Artificial Intelligence

Problem Solving

1111AI03
MBA, IM, NTPU (M6132) (Fall 2022)
Wed 2, 3, 4 (9:10-12:00) (B8F40)

Min-Yuh Day, Ph.D,
Associate Professor
Institute of Information Management, National Taipei University
https://web.ntpu.edu.tw/~myday

https://meet.google.com/miy-fbif-max

2022-09-28

Syllabus

Week	Date	Subject/Topics
1	2022/09/14	Introduction to Artificial Intelligence
2	2022/09/21	Artificial Intelligence and Intelligent Agents
3	2022/09/28	Problem Solving
4	2022/10/05	Knowledge, Reasoning and Knowledge Representation; Uncertain Knowledge and Reasoning
5	2022/10/12	Case Study on Artificial Intelligence I
6	2022/10/19	Machine Learning: Supervised and Unsupervised Learning

Syllabus

Week Date Subject/Topics

7 2022/10/26 The Theory of Learning and Ensemble Learning

8 2022/11/02 Midterm Project Report

9 2022/11/09 Deep Learning and Reinforcement Learning

10 2022/11/16 Deep Learning for Natural Language Processing

11 2022/11/23 Invited Talk: AI for Information Retrieval

12 2022/11/30 Case Study on Artificial Intelligence II

Syllabus

Week	Date	Subject/Topics
13	2022/12/07	Computer Vision and Robotics
14	2022/12/14	Philosophy and Ethics of AI and the Future of AI
15	2022/12/21	Final Project Report I
16	2022/12/28	Final Project Report II
17	2023/01/04	Self-learning
18	2023/01/11	Self-learning

Artificial Intelligence

Problem Solving

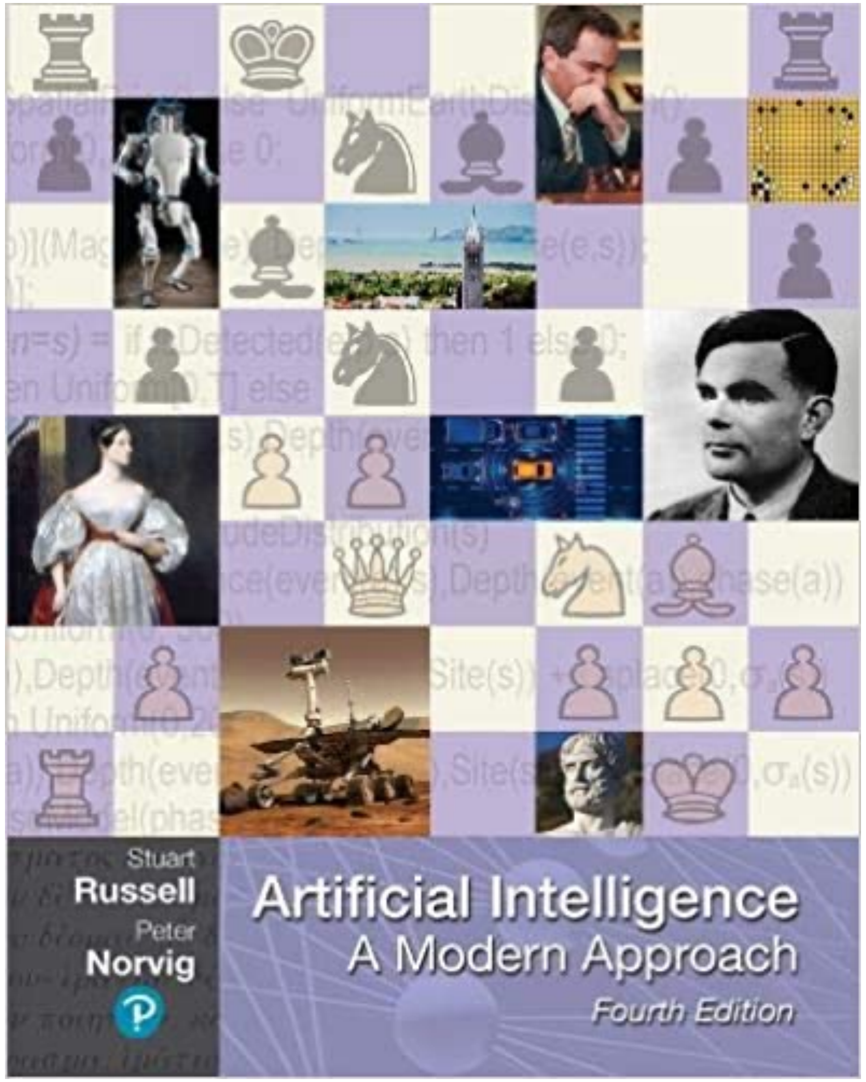
Outline

- **Solving Problems by Searching**
- **Search in Complex Environments**
- **Adversarial Search and Games**
- **Constraint Satisfaction Problems**

Stuart Russell and Peter Norvig (2020),

Artificial Intelligence: A Modern Approach,

4th Edition, Pearson



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

https://www.amazon.com/Artificial-Intelligence-A-Modern-Approach/dp/0134610997/

Artificial Intelligence: A Modern Approach

1. Artificial Intelligence
2. Problem Solving
3. Knowledge and Reasoning
4. Uncertain Knowledge and Reasoning
5. Machine Learning
6. Communicating, Perceiving, and Acting
7. Philosophy and Ethics of AI

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
Problem Solving

Artificial Intelligence:
2. Problem Solving

- Solving Problems by Searching
- Search in Complex Environments
- Adversarial Search and Games
- Constraint Satisfaction Problems

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

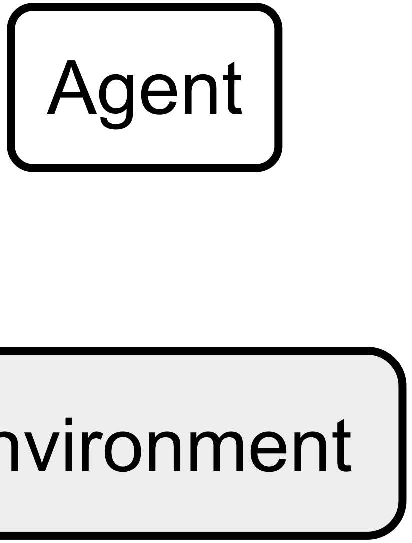
Intelligent Agents

4 Approaches of AI

2. Thinking Humanly: The Cognitive Modeling Approach	3. Thinking Rationally: The "Laws of Thought" Approach
1. Acting Humanly: The Turing Test Approach (1950)	4. Acting Rationally: The Rational Agent Approach

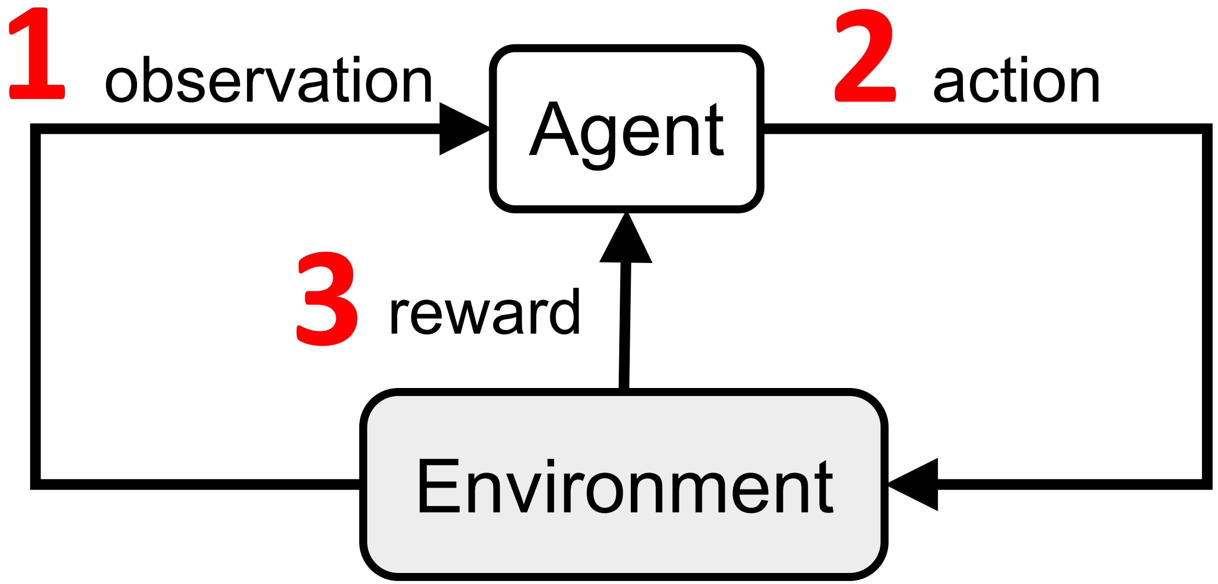
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Reinforcement Learning (DL)



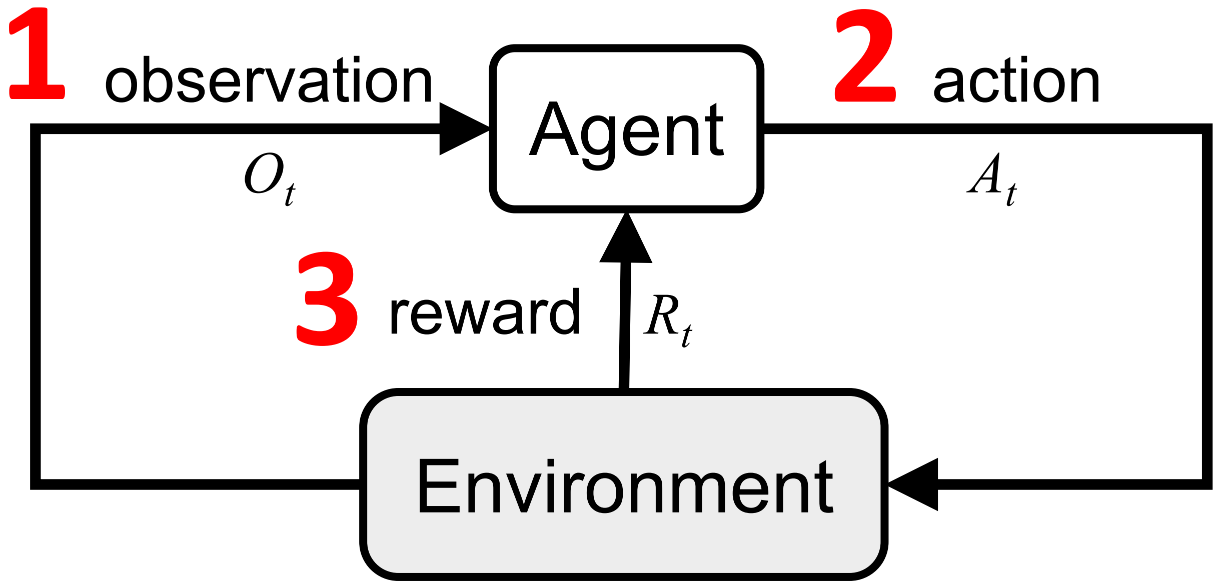
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)

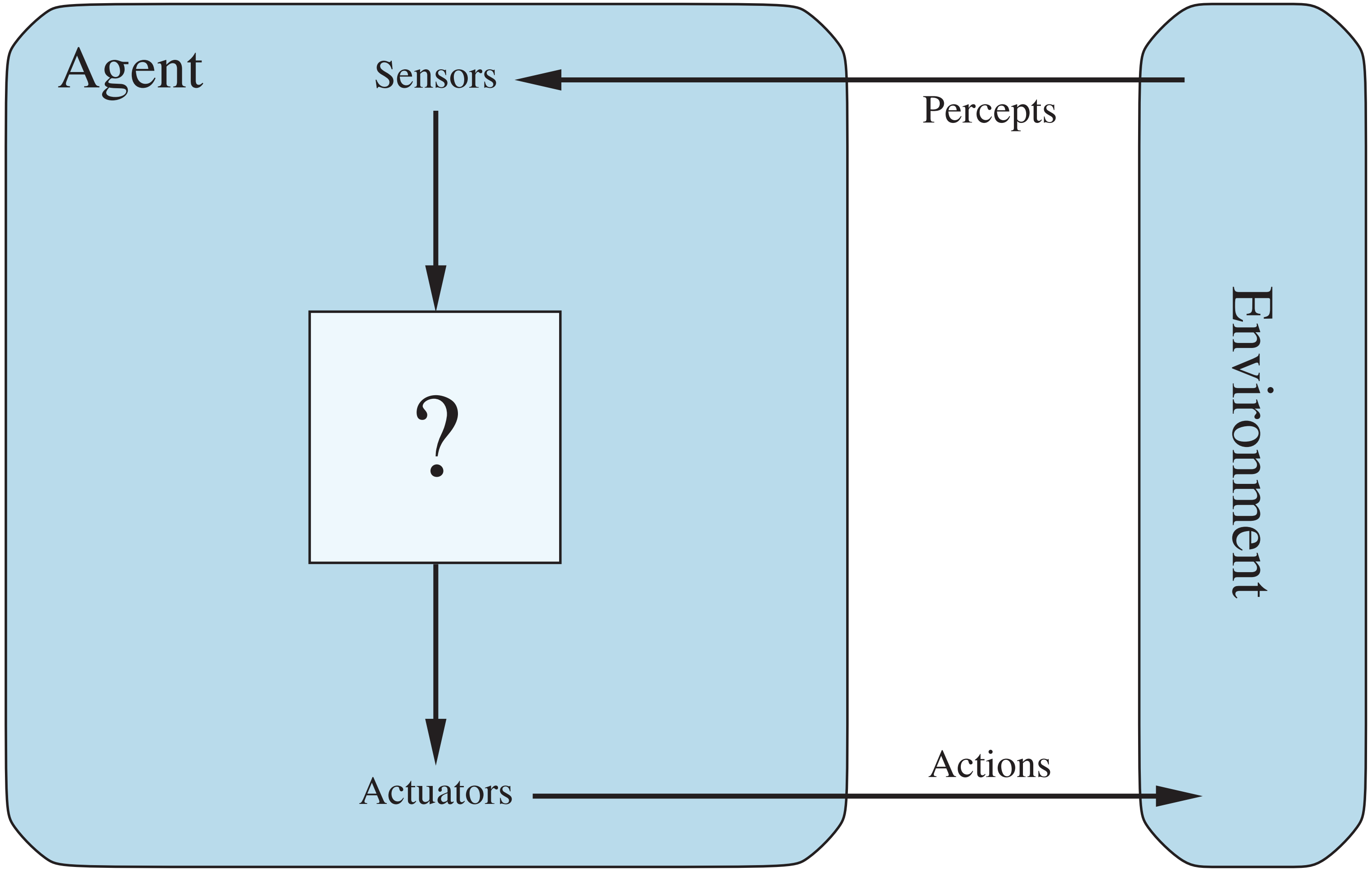


Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



Agents interact with environments through sensors and actuators

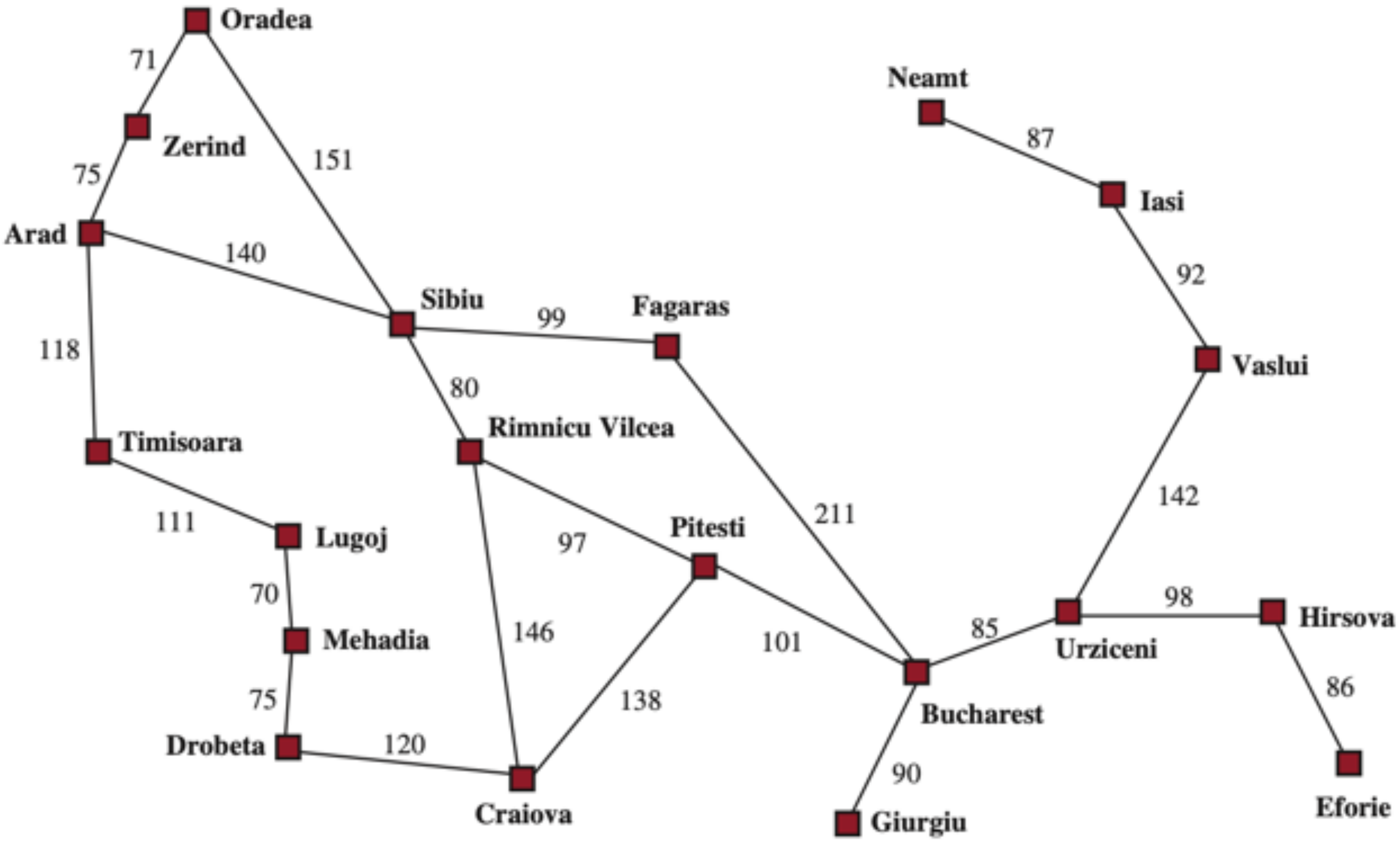


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Solving Problems by Searching

AI: Solving Problems by Searching

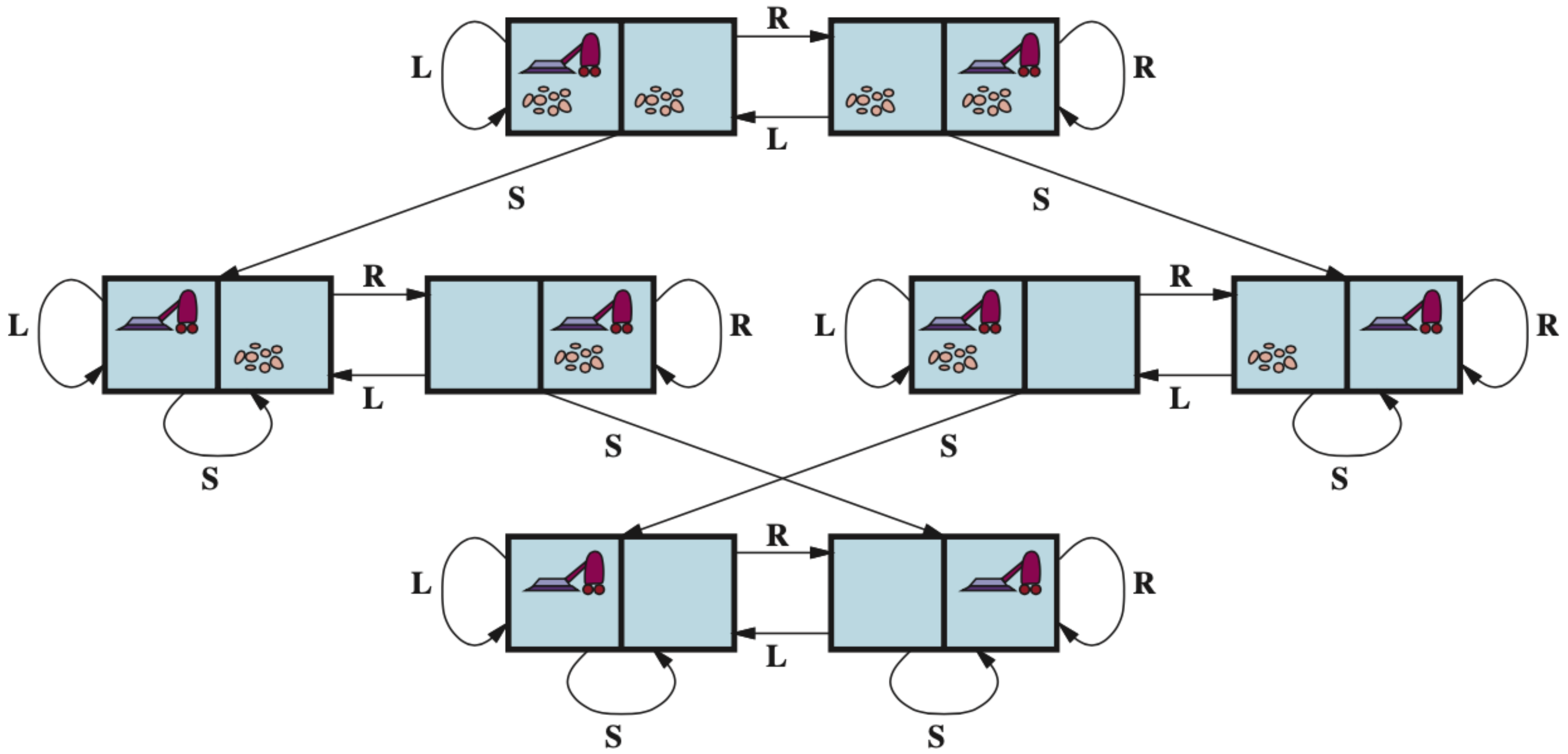
A simplified road map of part of Romania, with road distances in miles.



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

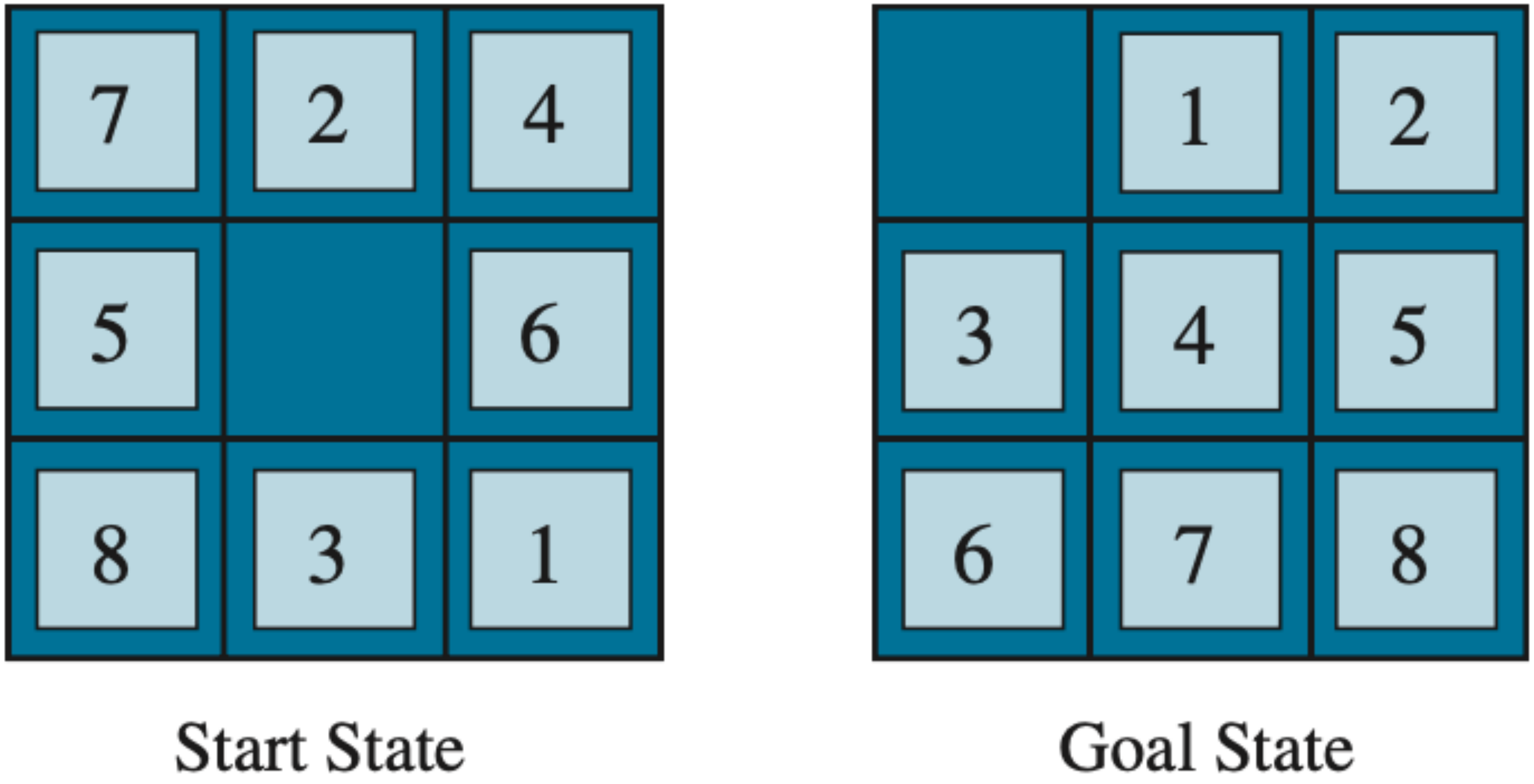
The state-space graph for the two-cell vacuum world

There are 8 states and three actions for each state:
L = *Left*, R = *Right*, S = *Suck*.



