episodic (sequential):

- The agent's experience is divided into atomic "episodes"
- each episode consists of the agent perceiving and then performing a single action, and
- the choice of action in each episode depends only on the episode itself.

Environment types

Static (vs. dynamic):

- The environment is unchanged while an agent is deliberating.
- The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does

Environment types

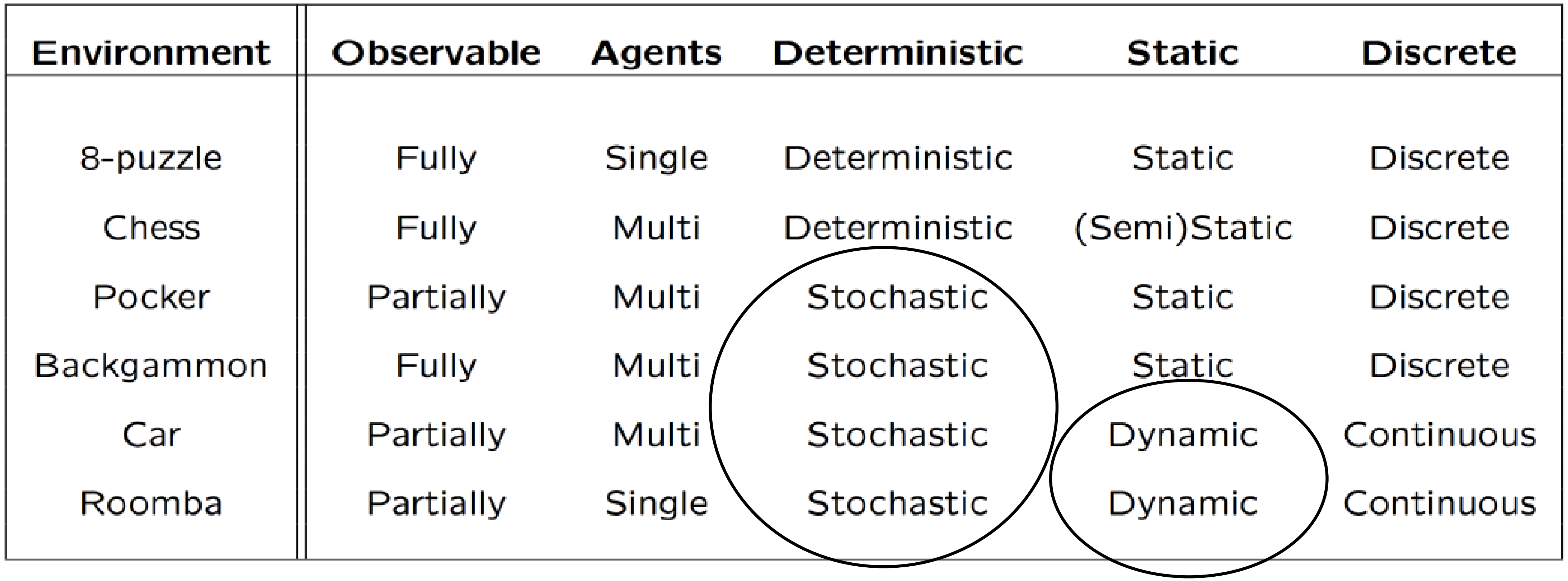
Discrete (vs. continuous):

- A limited number of distinct, clearly defined percepts and actions.
- e.g. checkers/draughts v self-driven car

Single agent (vs. multi-agent):

- An agent operating by itself in an environment.

Environment types



Environment	Observable	Agents	Deterministic	Static	Discrete
8-puzzle	Fully	Single	Deterministic	Static	Discrete
Chess	Fully	Multi	Deterministic	(Semi)Static	Discrete
Pocker	Partially	Multi	Stochastic	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Static	Discrete
Car	Partially	Multi	Stochastic	Dynamic	Continuous
Roomba	Partially	Single	Stochastic	Dynamic	Continuous

Randomness

Changing Environments

Agent types

Four basic types in order of increasing generality:

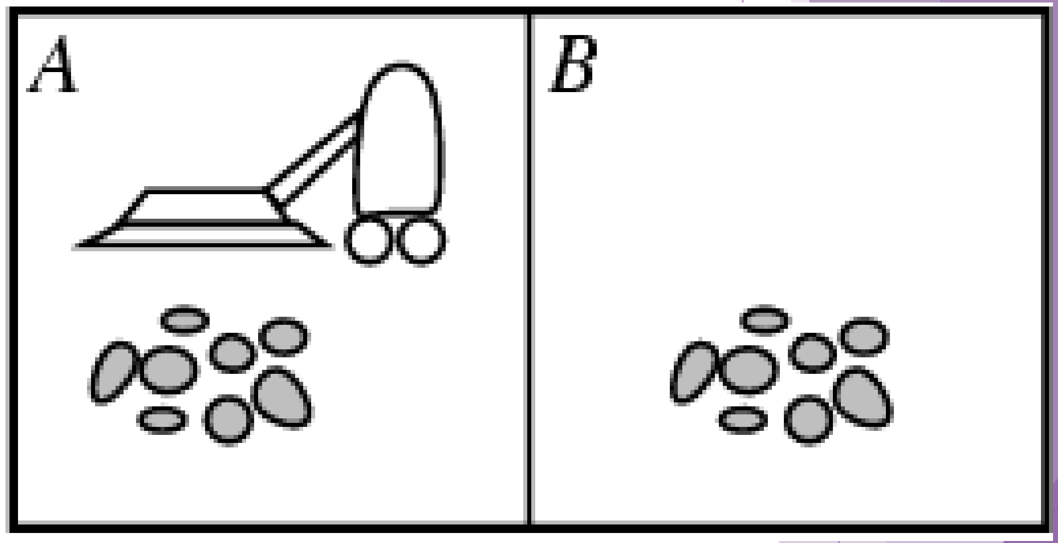
- Reflex agents (simple)
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

also

- Learning agents

Reflex agents

- action based on current state
- simple / limited
- fully-observed environment only
    - current percept



Percept	Action
[A, clean]	Right
[A, dirty]	Suck
[B, clean]	Left
[B, dirty]	Suck

Goal-based agents

- agents needs more information - goal
- affects actions
- considers future – what will happen if I do …?
- knowledge represented & modified

Utility-based agents

- maximizes performance
- utility function
- this is the performance measure
- can deal with uncertainty

What are search agents?

Formulate the problem

Initial state: the state in which the agent starts

States: All states reachable from the initial state by any sequence of actions

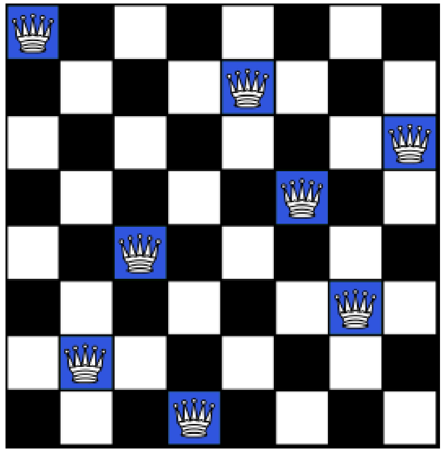
(State space)

Actions: possible actions available to the agent. At a state *s*, *Actions(s)* returns the set of actions that can be executed in state *s*.

(Action space)

e.g. go right or left in the maze

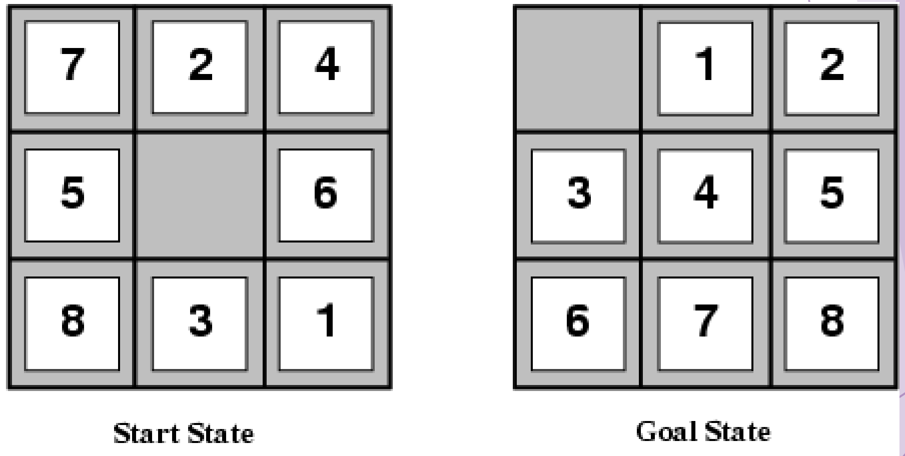
Examples



- **States:** all arrangements of 0 to 8 queens on the board.
- **Initial state:** No queen on the board
- **Actions:** Add a queen to any empty square
- **Transition model:** updated board
- **Goal test:** 8 queens on the board with none attacked

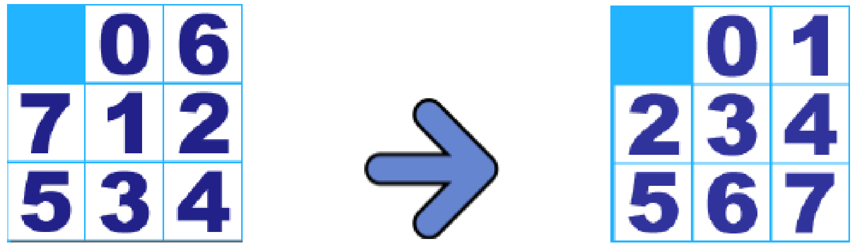
The 8-puzzle

states	locations of tiles
actions	move blank left, right, up, down
goal test	goal state (given)
path cost	1 per move



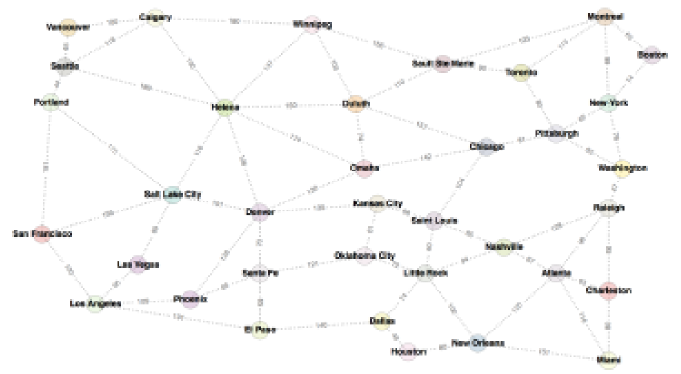
[Note: optimal solution of *n*-Puzzle family is NP-hard]

The 8-puzzle



- **States:** Location of each of the 8 tiles in the 3x3 grid
- **Initial state:** Any state
- **Actions:** Move Left, Right, Up or Down
- **Transition model:** Given a state and an action, returns resulting state
- **Goal test:** state matches the goal state?
- **Path cost:** total moves, each move costs 1.

search example

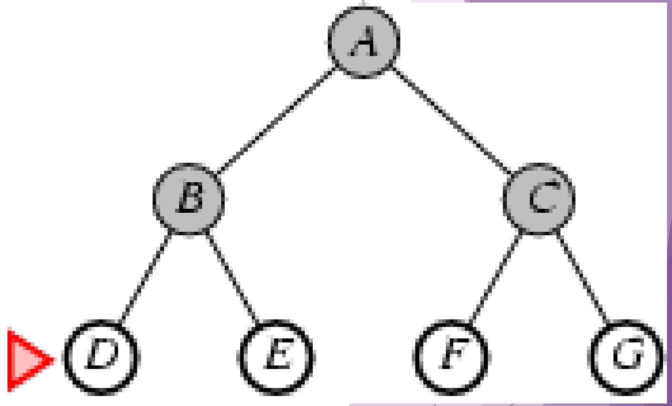


- **States:** In City where
  City ∈ {Los Angeles, San Francisco, Denver,...}
- **Initial state:** In Boston
- **Actions:** Go New York, etc.
- **Transition model:**
  Results (In (Boston), Go (New York)) = In(New York)
- **Goal test:** In(Denver)
- **Path cost:** path length in kilometers

Space

State space: a physical configuration

Search space: an abstract configuration represented by a search tree or graph of possible solutions.



Example: 8-puzzle

state space – all possible boards

search space – an abstract tree (or graph), nodes and edges

Space

Search tree: models the sequence of actions

- Root: initial state
- Branches: actions
- Nodes: results from actions. A node has: parent, children, depth, path cost, associated state in the state space.

Expand: A function that given a node, creates all children nodes

Space regions

The search space is divided into three regions:

1. Explored (a.k.a. Closed List, Visited Set)
2. Frontier (a.k.a. Open List, the Fringe)
3. Unexplored

Space regions

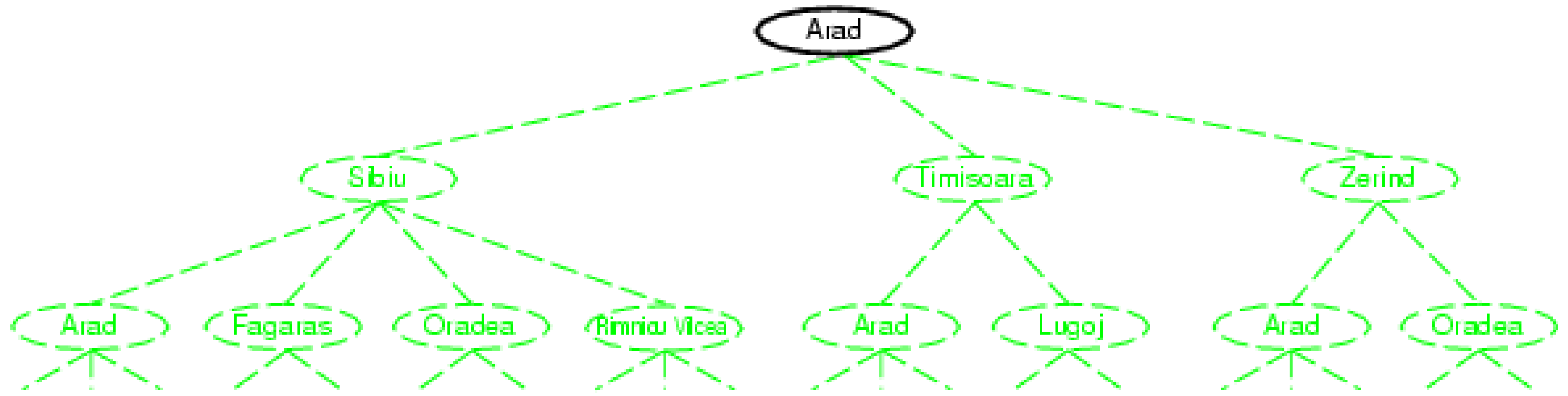
The essence of search is

- moving nodes from regions (3) to (2) to (1), and
- deciding the order of such moves.

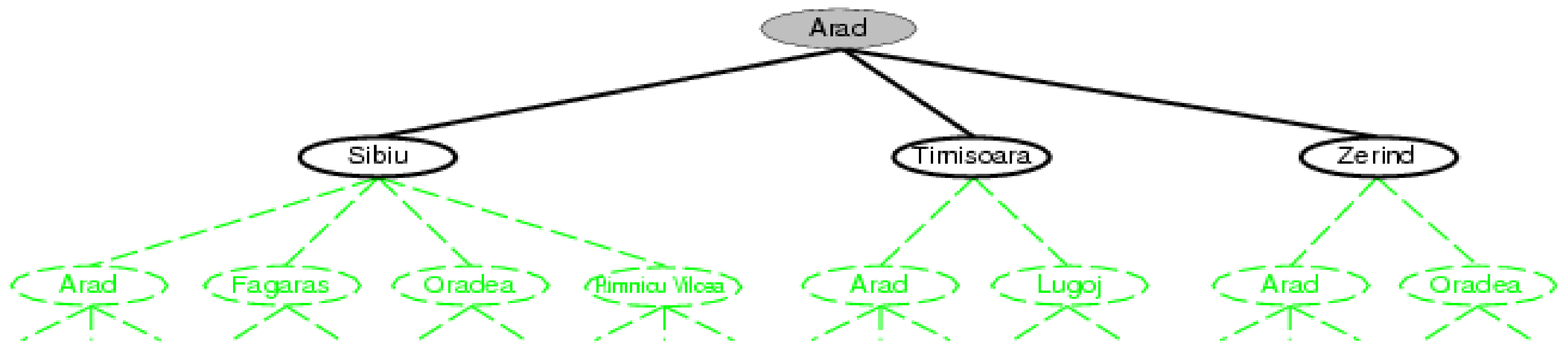
In the following we adopt the following color coding:

- green nodes are unexplored
- grey nodes are explored,
- white nodes are the frontier

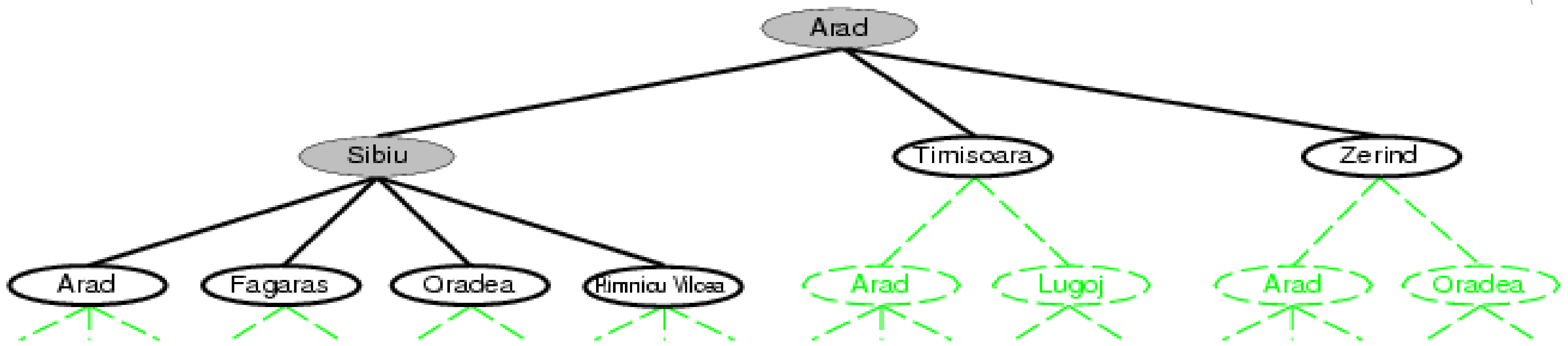
Tree search example



Tree search example



Tree search example



Search strategies

- order of node expansion
- Strategies are evaluated by:
  - completeness: does it always find a solution if one exists?
  - time complexity: number of nodes generated
  - space complexity: maximum number of nodes in memory
  - optimality: does it always find a least-cost solution?

Search strategies

- Time and space complexity are measured in terms of
  - *b*: maximum branching factor of the search tree
  - *d*: depth of the least-cost solution
  - *m*: maximum depth of the state space (may be ∞)

if domain knowledge – informed

else uninformed

Uninformed search strategies

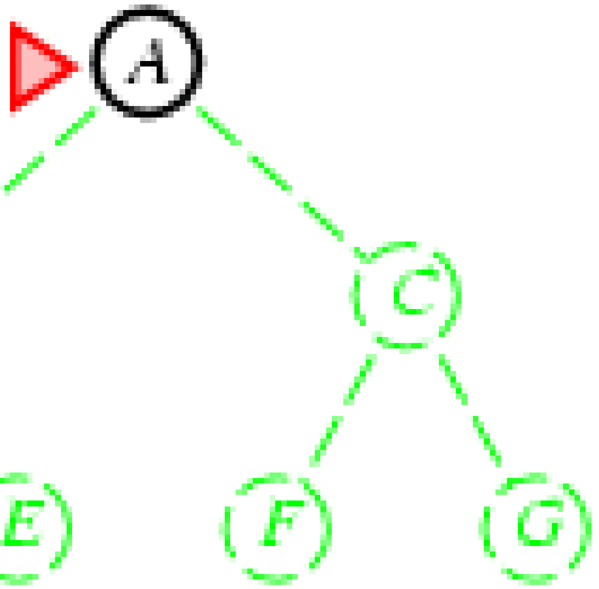
- Breadth-first search
- Depth-first search
- Depth-limited search
- Iterative deepening search
- Uniform-cost search

Uninformed search strategies

1. Breadth-first search (BFS):	Expand shallowest node
2. Depth-first search (DFS):	Expand deepest node
3. Depth-limited search (DLS):	Depth first with depth limit
4. Iterative-deepening (IDS):	DLS with increasing limit
5. Uniform-cost search (UCS):	Expand least cost node

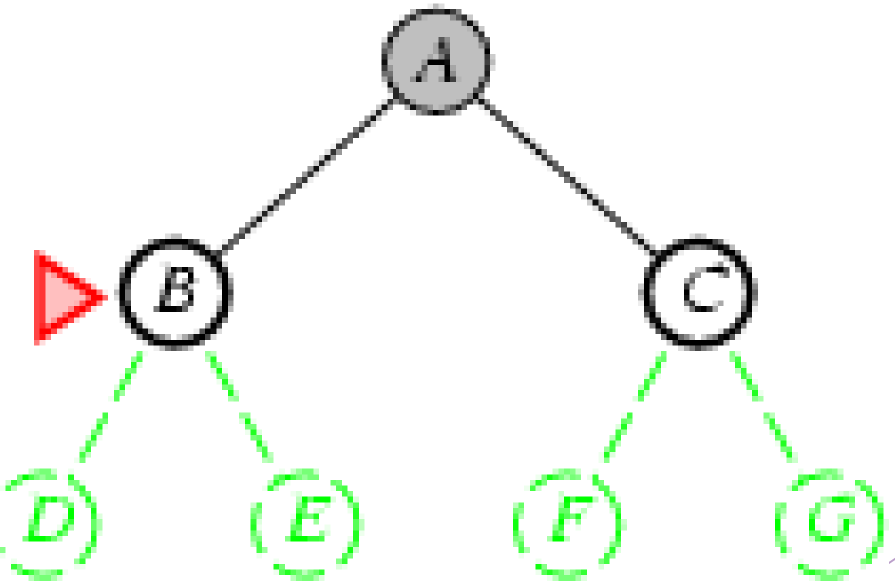
Breadth-first search

- Expand shallowest unexpanded node
- Implementation:
  - *fringe* is a FIFO queue, i.e., new successors go at end



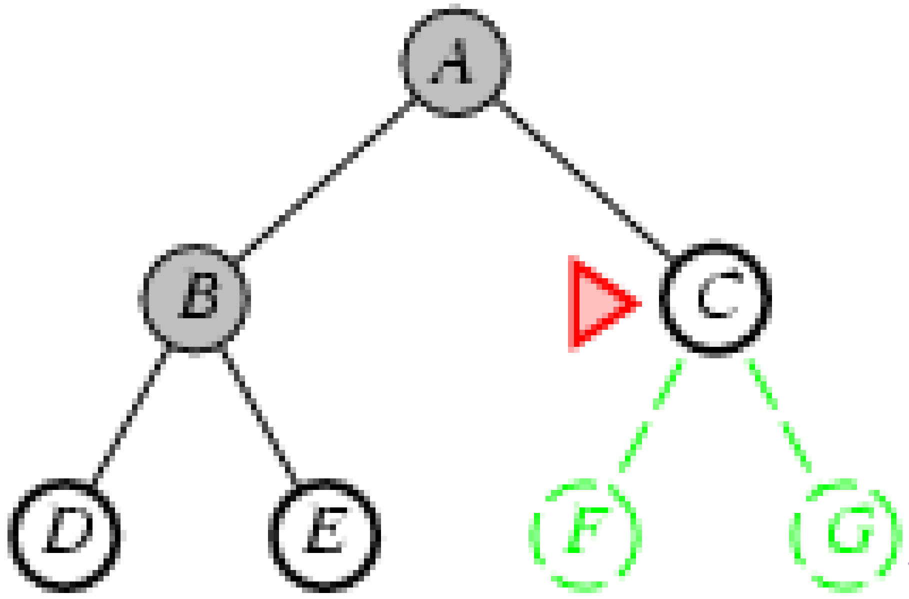
Breadth-first search

- Expand shallowest unexpanded node
- Implementation:
  - *fringe* is a FIFO queue, i.e., new successors go at end



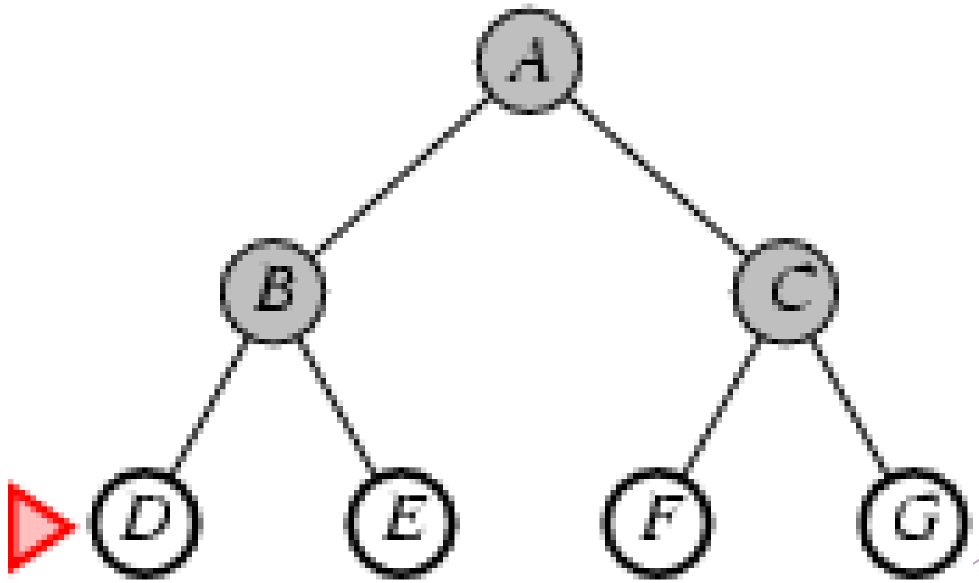
Breadth-first search

- Expand shallowest unexpanded node
- Implementation:
  - *fringe* is a FIFO queue, i.e., new successors go at end



Breadth-first search

- Expand shallowest unexpanded node
- Implementation:
  - *fringe* is a FIFO queue, i.e., new successors go at end



Properties of breadth-first search

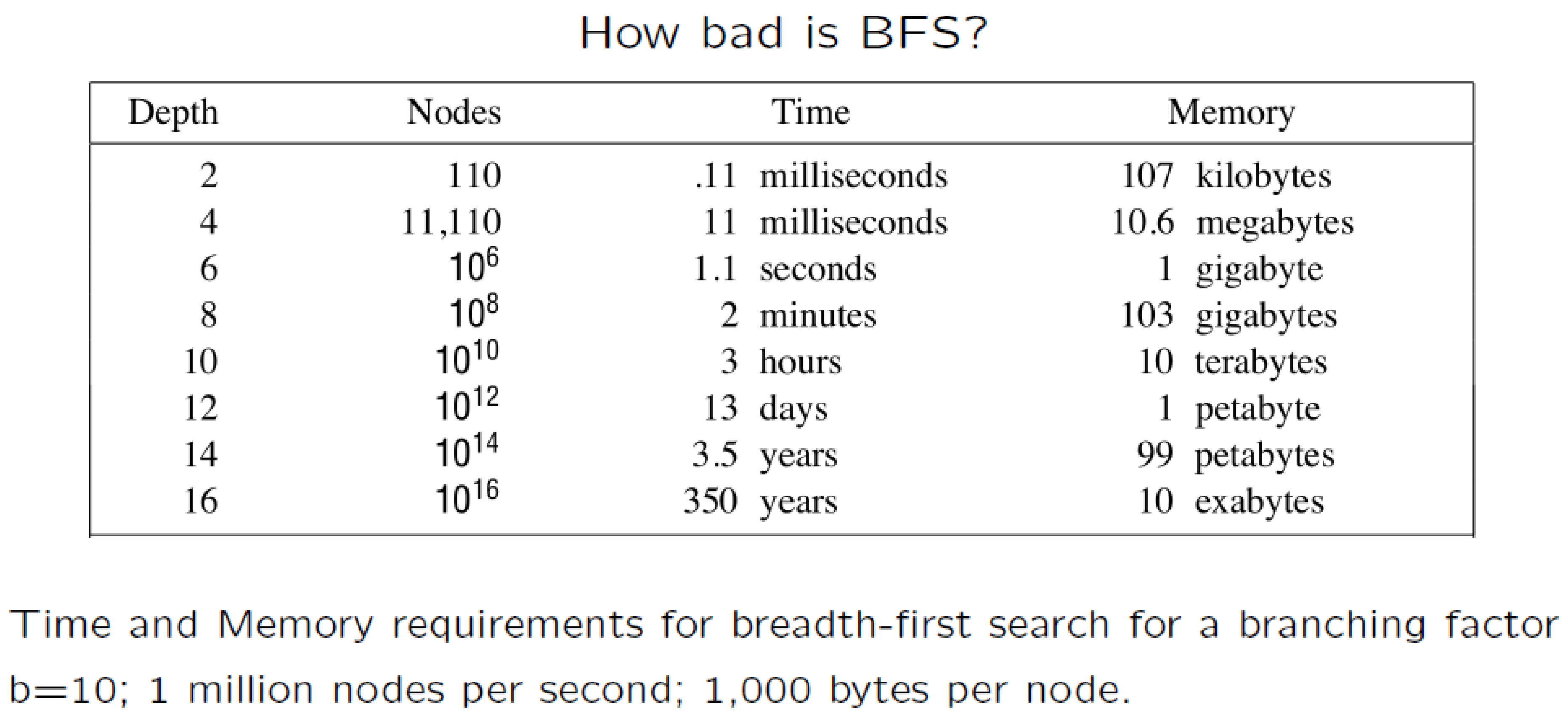
Complete	Yes	*(if b is finite)*
Time	$1 + b + b^2 + b^3 + \ldots + b^d + b(b^d - 1) = O(b^{d+1})$	
Space	$O(b^{d+1})$	
		(keeps every node in memory)
Optimal	Yes	*(if cost = 1 per step)*

Space is the bigger problem (more than time)

Why use it (if exponential time & space)?