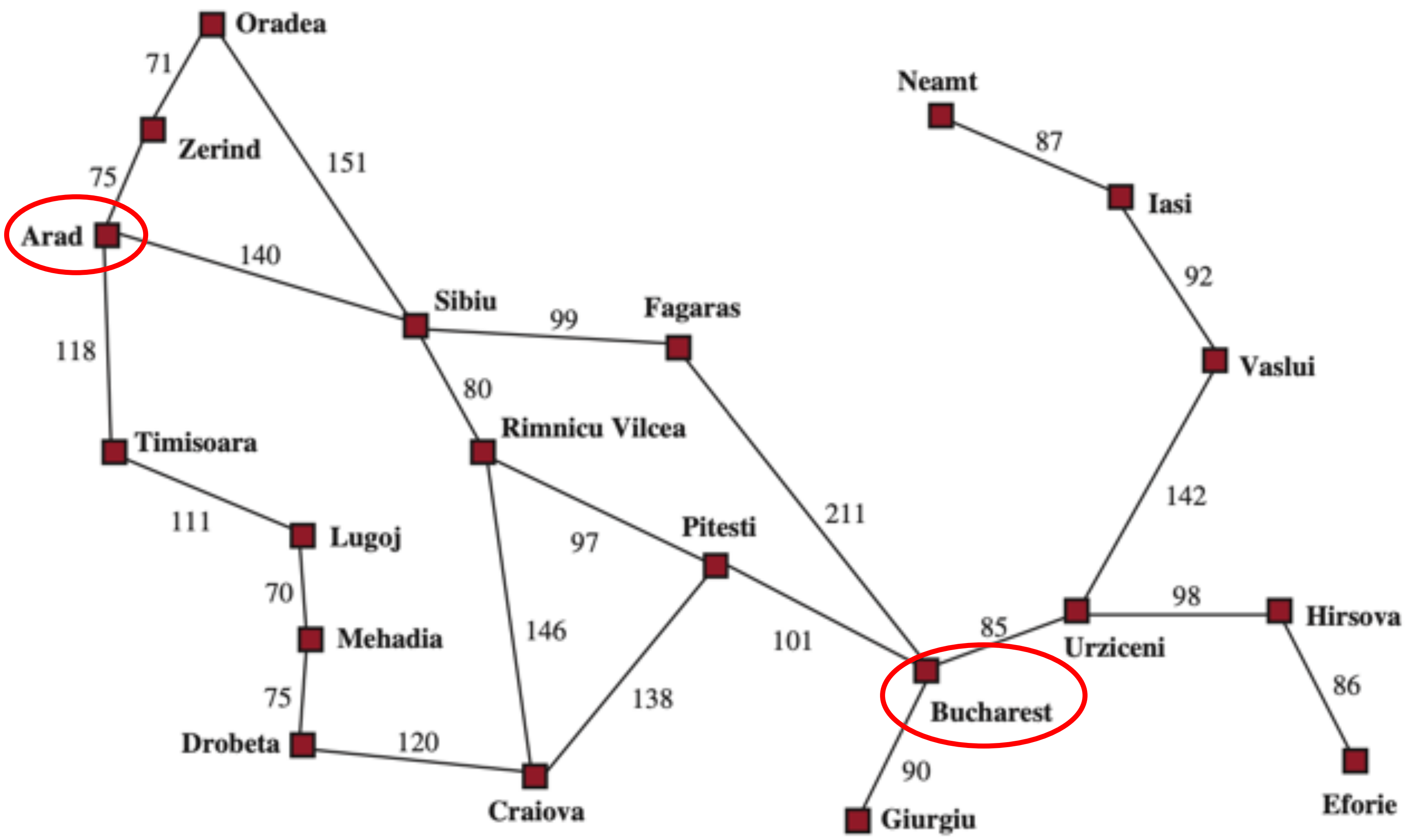
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A typical instance of the 8-puzzle



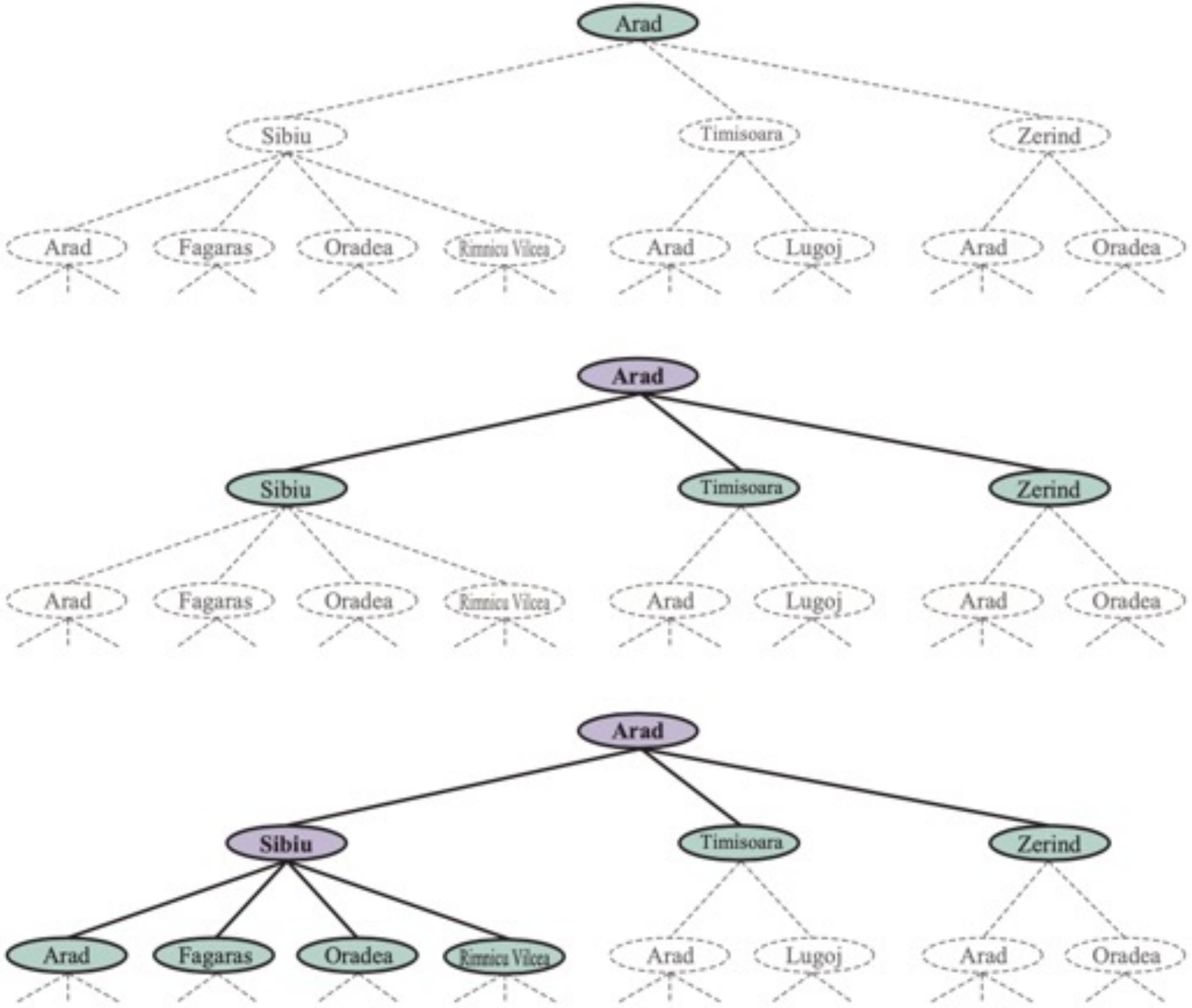
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Arad to Bucharest



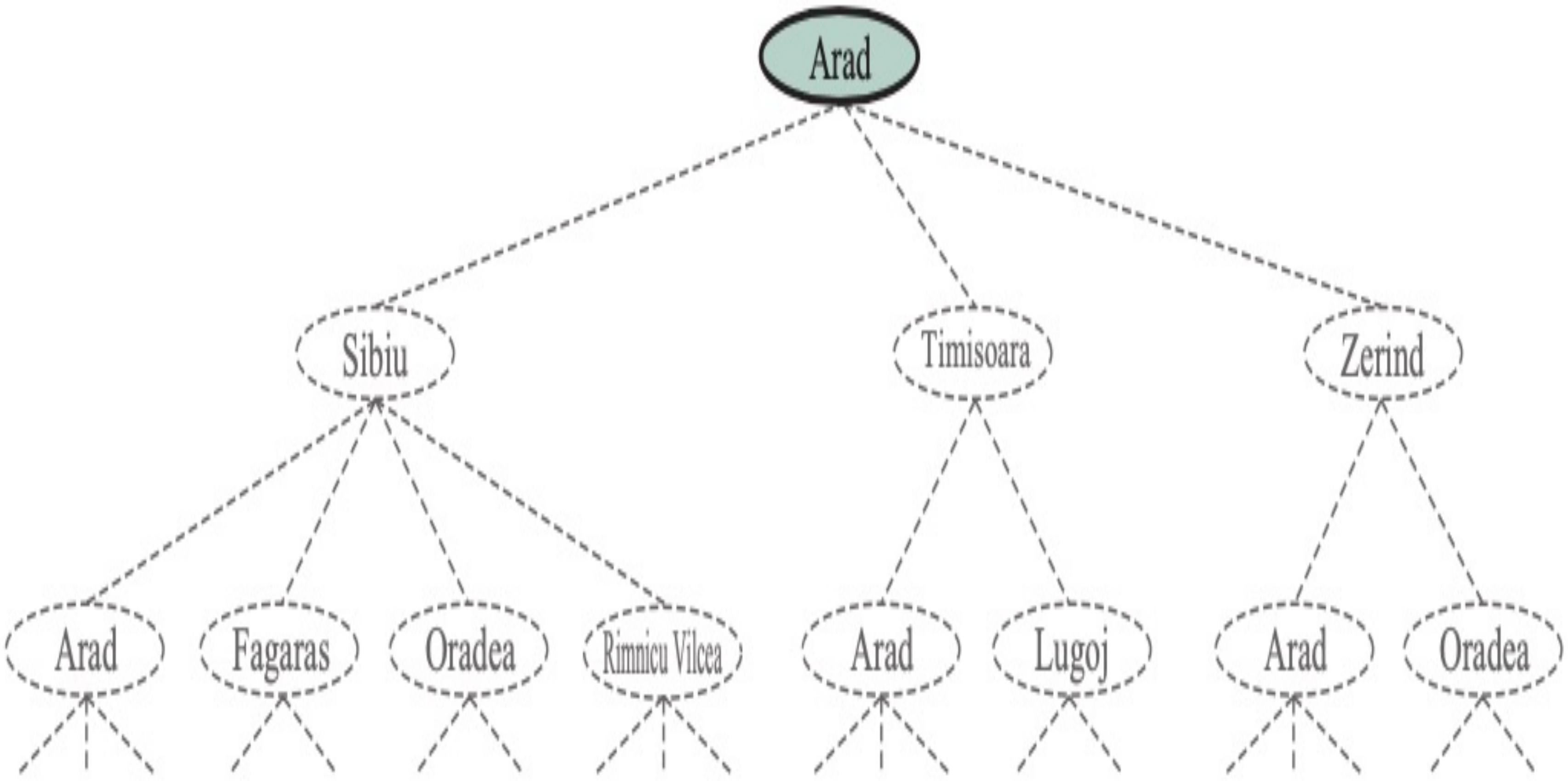
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Three partial search trees for finding a route from Arad to Bucharest



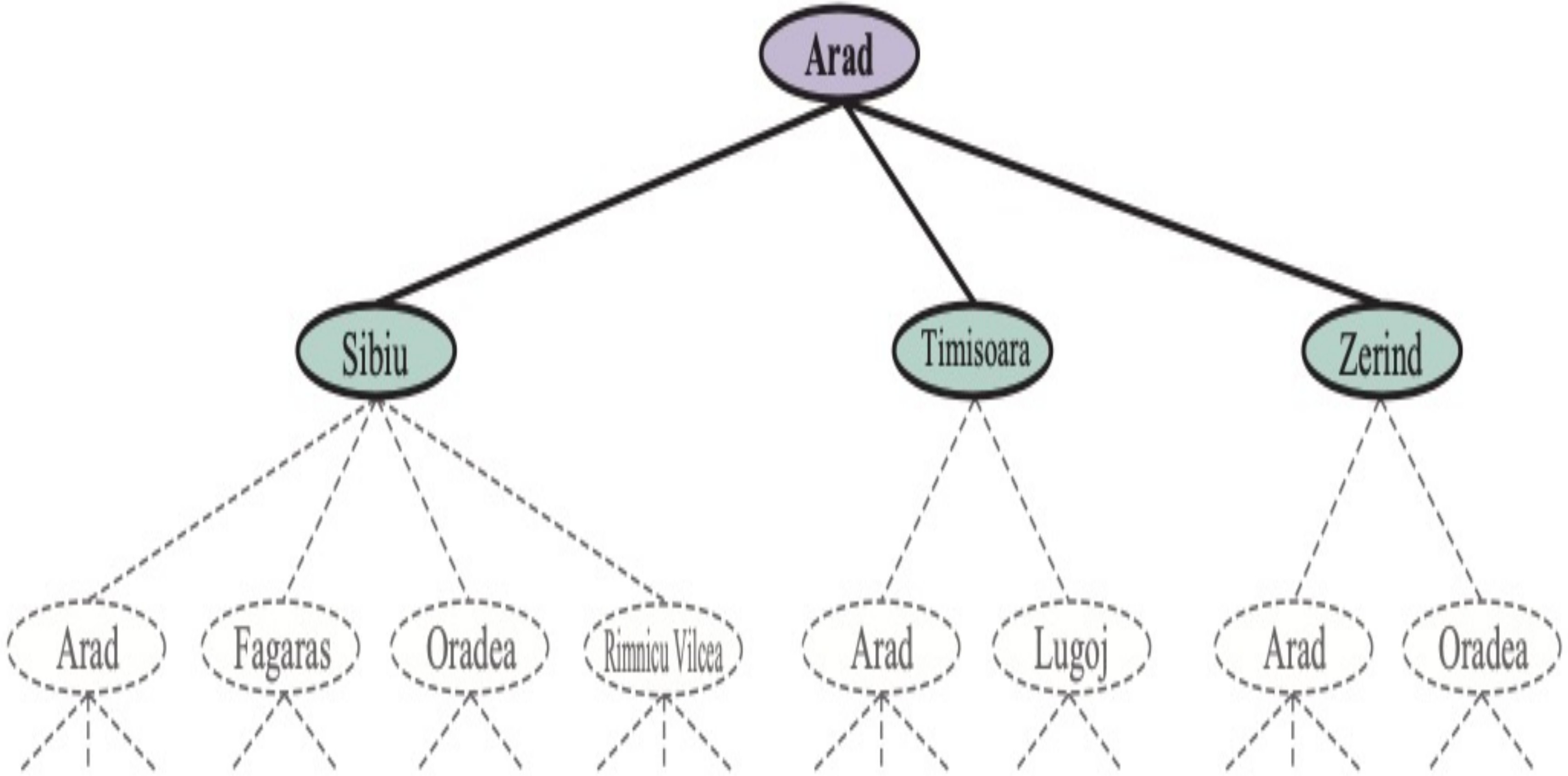
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Three partial search trees for finding a route from Arad to Bucharest



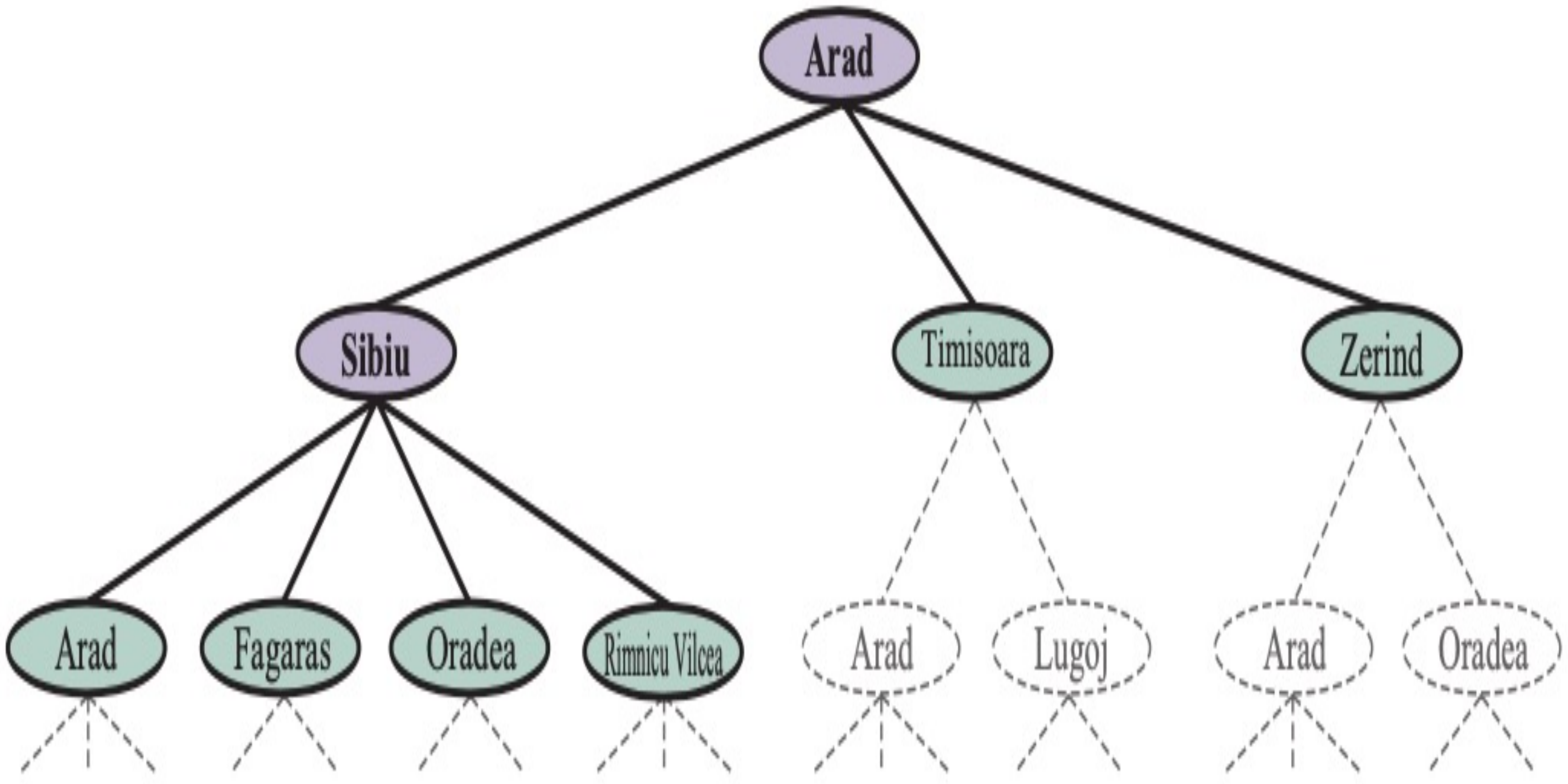
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Three partial search trees for finding a route from Arad to Bucharest



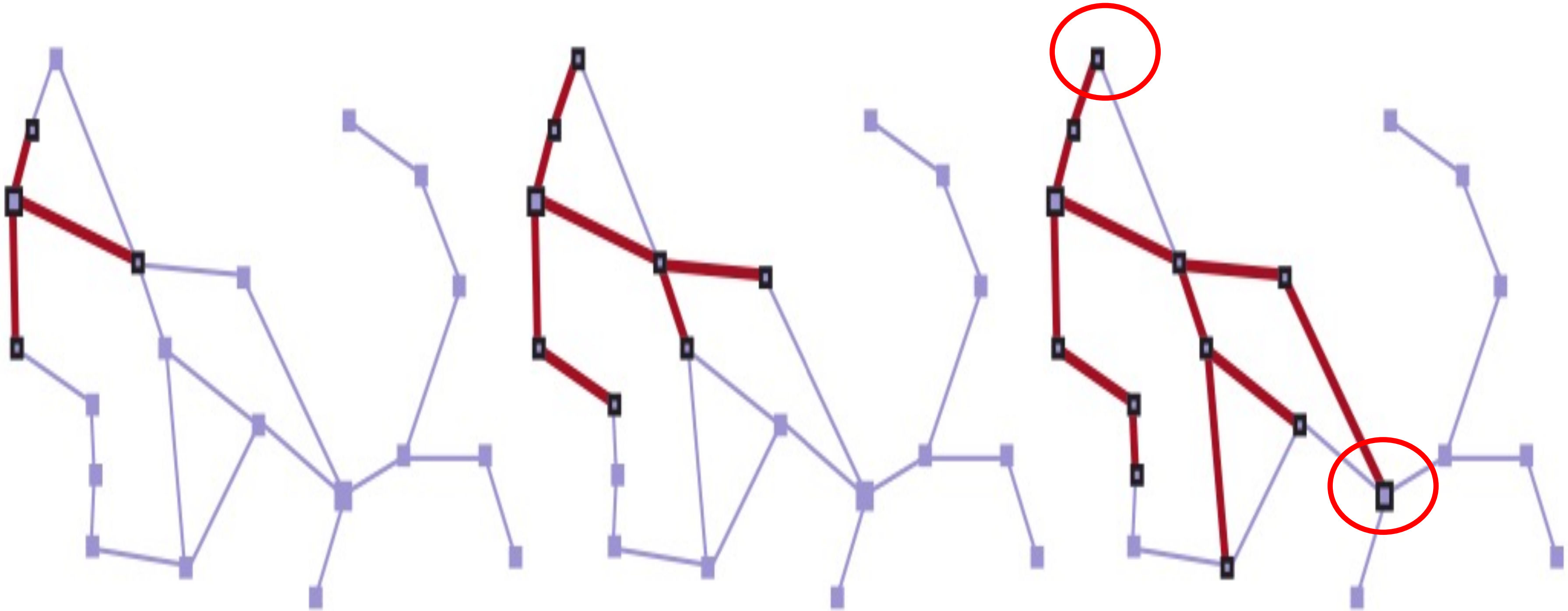
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Three partial search trees for finding a route from Arad to Bucharest



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

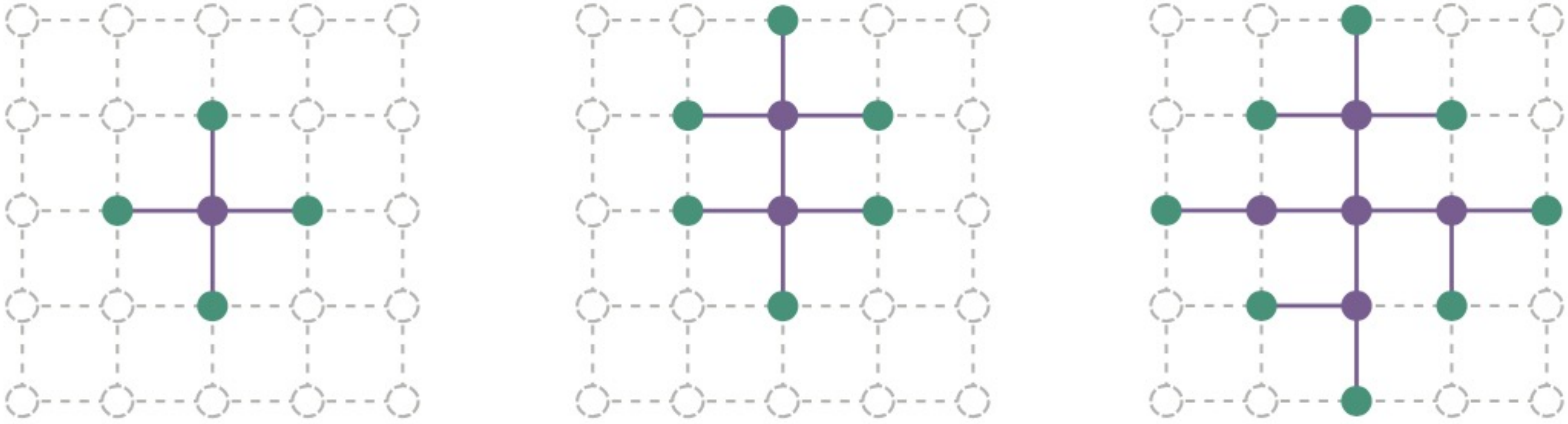
A sequence of search trees generated by a graph search on the Romania problem



The Separation Property of Graph Search

illustrated on a rectangular-grid problem

The frontier (green) separates
the interior (lavender) from
the exterior (faint dashed)



(a) (b) (c)

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The Best-First Search (BFS) Algorithm

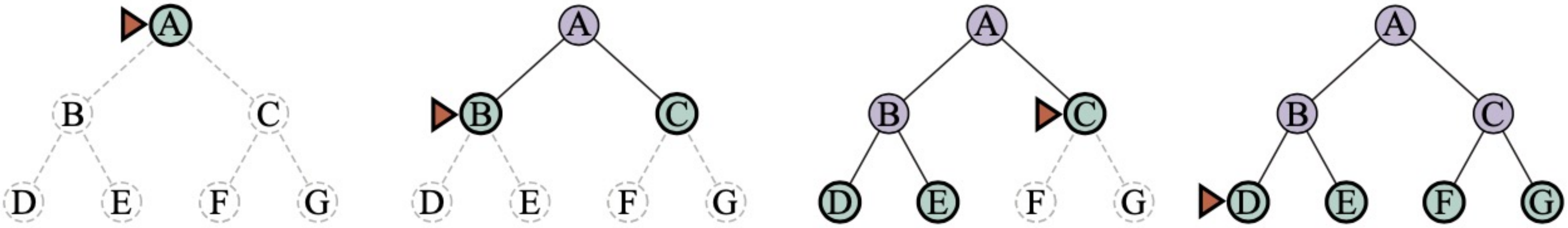
```
function BEST-FIRST-SEARCH(problem, f) returns a solution node or failure
  node ← NODE(STATE=problem.INITIAL)
  frontier ← a priority queue ordered by f, with node as an element
  reached ← a lookup table, with one entry with key problem.INITIAL and value node
  while not IS-EMPTY(frontier) do
    node ← POP(frontier)
    if problem.IS-GOAL(node.STATE) then return node
    for each child in EXPAND(problem, node) do
      s ← child.STATE
      if s is not in reached or child.PATH-COST < reached[s].PATH-COST then
        reached[s] ← child
        add child to frontier
  return failure

function EXPAND(problem, node) yields nodes
  s ← node.STATE
  for each action in problem.ACTIONS(s) do
    s′ ← problem.RESULT(s, action)
    cost ← node.PATH-COST + problem.ACTION-COST(s, action, s′)
    yield NODE(STATE=s′, PARENT=node, ACTION=action, PATH-COST=cost)
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Breadth-First Search on a Simple Binary Tree

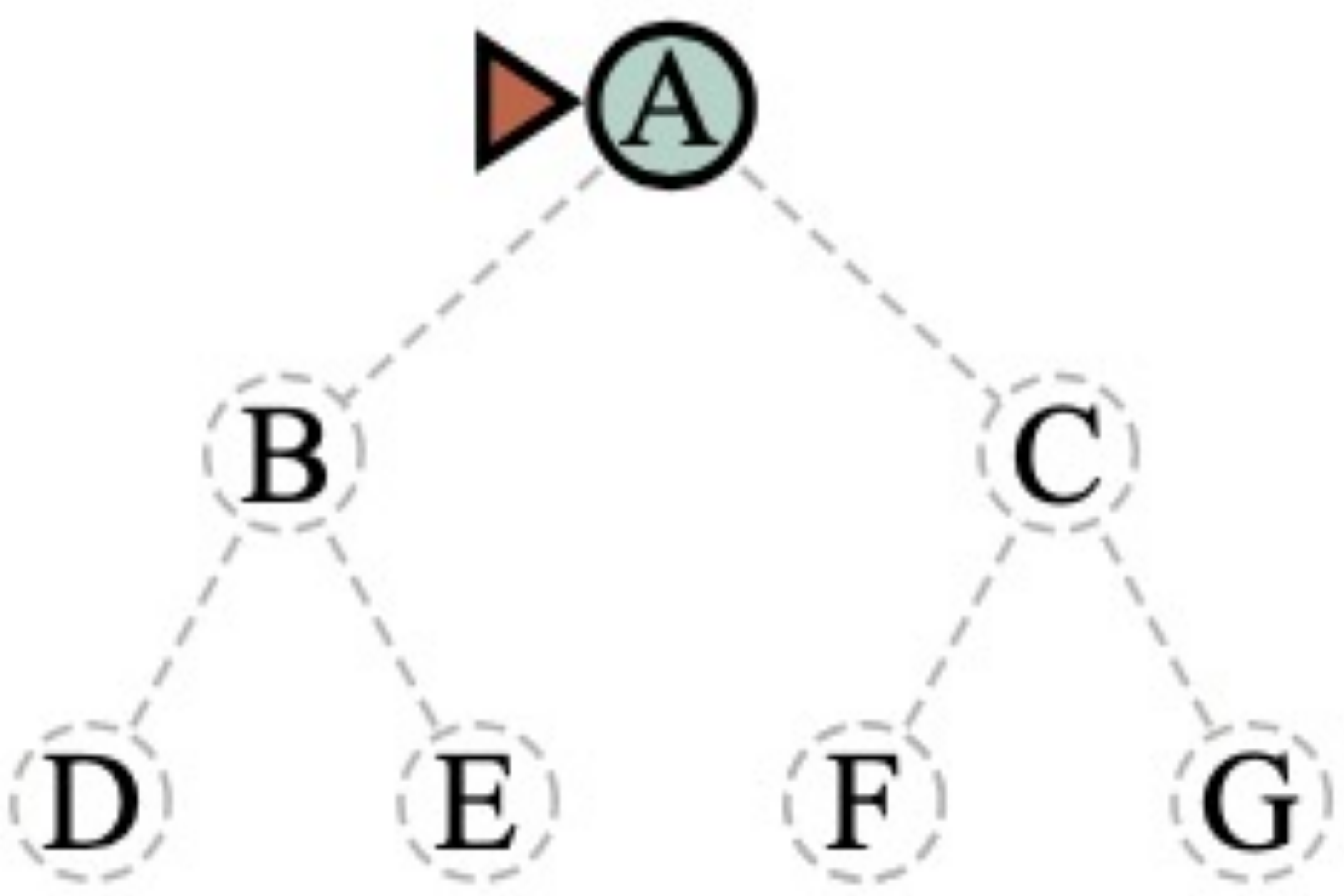
Bread-First Search (BFS)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Breadth-First Search on a Simple Binary Tree

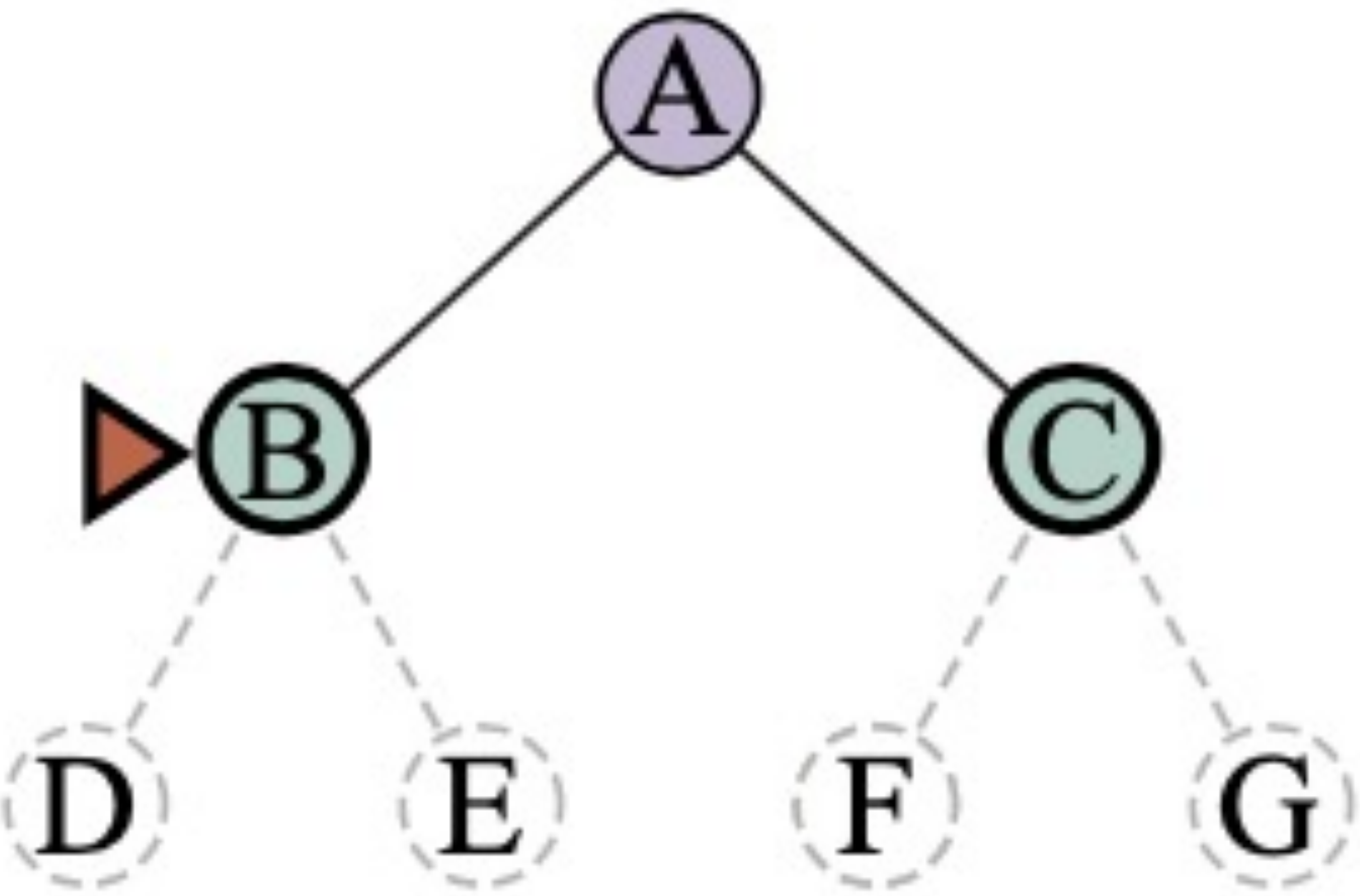
Bread-First Search (BFS)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Bread-First Search (BFS)

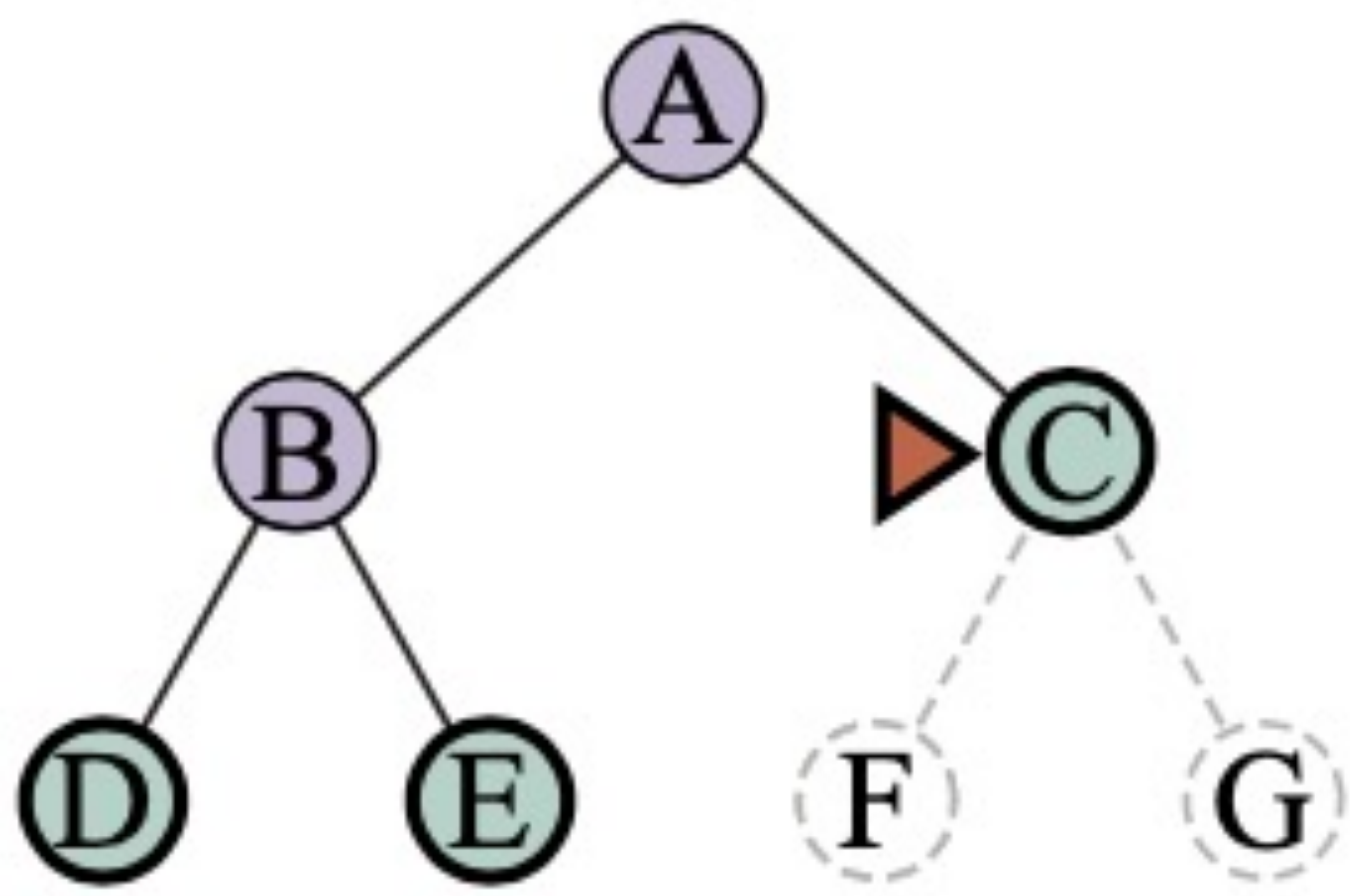
Breadth-First Search on a Simple Binary Tree



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Breadth-First Search on a Simple Binary Tree

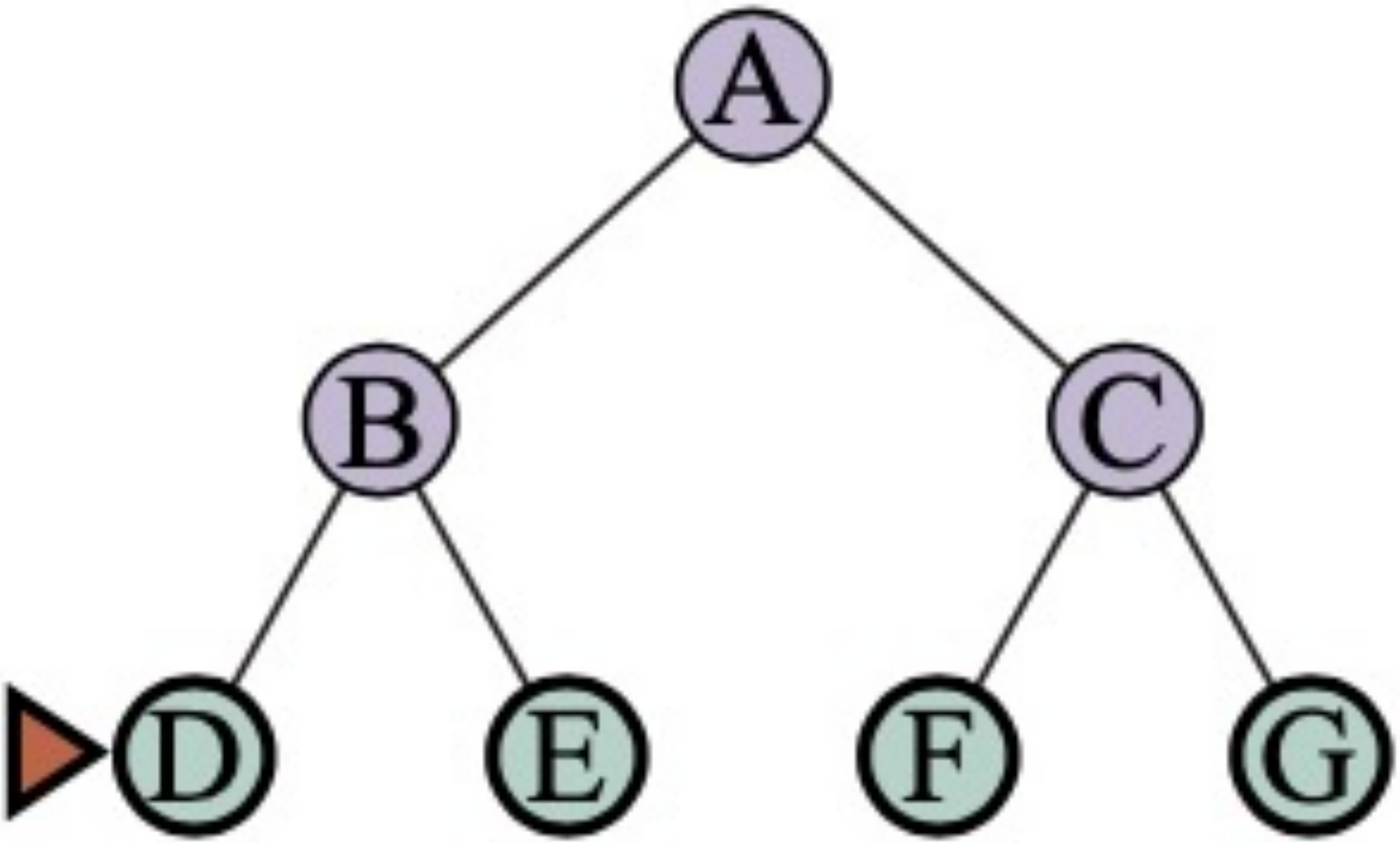
Bread-First Search (BFS)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Breadth-First Search on a Simple Binary Tree

Bread-First Search (BFS)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

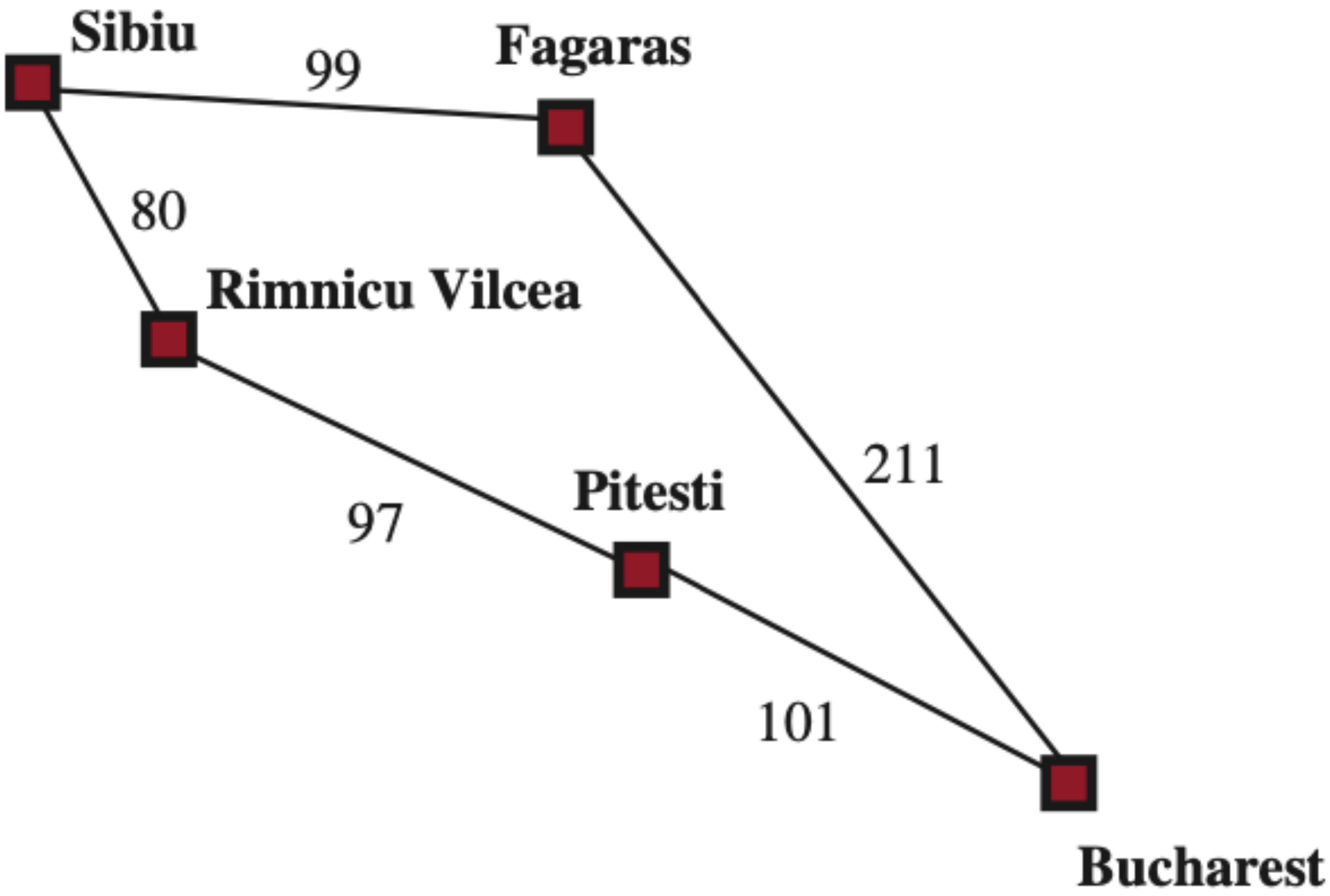
Breadth-First Search and Uniform-Cost Search Algorithms

```
function BREADTH-FIRST-SEARCH(problem) returns a solution node or failure
  node ← NODE(problem.INITIAL)
  if problem.IS-GOAL(node.STATE) then return node
  frontier ← a FIFO queue, with node as an element
  reached ← {problem.INITIAL}
  while not IS-EMPTY(frontier) do
    node ← POP(frontier)
    for each child in EXPAND(problem, node) do
      s ← child.STATE
      if problem.IS-GOAL(s) then return child
      if s is not in reached then
        add s to reached
        add child to frontier
  return failure

function UNIFORM-COST-SEARCH(problem) returns a solution node, or failure
  return BEST-FIRST-SEARCH(problem, PATH-COST)
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

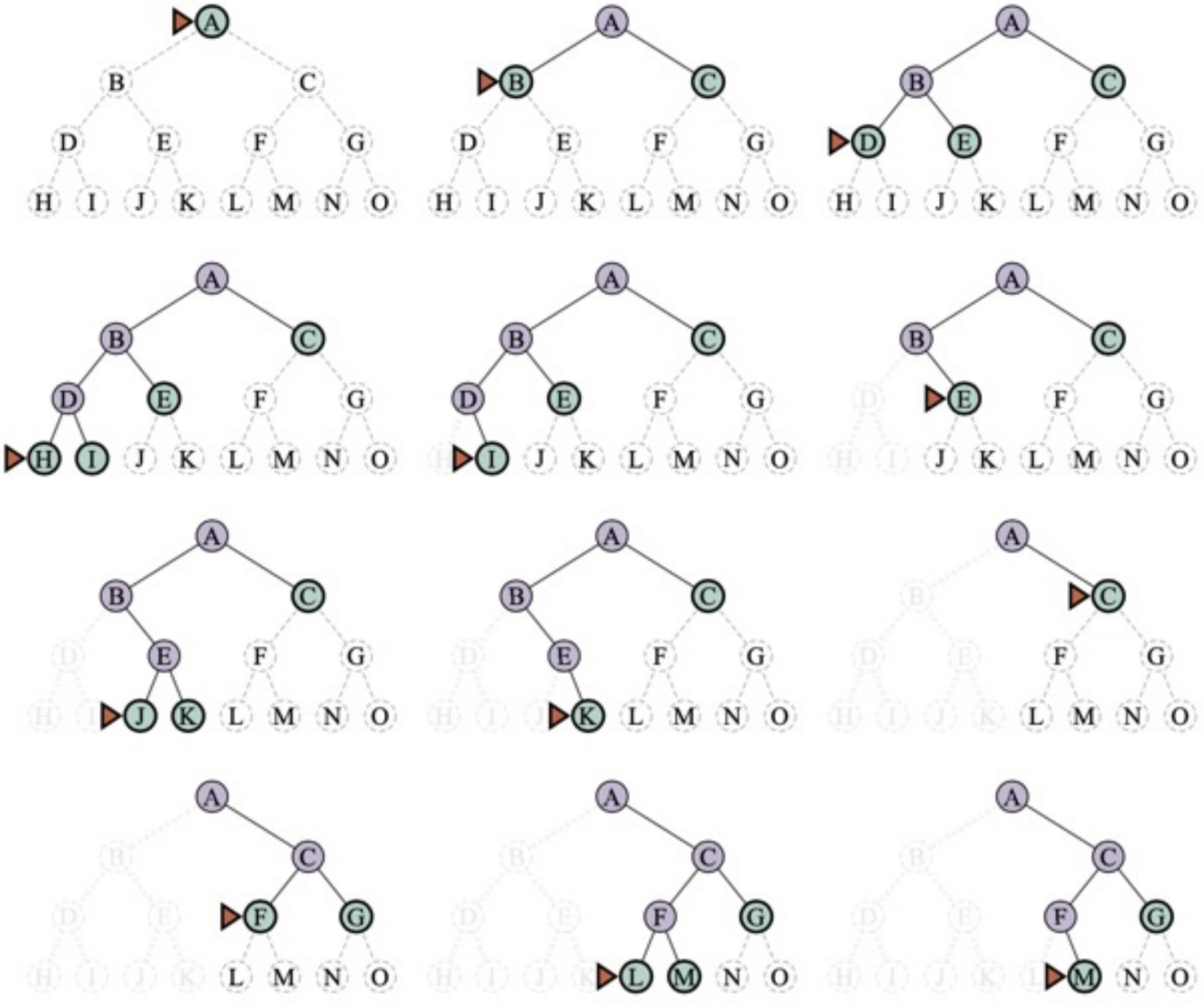
Part of the Romania State Space
Uniform-Cost Search



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Depth-First Search (DFS)

Depth-First Search (DFS)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

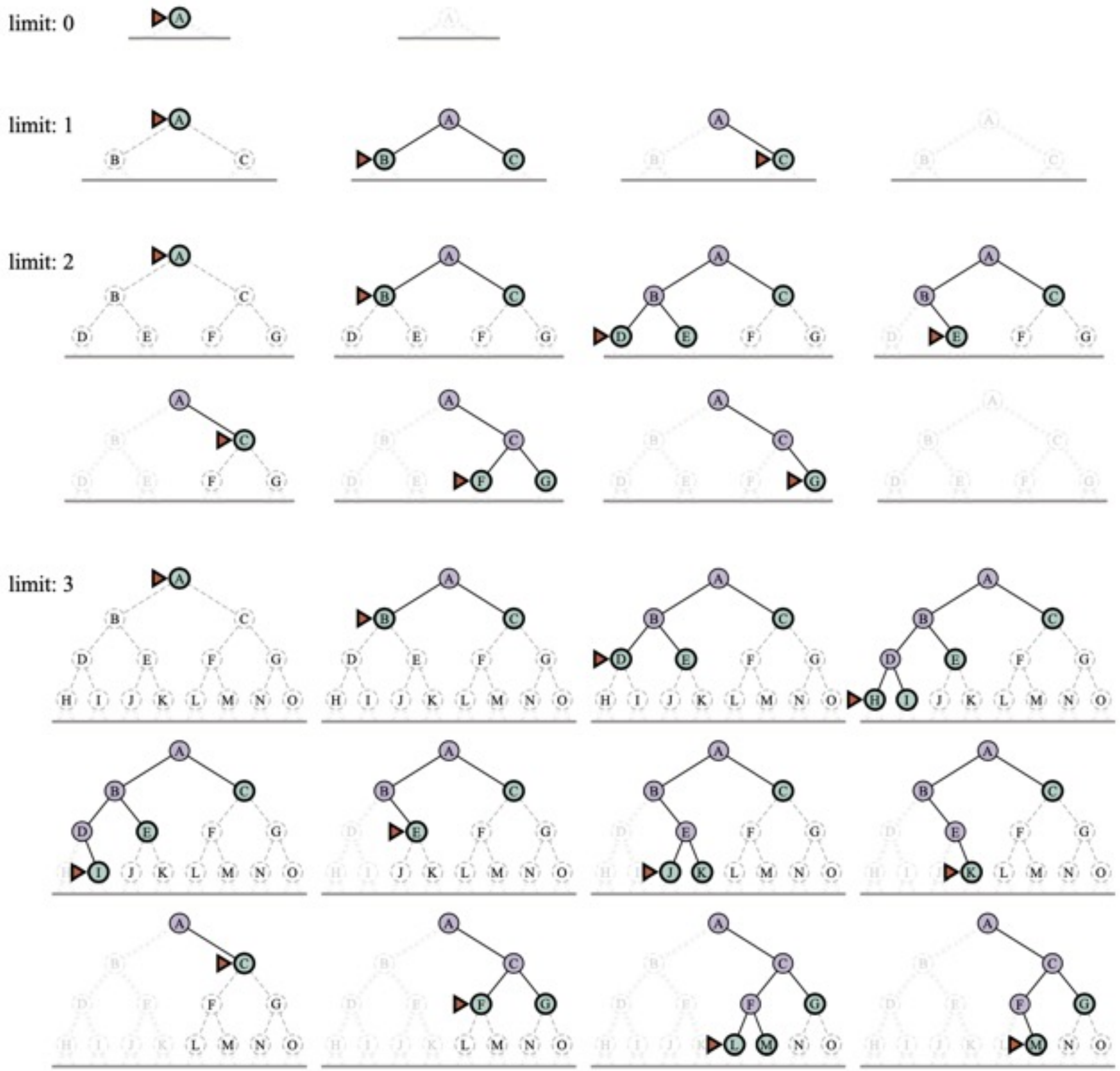
Iterative deepening and depth-limited tree-like search

```
function ITERATIVE-DEEPENING-SEARCH(problem) returns a solution node or failure
  for depth = 0 to ∞ do
    result ← DEPTH-LIMITED-SEARCH(problem, depth)
    if result ≠ cutoff then return result

function DEPTH-LIMITED-SEARCH(problem, ℓ) returns a node or failure or cutoff
  frontier ← a LIFO queue (stack) with NODE(problem.INITIAL) as an element
  result ← failure
  while not IS-EMPTY(frontier) do
    node ← POP(frontier)
    if problem.IS-GOAL(node.STATE) then return node
    if DEPTH(node) > ℓ then
      result ← cutoff
    else if not IS-CYCLE(node) do
      for each child in EXPAND(problem, node) do
        add child to frontier
  return result
```

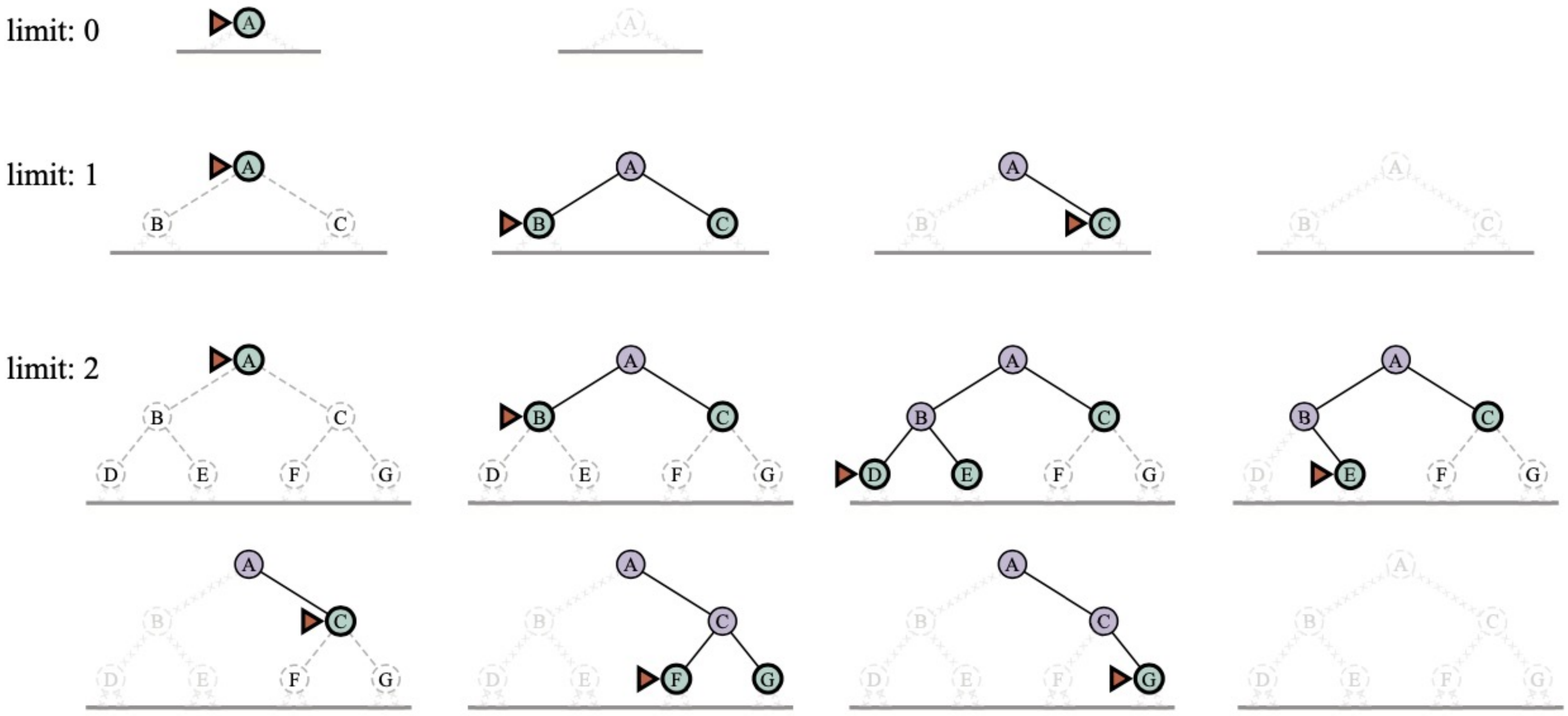
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Four iterations of iterative deepening search