- shallow problems, NLP

Properties of breadth-first search



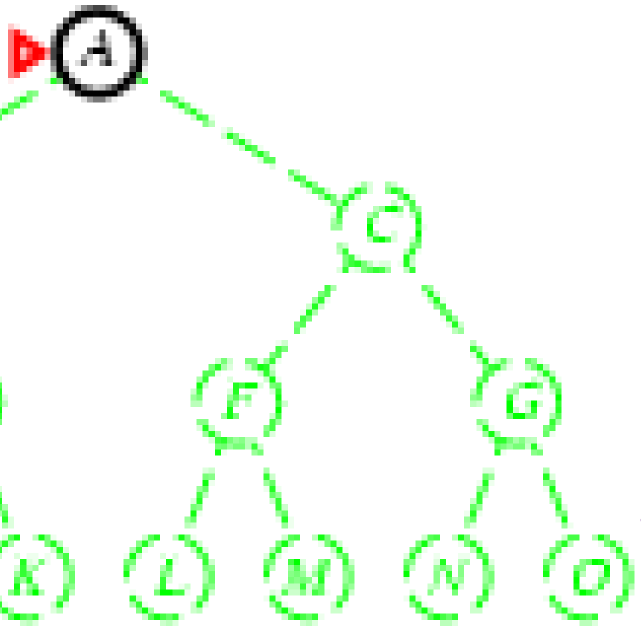
How bad is BFS?

Depth	Nodes	Time	Memory
2	110	.11 milliseconds	107 kilobytes
4	11,110	11 milliseconds	10.6 megabytes
6	10^6	1.1 seconds	1 gigabyte
8	10^8	2 minutes	103 gigabytes
10	10^{10}	3 hours	10 terabytes
12	10^{12}	13 days	1 petabyte
14	10^{14}	3.5 years	99 petabytes
16	10^{16}	350 years	10 exabytes

Time and Memory requirements for breadth-first search for a branching factor b=10; 1 million nodes per second; 1,000 bytes per node.

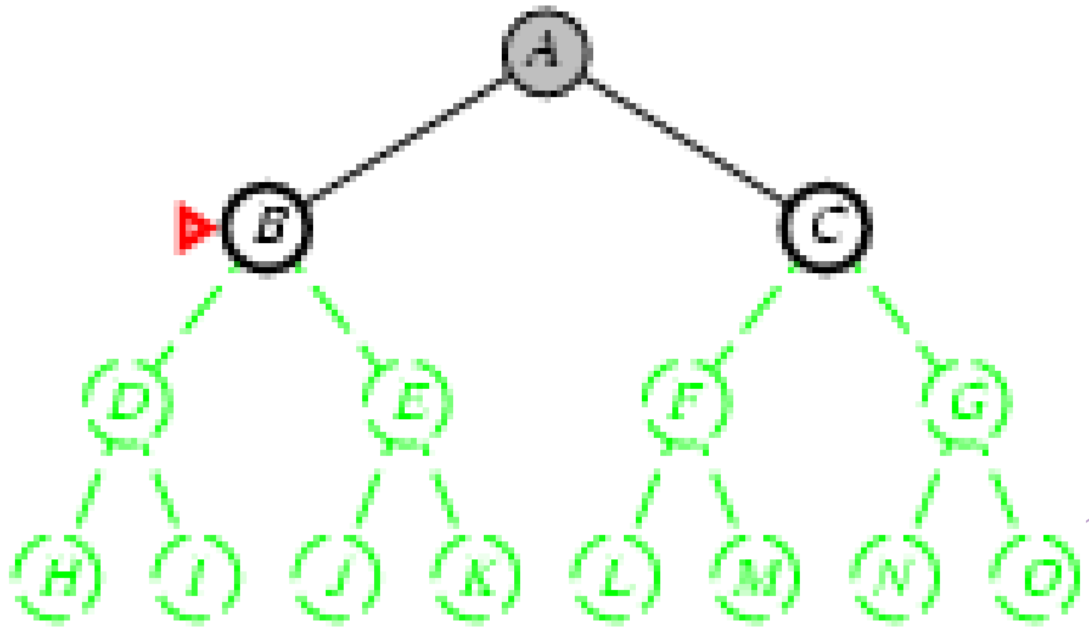
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



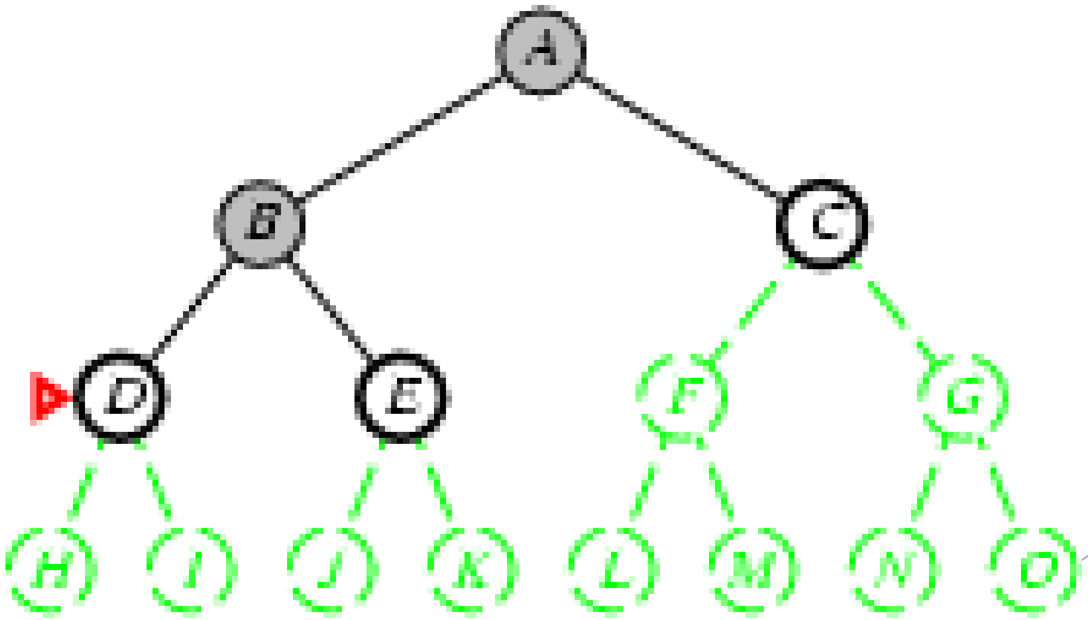
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



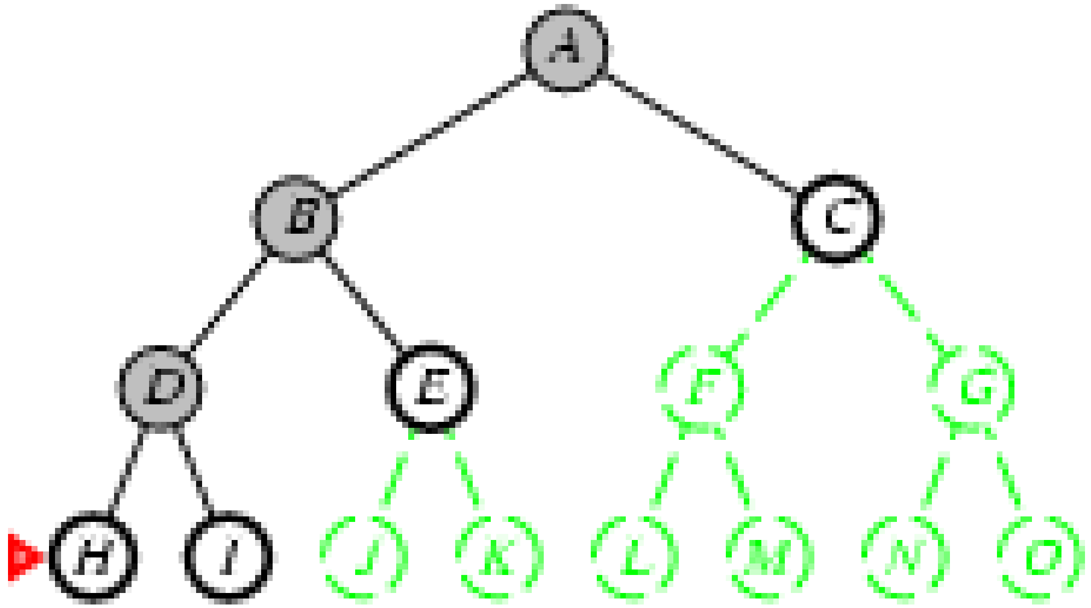
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



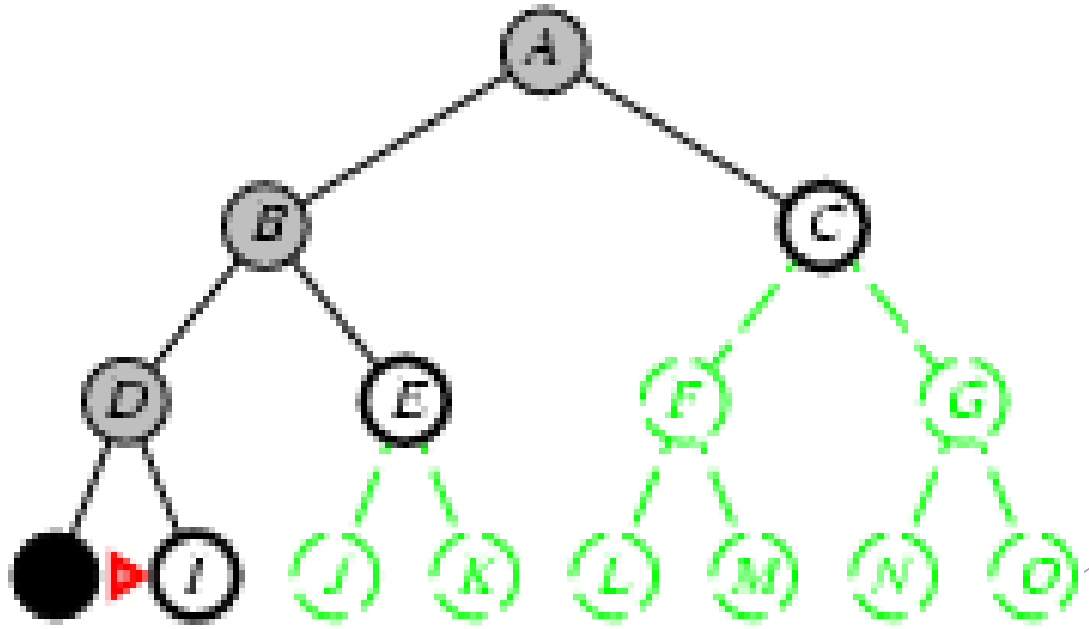
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



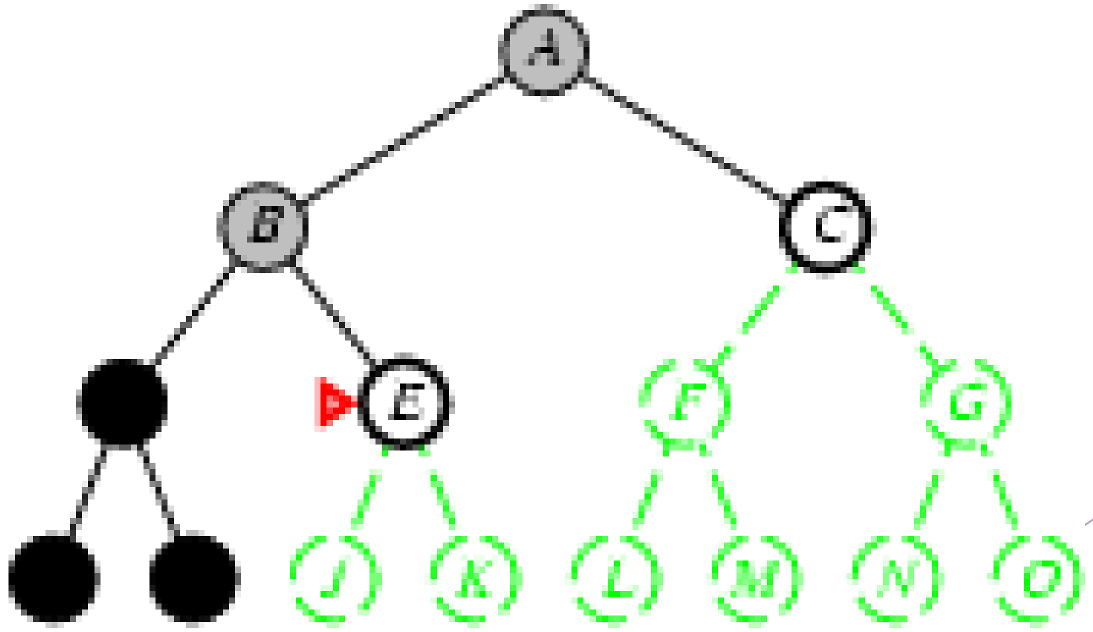
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



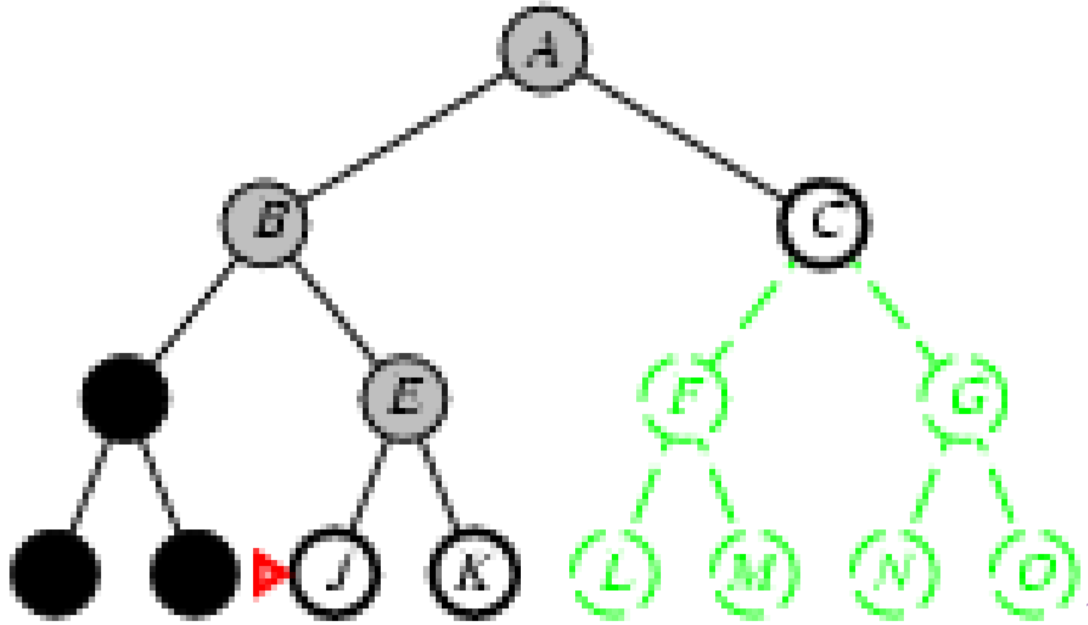
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



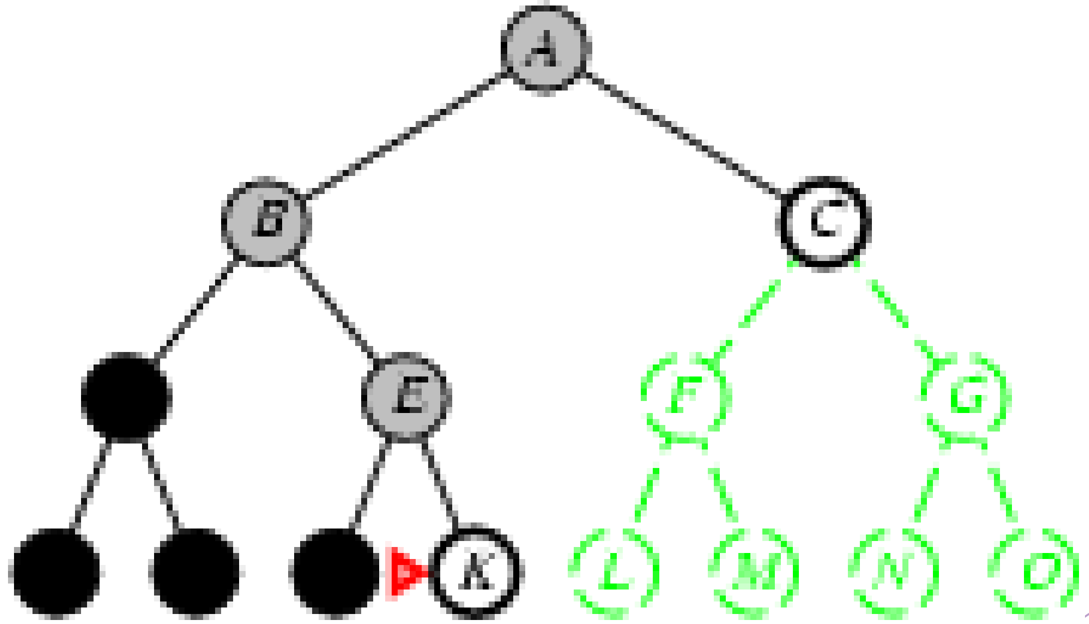
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



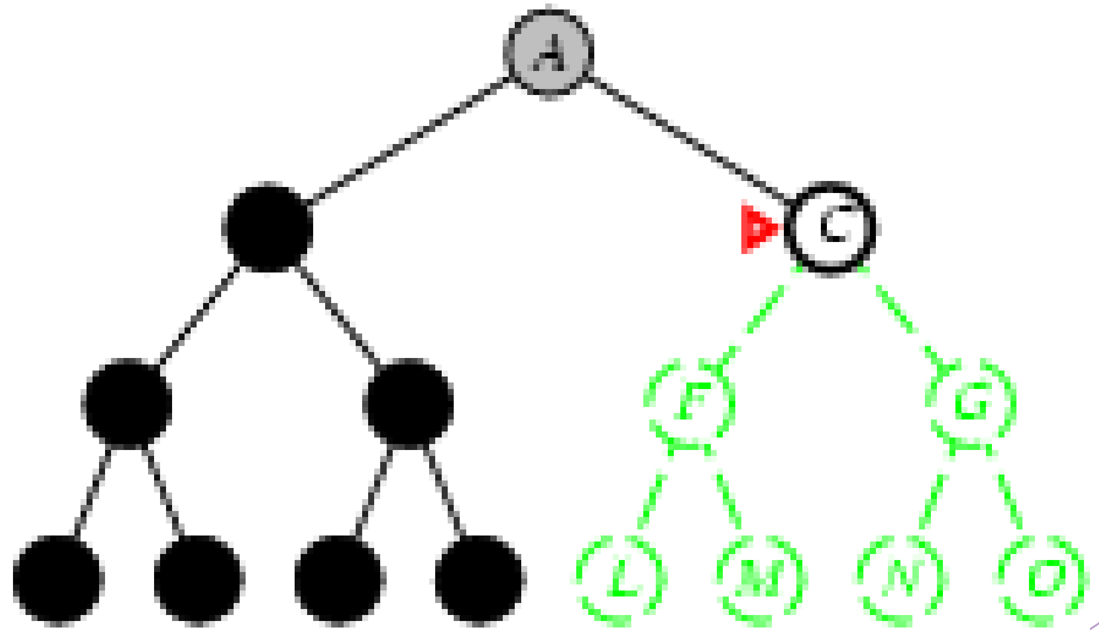
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



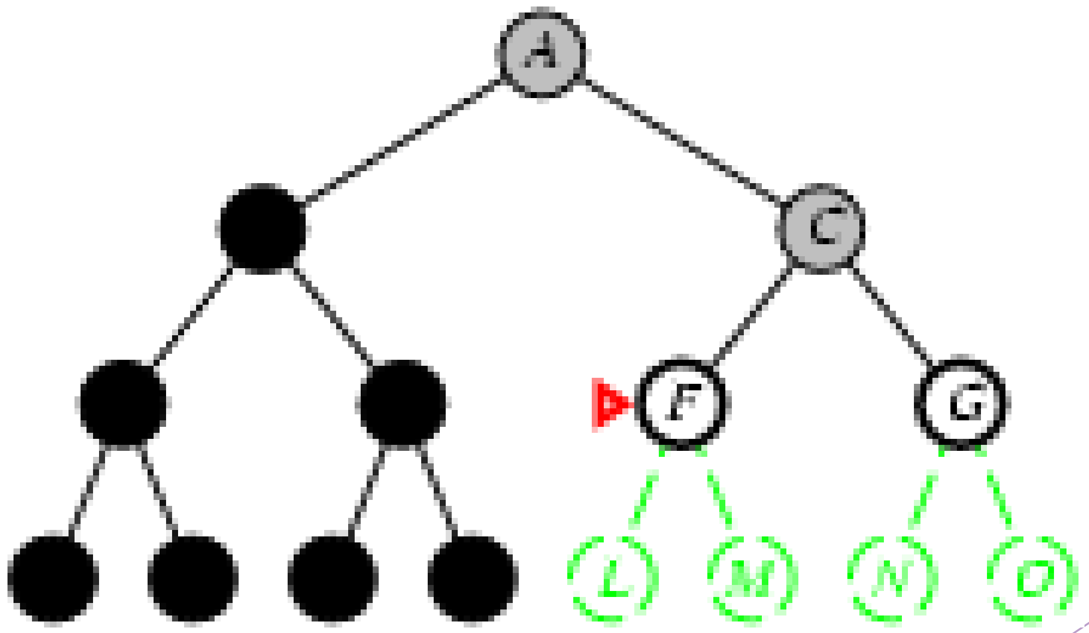
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