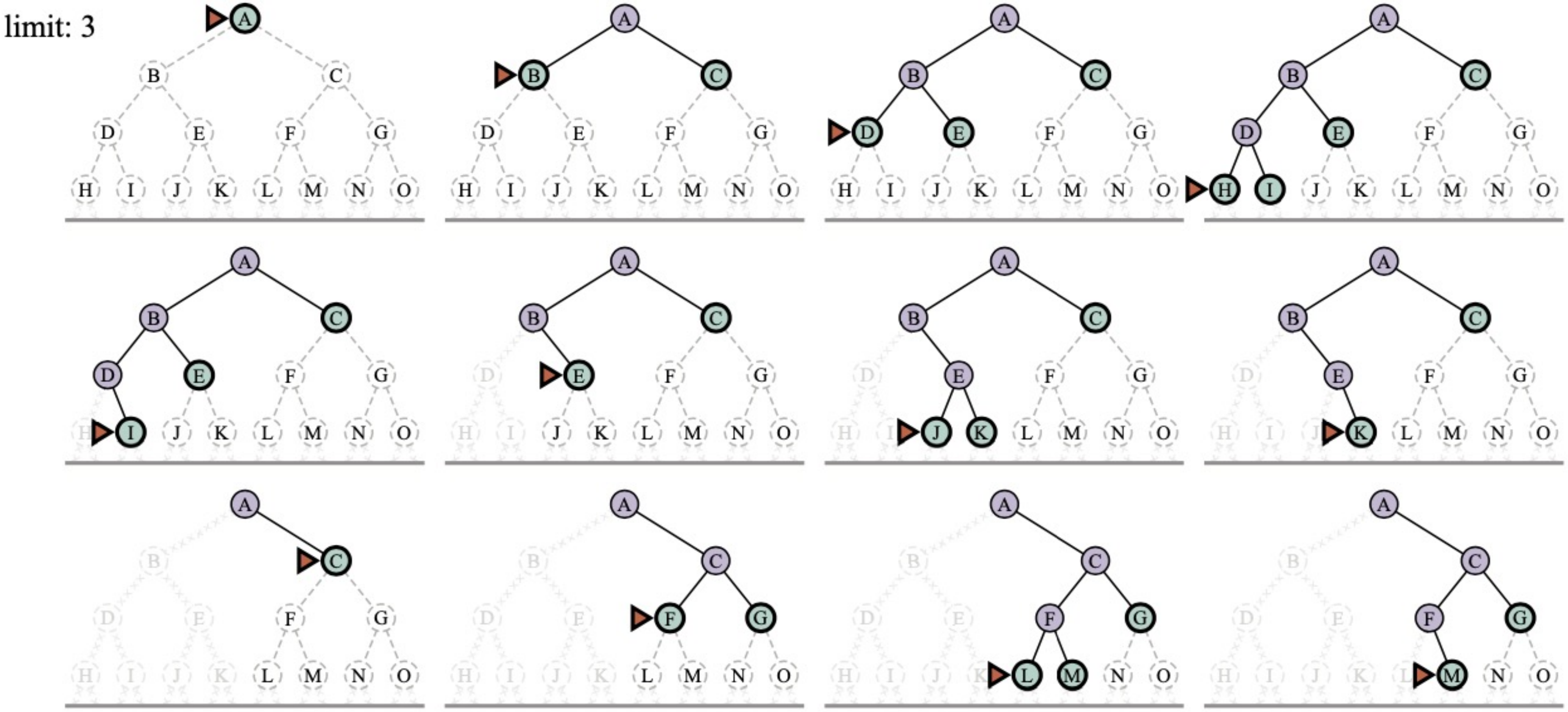
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Four iterations of iterative deepening search



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Four iterations of iterative deepening search



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Bidirectional Best-First Search

keeps two frontiers and two tables of reached states

function BIBF-SEARCH($problem_F$, f_F, $problem_B$, f_B) **returns** a solution node, or *failure*
    $node_F \leftarrow$ NODE($problem_F$.INITIAL)   *// Node for a start state*
    $node_B \leftarrow$ NODE($problem_B$.INITIAL)   *// Node for a goal state*
    $frontier_F \leftarrow$ a priority queue ordered by f_F, with $node_F$ as an element
    $frontier_B \leftarrow$ a priority queue ordered by f_B, with $node_B$ as an element
    $reached_F \leftarrow$ a lookup table, with one key $node_F$.STATE and value $node_F$
    $reached_B \leftarrow$ a lookup table, with one key $node_B$.STATE and value $node_B$
    $solution \leftarrow failure$
    while not TERMINATED($solution$, $frontier_F$, $frontier_B$) **do**
        if f_F(TOP($frontier_F$)) $<$ f_B(TOP($frontier_B$)) **then**
            $solution \leftarrow$ PROCEED(F, $problem_F$ $frontier_F$, $reached_F$, $reached_B$, $solution$)
        else $solution \leftarrow$ PROCEED(B, $problem_B$, $frontier_B$, $reached_B$, $reached_F$, $solution$)
    return $solution$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Bidirectional Best-First Search

keeps two frontiers and two tables of reached states

```
function PROCEED(dir, problem, frontier, reached, reached2, solution) returns a solution
      // Expand node on frontier; check against the other frontier in reached2.
      // The variable "dir" is the direction: either F for forward or B for backward.
  node ← POP(frontier)
  for each child in EXPAND(problem, node) do
    s ← child.STATE
    if s not in reached or PATH-COST(child) < PATH-COST(reached[s]) then
      reached[s] ← child
      add child to frontier
      if s is in reached2 then
        solution2 ← JOIN-NODES(dir, child, reached2[s]))
        if PATH-COST(solution2) < PATH-COST(solution) then
          solution ← solution2
  return solution
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Evaluation of search algorithms

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[1]	Yes[1,2]	No	No	Yes[1]	Yes[1,4]
Optimal cost?	Yes[3]	Yes	No	No	Yes[3]	Yes[3,4]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$

b is the branching factor; m is the maximum depth of the search tree;
d is the depth of the shallowest solution, or is m when there is no solution;
ℓ is the depth limit

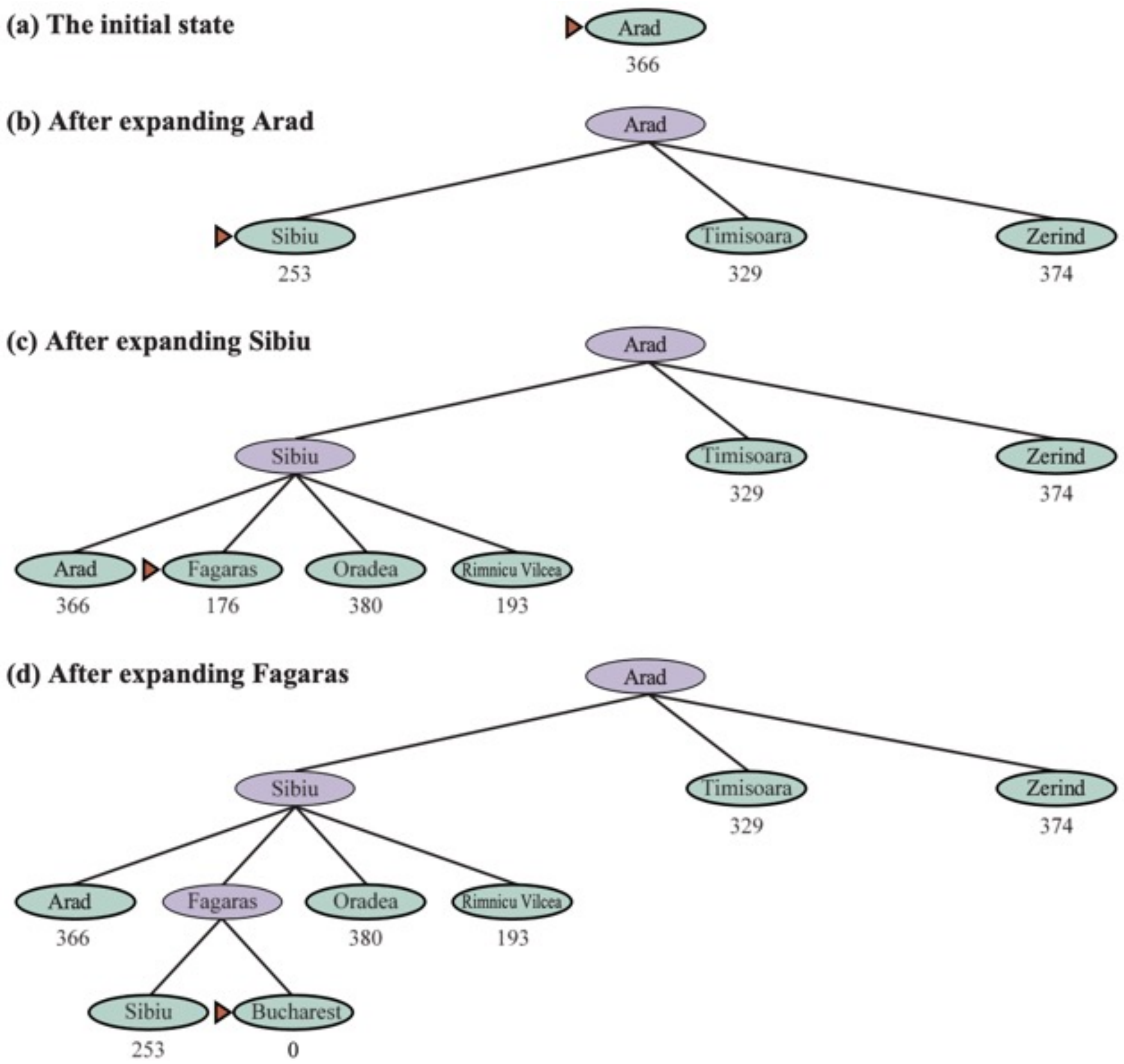
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Values of *hSLD*
—straight-line distances to Bucharest.

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

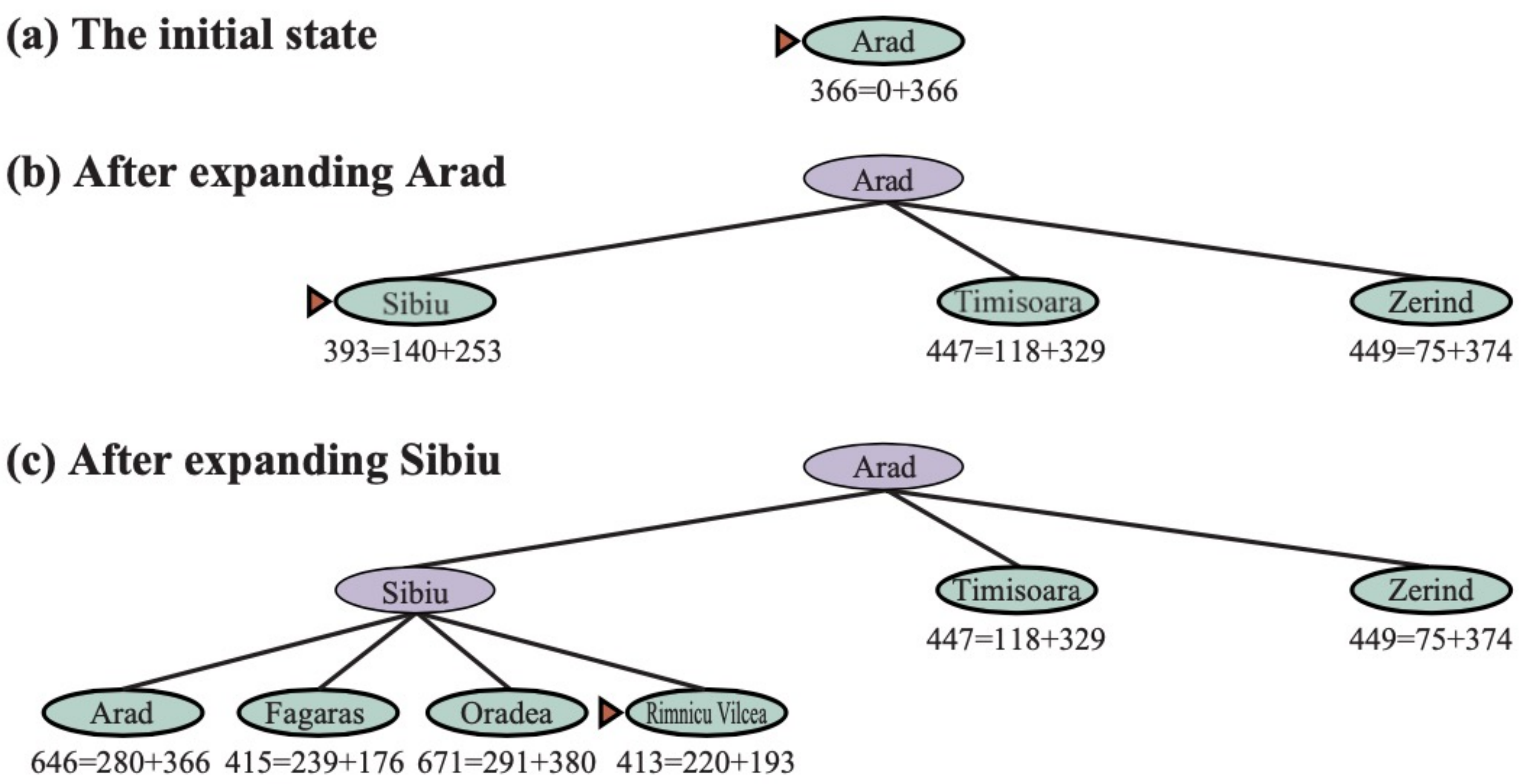
A∗ search



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A∗ search

Nodes are labeled with $f = g + h$.
The h values are the Straight-Line Distances heuristic h_{SLD}



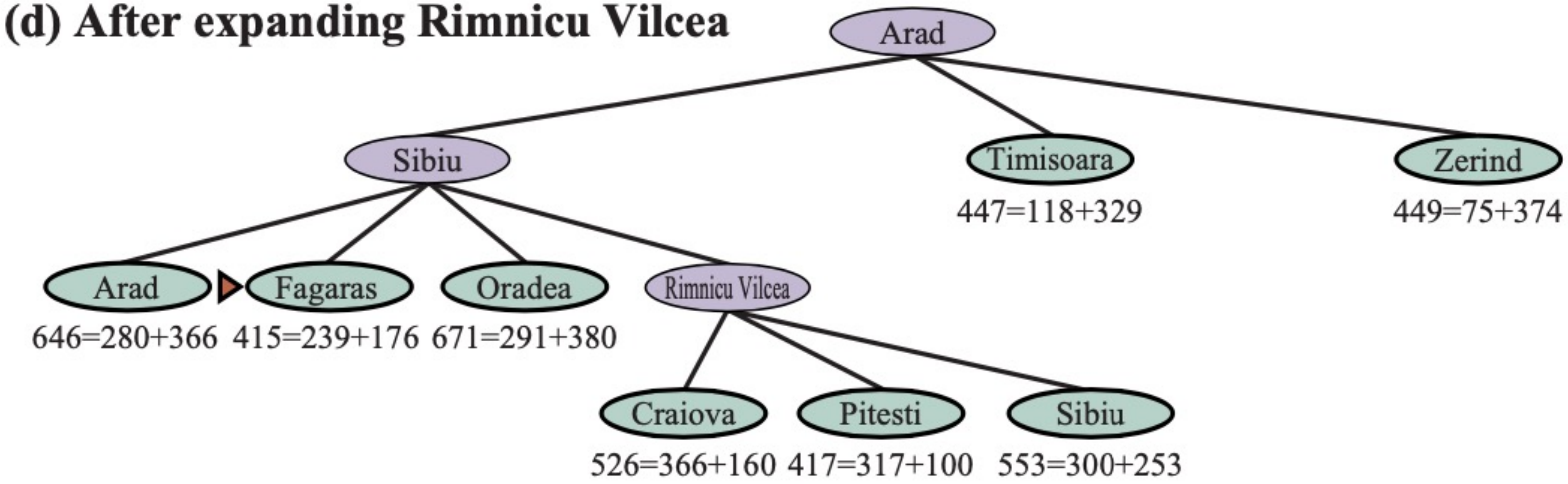
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A∗ search

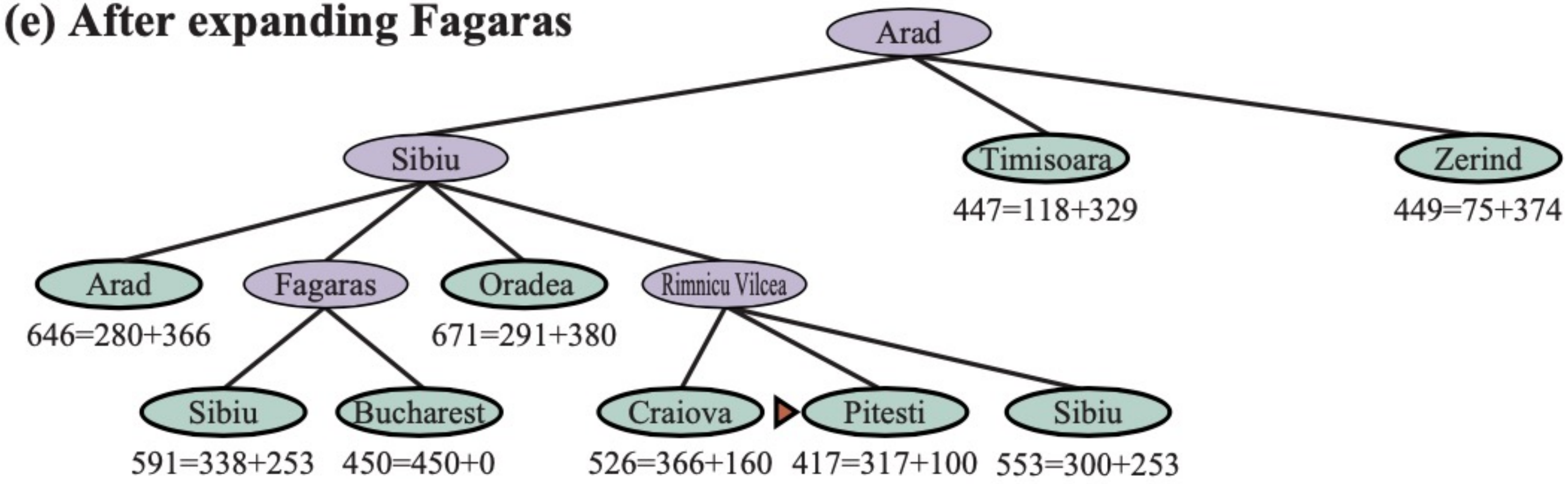
Nodes are labeled with $f = g + h$.

The h values are the Straight-Line Distances heuristic h_{SLD}

(d) After expanding Rimnicu Vilcea



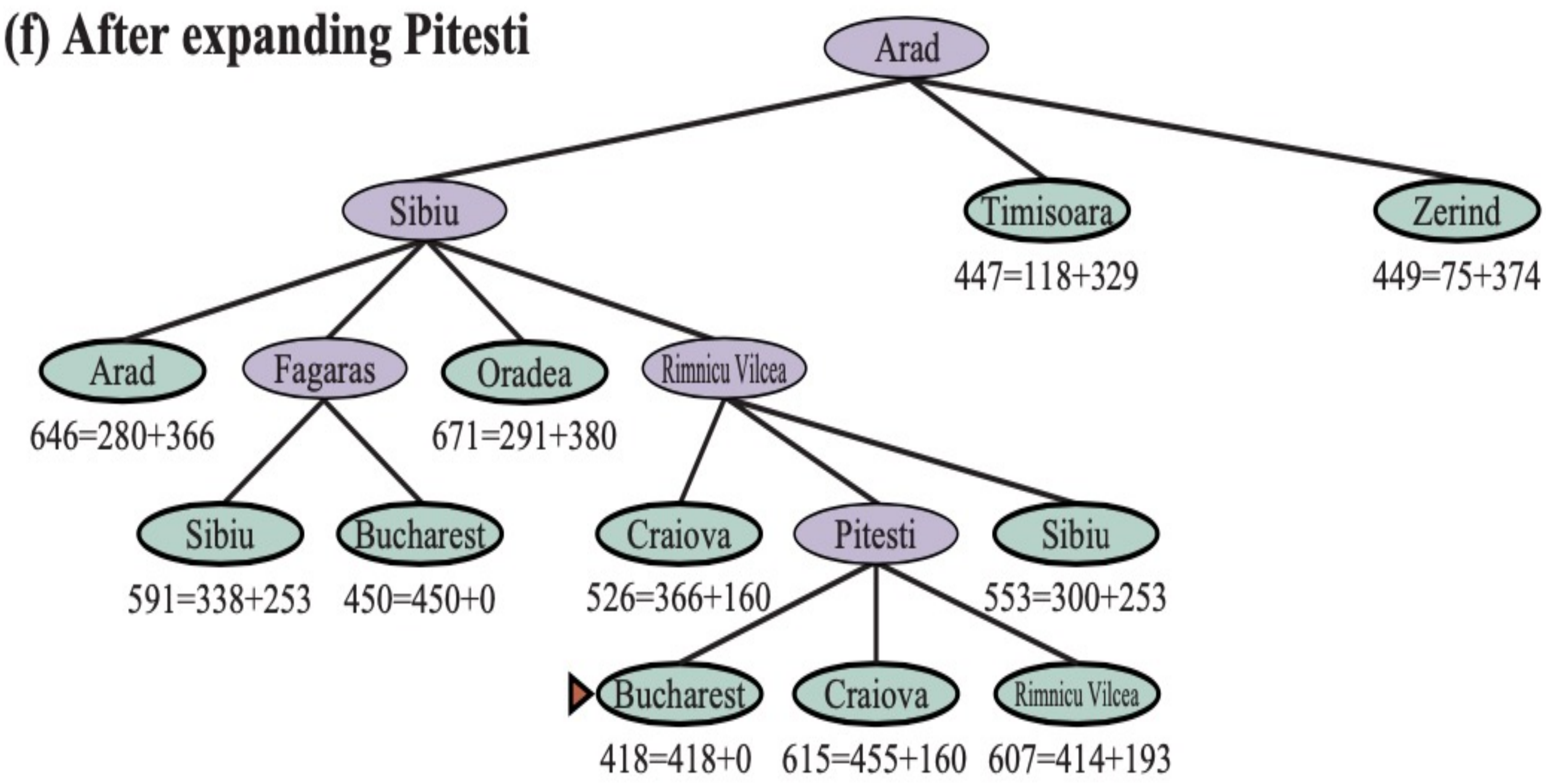
(e) After expanding Fagaras



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

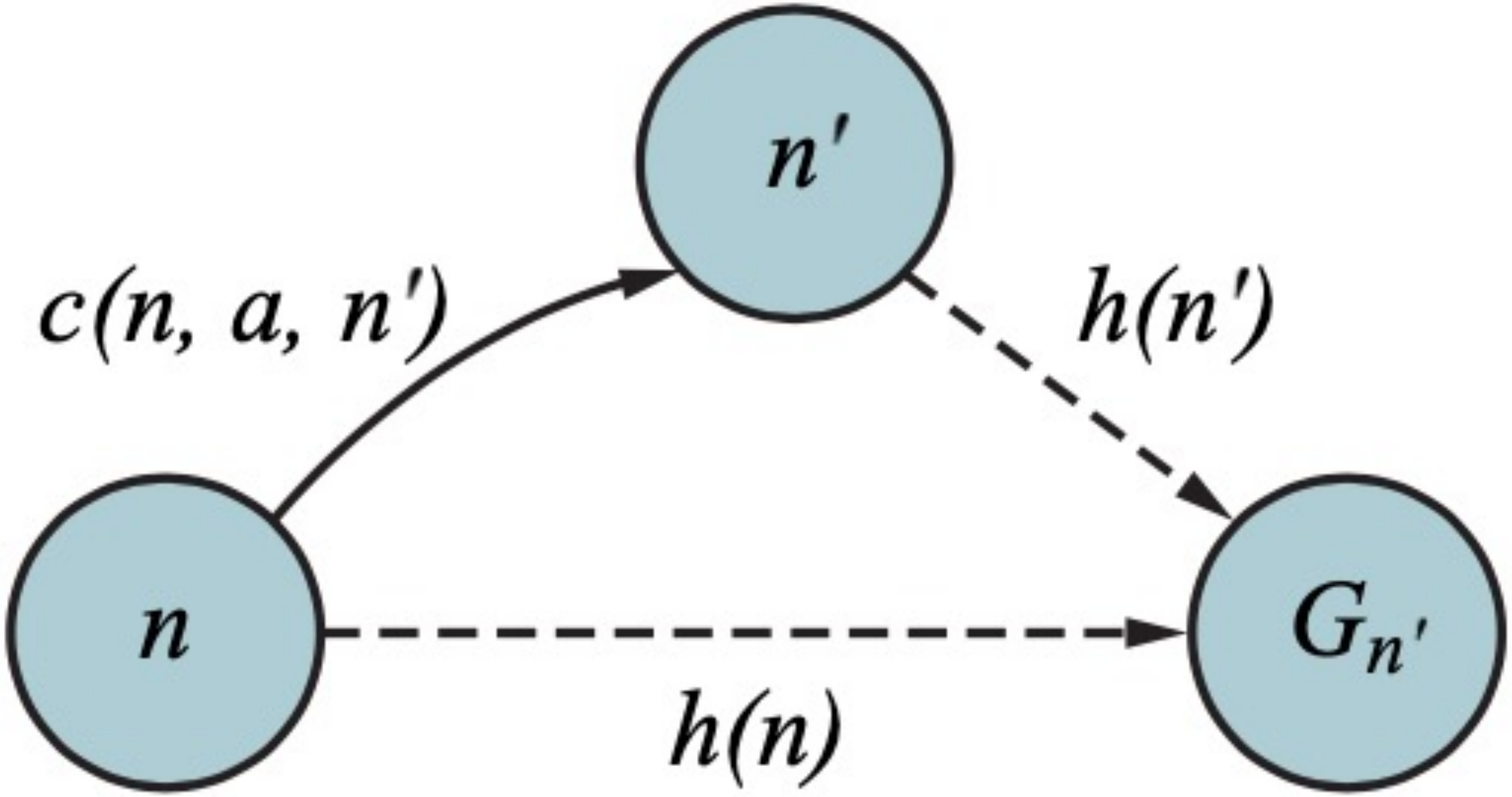
A∗ search

Nodes are labeled with $f = g + h$.
The h values are the Straight-Line Distances heuristic h_{SLD}



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

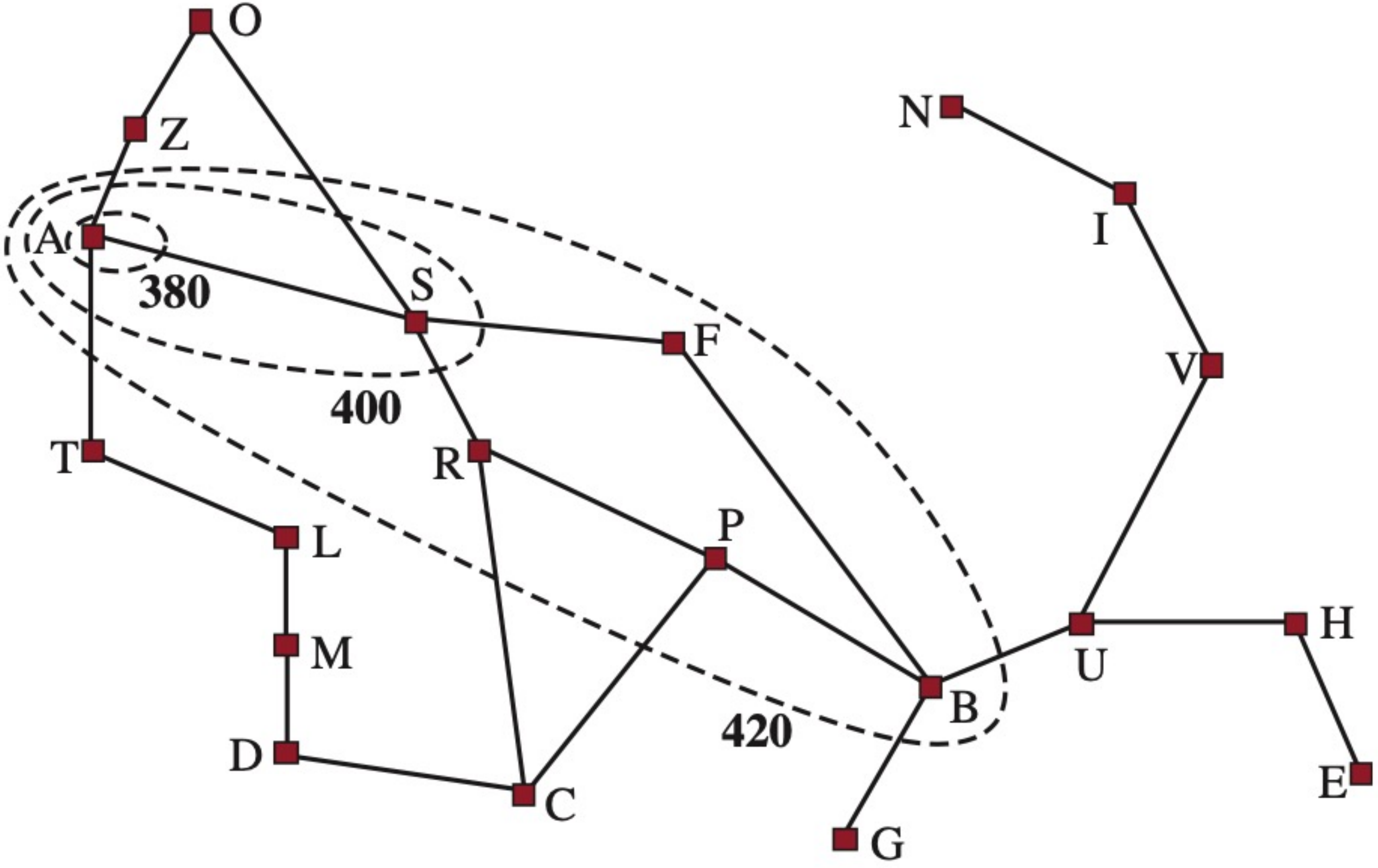
Triangle Inequality



If the heuristic h is consistent, then the single number $h(n)$ will be less than the sum of the cost c(n, a, a′) of the action from n to n′ plus the heuristic estimate h(n′).

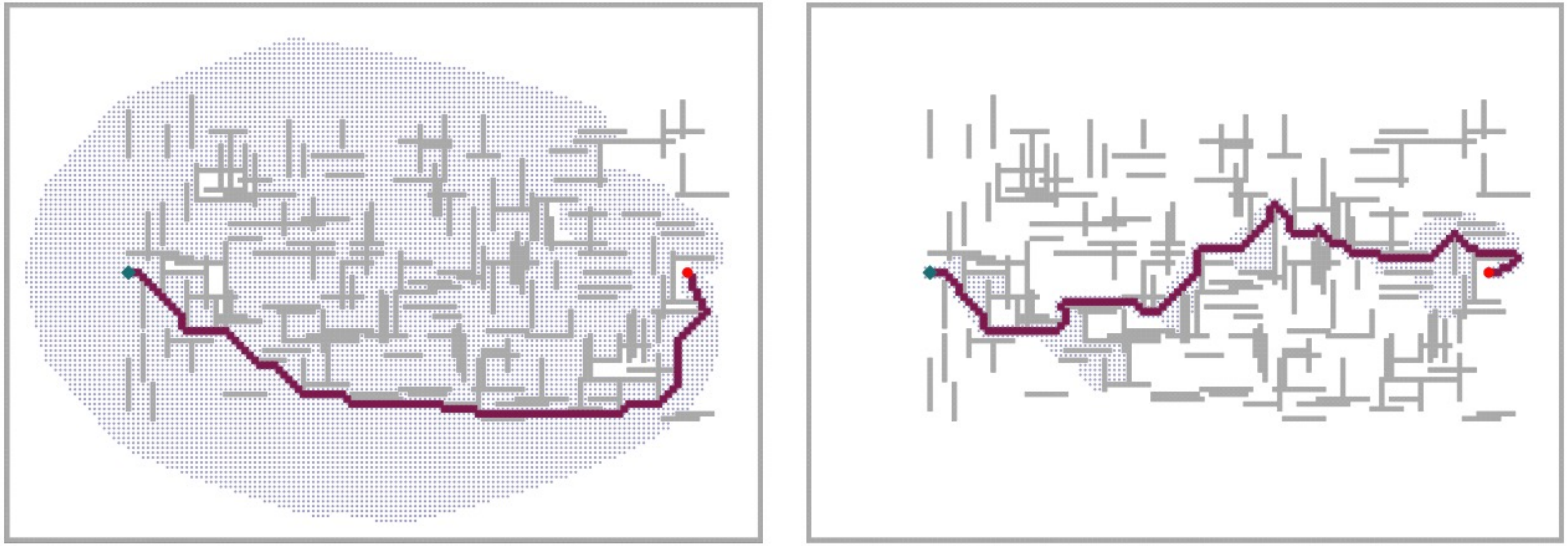
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Map of Romania showing contours *at f = 380, f = 400*, and *f = 420*, with Arad as the start state



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

(a) A∗ Search
(b) Weighted A∗ Search



(a) (b)

The gray bars are obstacles, the purple line is the path from the green start to red goal, and the small dots are states that were reached by each search.
On this particular problem, weighted A* explores 7 times fewer states and finds a path that is 5% more costly.

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

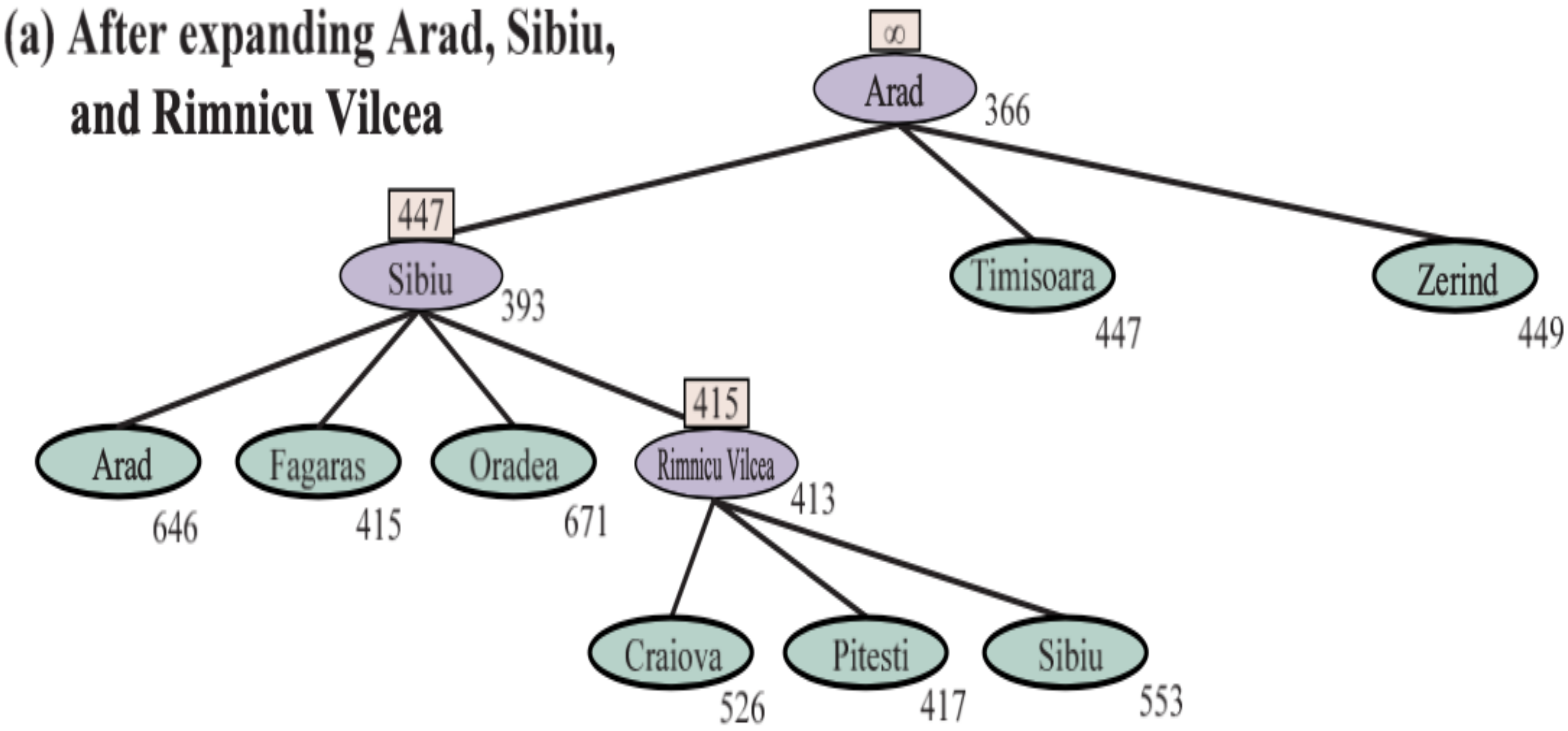
Recursive Best-First Search (RBFS) Algorithm