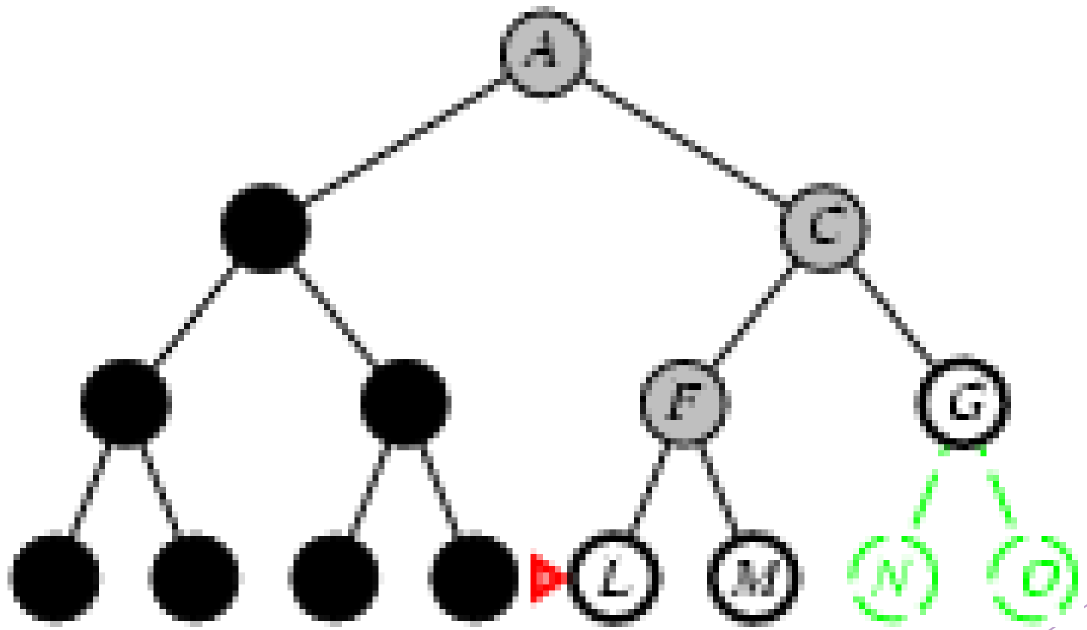
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



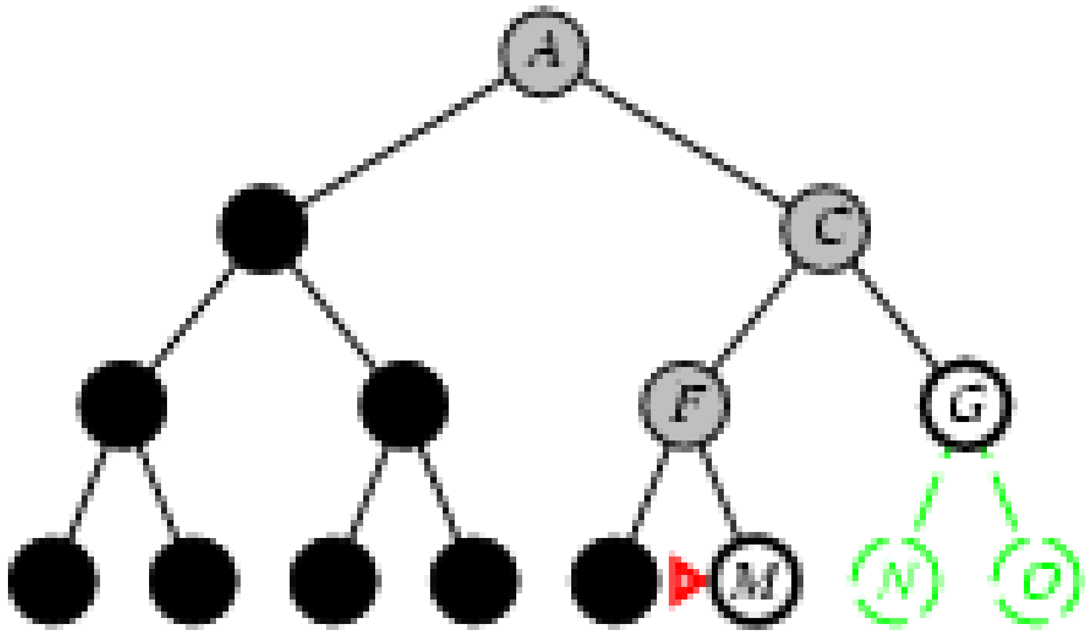
Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



Depth-first search

- Expand deepest unexpanded node
- Implementation:
  - *fringe* = LIFO queue, i.e., put successors at front



Properties of depth-first search

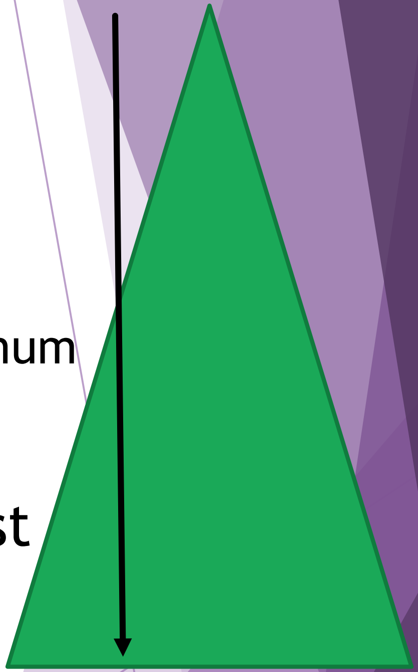
- Complete No:

  fails in infinite-depth spaces, spaces with loops
  - if modify to avoid repeated states along path

    → yes - complete in finite spaces
- Time $O(b^m)$: terrible if m is much larger than d
  - but if solutions are dense, may be much faster than breadth-first
- Space $O(bm)$, i.e., linear space complexity

  only store a single path (root-node)
- Optima No

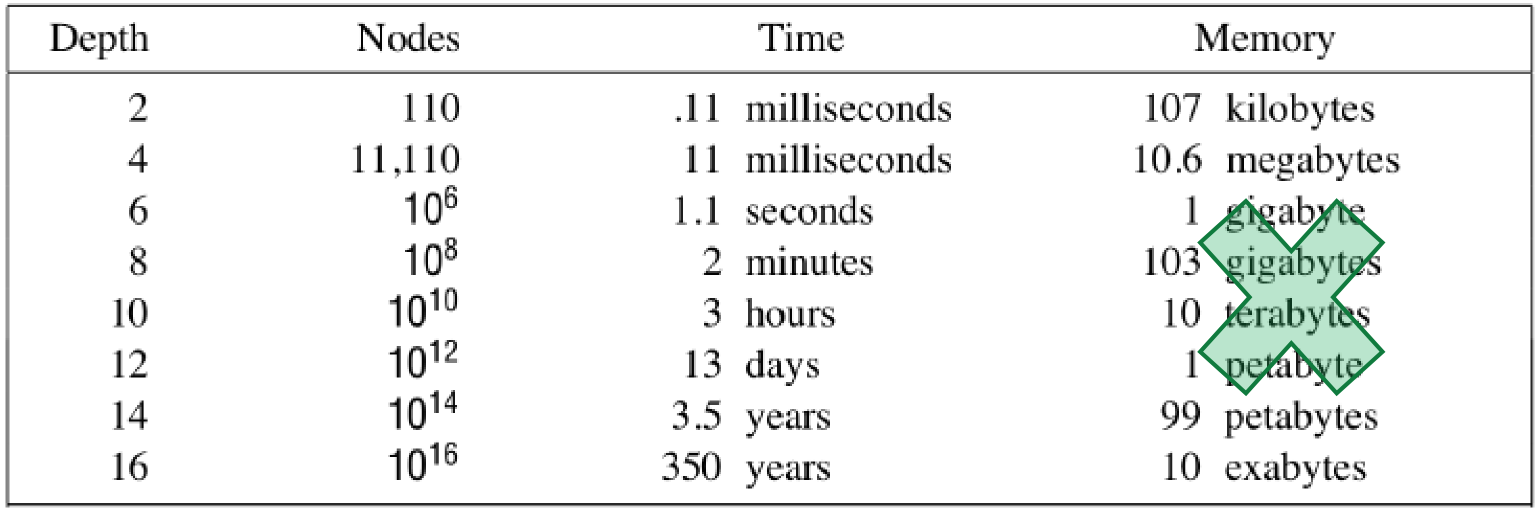
m = maximum depth



Properties of depth-first search

How bad is DFS?

Recall for BFS...



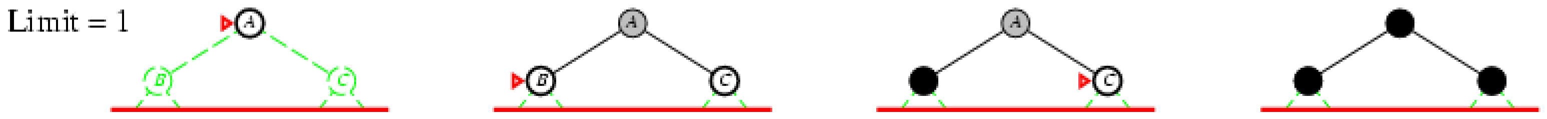
Depth	Nodes	Time	Memory
2	110	.11 milliseconds	107 kilobytes
4	11,110	11 milliseconds	10.6 megabytes
6	10^6	1.1 seconds	1 gigabyte
8	10^8	2 minutes	103 gigabytes
10	10^{10}	3 hours	10 terabytes
12	10^{12}	13 days	1 petabyte
14	10^{14}	3.5 years	99 petabytes
16	10^{16}	350 years	10 exabytes

Depth =16.

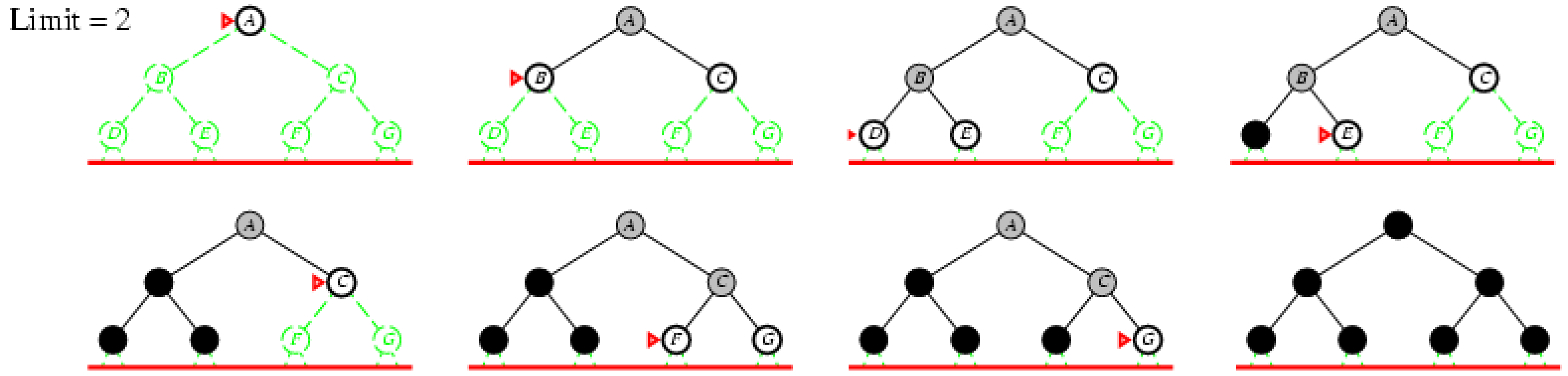
We go down from 10 exabytes in BFS to 156 kilobytes in DFS!

Iterative deepening search *l* =0

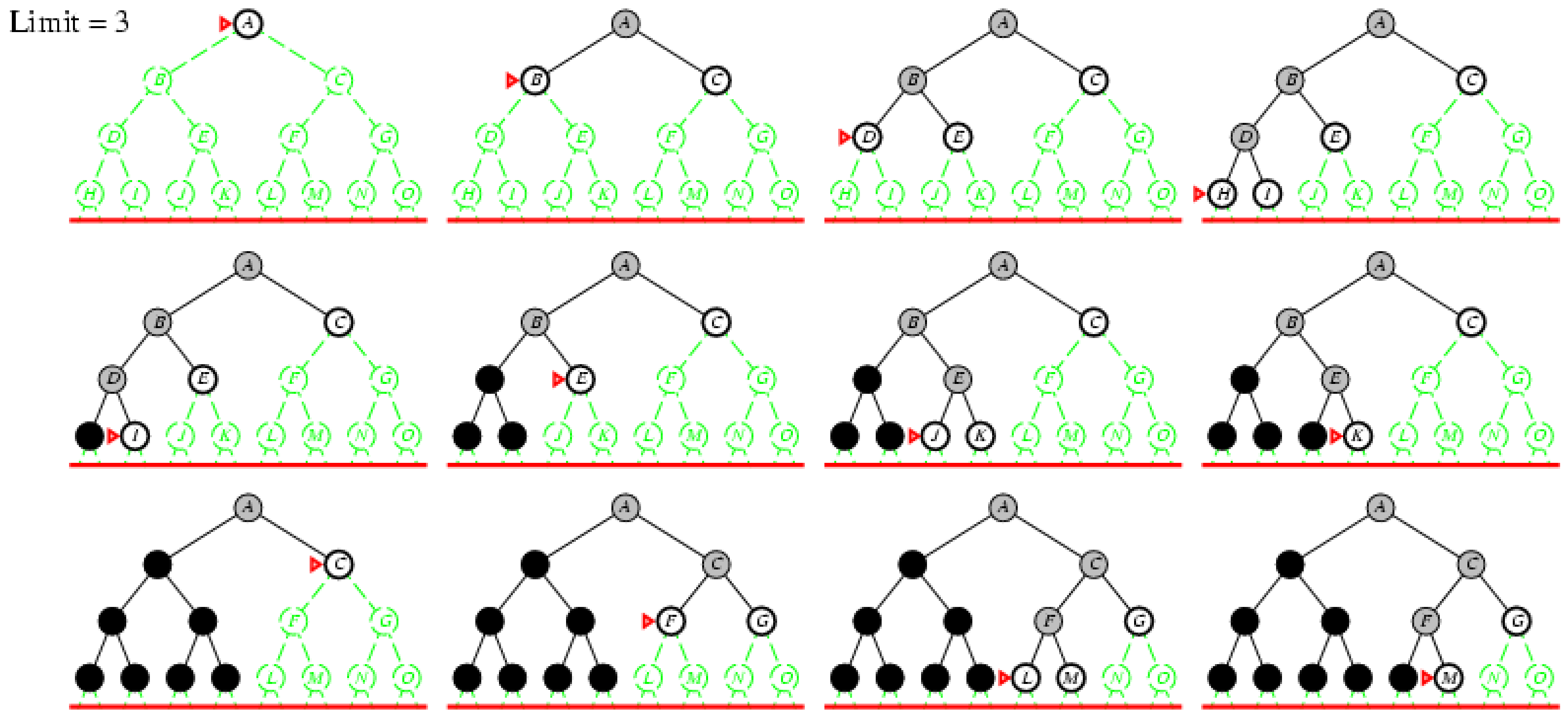
Iterative deepening search *l* =1



Iterative deepening search *l* =2



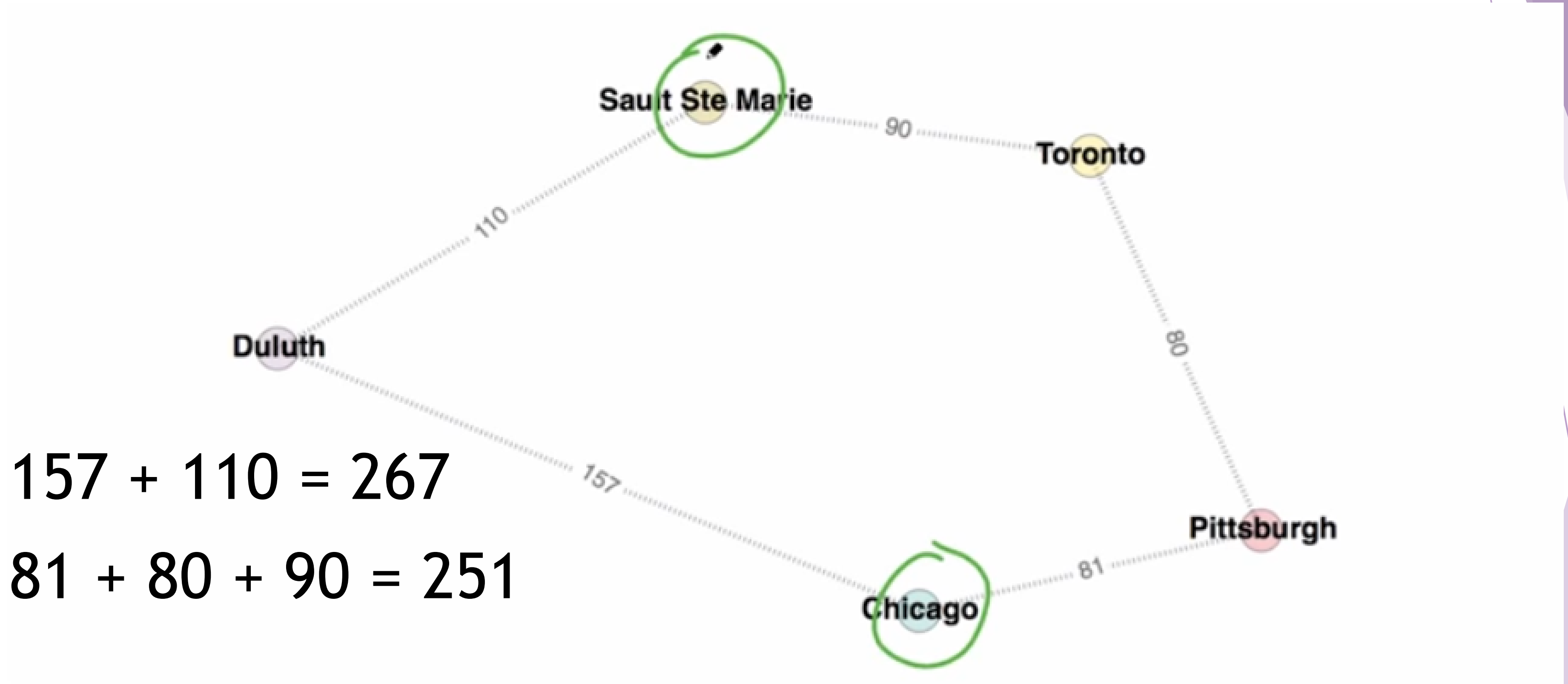
Iterative deepening search *l* =3



Properties of iterative deepening search

- Complete Yes
- Time $(d+1)b^0 + d\,b^1 + (d-1)b^2 + \ldots + b^d = O(b^d)$
- Space $O(bd)$
- Optimal Yes, if step cost = 1

Example



- 157 + 110 = 267
- 81 + 80 + 90 = 251

Go from Chicago to Sault Ste Marie. Using BFS, we would find Chicago-Duluth-Sault Ste Marie. However, using UCS, we would find Chicago-Pittsburgh-Toronto-Sault Ste Marie, which is actually the shortest path!

Uniform-cost search

Complete

Yes, if solution has a finite cost

Time

$O(b^{(C^*/\varepsilon)})$ where C^* is the cost of the optimal solution, $^\varepsilon$ is the max cost of an action

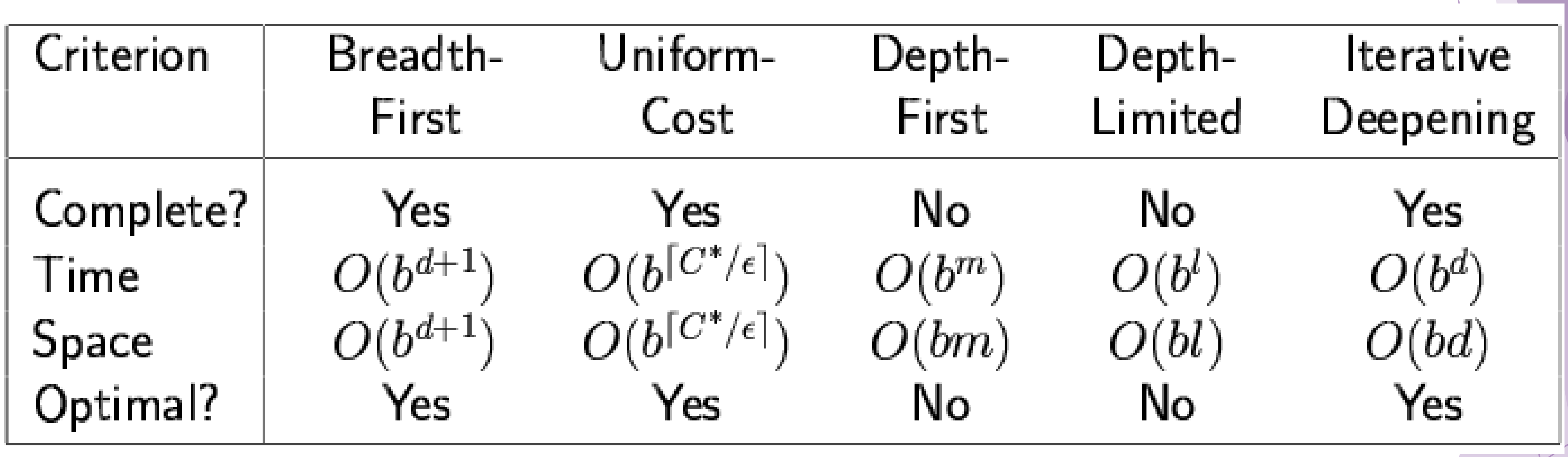
Space

# of nodes with $g \leq$ cost of optimal solution, $O(b^{(C^*/\ \varepsilon)})$

Optimal

Yes – nodes expanded in increasing order of $g(n)$

Summary of algorithms



Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening
Complete?	Yes	Yes	No	No	Yes
Time	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil})$	$O(b^m)$	$O(b^l)$	$O(b^d)$
Space	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil})$	$O(bm)$	$O(bl)$	$O(bd)$
Optimal?	Yes	Yes	No	No	Yes

What are informed search agents?

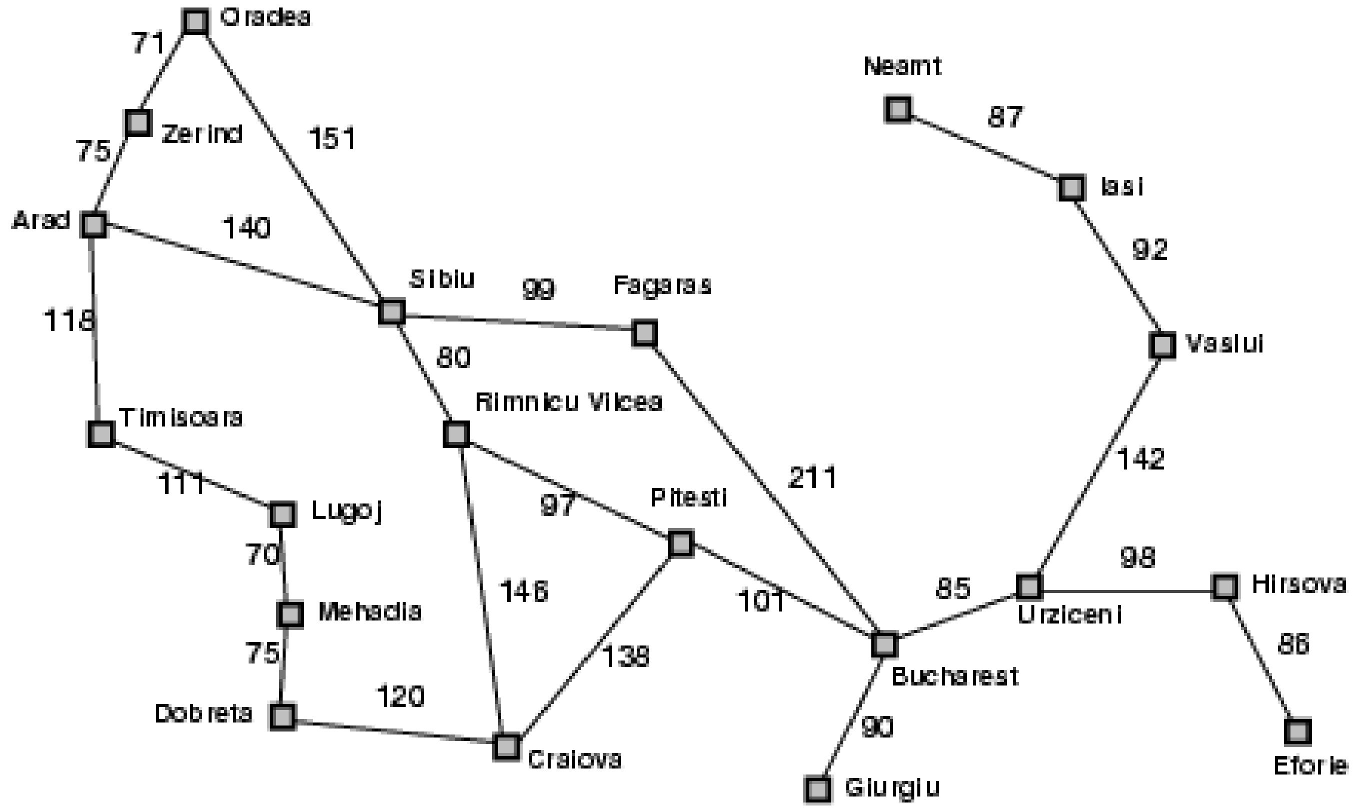
Informed search

Use domain knowledge!

- Are we getting close to the goal?
- Use a heuristic function that estimates how close a state is to the goal
- A heuristic does NOT have to be perfect!
- Example of strategies:

1. Greedy best-first search
2. A* search
3. IDA* (alternative type of A* search)

Romania with step costs in km



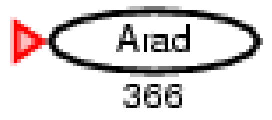
Straight–line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

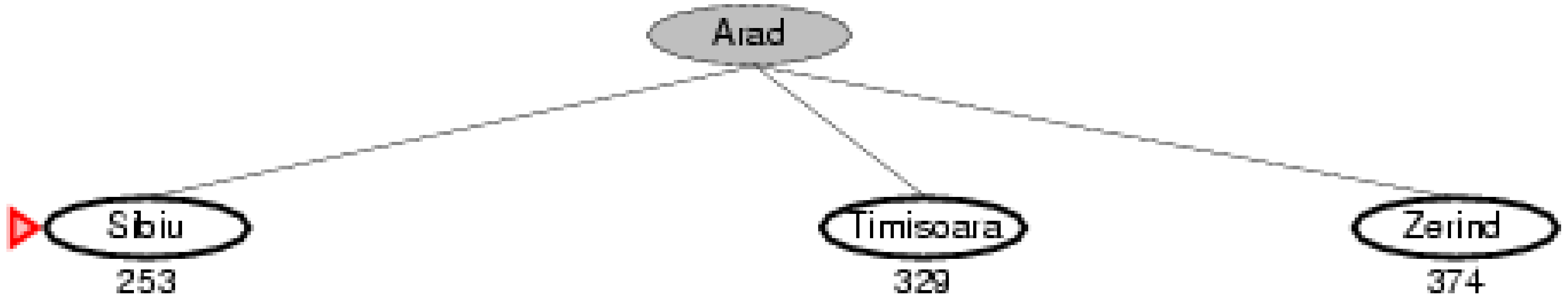
Greedy best-first search

- Evaluation function $f(n) = h(n)$ (heuristic)
- = estimate of cost from *n* to *goal*
- e.g., $h_{SLD}(n)$ = straight-line distance from *n* to Bucharest
- Greedy best-first search expands the node that appears to be closest to goal
- Example: there are 3 possible cities, which city is the closest, in km, to the goal (destination city)

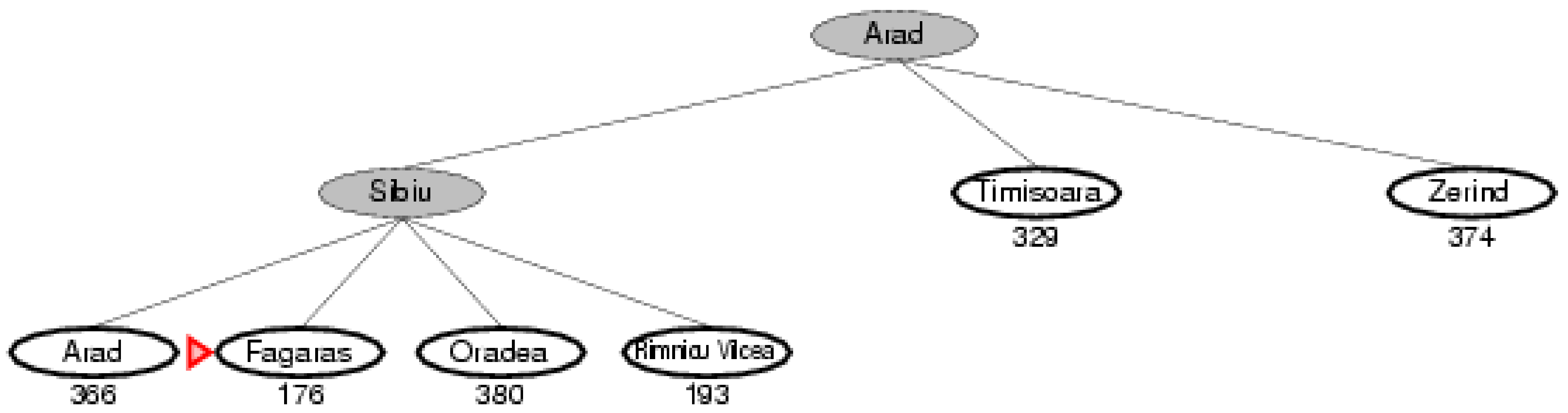
Greedy best-first search example



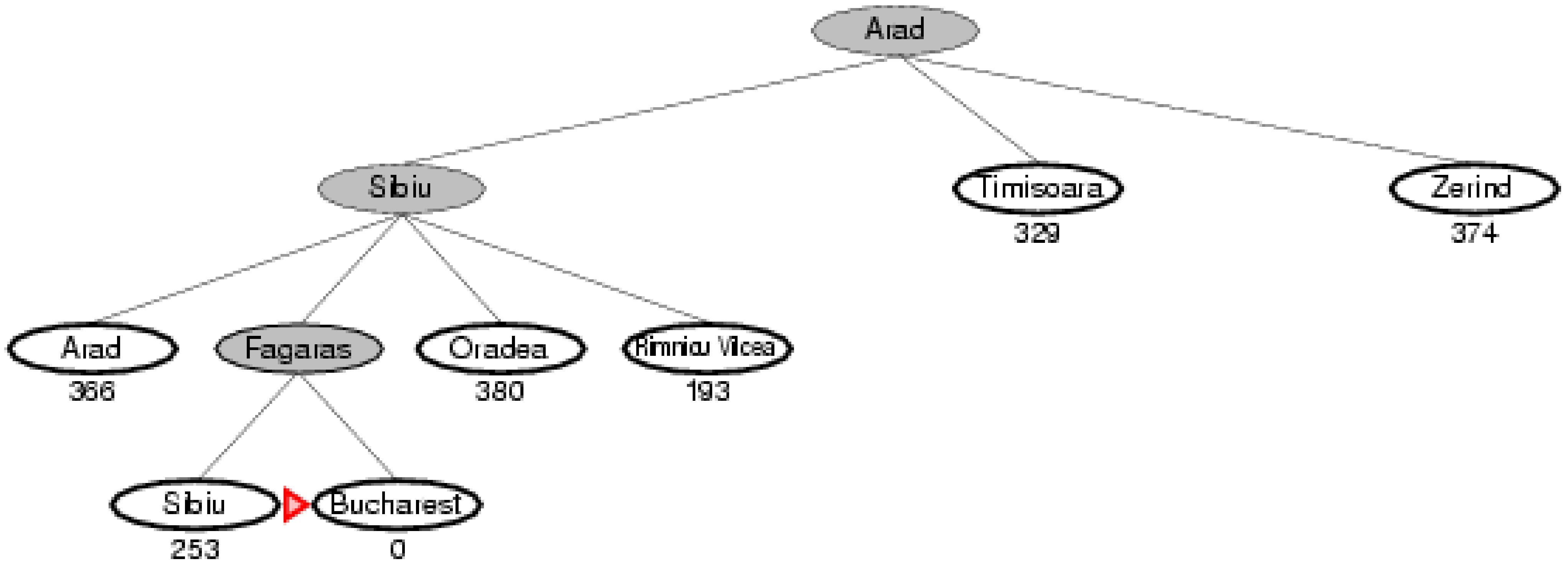
Greedy best-first search example



Greedy best-first search example



Greedy best-first search example



Properties of greedy best-first search

- Complete? No – can get stuck in loops
- Time? *O(b^m)*, but a good heuristic can give dramatic improvement
- Space? *O(b^m)* -- keeps all nodes in memory
- Optimal? No

A^* search

Minimize the total estimated solution cost

Combines:

- g(n): cost to reach node n
- h(n): cost to get from n to the goal
- f(n) = g(n)+h(n)

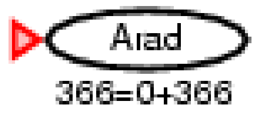
- f(n) is the estimated cost of the cheapest solution through n

A^* search

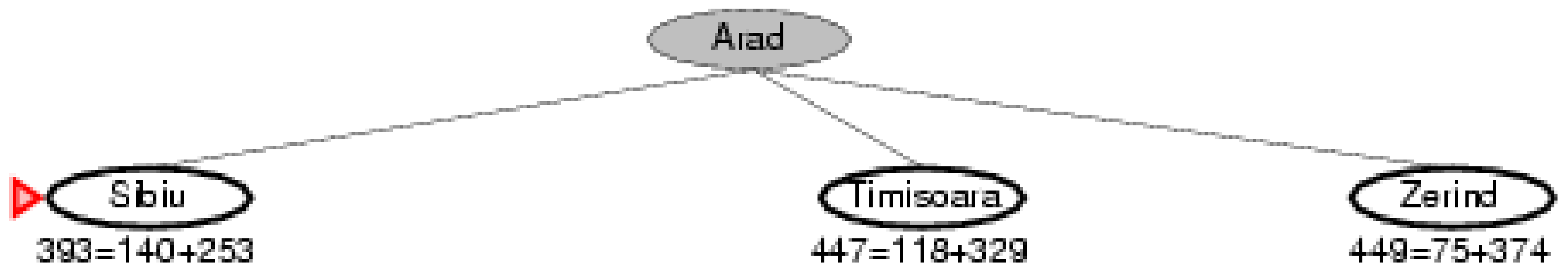
- Idea: avoid expanding paths that are already expensive
- Evaluation function $f(n) = g(n) + h(n)$

calculate total cost = cost so far + estimated cost to goal

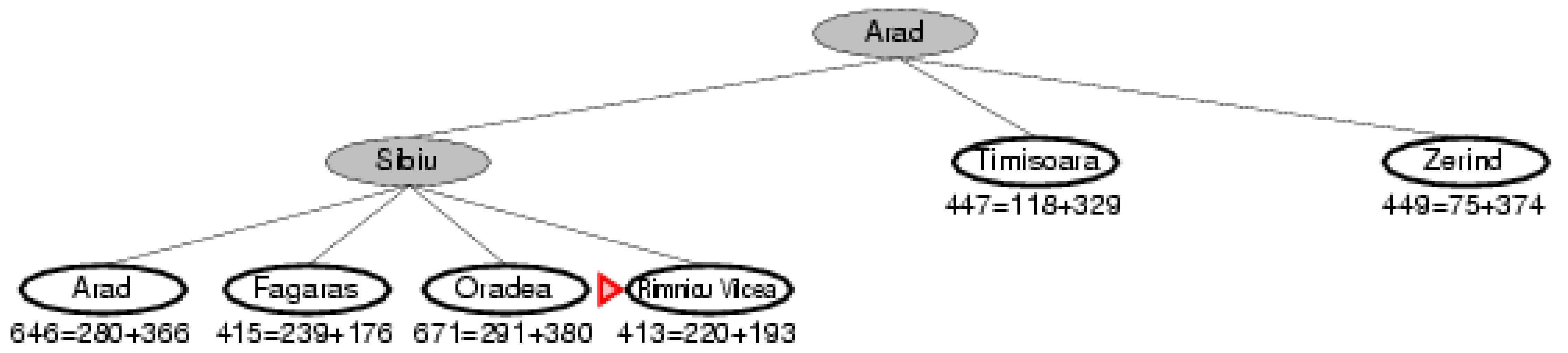
A* search example



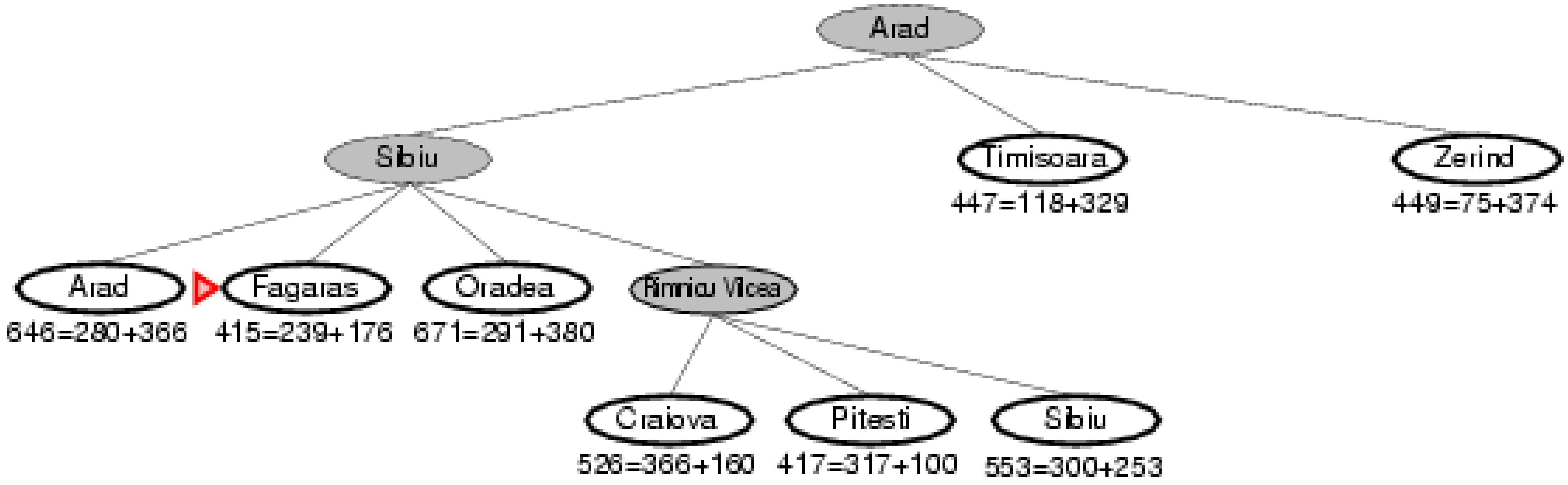
A* search example



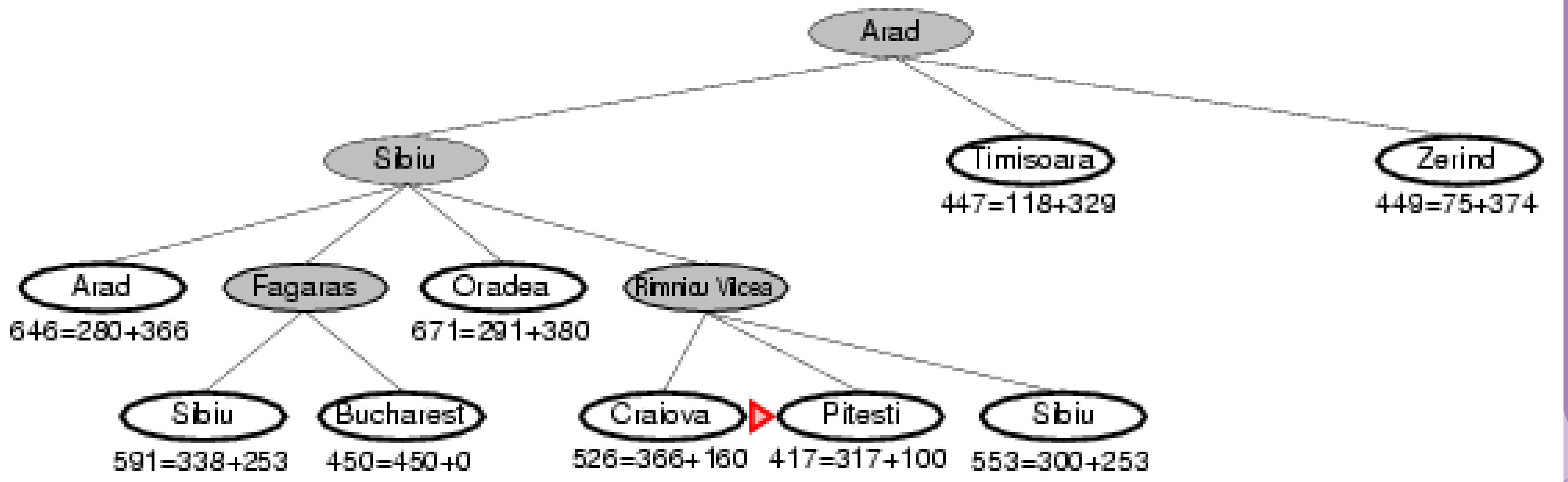
A* search example



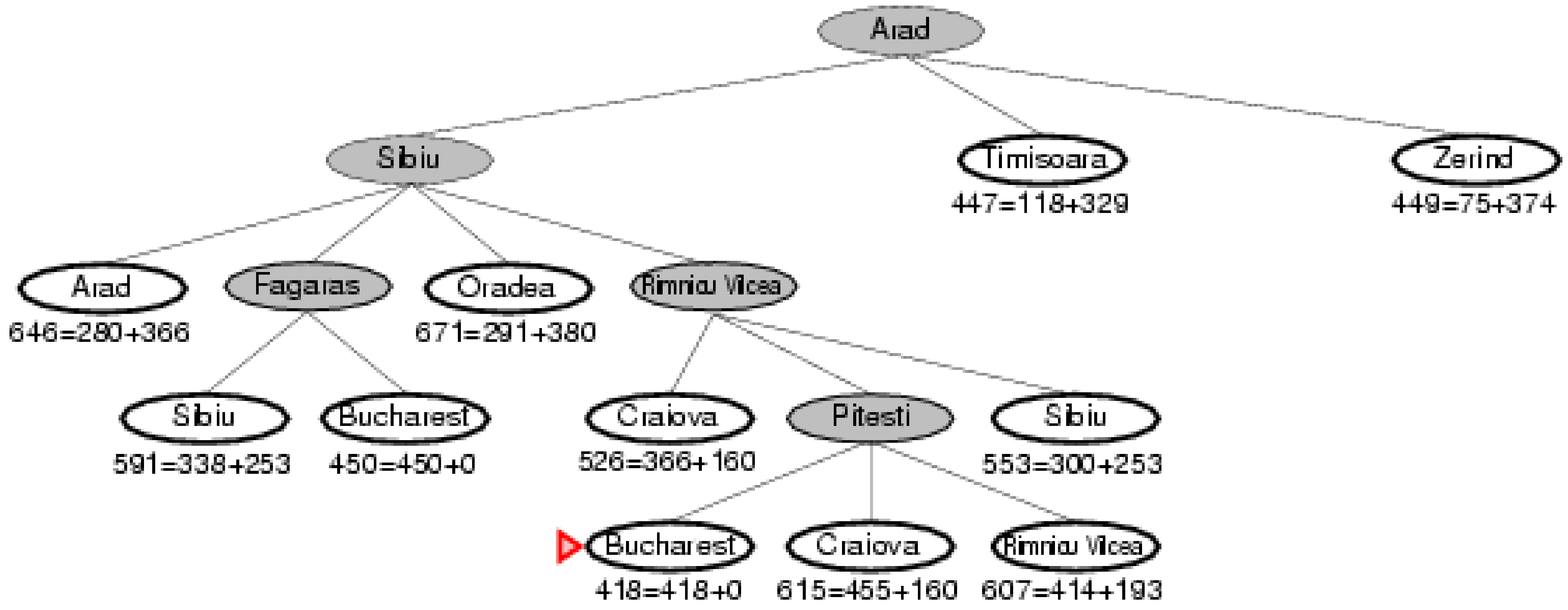
A^* search example



A* search example



A* search example



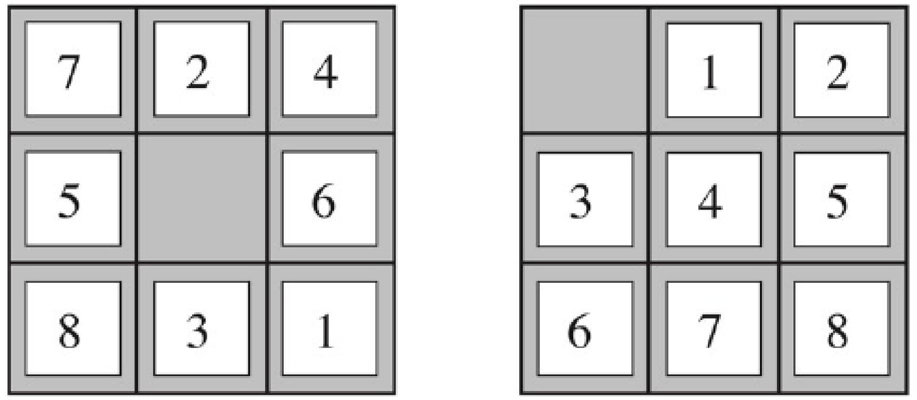
A^* search

A good heuristic can be powerful.

Only if it is of a “good quality”

An admissible heuristic never overestimates the cost to reach the goal, i.e., it is optimistic

A* search

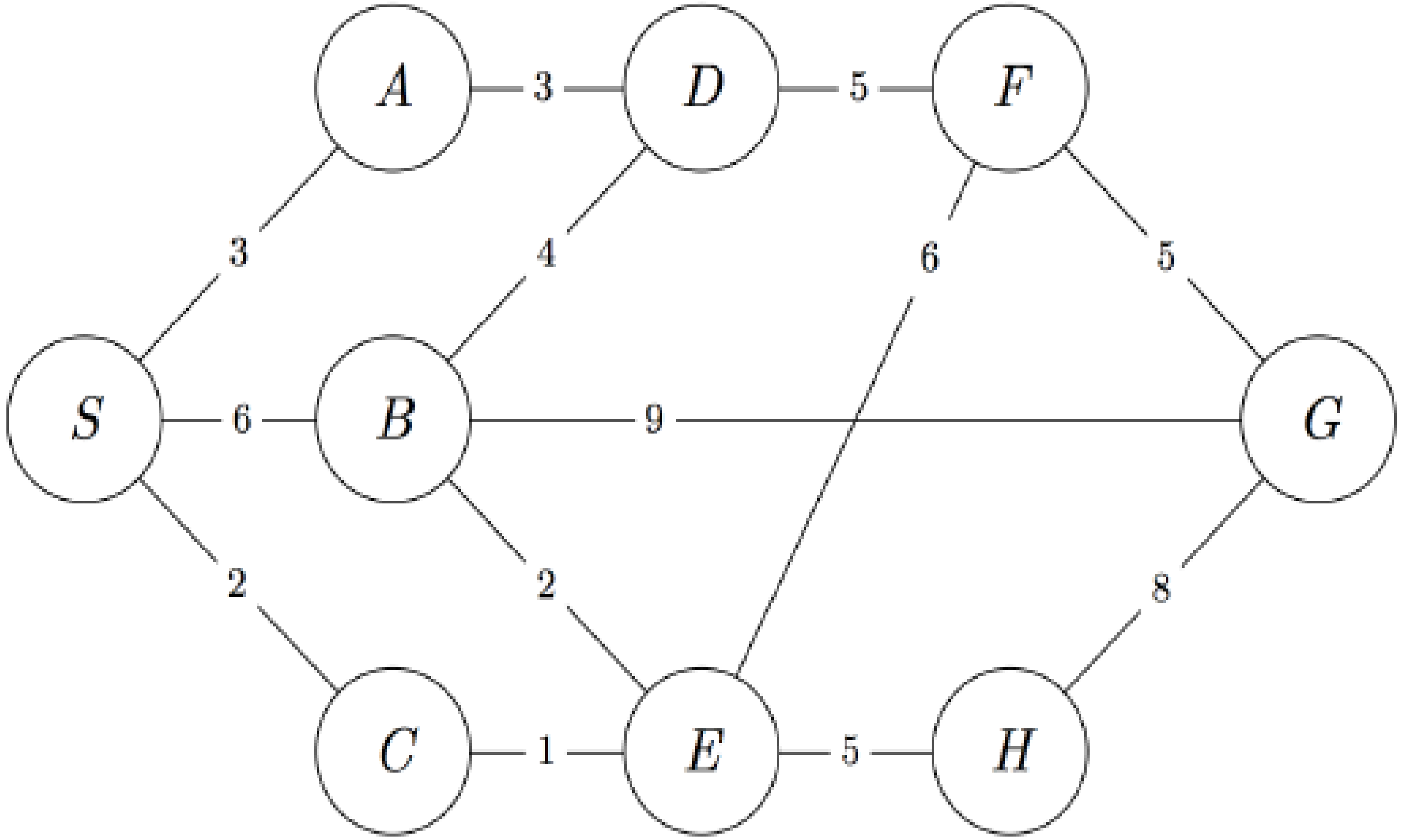


Start State Goal State

- The solution is 26 steps long.
- $h_1(n) =$ number of misplaced tiles
- $h_2(n) =$total Manhattan distance (sum of the horizontal and vertical distances).
- $h_1(n) = 8$
- Tiles 1 to 8 in the start state gives: $h_2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$ which does not overestimate the true solution.

A^* search

Complete?	Yes
Time?	exponential
Space?	keeps every node in memory - problem
Optimal?	Yes!


A
D
F
S
B
G
C
E
H
3
5
3
4
6
5
6
9
2
2
8
1
5

Games &
Satisfying Constraint Problems

Adversarial Search

- Adversarial search problems = games
- They occur in multi-agent competitive environments
- There is an opponent we can’t control planning again us!