```
function RECURSIVE-BEST-FIRST-SEARCH(problem) returns a solution or failure
   solution, fvalue ← RBFS(problem, NODE(problem.INITIAL), ∞)
  return solution

function RBFS(problem, node, f_limit) returns a solution or failure, and a new f-cost limit
   if problem.IS-GOAL(node.STATE) then return node
   successors ← LIST(EXPAND(node))
   if successors is empty then return failure, ∞
   for each s in  successors do        // update f with value from previous search
      s.f ← max(s.PATH-COST + h(s), node.f))
   while true do
      best ← the node in successors with lowest f-value
      if best.f > f_limit then return failure, best.f
      alternative ← the second-lowest f-value among successors
      result, best.f ← RBFS(problem, best, min(f_limit, alternative))
      if result ≠ failure then return result, best.f
```

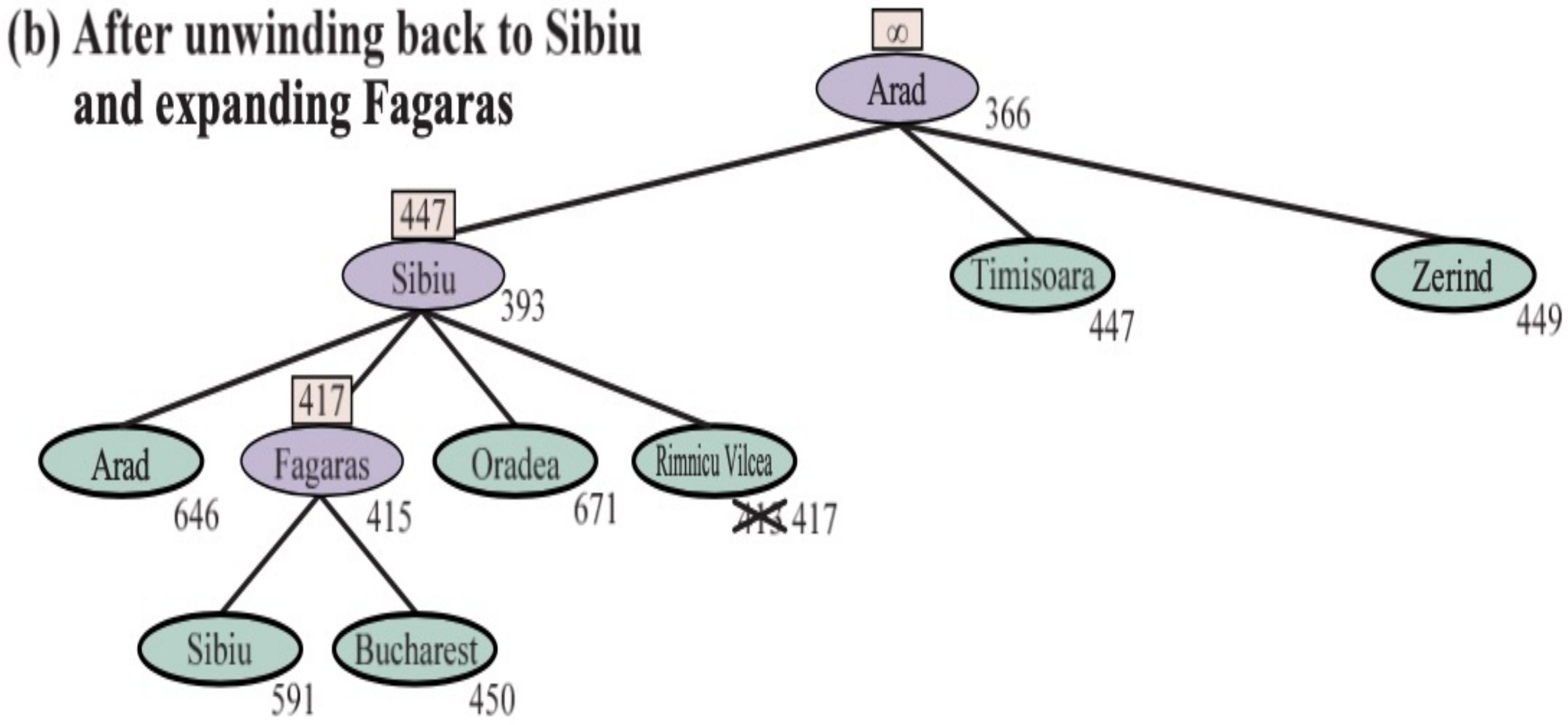
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Recursive Best-First Search (RBFS)



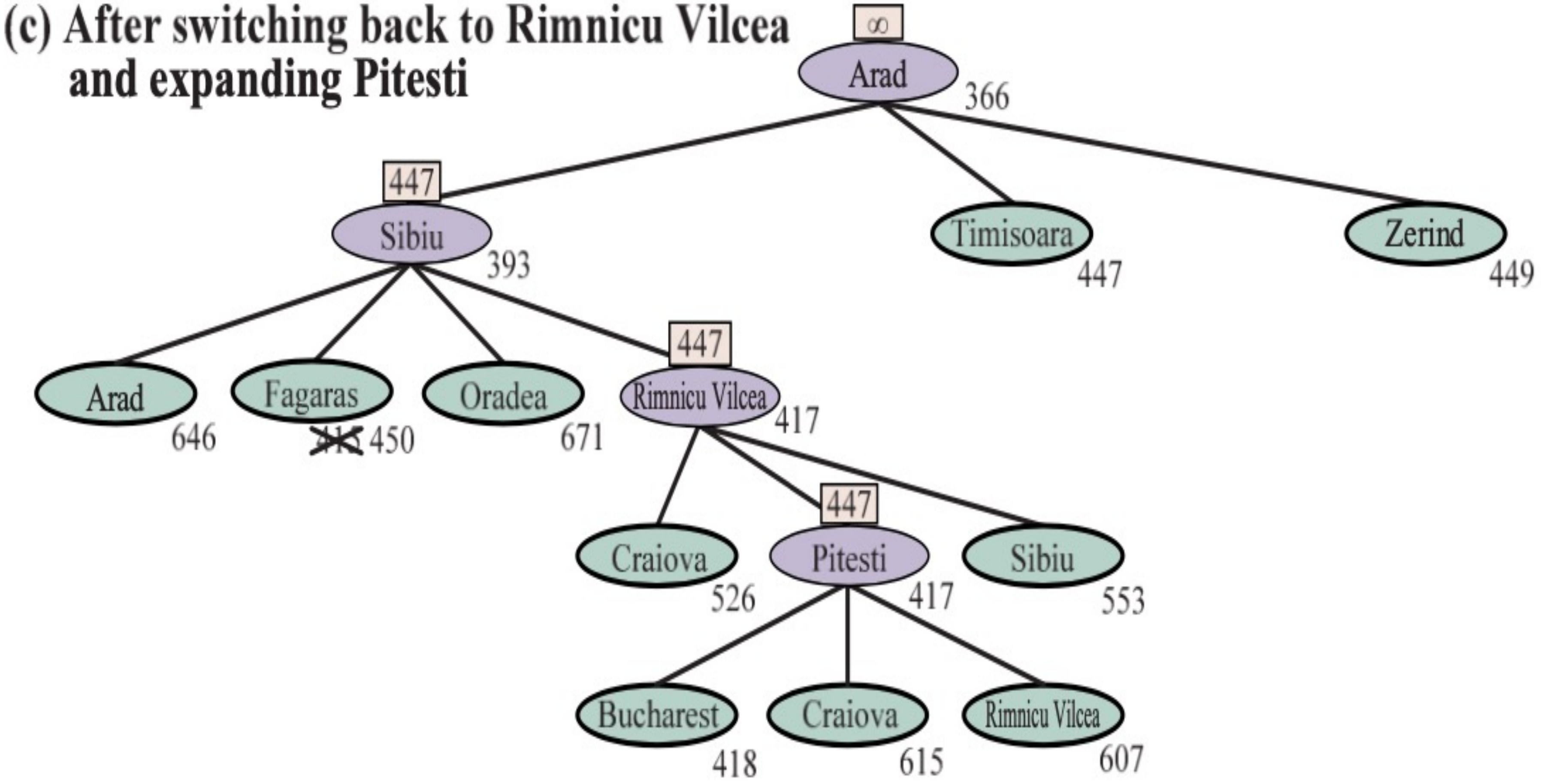
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Recursive Best-First Search (RBFS)

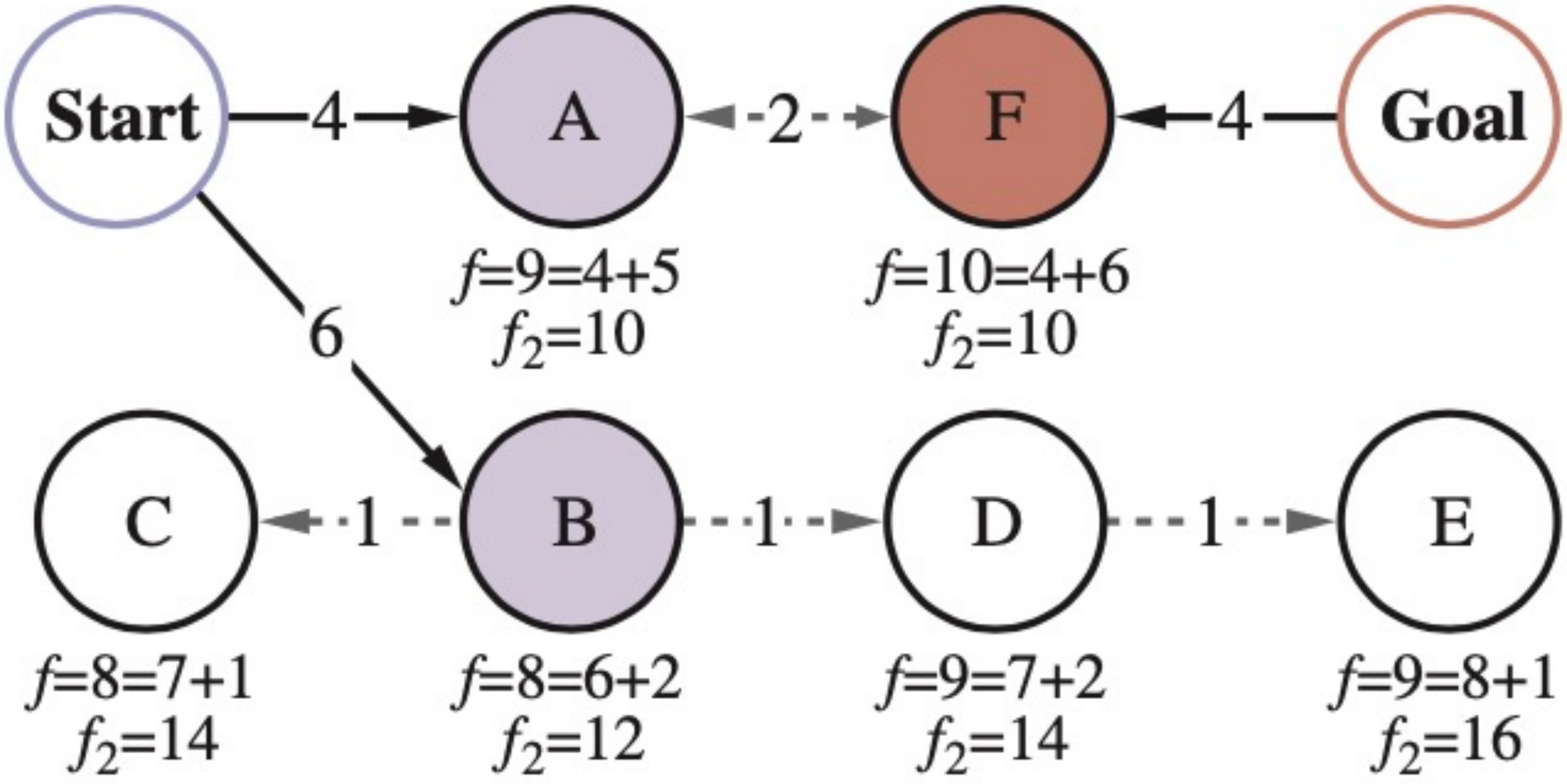


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Recursive Best-First Search (RBFS)



Bidirectional Search maintains two frontiers

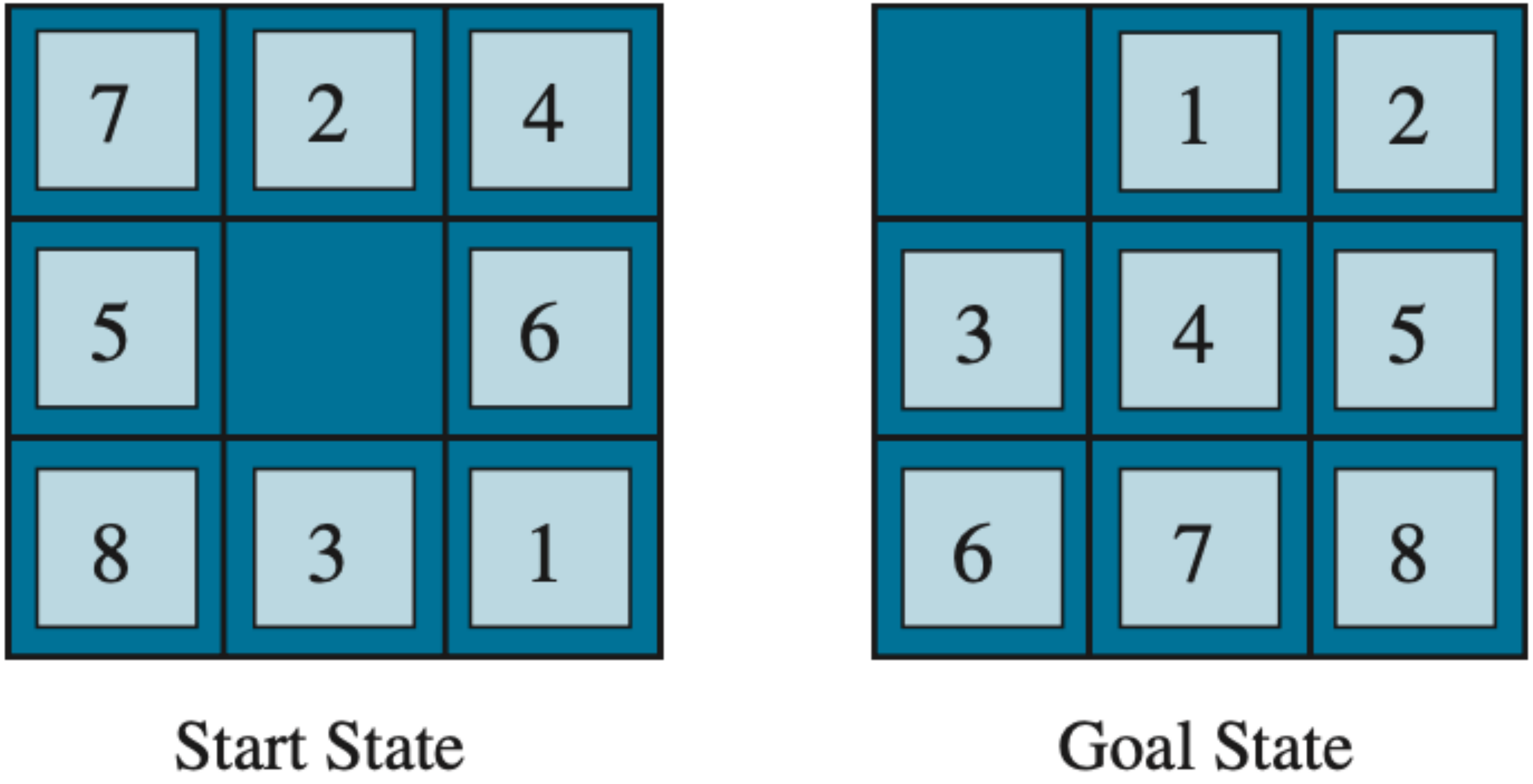


On the left, nodes A and B are successors of Start;
on the right, node F is an inverse successor of Goal

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A typical instance of the 8-puzzle

The shortest solution is 26 actions long



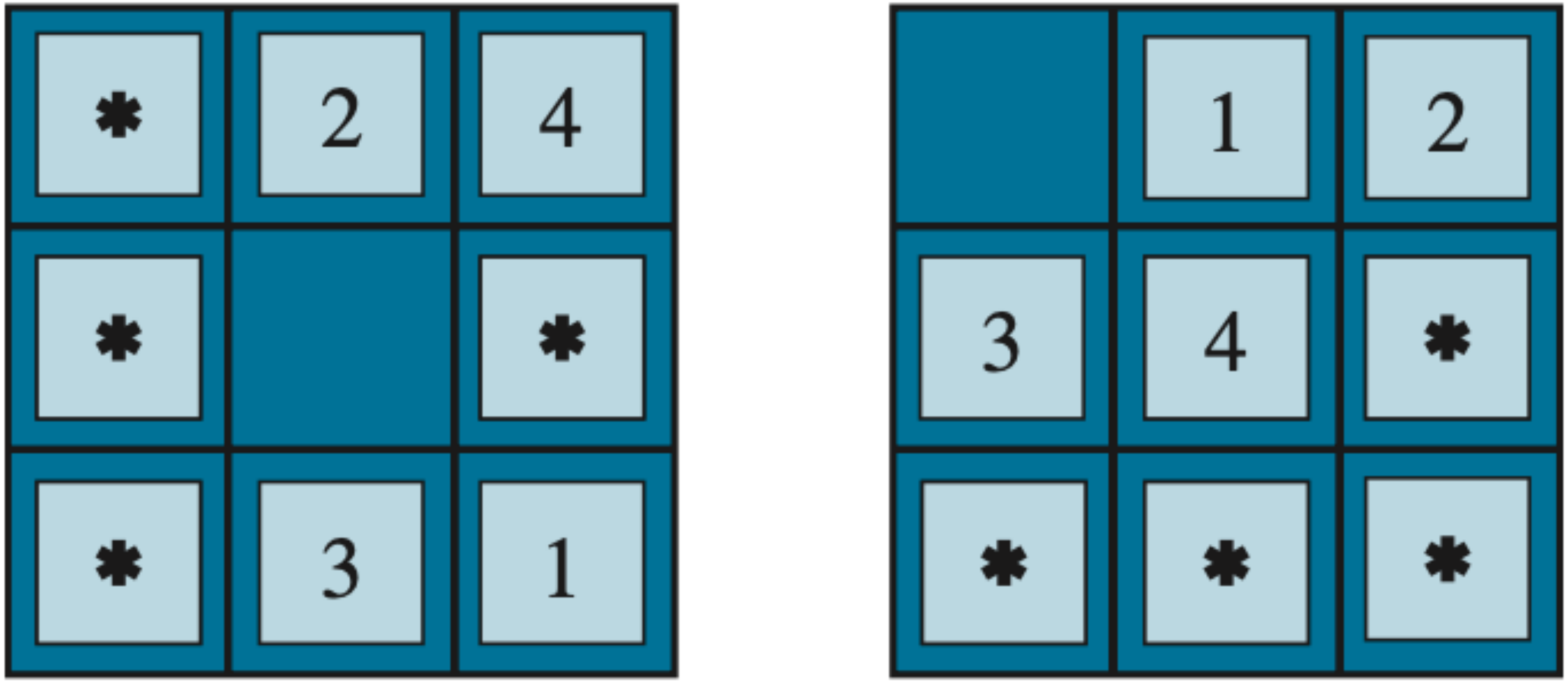
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Comparison of the search costs and effective branching factors for 8-puzzle problems

	Search Cost (nodes generated)			Effective Branching Factor		
d	BFS	$A^*(h_1)$	$A^*(h_2)$	BFS	$A^*(h_1)$	$A^*(h_2)$
6	128	24	19	2.01	1.42	1.34
8	368	48	31	1.91	1.40	1.30
10	1033	116	48	1.85	1.43	1.27
12	2672	279	84	1.80	1.45	1.28
14	6783	678	174	1.77	1.47	1.31
16	17270	1683	364	1.74	1.48	1.32
18	41558	4102	751	1.72	1.49	1.34
20	91493	9905	1318	1.69	1.50	1.34
22	175921	22955	2548	1.66	1.50	1.34
24	290082	53039	5733	1.62	1.50	1.36
26	395355	110372	10080	1.58	1.50	1.35
28	463234	202565	22055	1.53	1.49	1.36

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A subproblem of the 8-puzzle



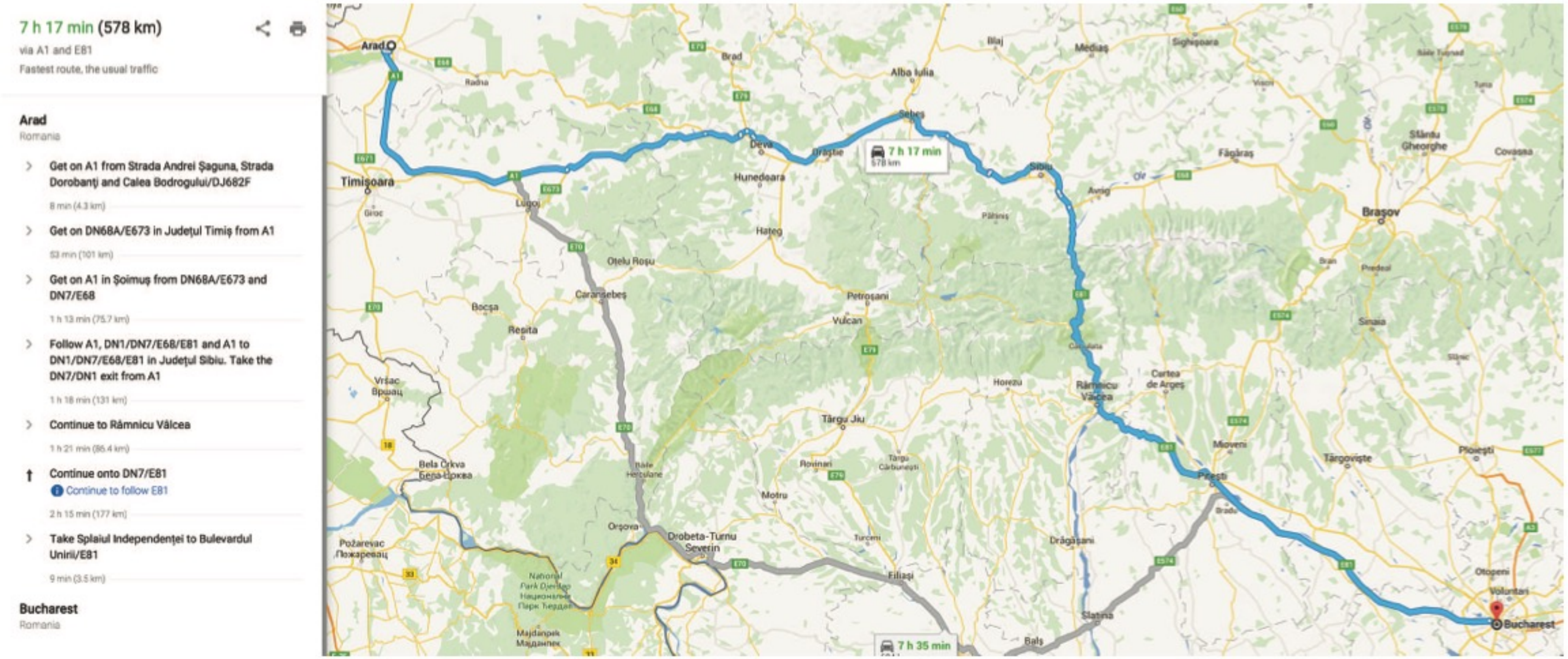
Start State

Goal State

The task is to get tiles 1, 2, 3, 4, and the blank into their correct positions, without worrying about what happens to the other tiles

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

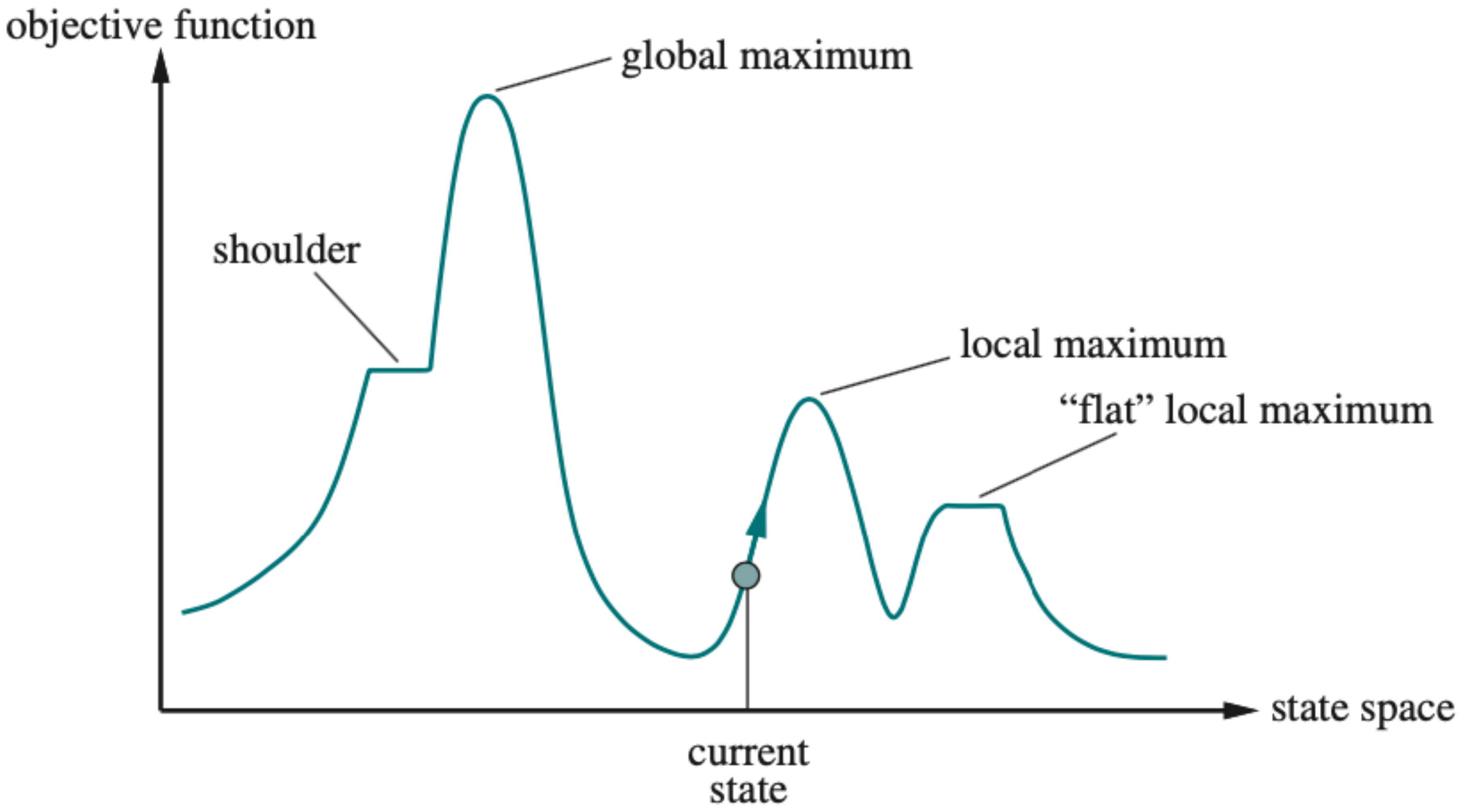
A Web service providing driving directions, computed by a search algorithm.



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Search in Complex Environments

A one-dimensional state-space landscape

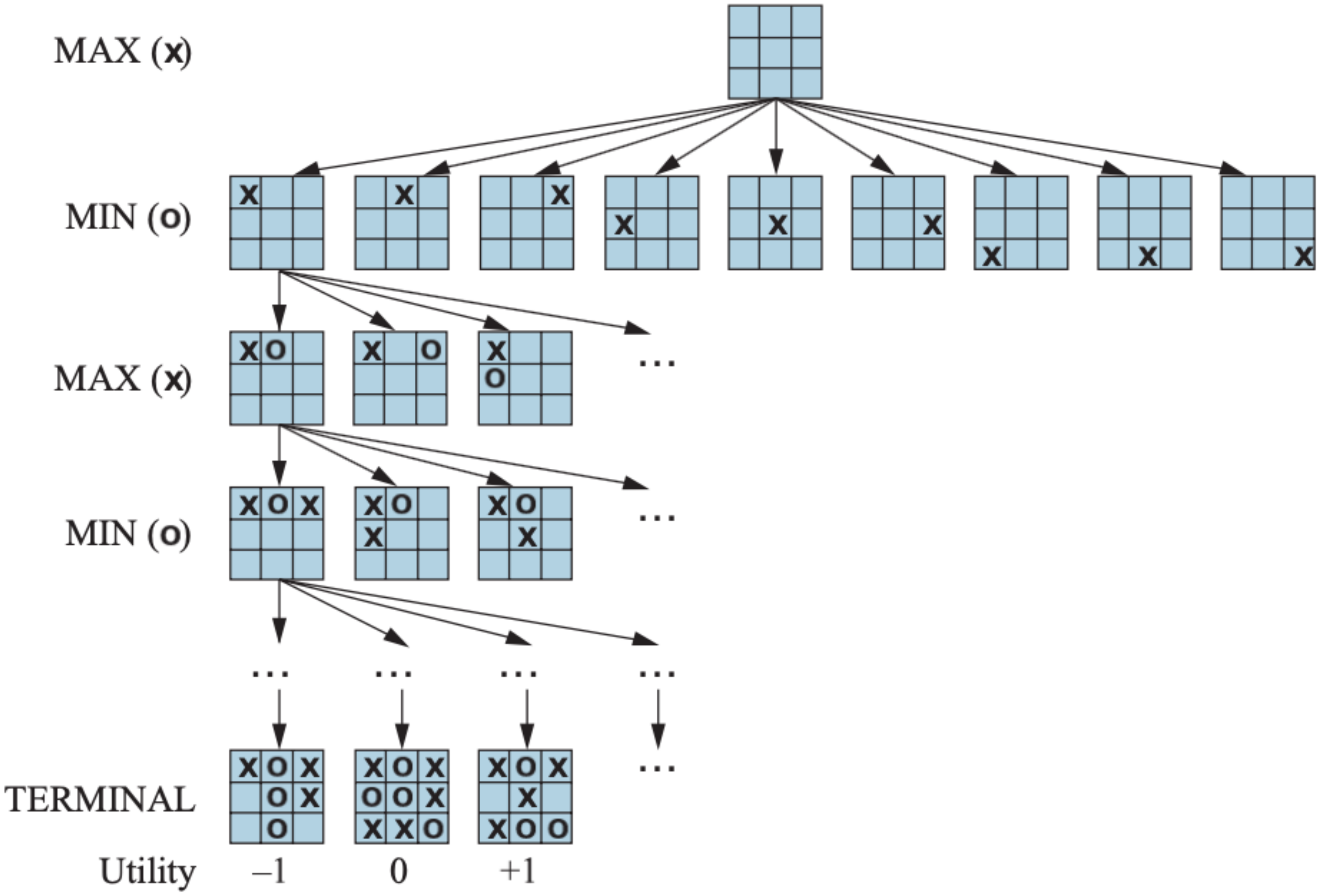


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Adversarial Search and Games

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Game Tree for the Game of Tic-tac-toe

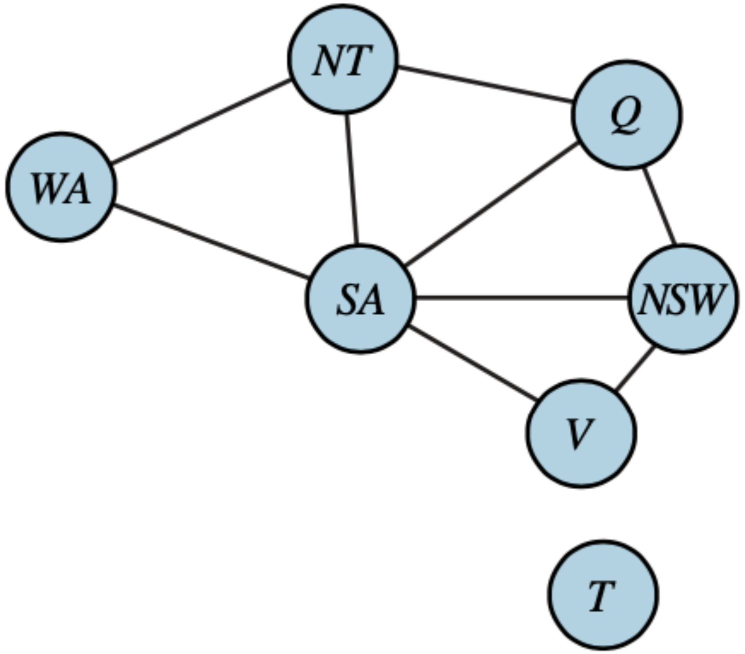


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Constraint Satisfaction Problems

The Map-Coloring Problem Represented as a Constraint Graph

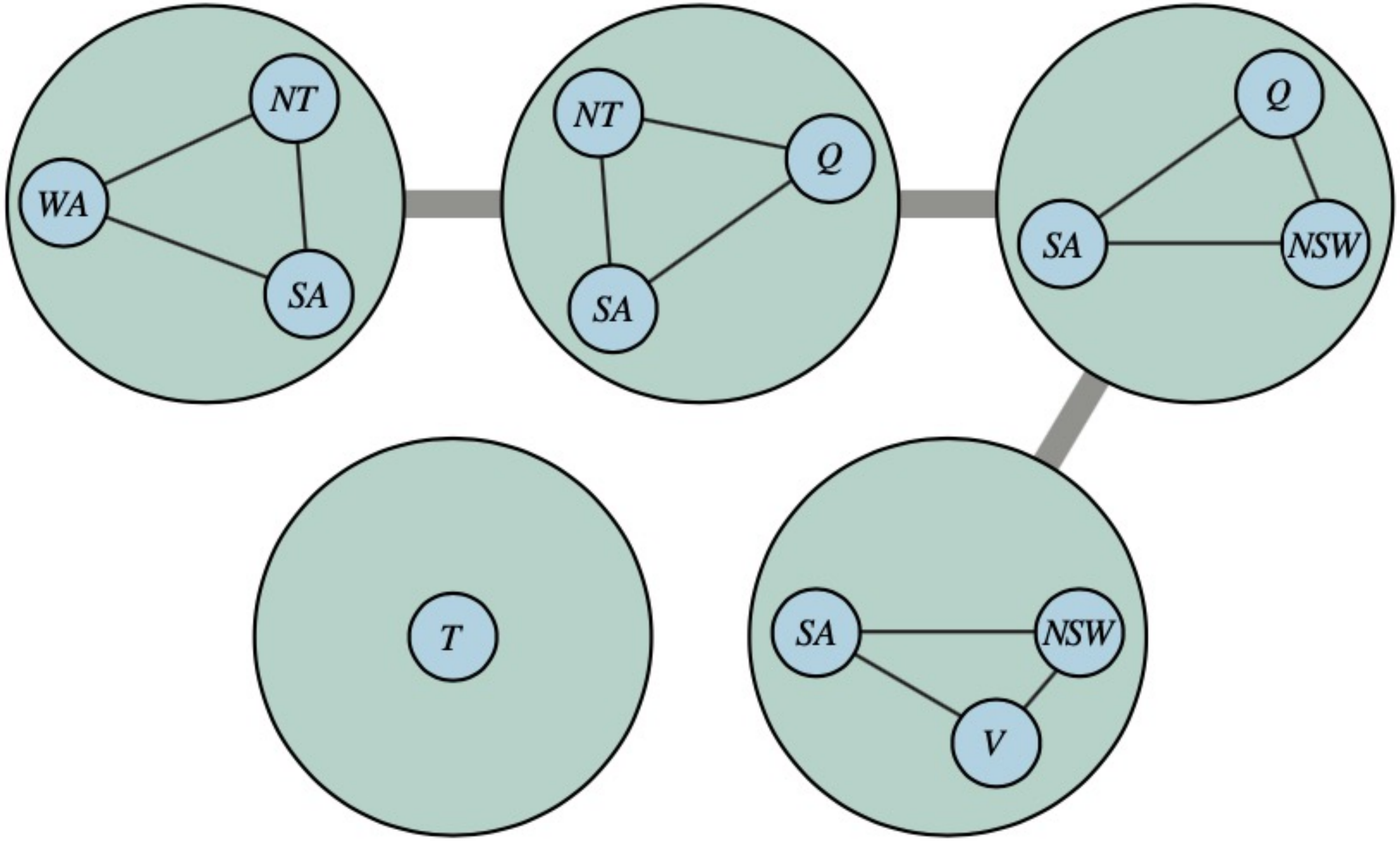
(a)



(b)

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A Tree Decomposition of the Constraint Graph



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence: A Modern Approach (AIMA)

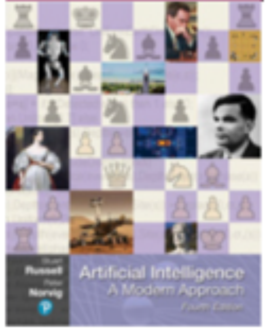
- **Artificial Intelligence: A Modern Approach (AIMA)**
  - http://aima.cs.berkeley.edu/
- **AIMA Python**
  - http://aima.cs.berkeley.edu/python/readme.html
  - https://github.com/aimacode/aima-python
- **Search**
  - http://aima.cs.berkeley.edu/python/search.html
- **Games: Adversarial Search**

  http://aima.cs.berkeley.edu/python/games.html
- **CSP (Constraint Satisfaction Problems)**
  - http://aima.cs.berkeley.edu/python/csp.html

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence: A Modern Approach (AIMA)

- ⌂ US Edition
- ⌂ Global Edition
- Acknowledgements
- Code
- Courses
- Editions
- Errata
- Exercises
- Figures
- Instructors Page
- Pseudocode
- Reviews



Artificial Intelligence: A Modern Approach, 4th US ed.

by Stuart Russell and Peter Norvig

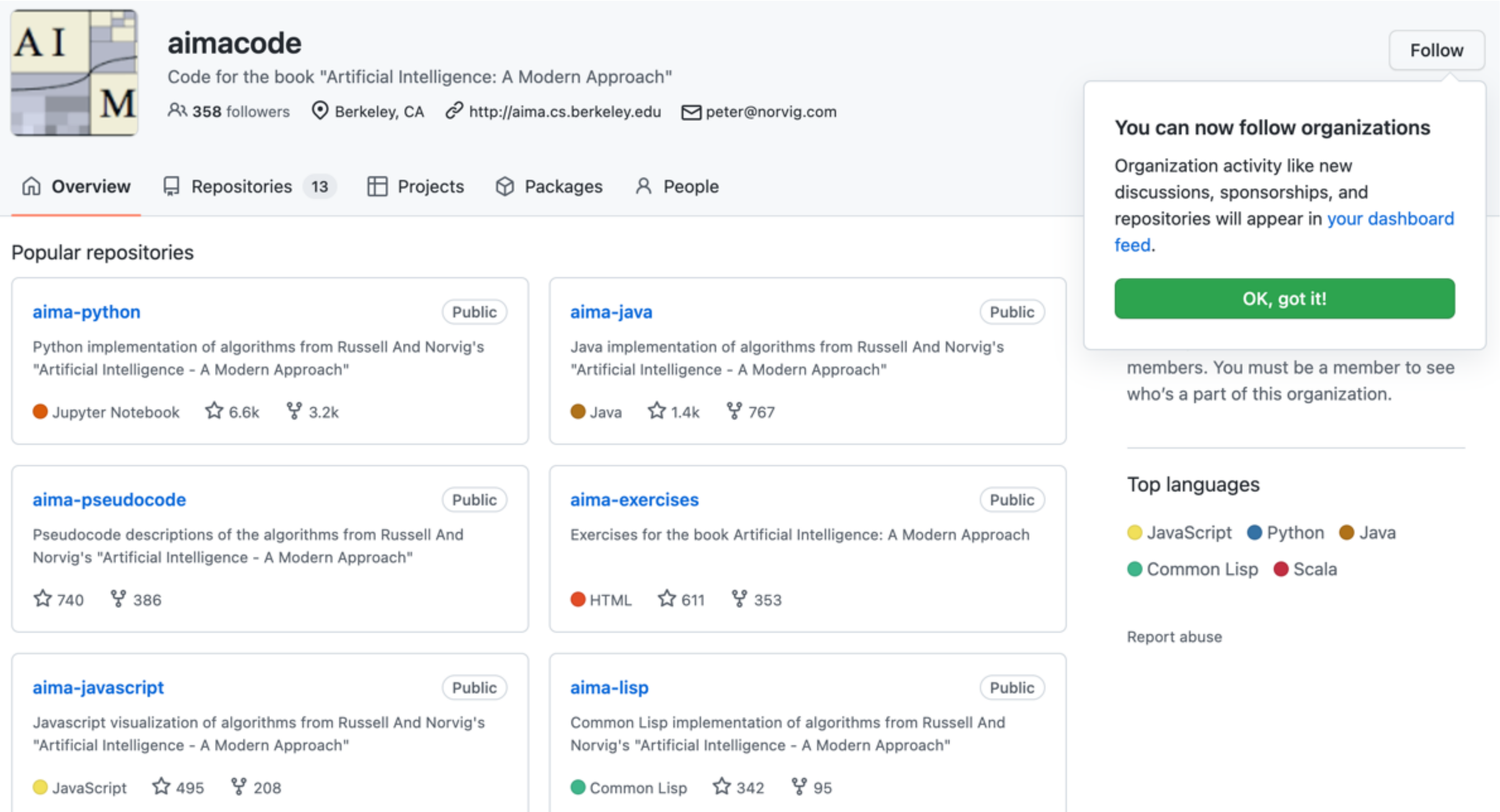
The authoritative, most-used AI textbook, adopted by over **1500** schools.

Table of Contents for the US Edition (or see the Global Edition)

Exercises (website)
Figures (pdf)
Code (website); Pseudocode (pdf)
Covers: US, Global

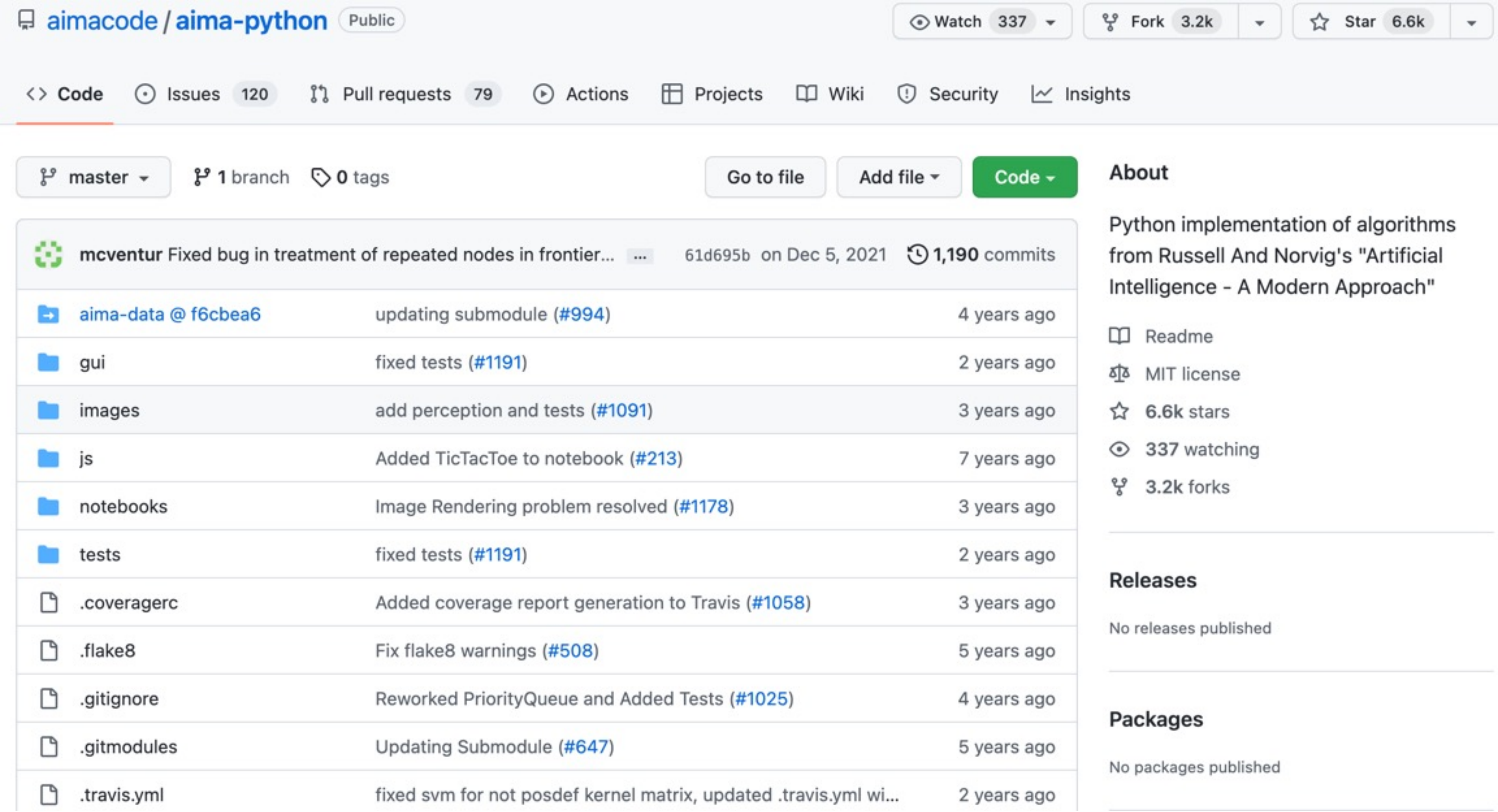
http://aima.cs.berkeley.edu/

AIMA Code



https://github.com/aimacode

AIMA Python



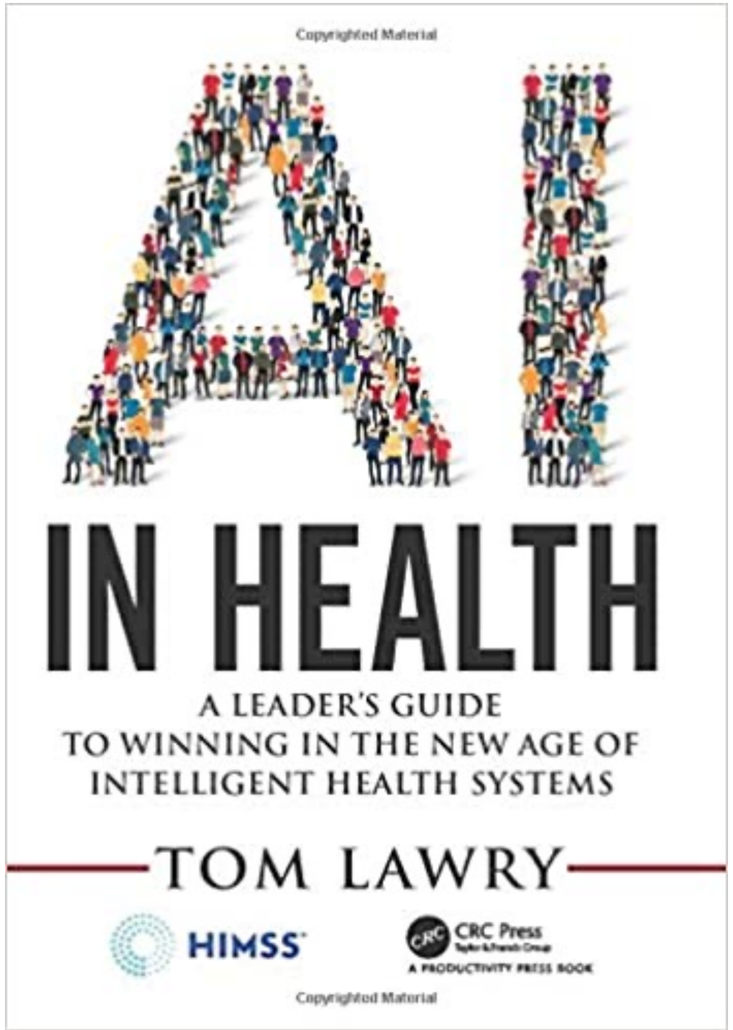
https://github.com/aimacode/aima-python

Tom Lawry (2020),

AI in Health:

A Leader’s Guide to Winning in the New Age of Intelligent Health Systems,

HIMSS Publishing



Source: Tom Lawry (2020), AI in Health: A Leader’s Guide to Winning in the New Age of Intelligent Health Systems, HIMSS Publishing

https://www.amazon.com/Health-HIMSS-Book-Tom-Lawry/dp/0367333716/

AI in Healthcare

Source: Secinaro, Silvana, Davide Calandra, Aurelio Secinaro, Vivek Muthurangu, and Paolo Biancone. "The role of artificial intelligence in healthcare: a structured literature review." BMC Medical Informatics and Decision Making 21, no. 1 (2021): 1-23.

Multimodal Fall Detection

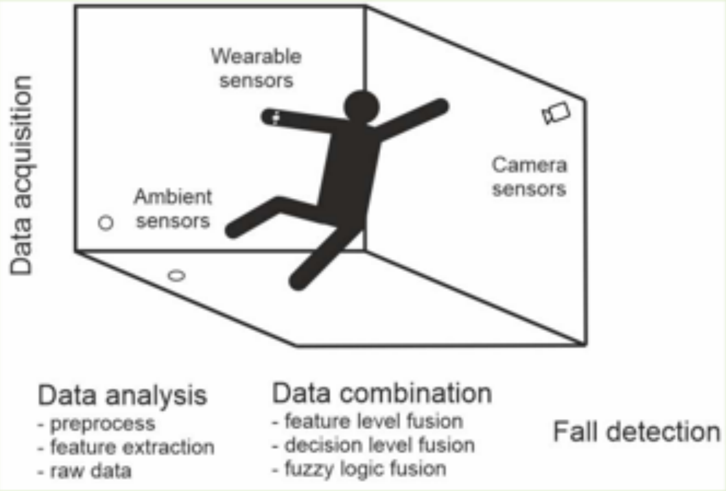
18398 IEEE SENSORS JOURNAL, VOL. 21, NO. 17, SEPTEMBER 1, 2021 IEEE Sensors Council

Performance, Challenges, and Limitations in Multimodal Fall Detection Systems: A Review

Vasileios-Rafail Xefteris, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris

Ambient Assisted Living (AAL)

***Abstract*—Fall events among older adults are a serious concern, having an impact on their health and well-being. The development of the Internet of Things (IoT) over the last years has led to the emergence of systems able to track abnormal body movements and falls, thus facilitating fall detection and in some cases prevention. Fusing information from multiple unrelated sources is one of the recent trends in healthcare systems. This work aims to provide a survey of recent methods and trends of multisensor data fusion in fall detection systems and discuss their performance, challenges, and limitations. The paper highlights the benefits of developing multimodal systems for fall detection compared to single-sensor approaches, categorizes the different methods applied to this field, and discusses issues and trends for future work.**



***Index Terms*—Data fusion, fall detection, multisensor fusion, non-wearable sensors, wearable sensors.**

Source: Xefteris, Vasileios-Rafail, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris. "Performance, challenges, and limitations in multimodal fall detection systems: a review." IEEE Sensors Journal (2021).

Multimodal Fall Detection

Ambient Assisted Living (AAL)

Sensor modalities	Intrusion	ROI specific	Accuracy	Power needs	Computational needs	Environment affected
Wearable	Obtrusive	No	Scenario dependent	High	Low/dependent	No
Ambient	No	Yes	Scenario dependent	Low	Low/dependent	Yes
Camera	Privacy	Yes	High	Low	High	Yes

Source: Xefteris, Vasileios-Rafail, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris. "Performance, challenges, and limitations in multimodal fall detection systems: a review." IEEE Sensors Journal (2021).

Challenges of Multimodal Fall Detection

Modalities combined	Performance	Response time	Power consumption	Unaddressed issues	Other advantages
Wearable	Reasonable accuracy.	Reasonably low time.	Up to 62 days.	Obtrusiveness.	Offer to other healthcare applications, continuous monitoring.
Non-wearable	High accuracy.	Reasonably low response time.	No action needed.	ROI restriction.	No recharge power needs.
Wearable and non-wearable	High accuracy.	Low response time.	No evidence.	Complexity.	Takes advantage of both modalities, no ROI restriction.

Source: Xefteris, Vasileios-Rafail, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris. "Performance, challenges, and limitations in multimodal fall detection systems: a review." IEEE Sensors Journal (2021).

Fall Detection
Non-Wearable Sensors Fusion

Reference	Year	Sensors	Method	Evaluation	Performance
[46]	2013	PIR and PM sensors.	Graph-theoretical concepts to track user and rule-based algorithm to detect falls.	Falls and ADLs from 5 healthy young subjects.	Accuracy: 82.86%
[47]	2014	Doppler radar sensor and PIR motion sensors.	SVM classifier on Doppler radar features, rule-based algorithm to correct false alarms using PIR data.	A week of continuous data monitoring of a volunteer.	Reduced false alarms by 63% with 100% detection rate.
[48]	2018	IR sensor and an ultrasonic distance sensor.	Thermal IR and ultrasonic features, SVM classifier.	180 falls and ADLs from 3 healthy young subjects, 6 continuous recordings.	Accuracy: 96.7% (discrete test), 90.3% (continuous test).
[52]	2018	Doppler radar sensor and RGB camera.	Multiple CNN, movement classification from radar, aspect ratio sequence from camera, max voting fusion.	1 type of fall and 3 types of ADLs from 3 subjects.	Accuracy: 99.85%
[53]	2019	Doppler radar and depth camera.	Joints' coordinates from depth camera, feature extraction from joints' coordinates and radar data, Linear Discriminant Classifier.	3 different datasets.	Sensitivity: 100% (FD).

Source: Xefteris, Vasileios-Rafail, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris. "Performance, challenges, and limitations in multimodal fall detection systems: a review." IEEE Sensors Journal (2021).

Fall Detection Datasets

Datasets	Posture samples	Subject					Type sensor	year
		Number	Height(cm)	Weight(kg)	Age(year)	Gender(M/F)		
Fall detection [4]	380	4	159-182	48-85	24-31	3M-1F	RGB camera	2007