Adversarial Search

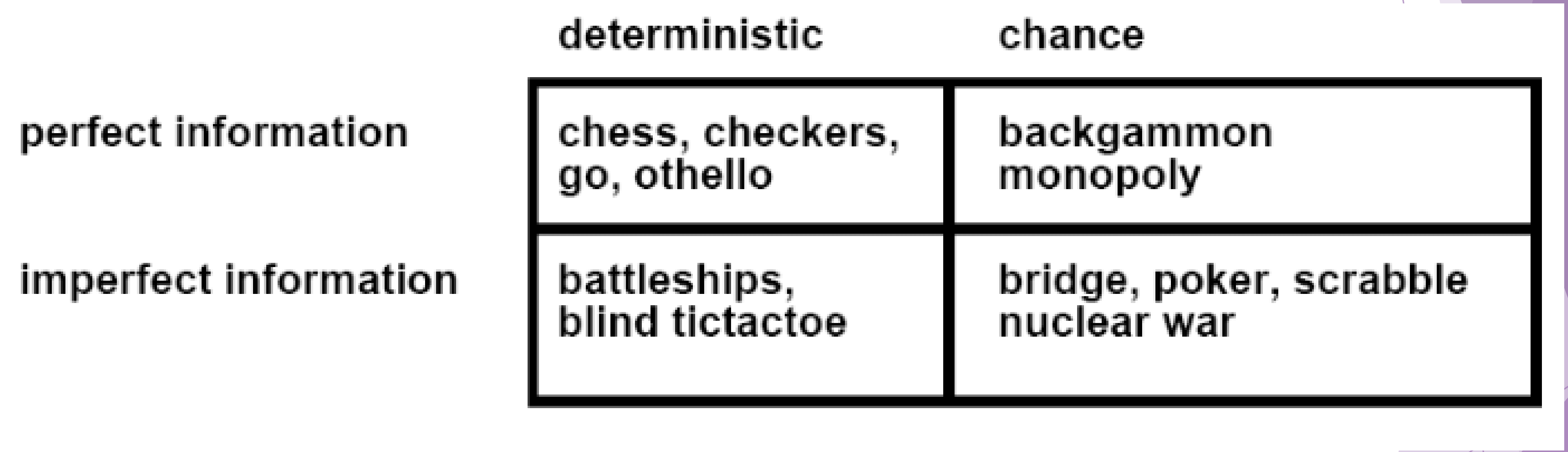
- Game vs. search:
  - optimal solution is not a sequence of actions
  - but a strategy (policy)
  - If opponent does a, agent does b, else
  - if opponent does c, agent does d, etc.
  - Tedious and fragile if hard-coded (i.e., rules)

We use 2 ingredients:

- search problems & heuristic evaluation

Type of Games

non-deterministic / stochastic games



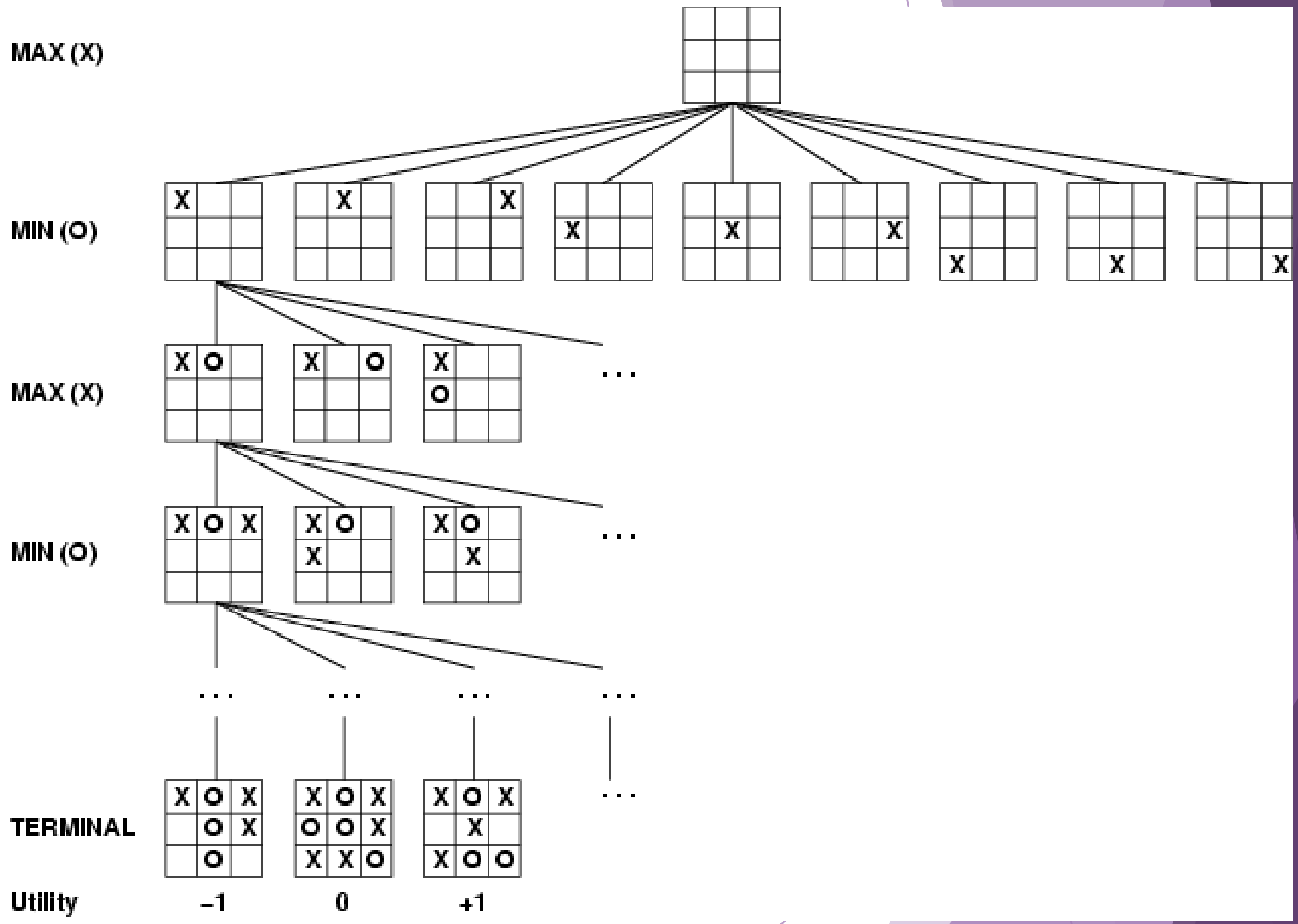
	deterministic	chance
perfect information	chess, checkers, go, othello	backgammon monopoly
imperfect information	battleships, blind tictactoe	bridge, poker, scrabble nuclear war

Adversarial Search

- Two players: Max and Min
- Players alternate turns
- Max moves first
- Max maximizes results
- Min minimizes the result
- Compute each node's minimax value's the best achievable utility against an optimal adversary
- Minimax value $\equiv$ best achievable payoff against best play

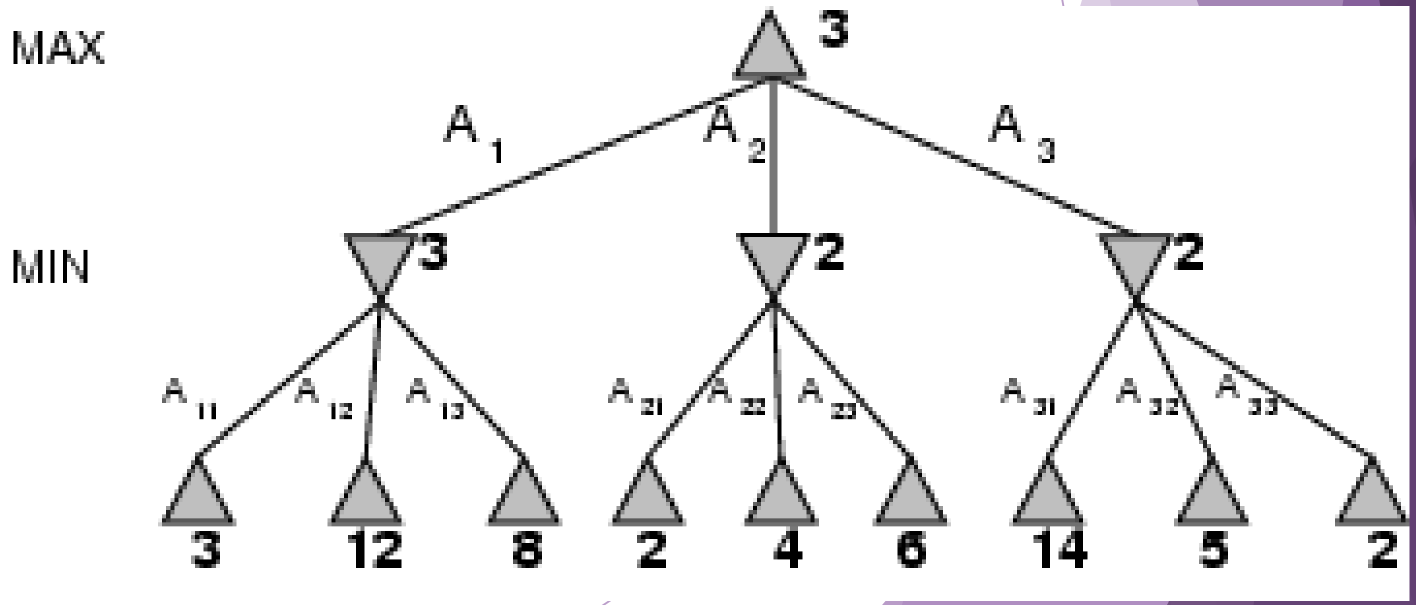
Game tree

2-player
Max & Min deterministic
take turns

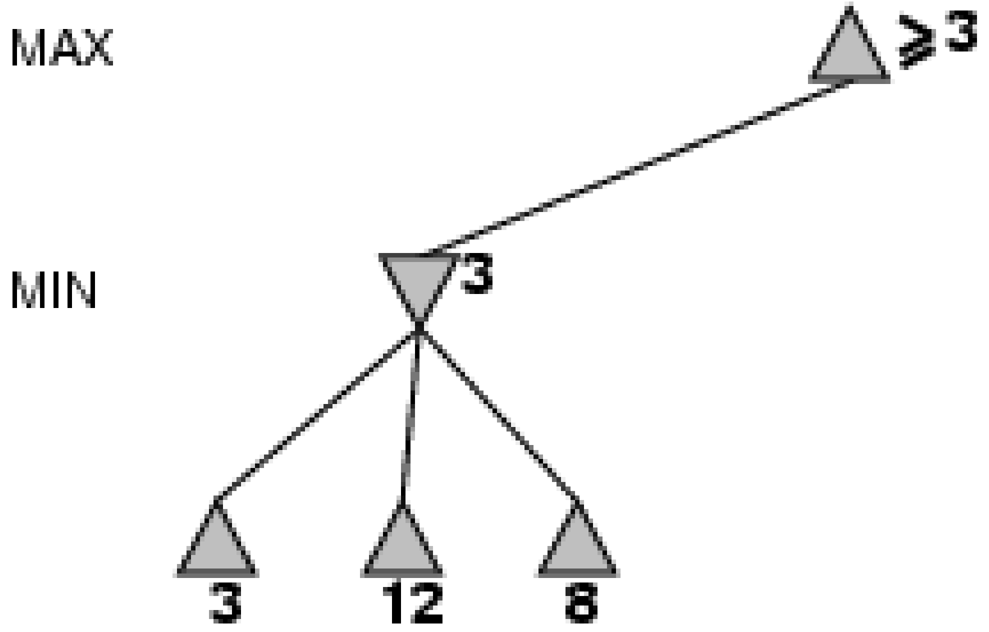


Minimax

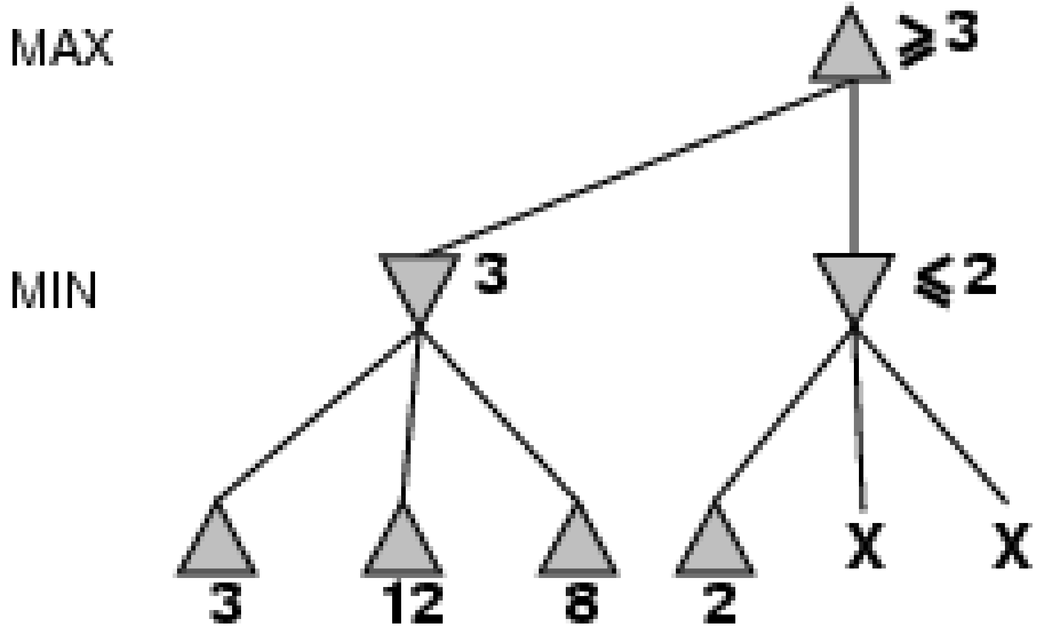
- Perfect play for deterministic games
- Idea: choose the move to position with highest minimax value
  - = best achievable payoff against best play
- E.g., 2-ply game:



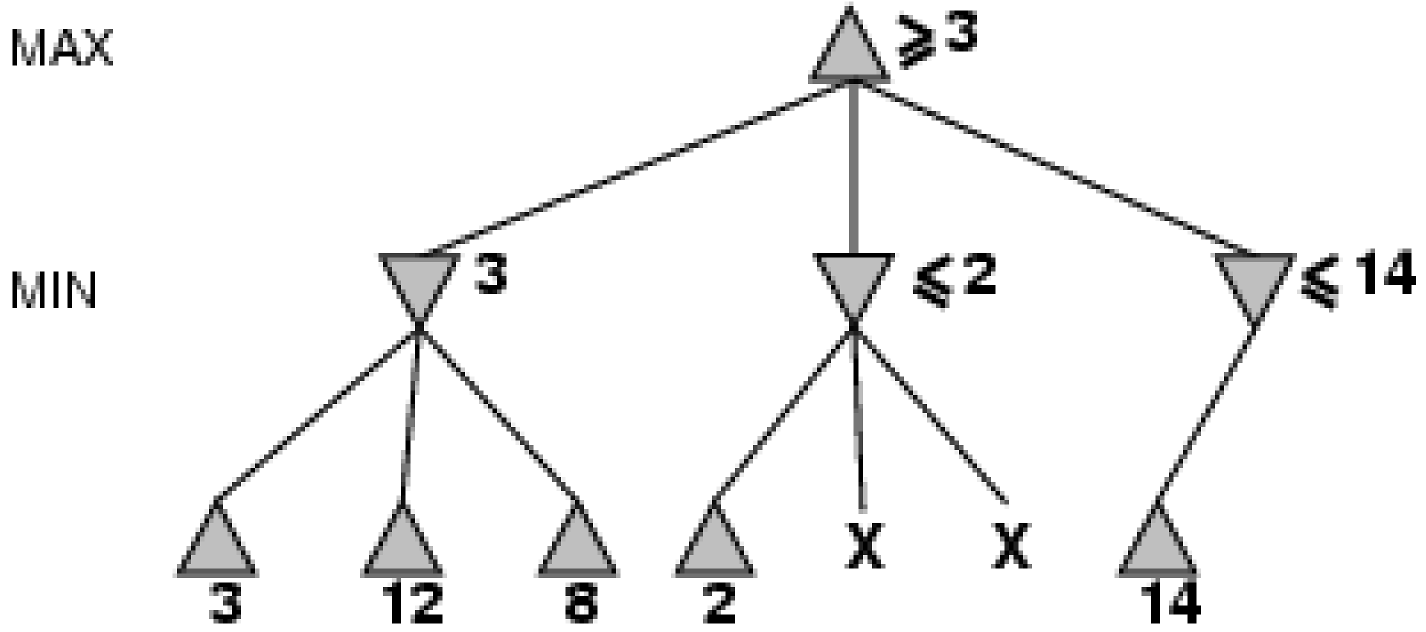
α-β pruning example



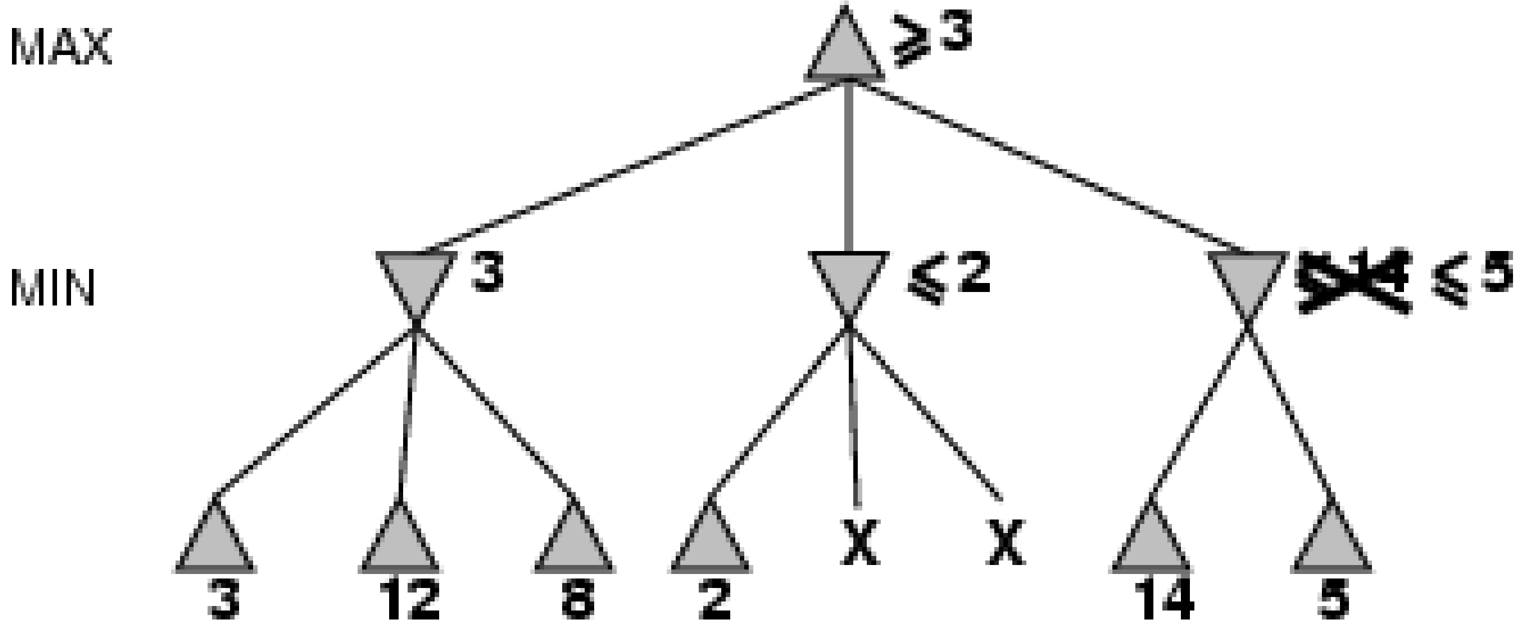
α-β pruning example



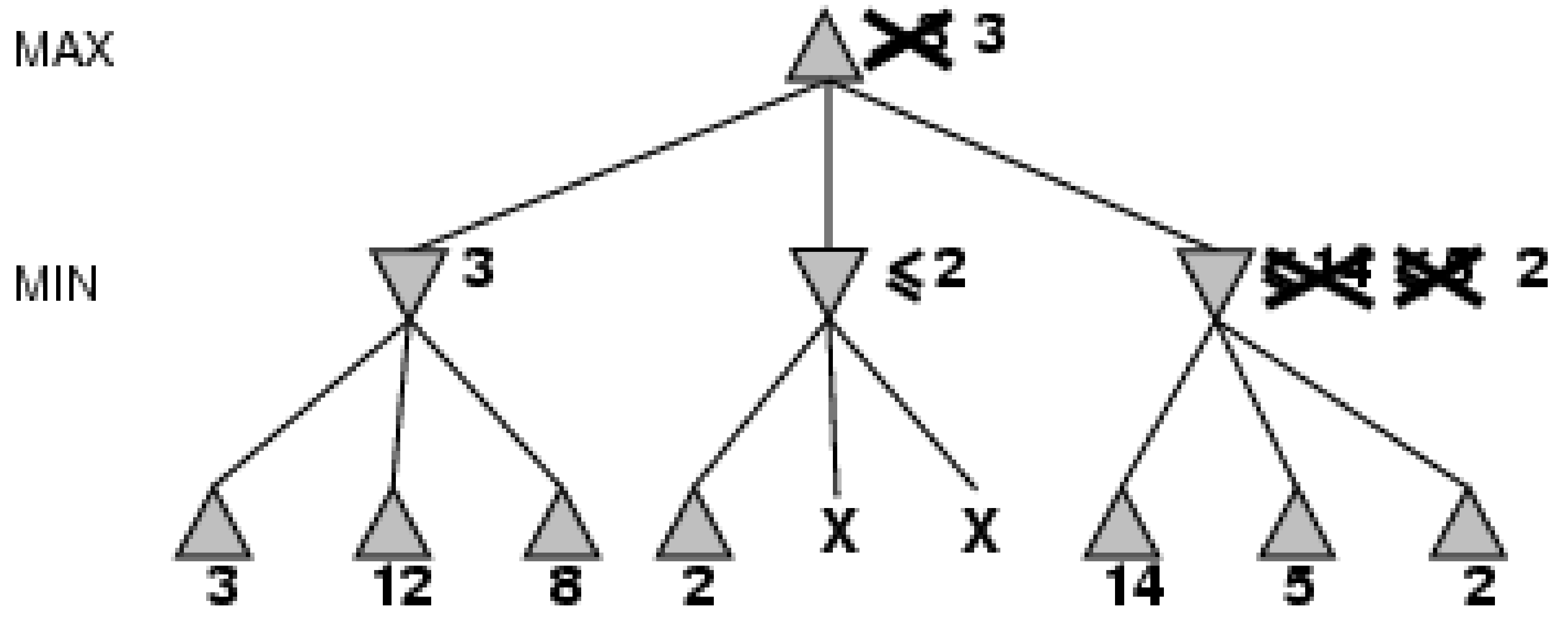
α-β pruning example



α-β pruning example

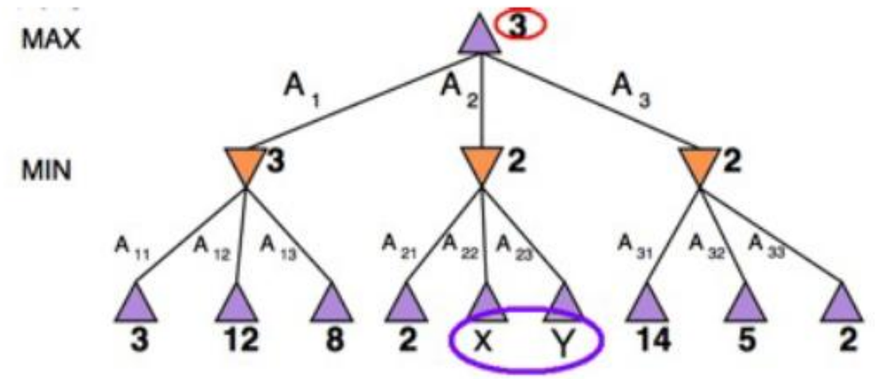


α-β pruning example



The values (X) are irrelevant as the max of (3, <=2, 2) is 3. The branches that stem from these two nodes will not affect the outcome therefore can be pruned.

calculations



$$\begin{aligned} Minimax(root) &= max(min(3,12,8), min(2,X,Y), min(14,5,2)) \\ &= max(3, min(2,X,Y), 2) \\ &= max(3, Z, 2) \qquad \text{where } Z = min(2,X,Y) \leq 2 \\ &= 3 \end{aligned}$$

Minimax decisions are independent of the values of X and Y.

Stochastic Games

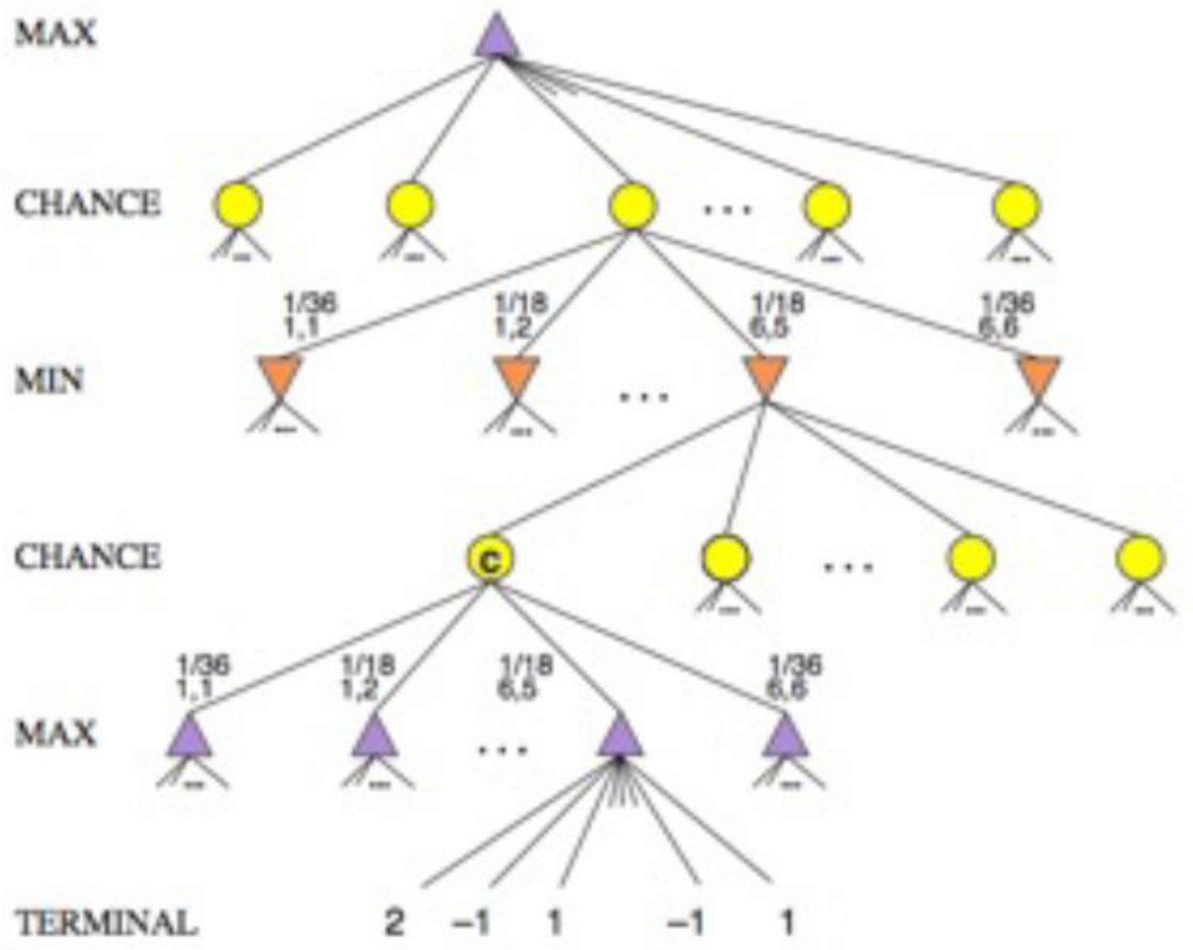
- Includes chance - a random element
  - e.g., throwing a die
- Include chance nodes.

Backgammon:

- old board game combining skills and chance.
- The goal is that each player tries to move all of his pieces off the board before his opponent does.