Fall detection[5]	72	2	N/A	N/A	N/A	2M	RGB camera	2008
Multicam Fall [6]	24	1	N/A	N/A	N/A	M	8 RGB camera	2010
Le2i[7]	249	10	N/A	N/A	N/A	N/A	RGB camera	2013
Thermal simulated fall [8]	35	10	N/A	N/A	N/A	N/A	Thermal camera	2016
SisFall[9]	154	45	149-183	42-102	19-75	23M-21F	RGB camera, 2 accelerometers, 1 gyroscope	2016
UR Fall Detection[10]	70	5	N/A	N/A	N/A	5M	2 Kinect camera, accelerometer	2016
NTU RGB+D Action Recognition [11]	56880	302	N/A	N/A	N/A	N/A	Kinect camera v2	2016
UMA Fall [12]	531	17	155-195	50-93	18-55	10M-7F	Mobility sensors (smartphone)	2017
CMD Fall [13]	20	50	N/A	N/A	21-40	30M-20F	Kinect camera, accelerometer	2018
TST Fall Detection Dataset V2[8]	264	11	N/A	N/A	N/A	N/A	Microsoft Kinect v2, accelerometer	2018
UP-Fall[14]	561	17	N/A	N/A	22-58	N/A	Infrared ,inertial measurement	2019

Note: N/A_Not Available; M_Male; F_Femal

Source: Oumaima, Guendoul, Ait Abdelali Hamd, Tabii Youness, Oulad Haj Thami Rachid, and Bourja Omar. "Vision-based fall detection and prevention for the elderly people: A review & ongoing research." In 2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS), pp. 1-6. IEEE, 2021.

Human Action Recognition (HAR)

Human Action Recognition from Various Data Modalities: A Review

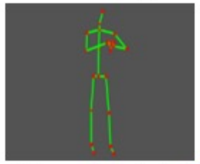
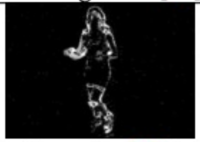
Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu

Abstract—Human Action Recognition (HAR) aims to understand human behavior and assign a label to each action. It has a wide range of applications, and therefore has been attracting increasing attention in the field of computer vision. Human actions can be represented using various data modalities, such as RGB, skeleton, depth, infrared, point cloud, event stream, audio, acceleration, radar, and WiFi signal, which encode different sources of useful yet distinct information and have various advantages depending on the application scenarios. Consequently, lots of existing works have attempted to investigate different types of approaches for HAR using various modalities. In this paper, we present a comprehensive survey of recent progress in deep learning methods for HAR based on the type of input data modality. Specifically, we review the current mainstream deep learning methods for single data modalities and multiple data modalities, including the fusion-based and the co-learning-based frameworks. We also present comparative results on several benchmark datasets for HAR, together with insightful observations and inspiring future research directions.

Index Terms—Human Action Recognition, Deep Learning, Data Modality, Single Modality, Multi-modality.

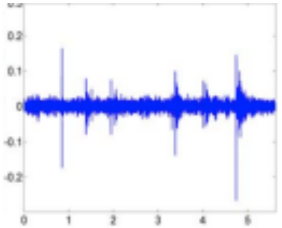
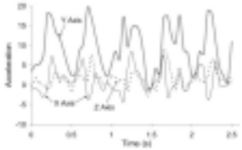
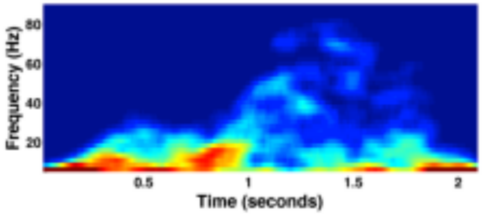
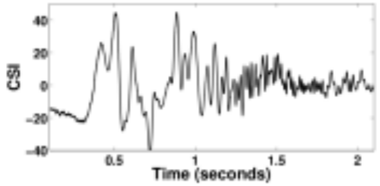
Source: Sun, Zehua, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. "Human action recognition from various data modalities: A review." IEEE transactions on pattern analysis and machine intelligence (2022).

Human Action Recognition (HAR) Modality

	Modality	Example	Pros	Cons
Visual Modality	RGB	 Hand-waving [27]	· Provide rich appearance information · Easy to obtain and operate · Wide range of applications	· Sensitive to viewpoint · Sensitive to background · Sensitive to illumination
	3D Skeleton	 Looking at watch [28]	· Provide 3D structural information of subject pose · Simple yet informative · Insensitive to viewpoint · Insensitive to background	· Lack of appearance information · Lack of detailed shape information · Noisy
	Depth	 Mopping floor [29]	· Provide 3D structural information · Provide geometric shape information	· Lack of color and texture information · Limited workable distance
	Infrared Sequence	 Pushing [30]	· Workable in dark environments	· Lack of color and texture information · Susceptible to sunlight
	Point Cloud	 Bending over [31]	· Provide 3D information · Provide geometric shape information · Insensitive to viewpoint	· Lack of color and texture information · High computational complexity
	Event Stream	 Running [32]	· Avoid much visual redundancy · High dynamic range · No motion blur	· Asynchronous output · Spatio-temporally sparse · Capturing device is relatively expensive

Source: Sun, Zehua, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. "Human action recognition from various data modalities: A review." IEEE transactions on pattern analysis and machine intelligence (2022).

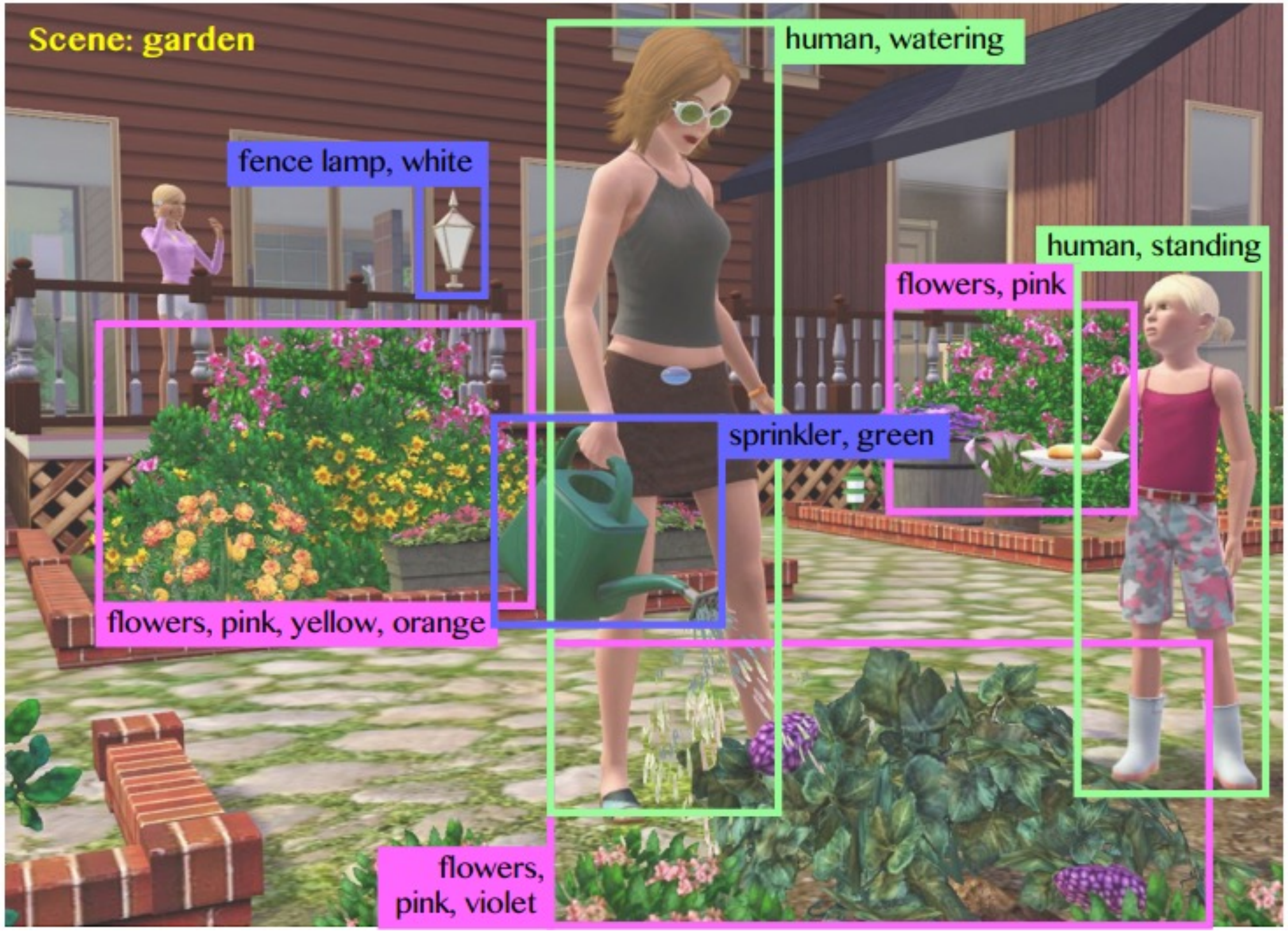
Human Action Recognition (HAR) Modality

Non-visual Modality	Audio	 Audio wave of jumping [33]	· Easy to locate actions in temporal sequence	· Lack of appearance information
	Acceleration	 Acceleration measurements of walking [34]	· Can be used for fine-grained HAR · Privacy protecting · Low cost	· Lack of appearance information · Capturing device needs to be carried by subject
	Radar	 Spectrogram of falling [35]	· Can be used for through-wall HAR · Insensitive to illumination · Insensitive to weather · Privacy protecting	· Lack of appearance information · Capturing device is relatively expensive
	WiFi	 CSI waveform of falling [35]	· Simple and convenient · Privacy protecting · Low cost	· Lack of appearance information · Sensitive to environments · Noisy

Source: Sun, Zehua, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. "Human action recognition from various data modalities: A review." IEEE transactions on pattern analysis and machine intelligence (2022).

Computer Vision in the Metaverse

with scene understanding, object detection, and human action/activity recognition



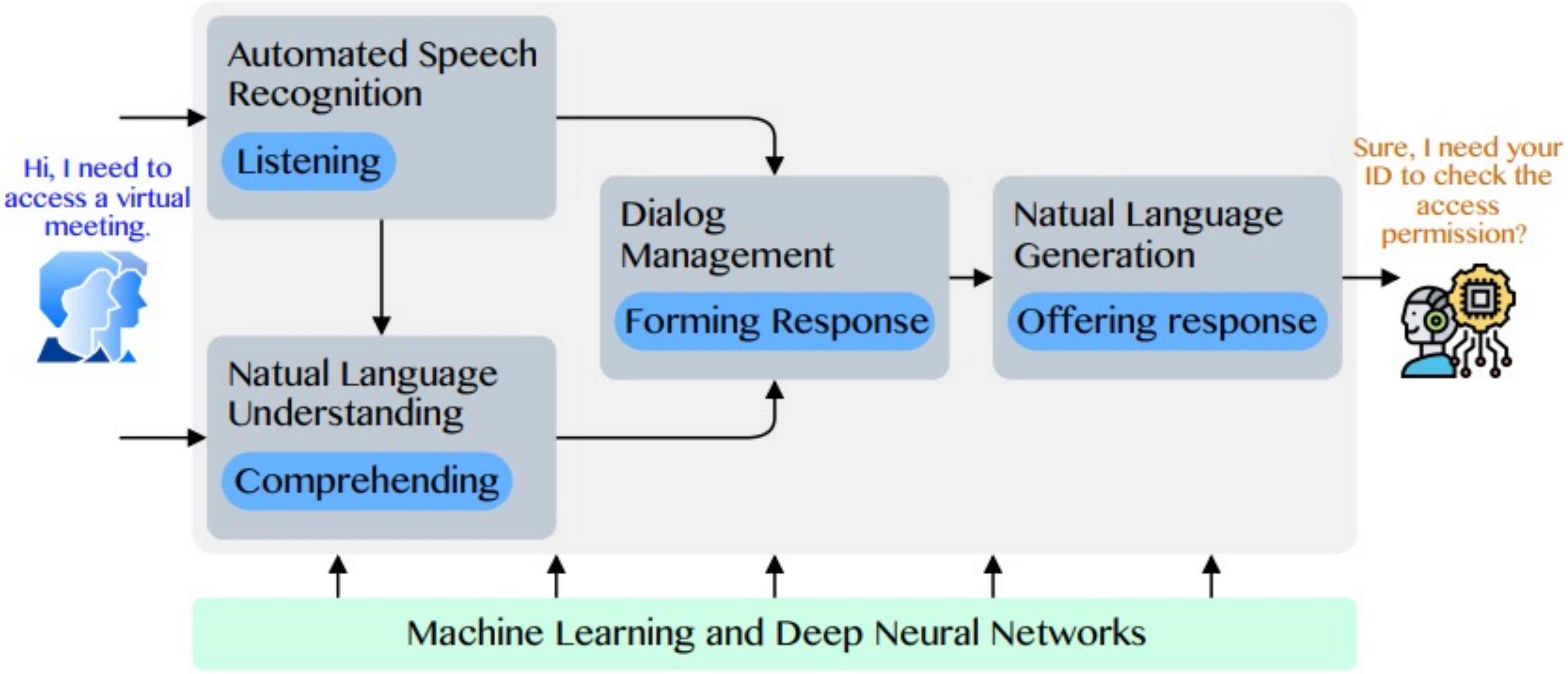
Source: Huynh-The, Thien, Quoc-Viet Pham, Xuan-Qui Pham, Thanh Thi Nguyen, Zhu Han, and Dong-Seong Kim (2022). "Artificial Intelligence for the Metaverse: A Survey." arXiv preprint arXiv:2202.10336.

Fall Detection

Source: https://www.ankecare.com/article/1929-2022-05-09-08-29-47

Conversational AI

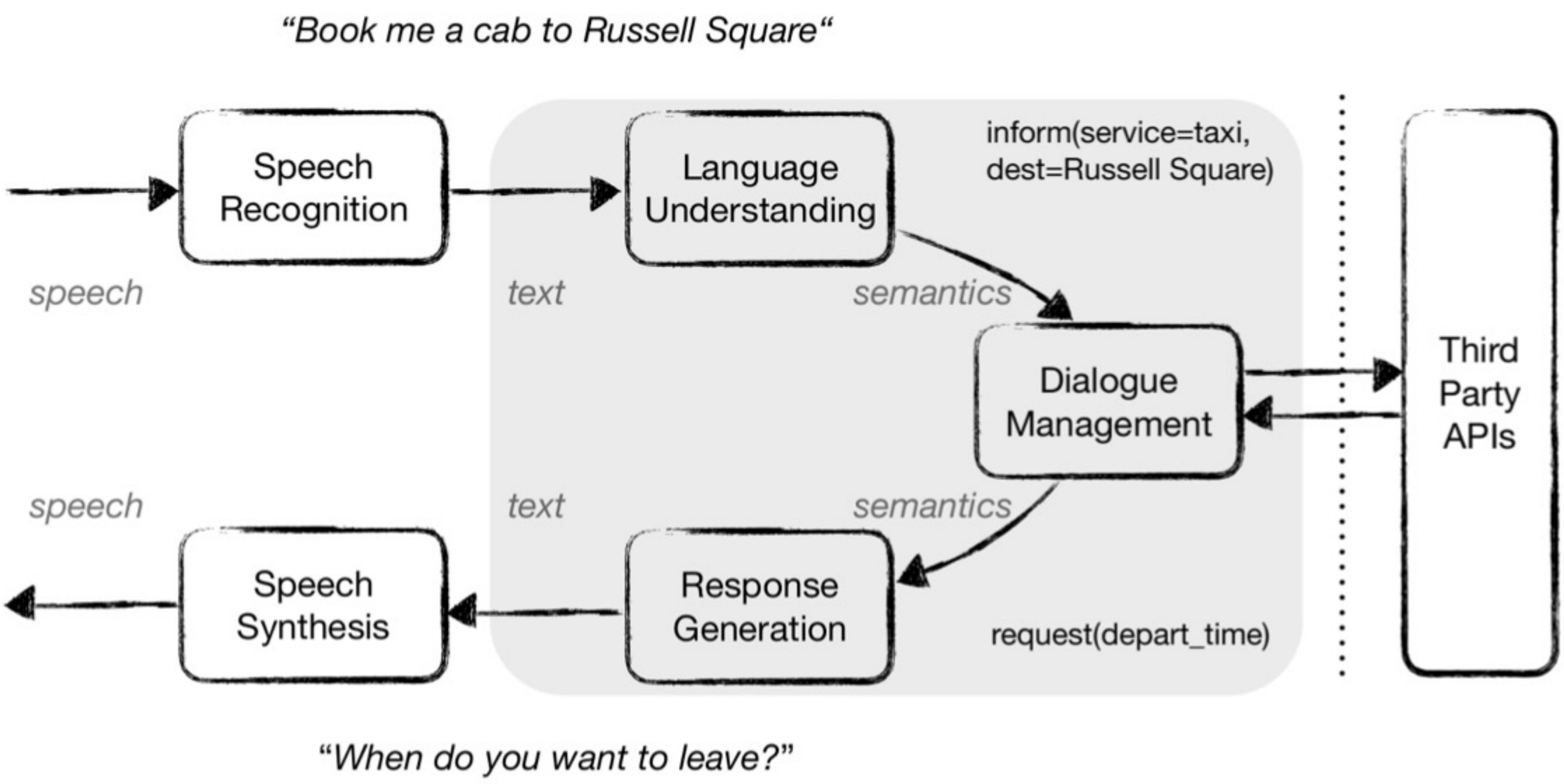
to deliver contextual and personal experience to users



Source: Huynh-The, Thien, Quoc-Viet Pham, Xuan-Qui Pham, Thanh Thi Nguyen, Zhu Han, and Dong-Seong Kim (2022). "Artificial Intelligence for the Metaverse: A Survey." arXiv preprint arXiv:2202.10336.

Task-Oriented Dialogue (ToD) System

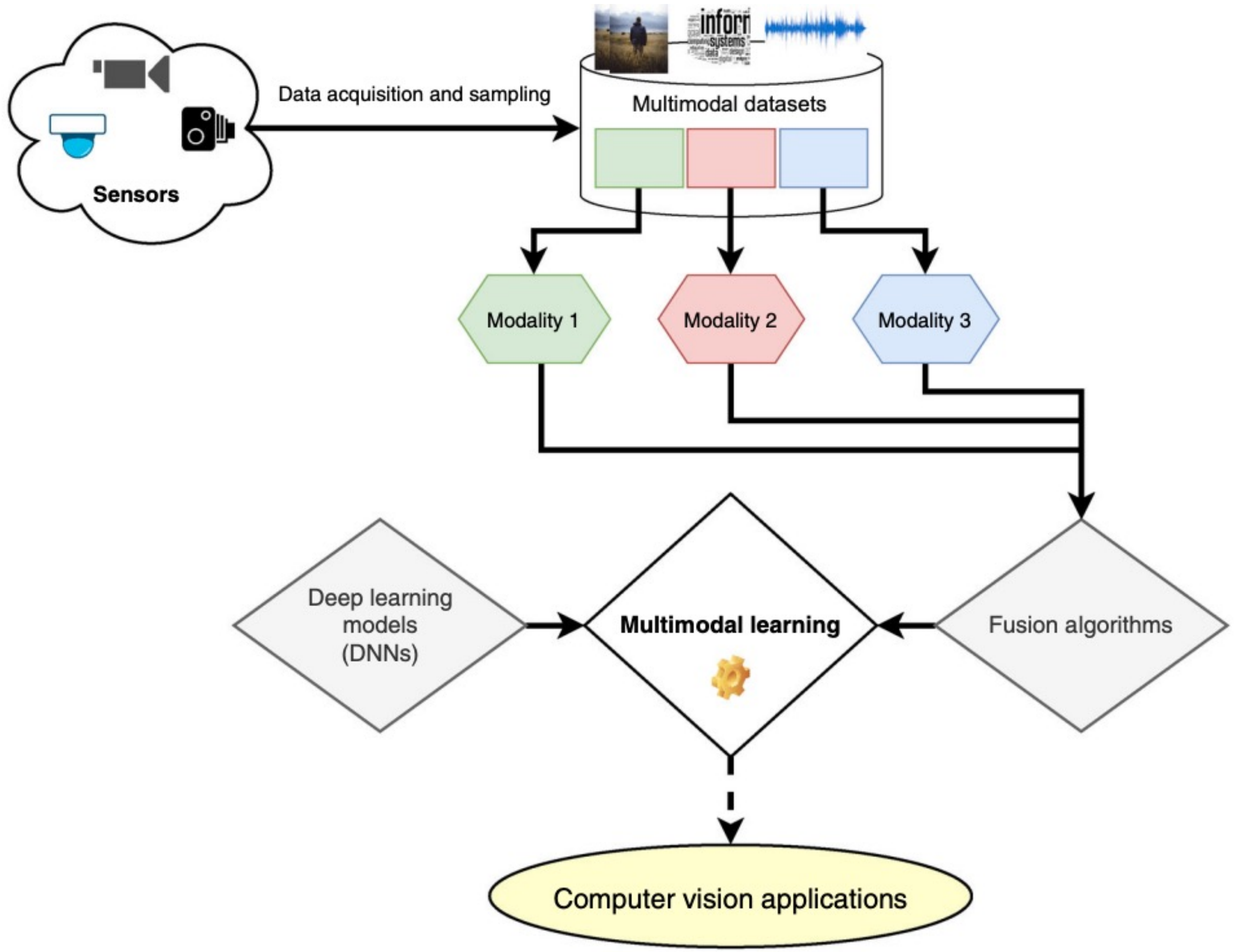
Speech, Text, NLP



Source: Razumovskaia, Evgeniia, Goran Glavas, Olga Majewska, Edoardo M. Ponti, Anna Korhonen, and Ivan Vulic.
"Crossing the conversational chasm: A primer on natural language processing for multilingual task-oriented dialogue systems." Journal of Artificial Intelligence Research 74 (2022): 1351-1402.

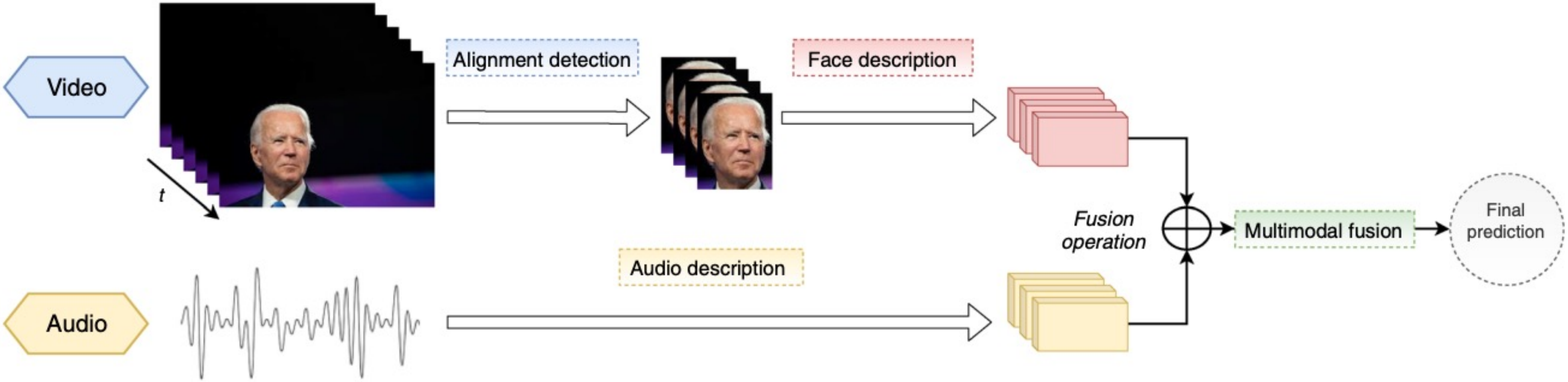
Multimodal Pipeline

that includes three different modalities (Image, Text. Audio)



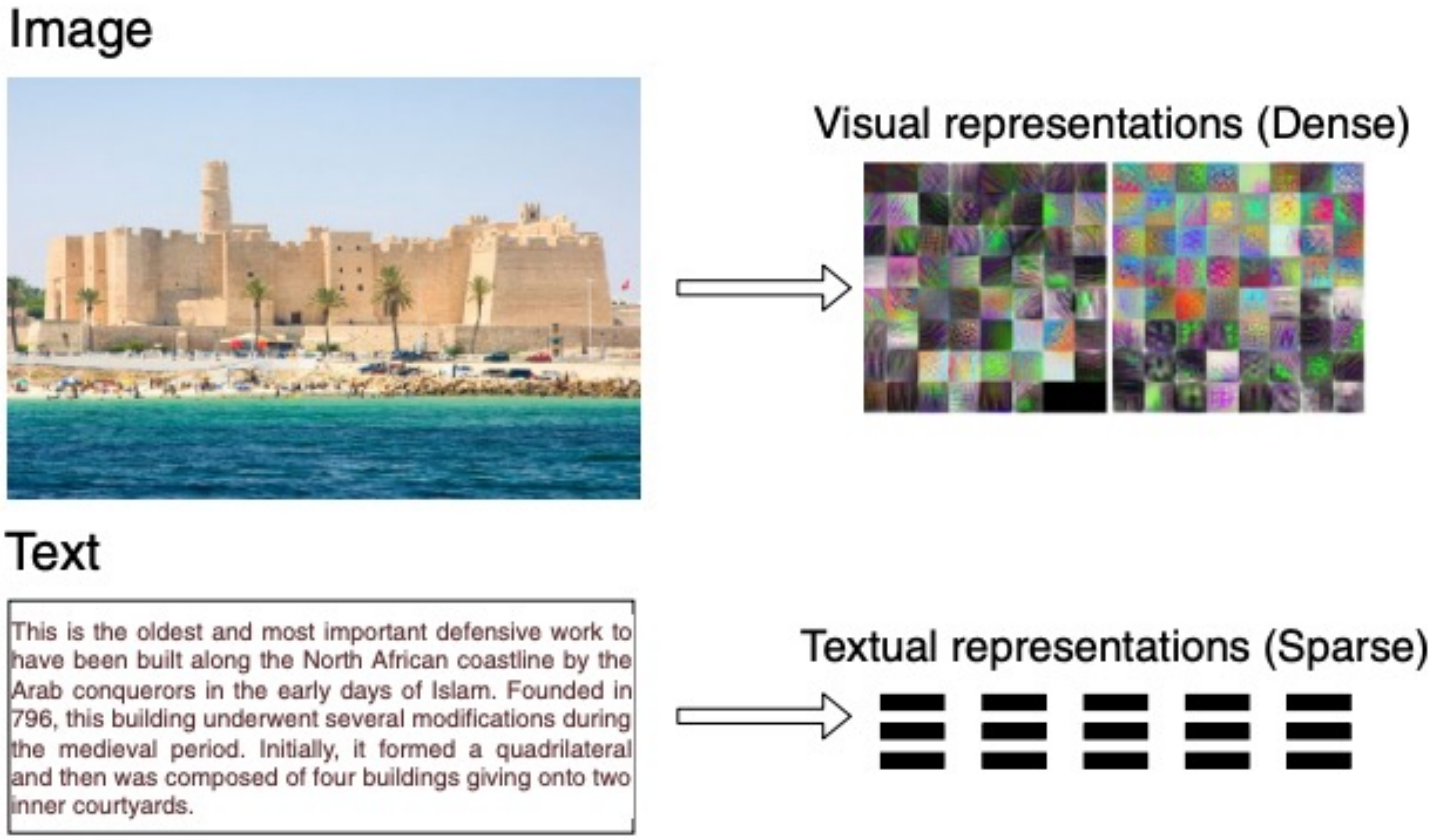
Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Video and Audio Multimodal Fusion



Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Visual and Textual Representation



Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Hybrid Multimodal Data Fusion

Text

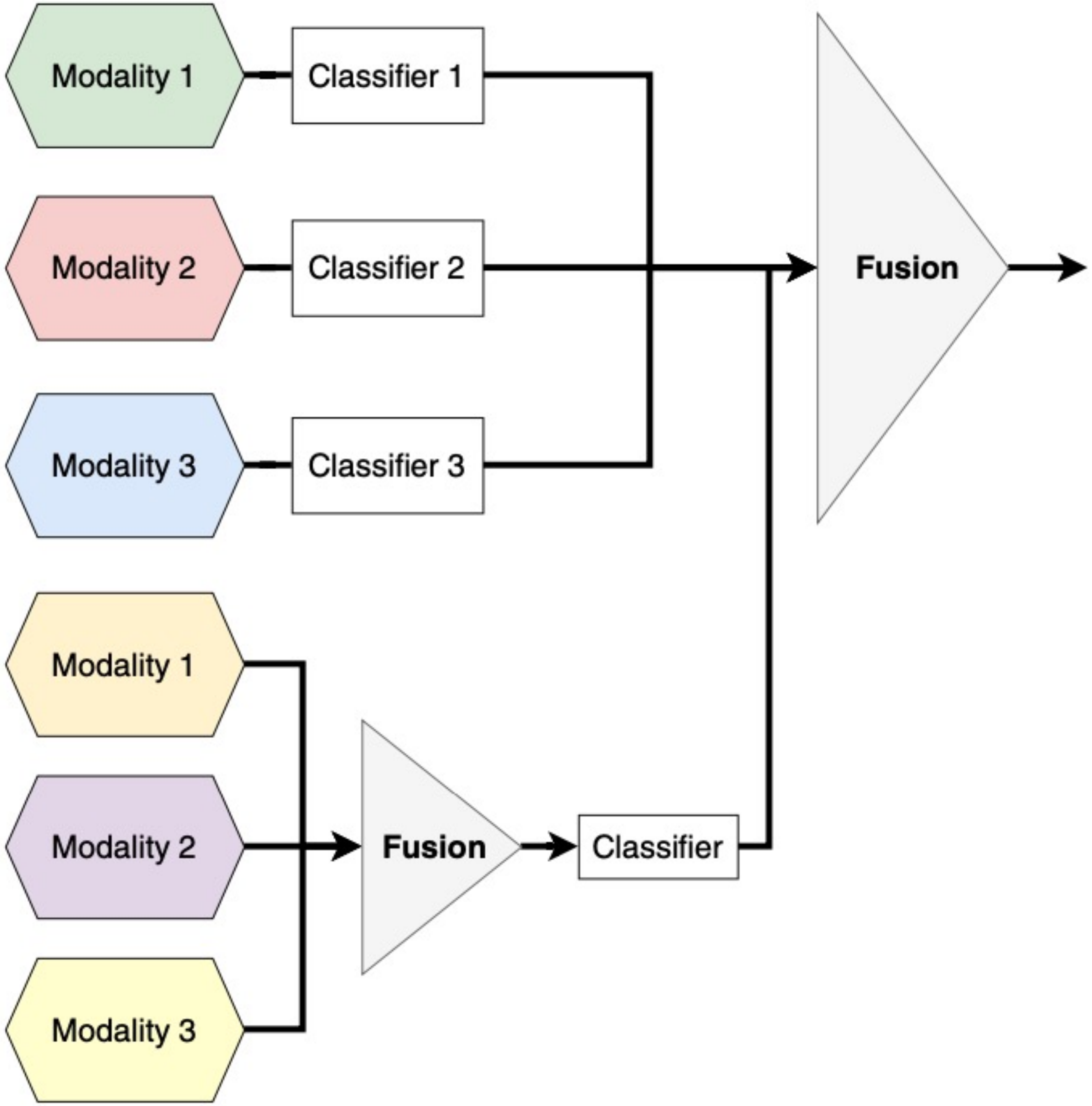
Audio

Image

Text

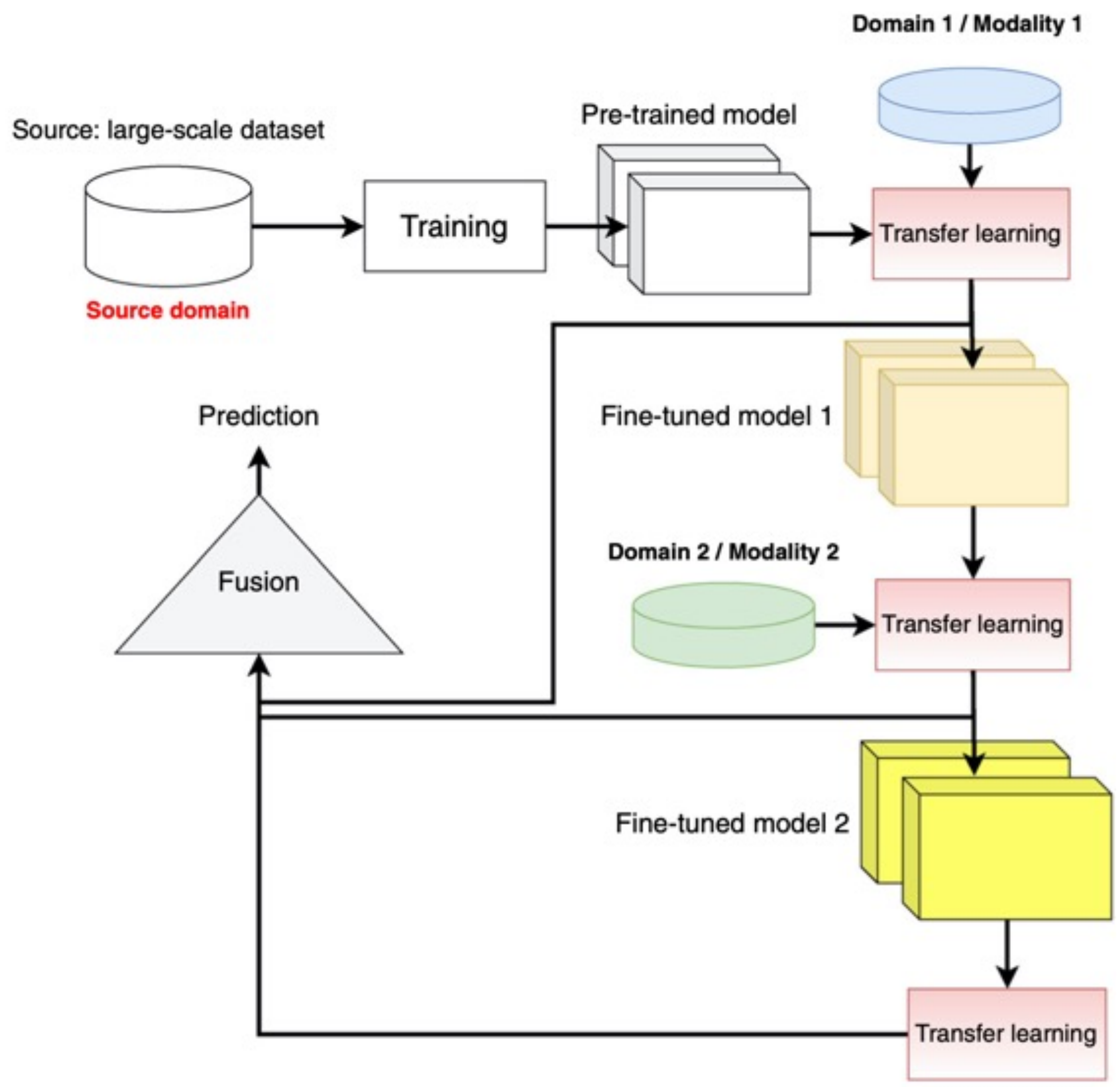
Speech

Video



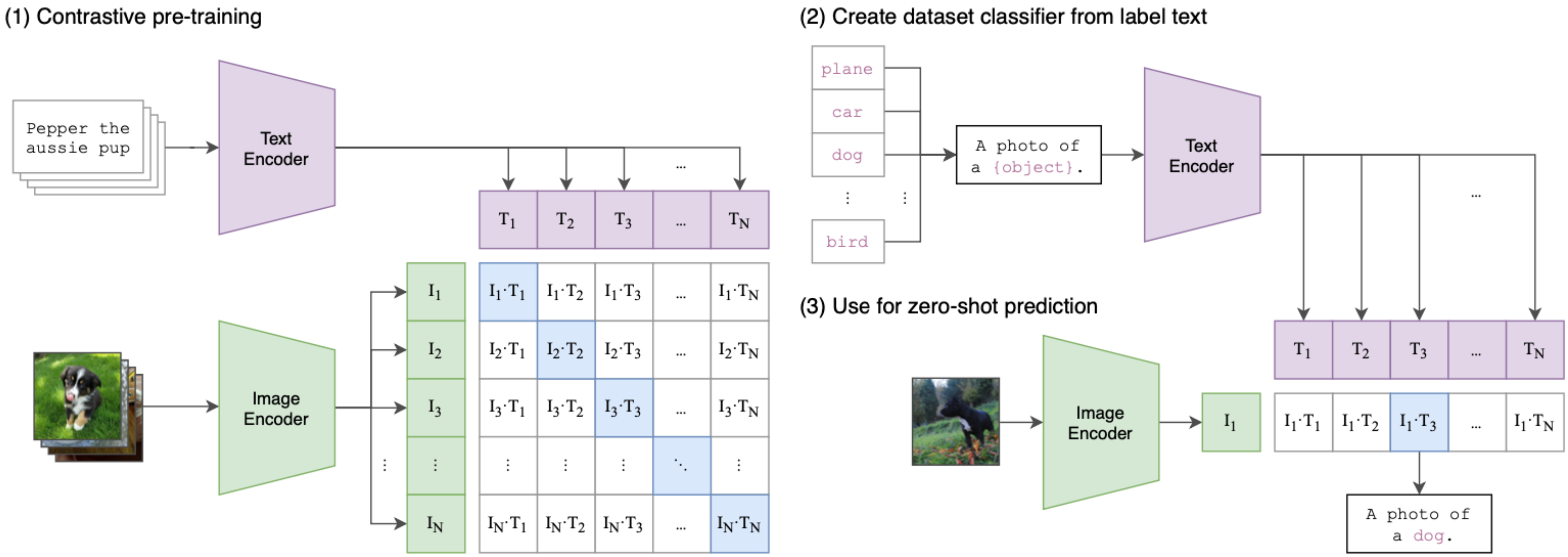
Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Multimodal Transfer Learning



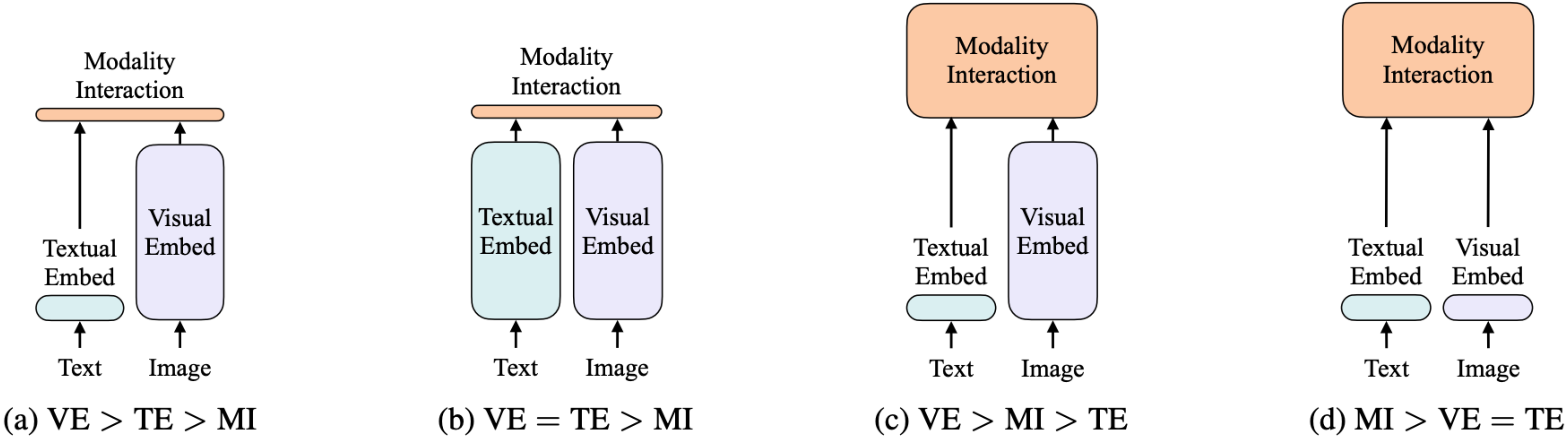
Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

CLIP: Learning Transferable Visual Models From Natural Language Supervision



Source: Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry et al. (2021) "Learning transferable visual models from natural language supervision." In International Conference on Machine Learning, pp. 8748-8763. PMLR.

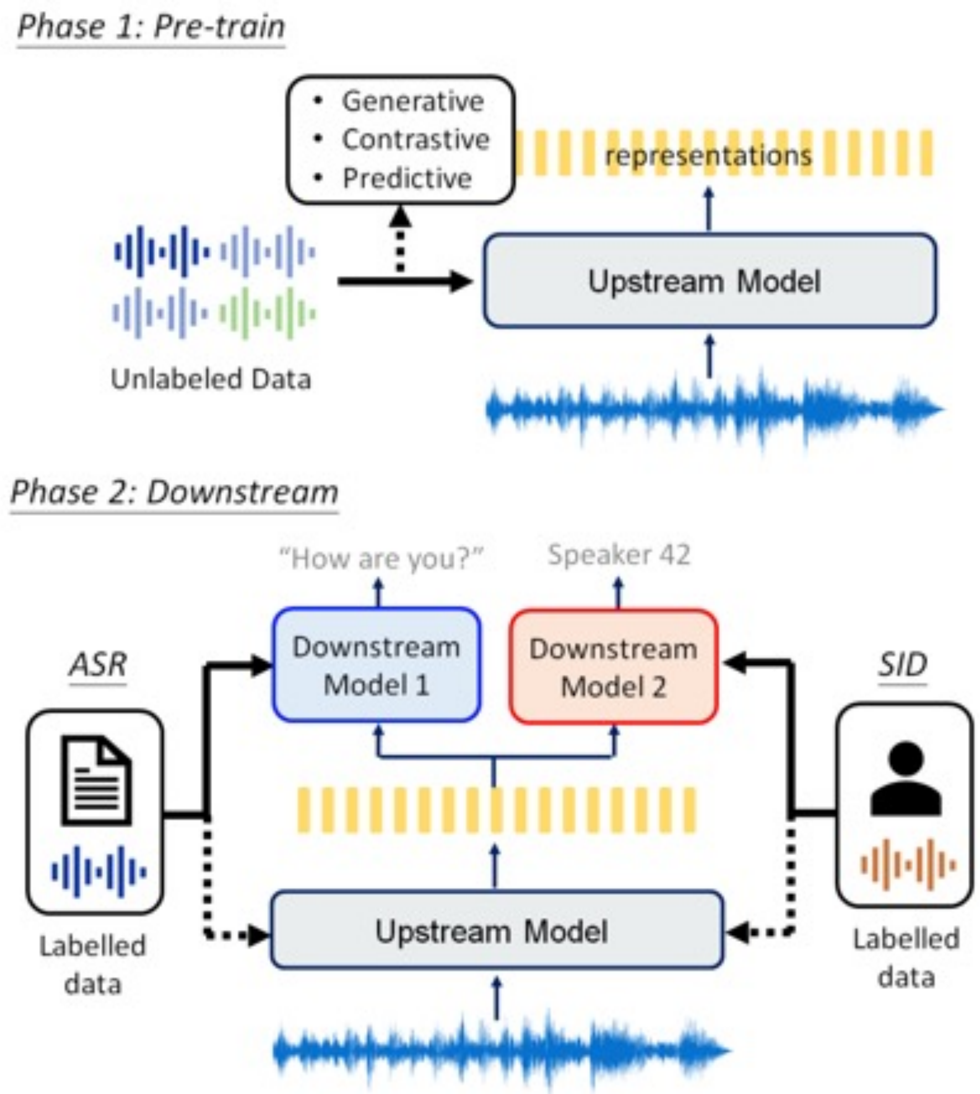
ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision



Source: Kim, Wonjae, Bokyung Son, and Ildoo Kim (2021). "Vilt: Vision-and-language transformer without convolution or region supervision." In International Conference on Machine Learning, pp. 5583-5594. PMLR.

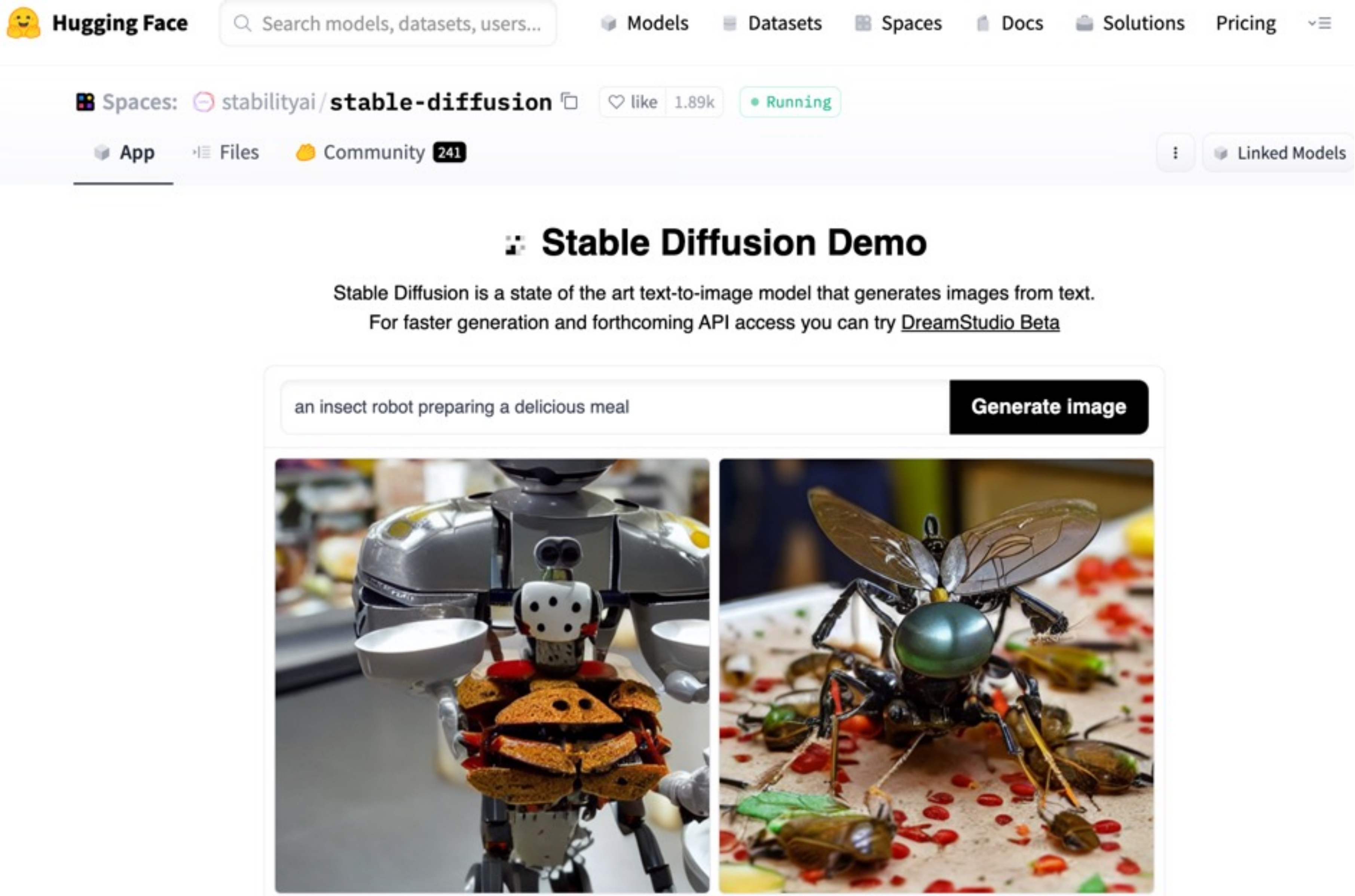
Self-Supervised Representation Learning in Speech Downstream Applications

Self-Supervised Learning (SSL)



Source: Mohamed, Abdelrahman, Hung-yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff et al. (2022) "Self-Supervised Speech Representation Learning: A Review." arXiv preprint arXiv:2205.10643.

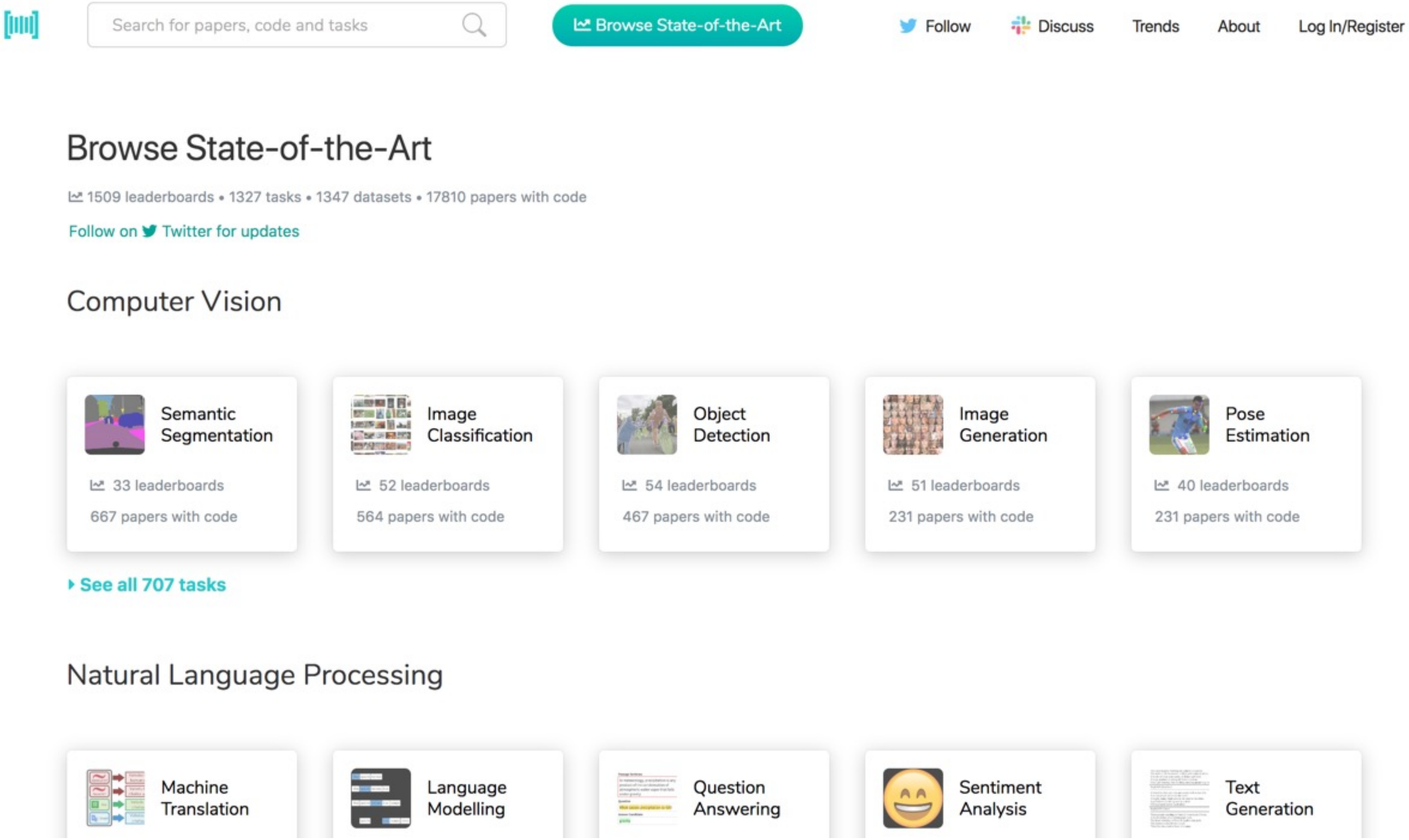
Stable Diffusion



https://huggingface.co/spaces/stabilityai/stable-diffusion

Papers with Code

State-of-the-Art (SOTA)



https://paperswithcode.com/sota

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb - Colaboratory

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=wsh36fLxDKC3

python101.ipynb

File Edit View Insert Runtime Tools Help

COMMENT SHARE

CODE TEXT CELL CELL CONNECTED EDITING

```python
# Future Value
pv = 100
r = 0.1
n = 7
fv = pv * ((1 + (r)) ** n)
print(round(fv, 2))
```

```
194.87
```

[11]

```python
amount = 100
interest = 10 #10% = 0.01 * 10
years = 7

future_value = amount * ((1 + (0.01 * interest)) ** years)
print(round(future_value, 2))
```

```
194.87
```

[12]

```python
# Python Function def
def getfv(pv, r, n):
    fv = pv * ((1 + (r)) ** n)
    return fv
fv = getfv(100, 0.1, 7)
print(round(fv, 2))
```

```
194.87
```

[13]

```python
# Python if else
score = 80
if score >=60 :
    print("Pass")
else:
    print("Fail")
```

```
Pass
```

https://tinyurl.com/aintpupython101

Summary

- **Solving Problems by Searching**
- **Search in Complex Environments**
- **Adversarial Search and Games**
- **Constraint Satisfaction Problems**

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O’Reilly Media.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Nithin Buduma, Nikhil Buduma, Joe Papa (2022), Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 2nd Edition, O'Reilly Media.