Ptkfgs [Public domain], via Wikimedia Commons

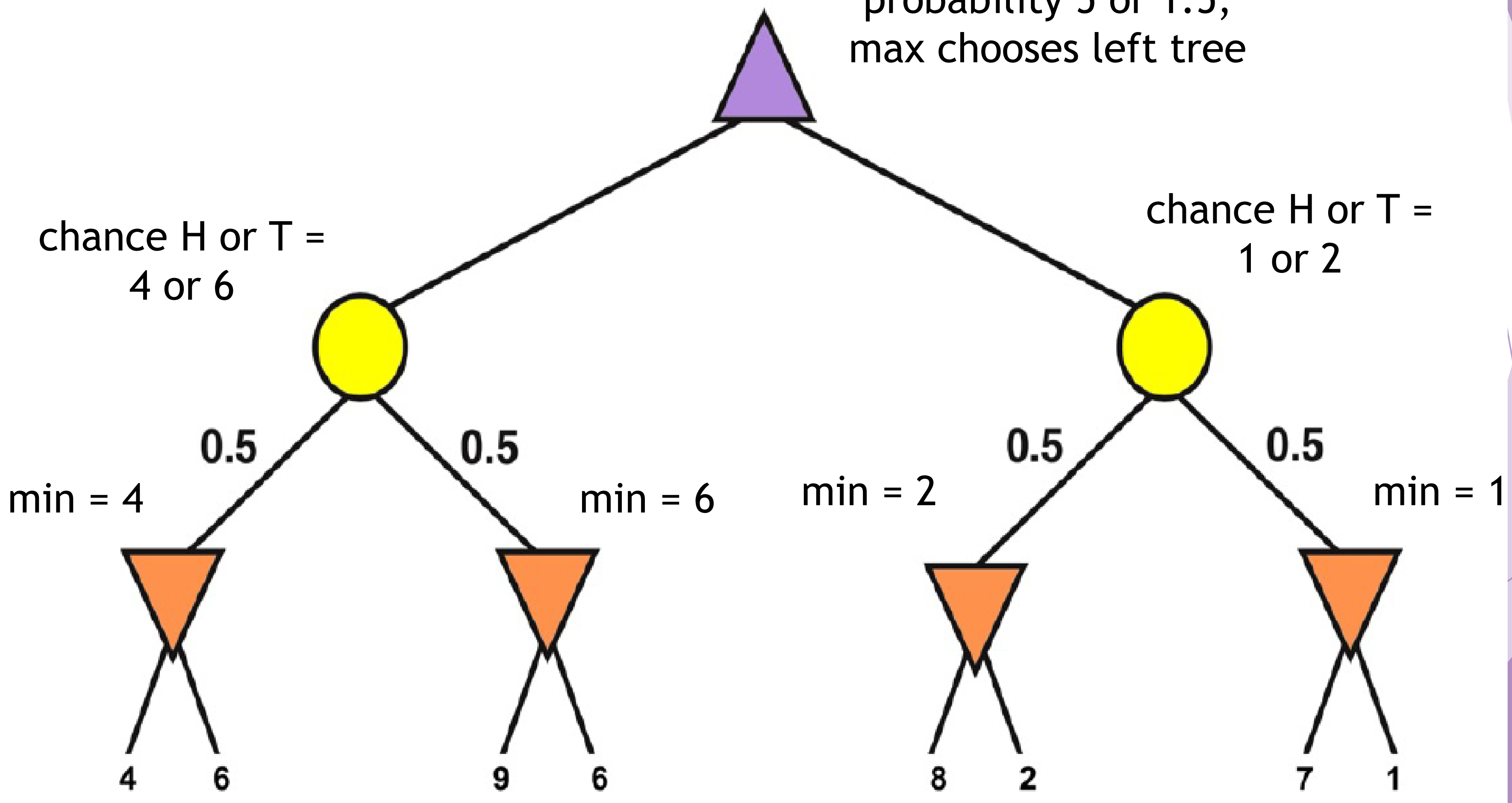
Chance Games



Partial game tree for Backgammon.

Chance Games

Example with coin-flipping:



CSP Definition

- A constraint satisfaction problem consists of three elements:

  – A set of variables, $X = \{X1, X2, \cdot \cdot \cdot Xn\}$

  – A set of domains for each variable: $D = \{D1, D2, \cdot \cdot \cdot Dn\}$

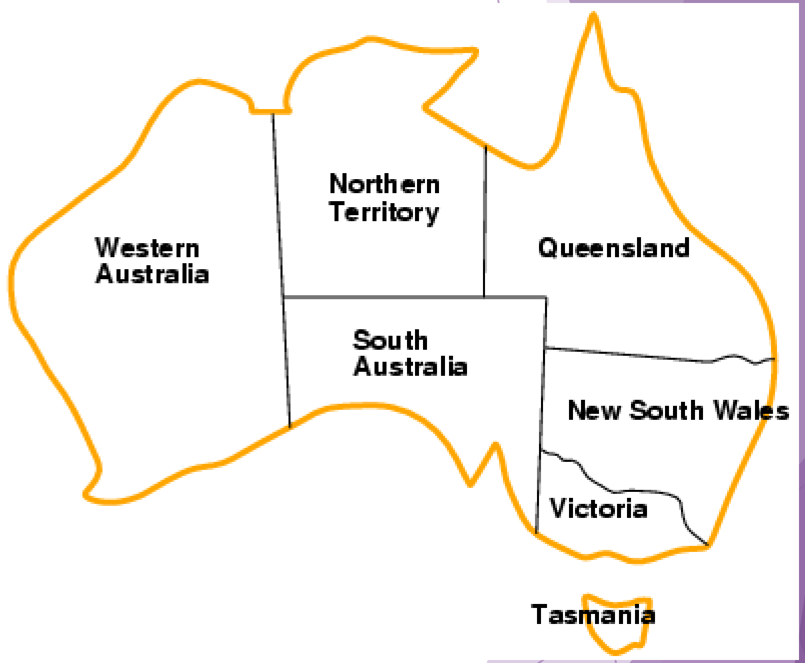
  – A set of constraints C that specify allowable combinations of values.

- Solving the CSP: finding the assignment(s) that satisfy all constraints.
- Concepts: problem formalization, backtracking search, arc consistency, etc.
- We call a solution, a consistent assignment.

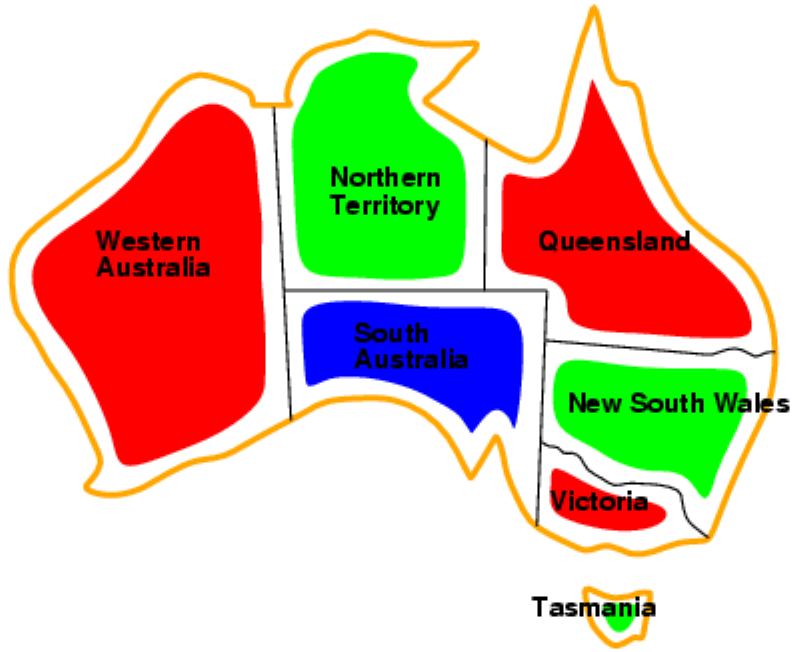
Example: Map-Coloring

- Variables *WA, NT, Q, NSW, V, SA, T*
- Domains D_i = {red,green,blue}
- Constraints: adjacent regions must have different colors

- e.g., WA ≠ NT, or (WA,NT) in {(red,green),(red,blue),(green,red), (green,blue),(blue,red),(blue,green)}



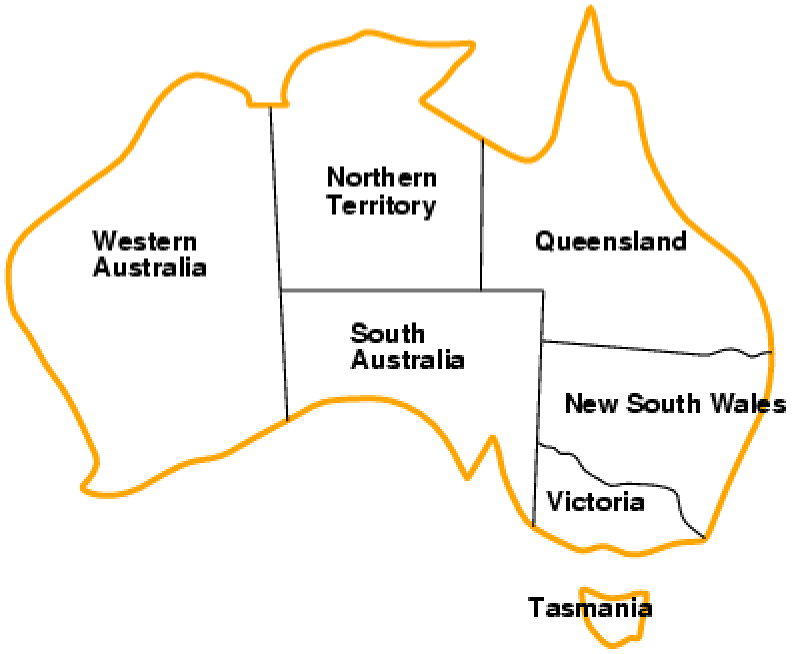
Example: Map-Coloring

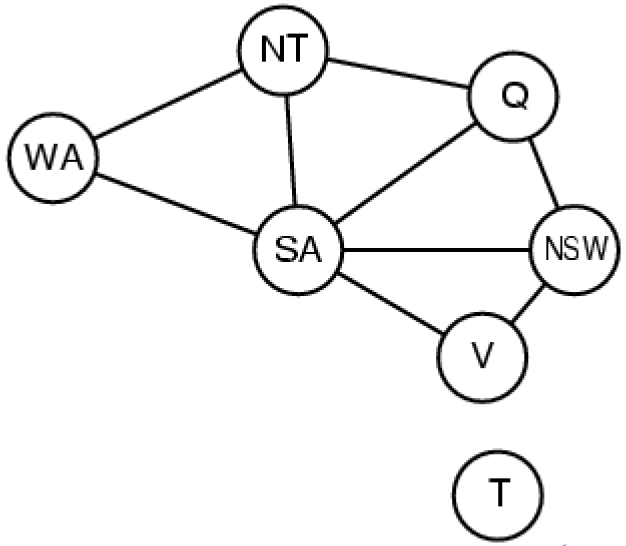


Solutions are complete and consistent assignments, e.g., WA = red, NT = green,Q = red,NSW = green,V = red,SA = blue,T = green

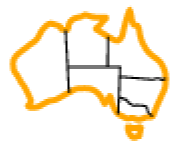
Constraint graph

- Binary CSP: each constraint relates two variables
- Constraint graph: nodes are variables, arcs are constraints

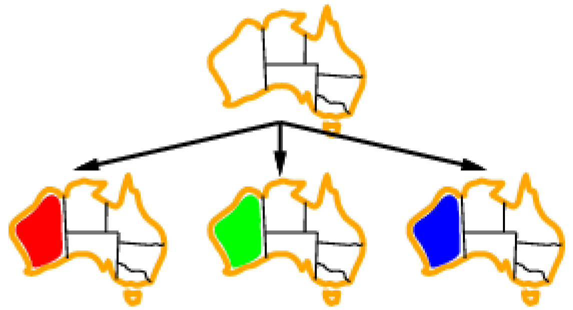




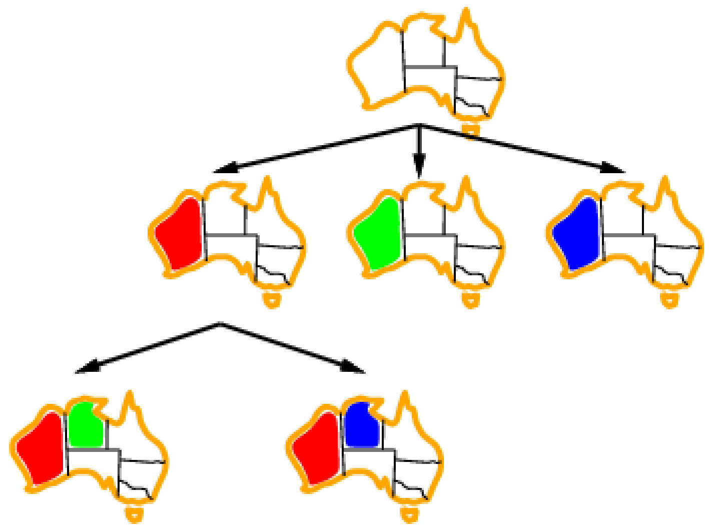
Backtracking example



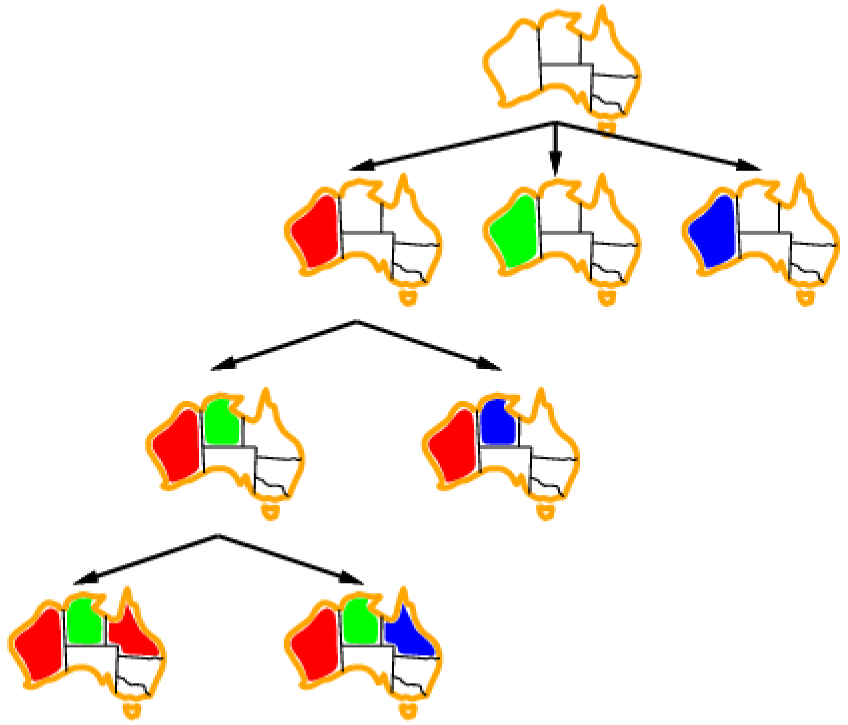
Backtracking example



Backtracking example



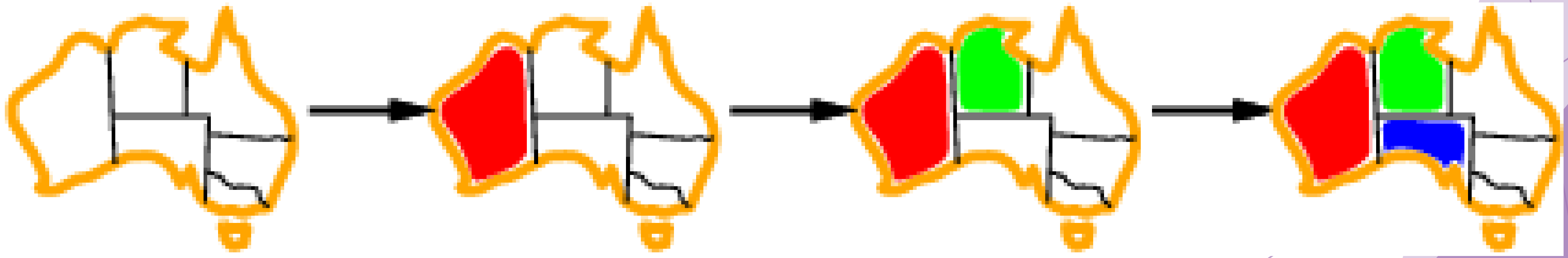
Backtracking example



Most constrained variable

- Most constrained variable:
  choose the variable with the fewest legal values
- a.k.a. minimum remaining values (MRV) heuristic

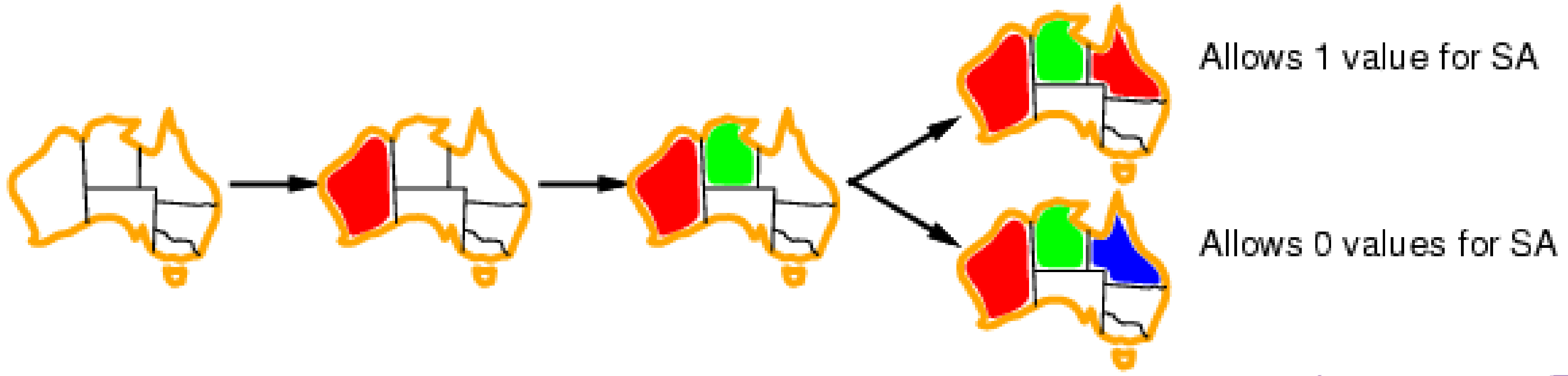
pick the hardest



Least constraining value

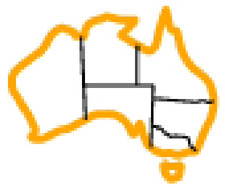
- Given a variable, choose the least constraining value:
  - the one that rules out the fewest values in the remaining variables
- Combining these heuristics makes 1000 queens feasible

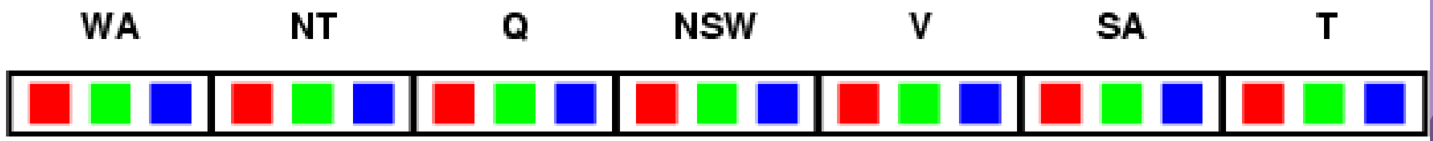
pick the ones that are likely to work



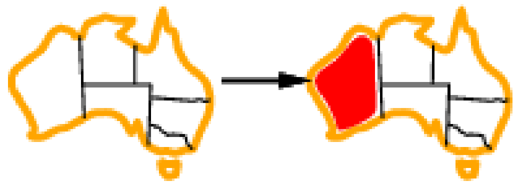
Forward checking

- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values



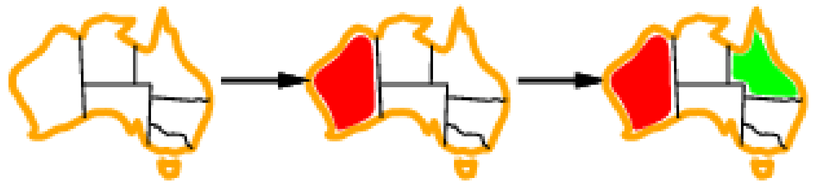


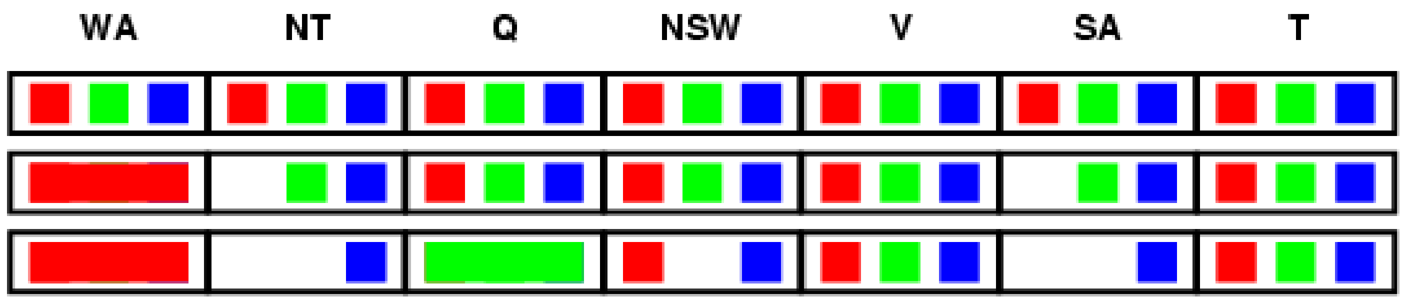
Forward checking



WA	NT	Q	NSW	V	SA	T
🟥 🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦
🟥🟥🟥	🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦	🟥 🟩 🟦	🟩 🟦	🟥 🟩 🟦

Forward checking

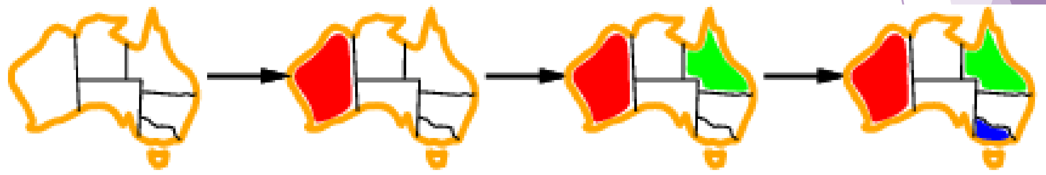




WA	NT	Q	NSW	V	SA	T
🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦
🟥	🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟩🟦	🟥🟩🟦
🟥	🟦	🟩	🟥🟦	🟥🟩🟦	🟦	🟥🟩🟦

Forward checking

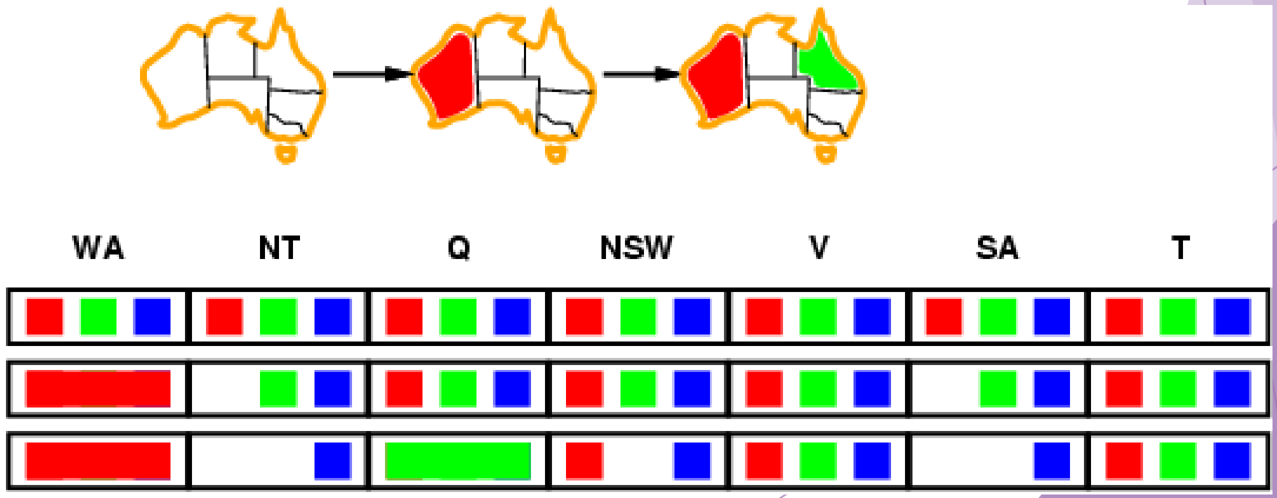
- Now Southern Australia has no options remaining – therefore this is not a possible solution



WA	NT	Q	NSW	V	SA	T
🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦
🟥	🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟩🟦	🟥🟩🟦
🟥	🟦	🟩	🟥🟦	🟥🟩🟦	🟦	🟥🟩🟦
🟥	🟦	🟩	🟥	🟦		🟥🟩🟦

Constraint propagation

- Forward checking propagates information from assigned to unassigned variables, but doesn't provide early detection for all failures:
- NT and SA cannot both be blue!
- Constraint propagation repeatedly enforces constraints locally

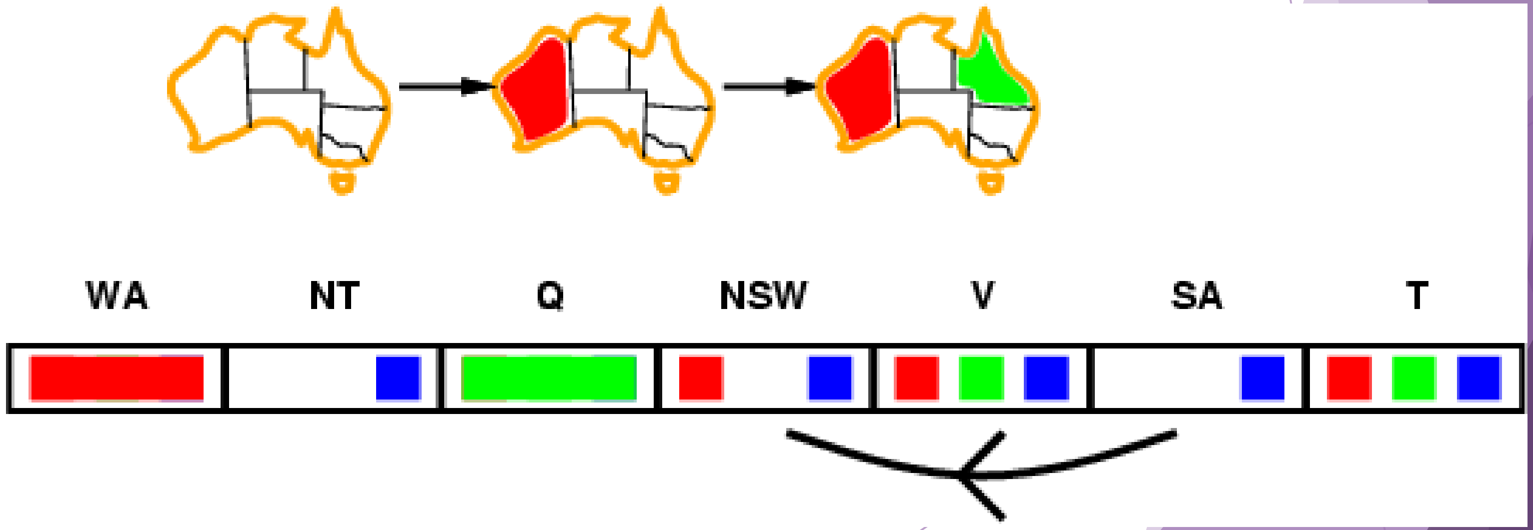


WA	NT	Q	NSW	V	SA	T
🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦
🟥	🟩🟦	🟥🟩🟦	🟥🟩🟦	🟥🟩🟦	🟩🟦	🟥🟩🟦
🟥	🟦	🟩	🟥🟦	🟥🟩🟦	🟦	🟥🟩🟦

Arc consistency

- Simplest form of propagation makes each arc consistent
- *X* → *Y* is consistent iff

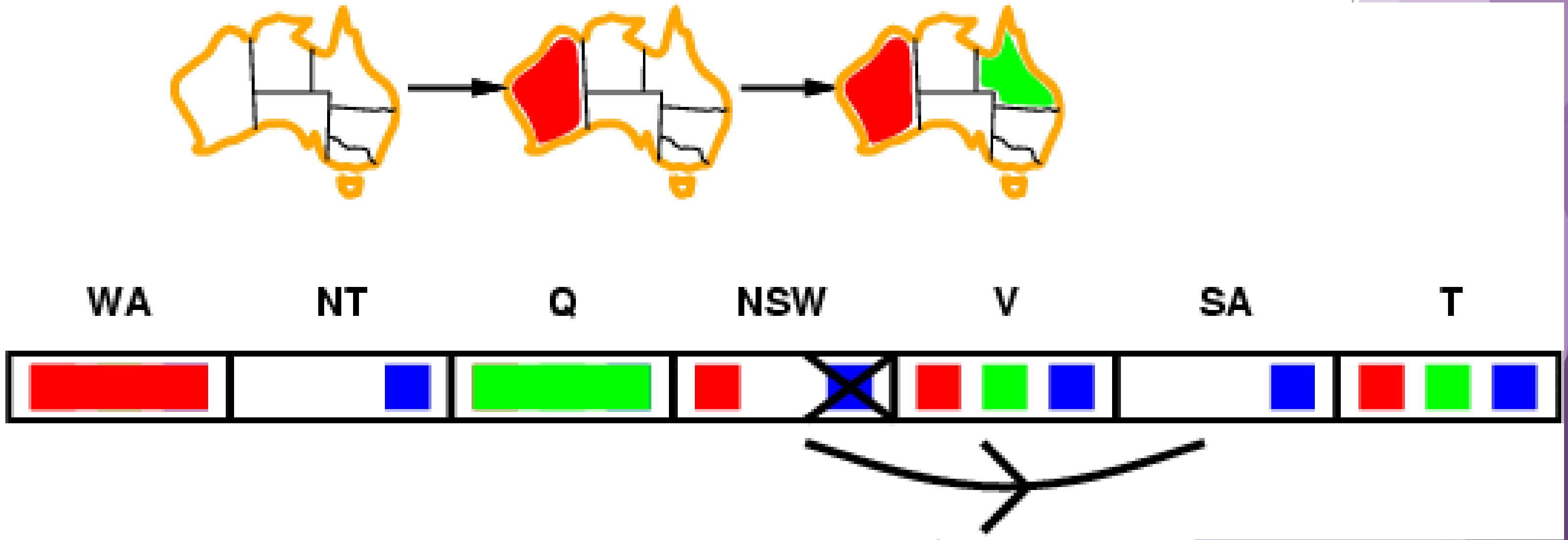
  for every value *x* of *X* there is some allowed *y*



Arc consistency

- Simplest form of propagation makes each arc consistent
- *X* →*Y* is consistent iff

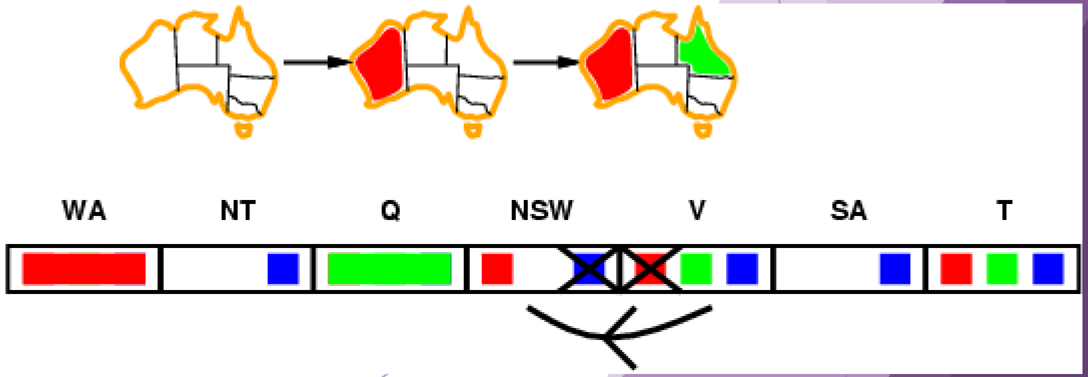
  for every value *x* of *X* there is some allowed *y*



Arc consistency

- Simplest form of propagation makes each arc consistent
- *X* → *Y* is consistent iff

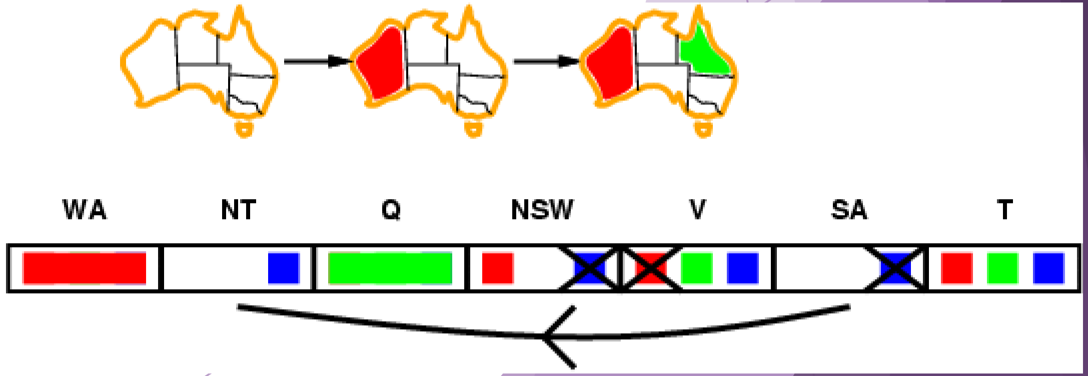
  for every value *x* of *X* there is some allowed *y*
- If *X* loses a value, neighbors of *X* need to be rechecked



Arc consistency

- Simplest form of propagation makes each arc consistent
- *X* →*Y* is consistent iff

  for every value *x* of *X* there is some allowed *y*
- If *X* loses a value, neighbors of *X* need to be rechecked
- Arc consistency detects failure earlier than forward checking
- Can be run as a preprocessor or after each assignment



Complexity

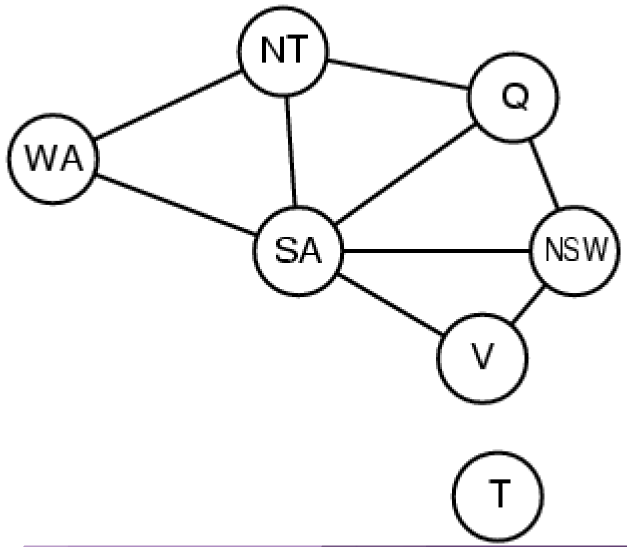
Example:

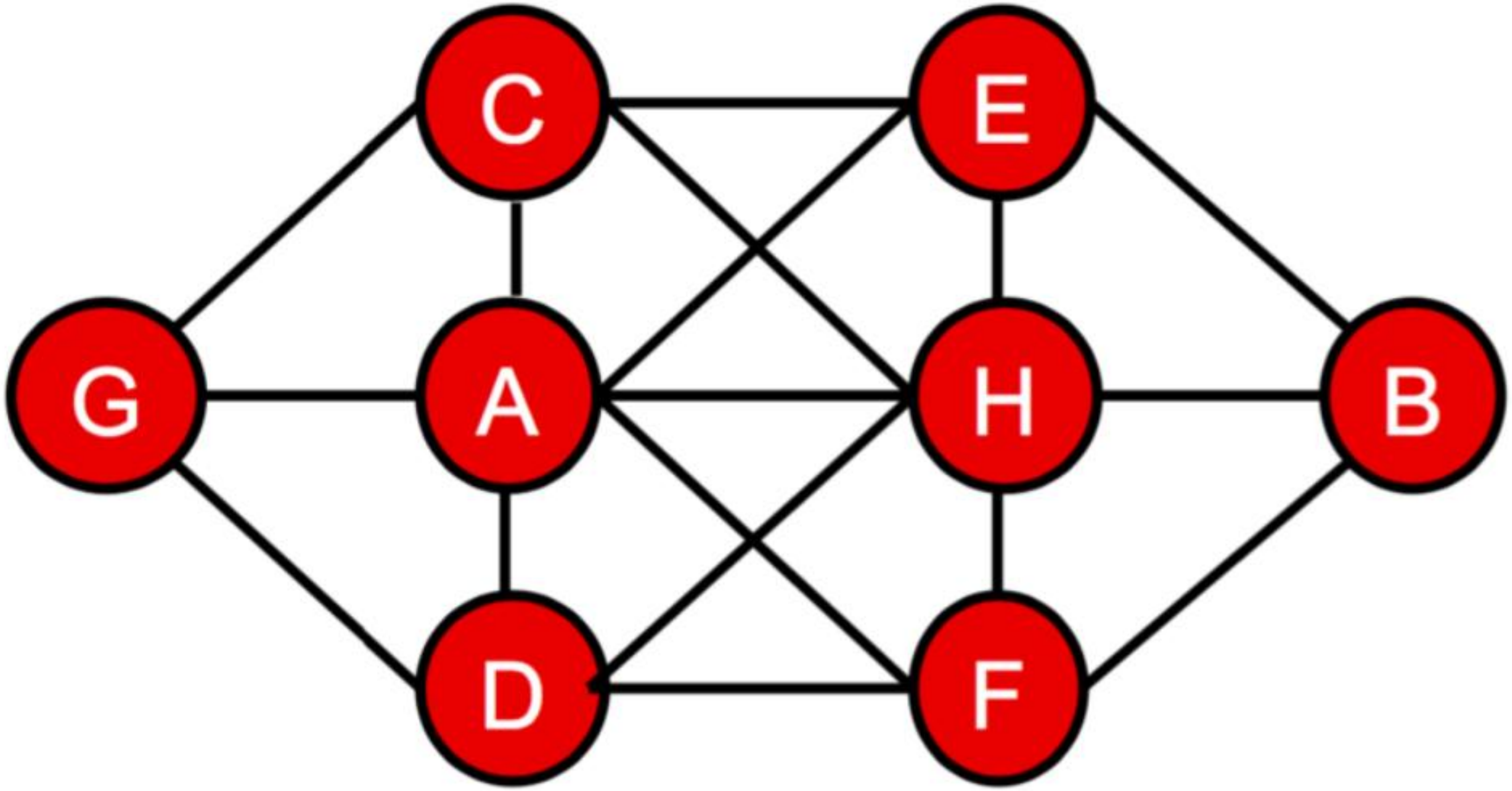
Assume n = 80, d = 2.

- Assume we can decompose into 4 subproblems with c = 20.
- Assume processing at 10 million nodes per second.
- Without decomposition of the problem we need:

  $2^{80} = 1.2*10^{24}$ - take 3.83 million years!
- With decomposition of the problem we need:

  $4 * 2^{20} = 4.2*10^{6-}$ reduced to 0.4 seconds




C
E
G
A
H
B
D
F

What is
Artificial Intelligence now?

we need to understand
Natural Language Processing

AI journey

When we think of Artificial Intelligence we think of ChatGPT (Large Language Models called LLMs)

but, where did they come from?

AI & NLP

what was AI/NLP research like 25 years ago?

Why? What is NLP?

Natural Language Processing (NLP) deals with the research of how to apply computational techniques on human language

- AI subfield
- ‘demanding’ area’ for research
- results notoriously difficult
- “can’t teach it to students as they would fail the course”

NATURAL LANGUAGE PROCESSING

Text Processing

- **Parsing**: Dividing text into meaningful units, like words or phrases, and structuring them to understand grammar and syntax
- **Semantic Analysis**: Determining the meaning of words in context (e.g. sentiment analysis)
- **Named Entity Recognition (NER)**: Identifying proper nouns like names of people, places, dates
- **Text Generation**: Creating new text based on input, such as summaries or articles (e.g., GPT models)

Speech Processing

- **Speech Recognition**: Converting spoken language to text
- **Speech Synthesis**: Converting text back to human-sounding speech (TTS)
- **Intent Recognition**: Understanding the purpose or request in user queries
- **Dialog Management**: Structuring back-and-forth conversation flows in a logical way.

Other NLP areas

- **Information Retrieval**
- **Information Extraction**
- **Machine Translation**
- **Question Answering Systems**
- **Co-reference Resolution**

Research Lab – research included

- information overload
  - overload of information online – information extraction
- pattern recognition
  - text-based (e.g. finding patterns in text)
  - image-based (e.g. snake skin)
  - anti-spam & content-based filtering
    - spam was a problem
    - inappropriate email (e.g. pornography)
- speech
  - speech markup language (XML)
  - Text-to-Speech systems (SAPI)
  - translation to sign language
- *other research included – NLP (WSD),Thai NLP, Computer Vision (VR gloves)*

According to a study by Websense in the early 2000s, **up to 30% of workplace internet use was unrelated to work tasks**, with some portion of this time spent on adult sites

2005 American Management Association survey reported **that 76% of companies monitored workers' website visits**

and Language?

it’s difficult

NLP / AI 10 years ago

What is a conversation agent?
HAL 9000 (from 2001: A Space Odyssey)

Dave: *Open the pod bay doors, Hal.*
HAL: *I'm sorry Dave, I'm afraid I can't do that*

Virtual assistants:
- Siri, Apple
- Cortana, Microsoft
- M, Facebook
- Alexa, Amazon

http://www.aljazeera.net/NR/exeres/8FD54E7F-56C5-49A0-B60A-89A67426F3B3.htm

نحضر لك الأخبار الساخنة أينما تكون
اشترك الآن

أنان في الناقورة لتفقد الخط الأزرق والمناطق المدمرة

الثلاثاء 5/8/1427 هـ - الموافق 29/8/2006 م (آخر تحديث) الساعة 17:44 (مكة المكرمة)، 14:44 (غرينتش)

محكمة إسرائيلية تمدد اعتقال وزير المالية الفلسطينية و13 نائبا من حماس
أنان يصل إسرائيل في ثاني محطة

استشهاد فلسطينيين وإصابة تسعة في غارات بالضفة والقطاع
أصيب تسعة فلسطينيين بينهم مدنيون في غارة جوية إسرائيلية على حي الشجاعية في قطاع غزة. يأتي ذلك مباشرة بعد استشهاد قائدين بارزين من كتائب شهداء الأقصى في عملية لقوات الاحتلال الإسرائيلي نفذها سلاح الجو وقوات المشاة في مخيم بلاطة بالضفة الغربية.

البشير يلتقي فريزر ومجلس الأمن لن يفرض قوات بدارفور
من المقرر أن يلتقي الرئيس السوداني عمر البشير جينداي فريزر مساعدة وزيرة الخارجية الأميركية التي تحاول في الخرطوم إقناع المسؤولين السودانيين بنشر قوة أممية بدارفور. من جانبه قال السفير الأميركي في الأمم المتحدة إنه لا نية لمجلس الأمن بفرض قوات في الإقليم.

رمسفيلد وتشيني يصران على إبقاء القوات الأميركية بالعراق
دعا وزير الدفاع الأميركي دونالد رمسفيلد الأميركيين إلى التحلي بالصبر بخصوص العراق. وانتهز ديك تشيني نائب الرئيس دعوات الديمقراطيين لسحب القوات الأميركية من العراق للربط بين الانسحاب المبكر واحتمال وقوع هجمات داخل الولايات المتحدة.

مقتل مدنيين وإصابة ضابط بهجوم انتحاري في أفغانستان
أعلنت القوة الدولية للمساعدة على إرساء الأمن (إيساف) مقتل مدنيين وإصابة ضابط أفغاني في هجوم استهدف قافلة لقوات الأطلسي بجنوب أفغانستان. وفي العاصمة كابل انفجرت قنبلة يدوية الصنع لدى مرور

I made her duck

1. I cooked *[animal-duck]* for her to eat
2. I cooked *[animal-duck]* belonging to her
3. I created the *[plastic-duck]* she owns
4. I caused her to quickly lower her head or body
5. I waved my magic wand and turned her into a *[animal-duck]*

Probabilistic Language Models

assign a probability to a sentence

- Machine Translation:
  - P(**high** winds tonight) > P(**large** winds tonight)

- Spelling Correction
  - The office is about fifteen **minuets** from my house
    - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)

- Speech Recognition
  - P(I saw a van) >> P(eyes awe of an)

Probabilistic Language Models

probability of an upcoming word:

$$P(w_5|w_1,w_2,w_3,w_4)$$

compute the probability of a sentence or sequence of words:

$$P(W) = P(w_1,w_2,w_3,w_4,w_5\ldots w_n)$$

Chain Rule of Probability

P(“water is so transparent”) =

P(is|water) × P(so|water is) × P(transparent|water is so)

we could count and divide? Too many possible sentences

Markov Assumption

P(the | its water is so transparent that) *similar to*

P(the | transparent that)

How many?

2 words = bigram

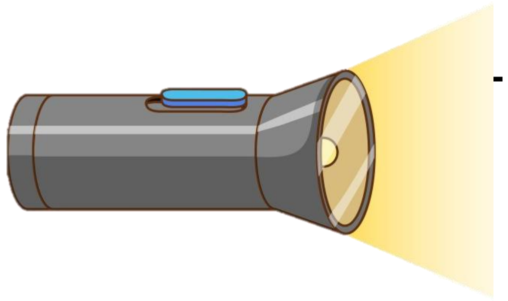
3 words = trigram

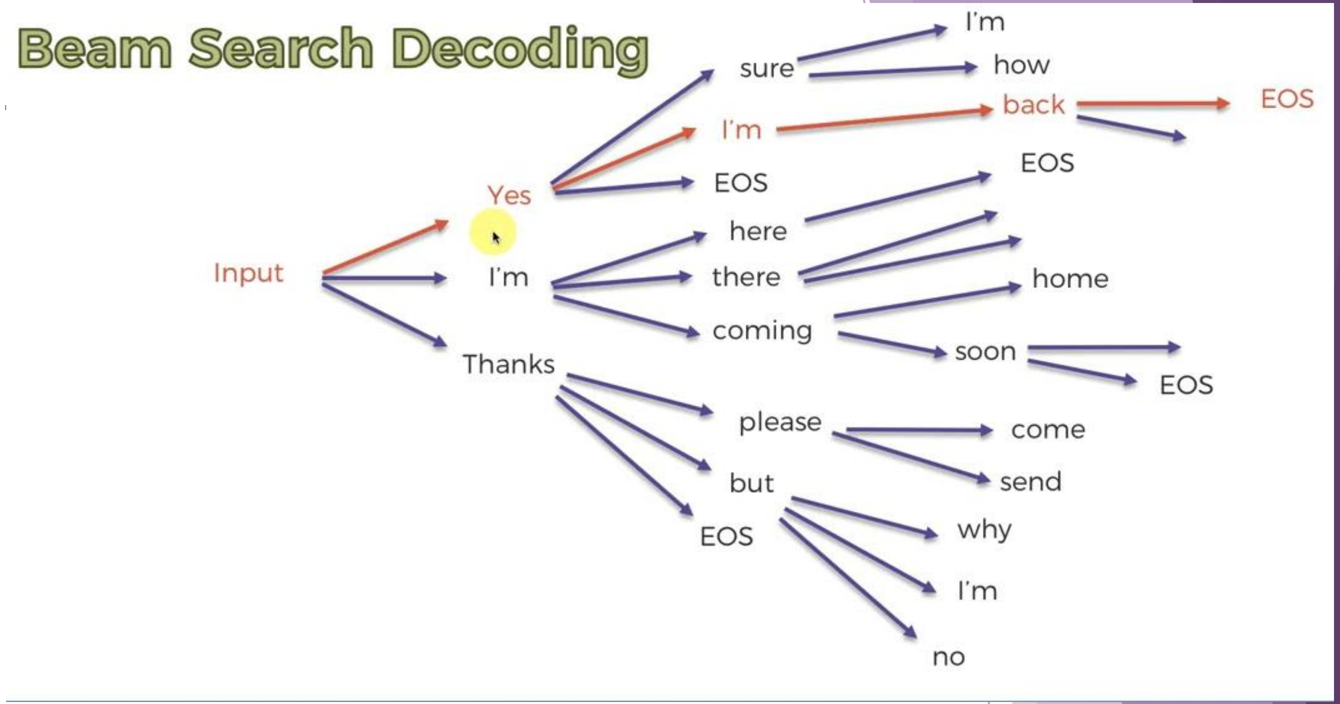
4 words = 4-gram

5 words = 5-gram

called n-grams

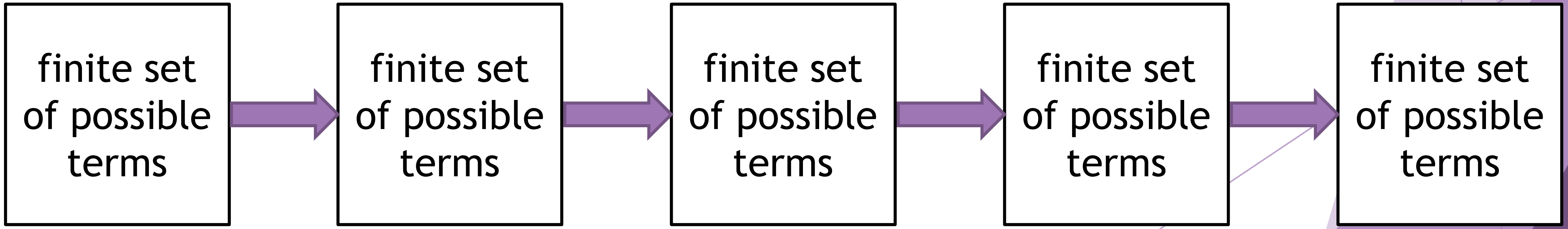
but we have long distance dependencies





- English 500,000+ word dictionaries
- probability of next word - too computationally expensive
- use beam search

$P(w_5|w_1,w_2,w_3,w_4)$


finite set of possible terms
finite set of possible terms
finite set of possible terms
finite set of possible terms
finite set of possible terms

but it has all changed!

the neural network revolution

Neural Networks

- We have known about neural networks for a long time
- It is said the first Artificial Neural Networks were **developed in the 1940s**
- 1983-1985 **Geoffrey Hinton** used tools from statistical physics to create the Boltzmann machine
- Even Recurrent neural networks (RNN) (used for NMT) was written about **in 1997 [1]**

But now we have the data and the computing power

[1] Castaño, Asunción; Casacuberta, Francisco (1997). A connectionist approach to machine translation. 5th European Conference on Speech Communication and Technology (Eurospeech 1997). Rhodes, Greece. pp. 91–94. doi:10.21437/Eurospeech.1997-50.

Thai-English Machine Translation

is a good example of what has happened to AI and NLP

the story

Would Google Translation stay number one?

Who could develop the first successful NMT system?

TPUs and how long to converge

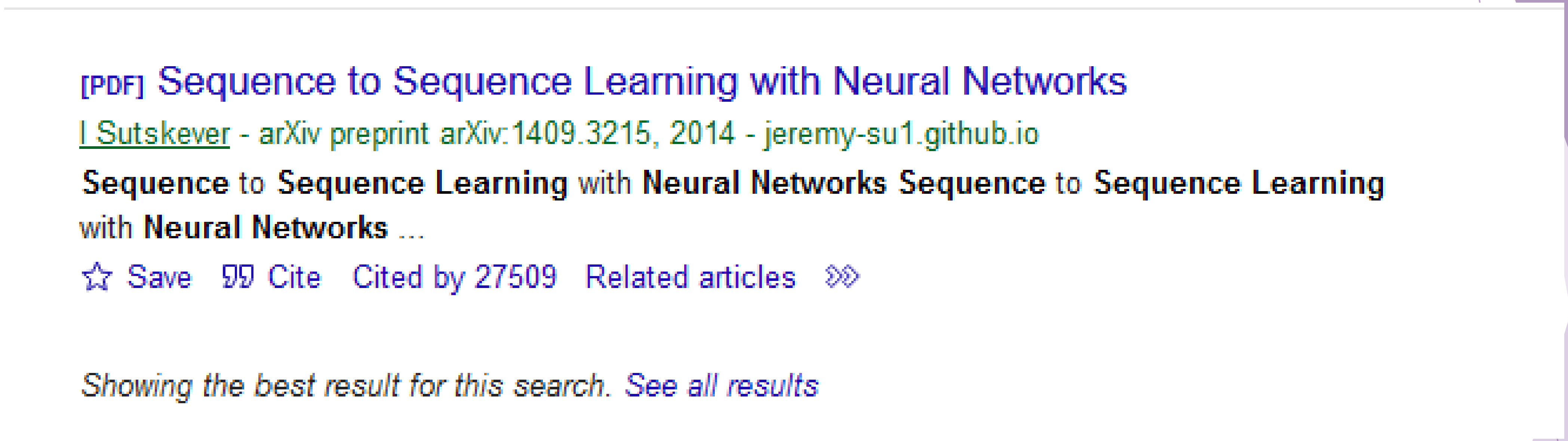
NMT = Neural Machine Translation
TPU = Tensor Processing Unit

the facts

- Google developed the first successful NMT system
- in 2016 Google launched its NMT-based Google Translate for several language pairs.
- Google did leverage extensive computational resources, including GPUs initially, and later integrated its proprietary TPUs to speed up inference and training times.
- However, there isn’t strong evidence that the initial NMT system specifically took three months to converge using TPUs.
- Research in Canada: The University of Montreal, led by **Yoshua Bengio** and contributions from **Kyunghyun Cho** and others, was among the pioneering teams in neural machine translation.
- They helped develop early NMT models and contributed to the broader research that Google later built upon.

the *(small)* AI world

Ilya Sutskever



[PDF] Sequence to Sequence Learning with Neural Networks

I Sutskever - arXiv preprint arXiv:1409.3215, 2014 - jeremy-su1.github.io

Sequence to **Sequence Learning** with **Neural Networks Sequence** to **Sequence Learning** with **Neural Networks** ...

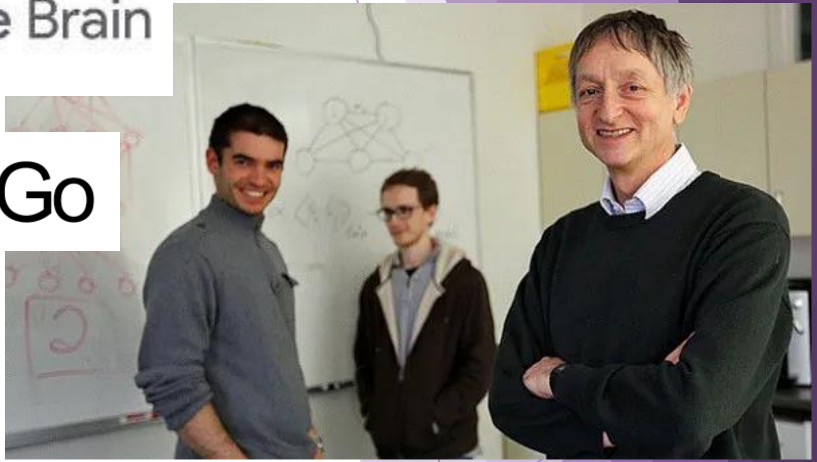
☆ Save 99 Cite Cited by 27509 Related articles »

Showing the best result for this search. See all results

Ilya Sutskever

In 2012, Sutskever spent about two months as a postdoc with **Andrew Ng** at Stanford University. He then returned to the University of Toronto and joined **Hinton's** new research company DNNResearch, a spinoff of **Hinton's** research group.

In 2013, Google acquired DNNResearch and hired Sutskever as a research scientist at Google Brain.
At **Google Brain**, Sutskever worked with Oriol Vinyals and Quoc Viet Le to create the sequence-to-sequence learning algorithm, and worked on TensorFlow.
He is also one of the AlphaGo paper's many co-authors.



At the end of 2015, Sutskever left Google to become cofounder and chief scientist of the newly founded organization **OpenAI.**
Sutskever is considered to have played a key role in the development of **ChatGPT.**

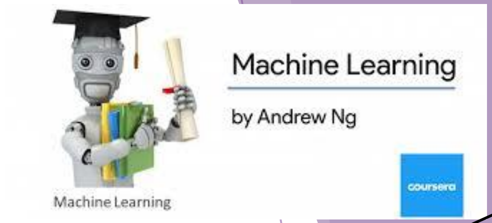
the ai world

early 2010's

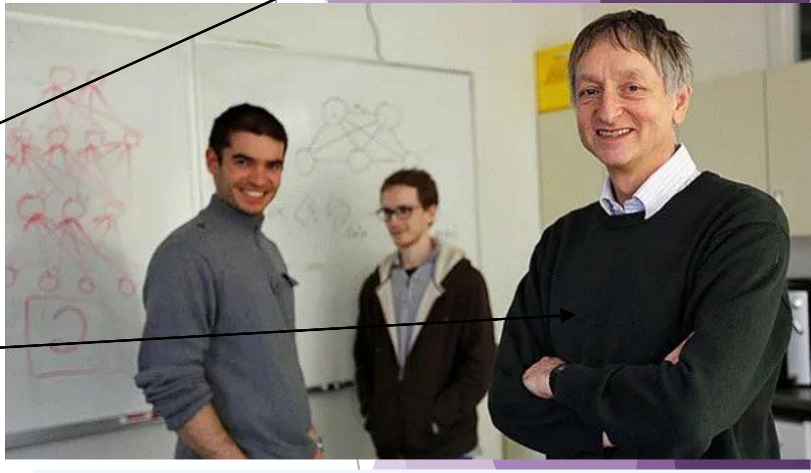
- studied machine learning online with **Andrew Ng**
- studied neural networks online with **Geoffrey Hinton**

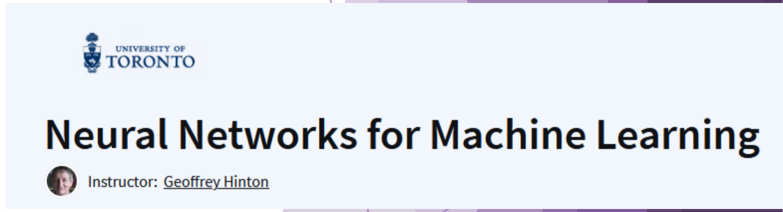
now

- retrieval augmented generation (RAG)
- AI safety and ethical considerations


Machine Learning

by Andrew Ng





Neural Networks for Machine Learning

Instructor: Geoffrey Hinton

progress

progress

iRobot Roomba (2002) The first commercially successful robot vacuum

https://vacuumwars.com/history-of-the-robot-vacuum-cleaner/

robot vacuum cleaners

22 years ago - not very good

22 years later - not very good

progress

- speech recognition (NLP in personal assistant)
  - except Scottish accent
  - but we still use keyboards & mice
- pattern recognition
  - object recognition – nearly at 100% world objects

ARTIFICIAL INTELLIGENCE

early 2010s

The adoption of **deep learning techniques**, led to significant advancements in NLP tasks such as machine translation, sentiment analysis, and speech recognition.

Deep learning models outperformed traditional methods, marking a shift towards more complex and capable AI systems

late 2010s

The introduction of **the transformer** (2017) allowed for more efficient processing of language data.

Leading to breakthroughs in translation and text generation tasks.

Transformers became the backbone for subsequent large language models (LLMs).

2020s

The emergence of LLMs transformed user interaction with AI.

The surge in generative AI applications has led to new creative possibilities in content generation.

This boom has also raised ethical concerns regarding misinformation and copyright issues, prompting discussions about responsible AI use and regulation.

The **widespread integration of AI technologies** into consumer products and services—such as virtual assistants, customer service bots, and content recommendation systems—has made AI a ubiquitous part of daily life.

This has increased public awareness and demand for further advancements and improvements.

MACHINE TRANSLATION

Statistical

- analyzed bilingual text corpora
- identify patterns and rules for translation
- breaks down sentences
- translating these units
- based on probabilities
- from training data
- dependent on the training data quality and quantity

Neural

- deep learning models
- used vast amounts of data
- handling long-range dependencies
- more natural-sounding results
- improving fluency and coherence

Attention

- focus on specific parts of the input
- when generating each word output
- improving accuracy for longer sentences
- has led to substantial improvements in translation quality,
- particularly for complex sentence structures

Transformers

- replaced recurrent neural networks
- with self-attention mechanisms allowing for parallelization
- significantly increasing efficiency
- Transformers have excelled in handling diverse languages and contexts,
- making them the current state-of-the-art.

LANGUAGE MODELS

Statistical

- Statistical Language Models operate on probabilistic principles,
- analyzing large corpora of text
- learn the likelihood of word sequences using n-grams.
- struggle with long-range dependencies
- resulting in less coherent and accurate outputs

Neural

- Neural Language Models created better language representation
- models could capture more complex patterns in data.
- They improved fluency and contextual understanding
- but still had scalability and language diversity problems.

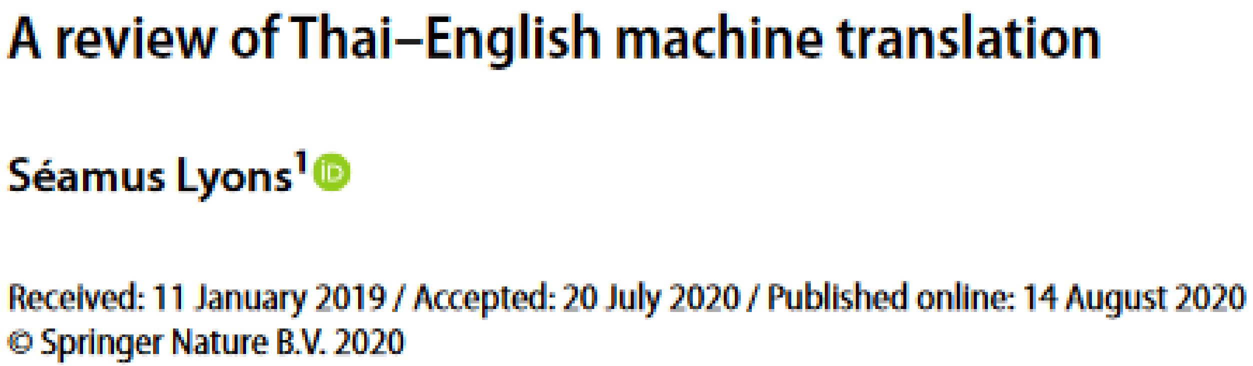
Attention

- allowed models to focus on relevant parts of the input sequence
- when generating outputs.
- improved how models handled context.
- enhanced performance in tasks requiring understanding of longer contexts
- leading to more coherent outputs

Transformers

- revolutionized language modelling
- enabled the training of much larger models (LLMs)
- with billions of parameters,
- leading to unprecedented improvements in various NLP tasks
- such as translation, summarization, and question-answering.

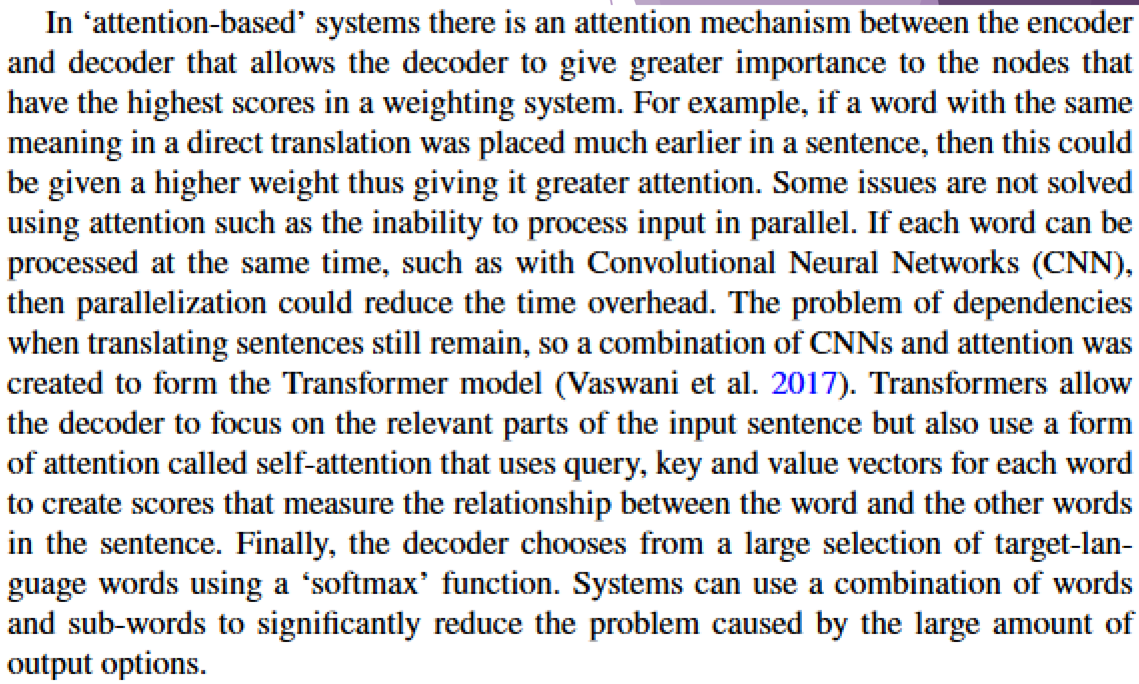
the speed of progress



A review of Thai–English machine translation

Séamus Lyons[1]

Received: 11 January 2019 / Accepted: 20 July 2020 / Published online: 14 August 2020



In 'attention-based' systems there is an attention mechanism between the encoder and decoder that allows the decoder to give greater importance to the nodes that have the highest scores in a weighting system. For example, if a word with the same meaning in a direct translation was placed much earlier in a sentence, then this could be given a higher weight thus giving it greater attention. Some issues are not solved using attention such as the inability to process input in parallel. If each word can be processed at the same time, such as with Convolutional Neural Networks (CNN), then parallelization could reduce the time overhead. The problem of dependencies when translating sentences still remain, so a combination of CNNs and attention was created to form the Transformer model (Vaswani et al. 2017). Transformers allow the decoder to focus on the relevant parts of the input sentence but also use a form of attention called self-attention that uses query, key and value vectors for each word to create scores that measure the relationship between the word and the other words in the sentence. Finally, the decoder chooses from a large selection of target-language words using a 'softmax' function. Systems can use a combination of words and sub-words to significantly reduce the problem caused by the large amount of output options.

17 months

required an emergency re-write of the NMT section

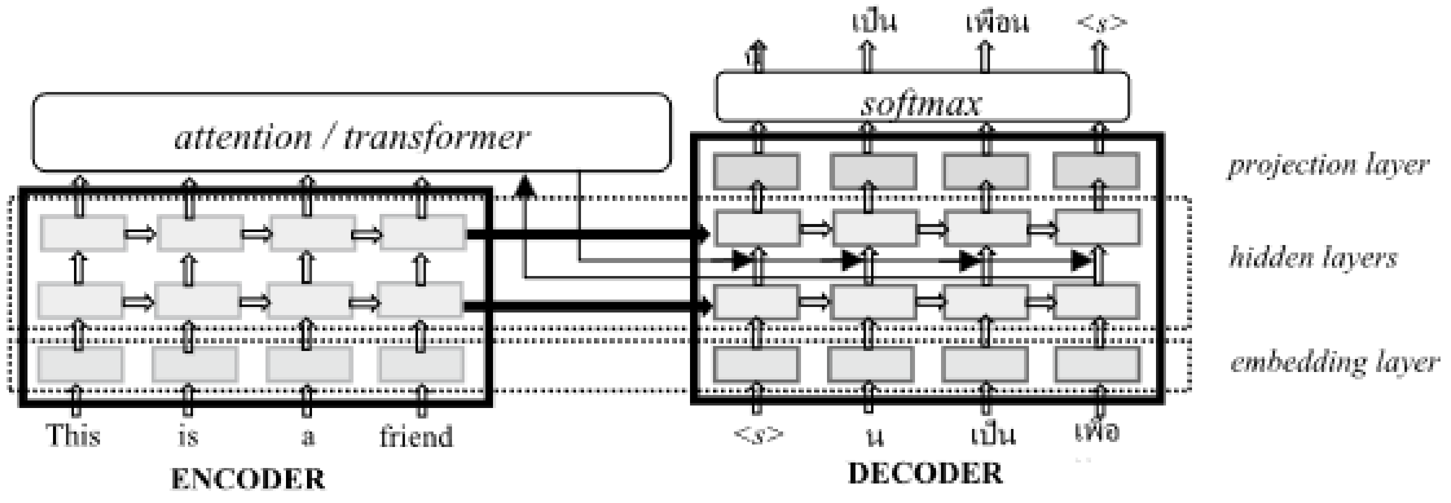
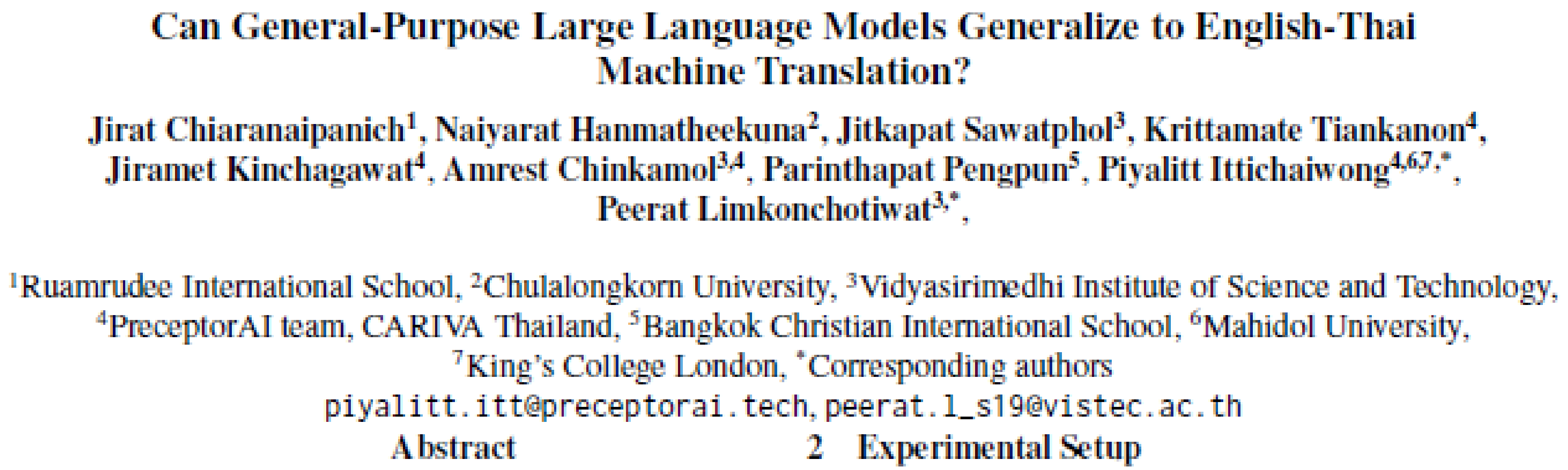


Fig. 5 A neural machine translation (NMT) system

and this week



Can General-Purpose Large Language Models Generalize to English-Thai Machine Translation?

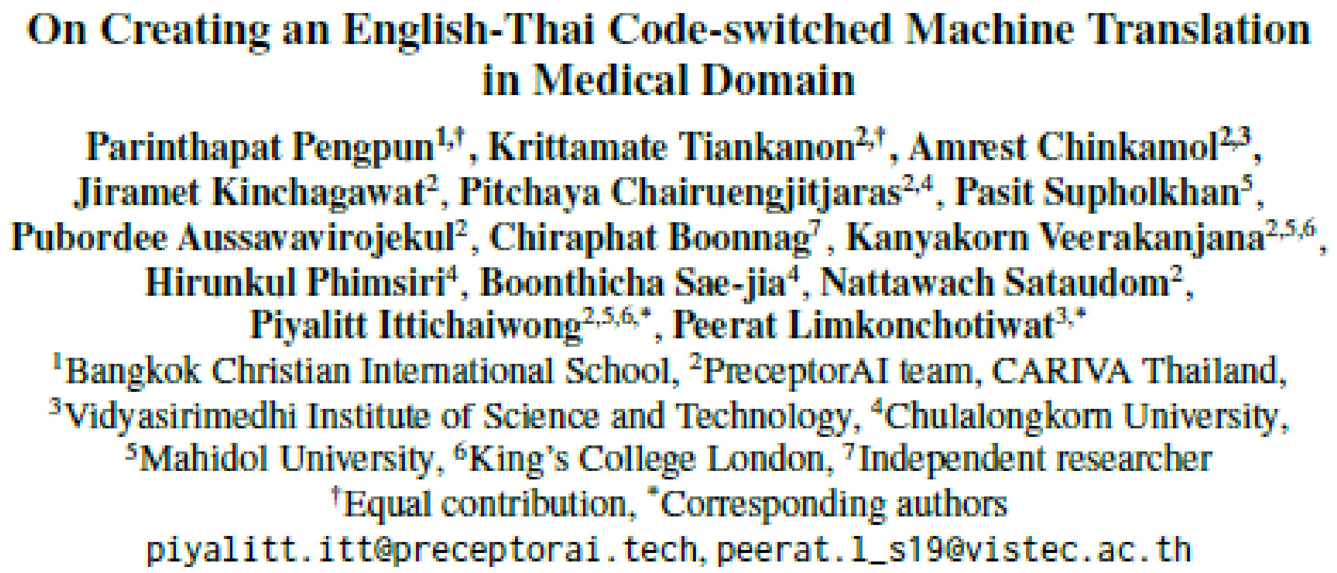
Jirat Chiaranaipanich[1], Naiyarat Hanmatheekuna[2], Jitkapat Sawatphol[3], Krittamate Tiankanon[4], Jiramet Kinchagawat[4], Amrest Chinkamol[3,4], Parinthapat Pengpun[5], Piyalitt Ittichaiwong[4,6,7,*], Peerat Limkonchotiwat[3,*],

[1]Ruamrudee International School, [2]Chulalongkorn University, [3]Vidyasirimedhi Institute of Science and Technology, [4]PreceptorAI team, CARIVA Thailand, [5]Bangkok Christian International School, [6]Mahidol University, [7]King's College London, [*]Corresponding authors

piyalitt.itt@preceptorai.tech, peerat.l_s19@vistec.ac.th

Abstract

2 Experimental Setup



On Creating an English-Thai Code-switched Machine Translation in Medical Domain

Parinthapat Pengpun[1,†], Krittamate Tiankanon[2,†], Amrest Chinkamol[2,3], Jiramet Kinchagawat[2], Pitchaya Chairuengjitjaras[2,4], Pasit Supholkhan[5], Pubordee Aussavavirojekul[2], Chiraphat Boonnag[7], Kanyakorn Veerakanjana[2,5,6], Hirunkul Phimsiri[4], Boonthicha Sae-jia[4], Nattawach Sataudom[2], Piyalitt Ittichaiwong[2,5,6,*], Peerat Limkonchotiwat[3,*]

[1]Bangkok Christian International School, [2]PreceptorAI team, CARIVA Thailand, [3]Vidyasirimedhi Institute of Science and Technology, [4]Chulalongkorn University, [5]Mahidol University, [6]King's College London, [7]Independent researcher

[†]Equal contribution, [*]Corresponding authors

piyalitt.itt@preceptorai.tech, peerat.l_s19@vistec.ac.th

Neural Networks

How have neural networks improved NLP?

01

Sentiment Analysis
Recurrent Neural Networks (RNNs) and **Transformers** have helped sentiment analysis evolve from simple word-based analysis to understanding nuanced opinions in text.
Models like **BERT** and **RoBERTa** capture context more effectively, distinguishing subtle sentiment in complex reviews or social media posts.

02

Question Answering (QA)
Neural networks have enabled models to read passages and provide answers to questions with high accuracy.
BERT, **GPT**, and other transformer-based models excel at QA tasks, leveraging their ability to understand and generate responses based on specific questions.

03

Text Summarization
Neural models generate concise summaries by identifying key information from larger documents.
Sequence-to-sequence (seq2seq) models and **attention mechanisms** are particularly effective, enabling both extractive and abstractive summarization approaches

04

Chatbots and Conversational Agents
Neural networks are behind more natural and coherent conversational agents.
Dialog systems like those in GPT and ChatGPT use transformers to manage context over long interactions, improving the relevance of responses in real-time conversations.

Neural Networks

How have neural networks improved NLP?

05

Named Entity Recognition (NER)
Neural networks have improved the accuracy of identifying names, dates, organizations, and other entities in text.
Bi-directional LSTM-CRF models, often combined with transformer embeddings, have set new benchmarks in extracting these details accurately.

06

Text Generation and Completion
Models like **GPT-3** can generate realistic text based on prompts, making significant strides in applications like automated writing, content generation, and code completion.
These models understand context well enough to complete or expand on text in a way that appears human-written.

07

Speech Recognition and Synthesis
Neural networks, particularly **convolutional neural networks (CNNs)** and **transformers**, have improved both transcription accuracy and natural-sounding speech synthesis.
Systems like **WaveNet** and **Tacotron** produce human-like voices, critical for virtual assistants and accessibility tools.

08

Grammar and Style Correction
Models like **T5** and **GPT** can detect and suggest corrections for grammar and style errors, often with human-like precision.
These models help in developing tools like Grammarly and Microsoft Editor, making grammar checks context-aware and more accurate.

so, what changed?

- from corpus-based research to Neural Networks

1. vast amounts of text (the internet)
2. more computational power

Neural Networks can converge

so advanced machine learning

without human input (features & weights)

References

- ColumbiaX (2017) Overview of AI, edX CSMM 101x Artificial Intelligence (AI) lecture notes [online video]. Available at: https://courses.edx.org/courses/course-v1:ColumbiaX+CSMM.101x+2T2017/courseware/72ad68365307491198cfcadeeb17baad/adbd7b28184343a1ab8532fbdadd7de4/?activate_block_id=blockv1%3AColumbiaX%2BCSMM.101x%2B2T2017%2Btype%40sequential%2Bblock%40adbd7b28184343a1ab8532fbdadd7de4 [Accessed 26th July 2017].
- ColumbiaX (2017) Applications of AI, edX CSMM 101x Artificial Intelligence (AI) lecture notes [online video]. Available at: https://courses.edx.org/courses/course-v1:ColumbiaX+CSMM.101x+2T2017/courseware/72ad68365307491198cfcadeeb17baad/835095fd0f704a4eb141c14ff16ec4de/?child=last[Accessed 26th July 2017].

References

- ColumbiaX (2017) AI Foundation and History, edX CSMM 101x Artificial Intelligence (AI) lecture notes [online video]. Available at: https://courses.edx.org/courses/course-v1:ColumbiaX+CSMM.101x+2T2017/courseware/72ad68365307491198cfcadeeb17baad/c9da1f2103584d93a4ef9d6275abd448/?child=first [Accessed 26th July 2017].
- Russell, S., Norvig, P. and Intelligence, A., 1995. A modern approach. Artificial Intelligence. International Edition (third). Prentice-Hall

Thank you!

any questions?