National Taipei University

Artificial Intelligence and Intelligent Agents; Problem Solving

1141AI02
MBA, IM, NTPU (M5276) (Fall 2025)
Tue 2, 3, 4 (9:10-12:00) (B3F17)

https://meet.google.com/paj-zhhj-mya

NVIDIA University Ambassador
Certified Instructor
aws educate Cloud Ambassador 2020 Cohort
aws academy Accredited Educator
aws certified Cloud Practitioner
aws certified Solutions Architect Associate

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

2025-09-16

Syllabus

Week Date Subject/Topics

1 2025/09/09 Introduction to Artificial Intelligence

2 2025/09/16 Artificial Intelligence and Intelligent Agents;
Problem Solving

3 2025/09/23 Knowledge, Reasoning and Knowledge Representation;
Uncertain Knowledge and Reasoning

4 2025/09/30 Case Study on Artificial Intelligence I

5 2025/10/07 Machine Learning: Supervised and Unsupervised Learning;
The Theory of Learning and Ensemble Learning

Syllabus

Week Date Subject/Topics

6 2025/10/14 NVIDIA Fundamentals of Deep Learning I:
Deep Learning; Neural Networks

7 2025/10/21 NVIDIA Fundamentals of Deep Learning II:
Convolutional Neural Networks;
Data Augmentation and Deployment

8 2025/10/28 Self-Learning

9 2025/11/04 Midterm Project Report

10 2025/11/11 NVIDIA Fundamentals of Deep Learning III:
Pre-trained Models; Natural Language Processing

Syllabus

Week Date Subject/Topics

11 2025/11/18 Case Study on Artificial Intelligence II

12 2025/11/25 Computer Vision and Robotics

13 2025/12/02 Generative AI, Agentic AI, and Physical AI

14 2025/12/09 Philosophy and Ethics of AI and the Future of AI

15 2025/12/16 Final Project Report I

16 2025/12/23 Final Project Report II

Artificial Intelligence
Intelligent Agents
Problem Solving

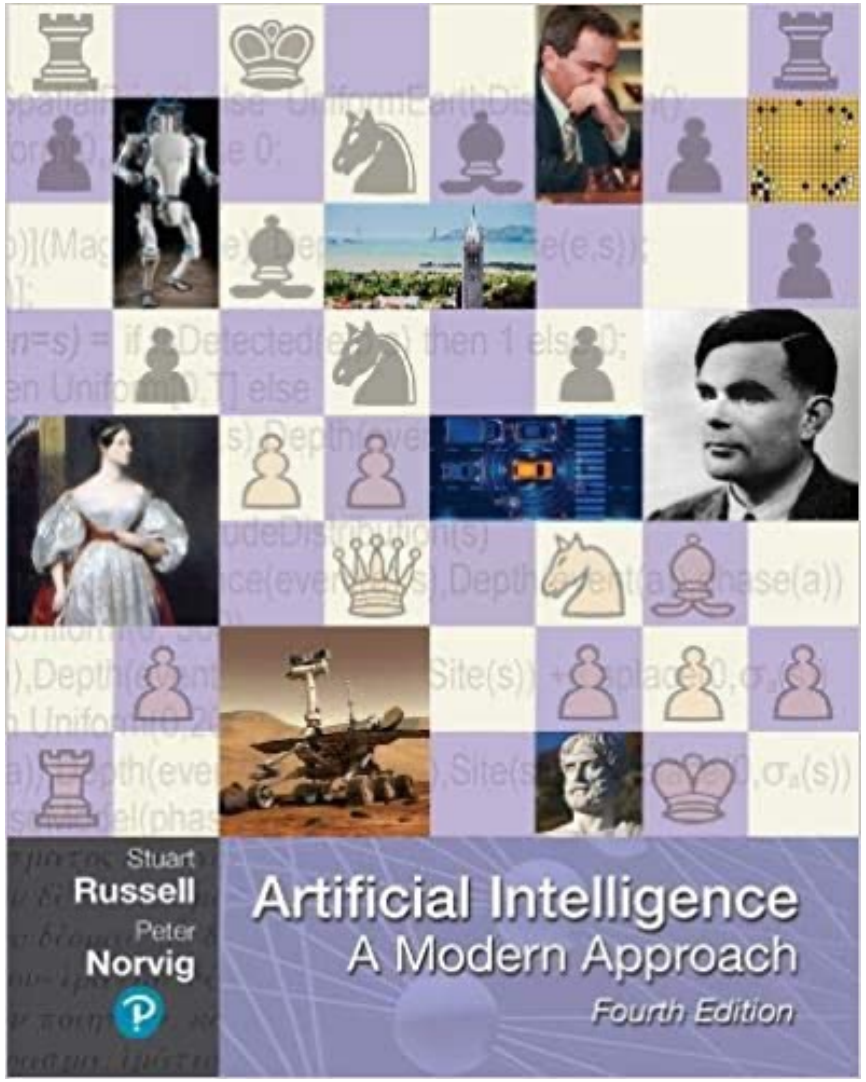
Outline

- **Artificial Intelligence**
- **Intelligent Agents**
- **Problem Solving**

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: Intelligent Agents

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

Artificial Intelligence: 3. Knowledge and Reasoning

- Logical Agents
- First-Order Logic
- Inference in First-Order Logic
- Knowledge Representation
- Automated Planning

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

Artificial Intelligence:
4. Uncertain Knowledge and Reasoning

- Quantifying Uncertainty
- Probabilistic Reasoning
- Probabilistic Reasoning over Time
- Probabilistic Programming
- Making Simple Decisions
- Making Complex Decisions
- Multiagent Decision Making

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

Artificial Intelligence:
5. Machine Learning

- **Learning from Examples**
- **Learning Probabilistic Models**
- **Deep Learning**
- **Reinforcement Learning**

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

Artificial Intelligence:

6. Communicating, Perceiving, and Acting

- Natural Language Processing
- Deep Learning for Natural Language Processing
- Computer Vision
- Robotics

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

Artificial Intelligence:

Philosophy and Ethics of AI
The Future of AI

NVIDIA Developer Program

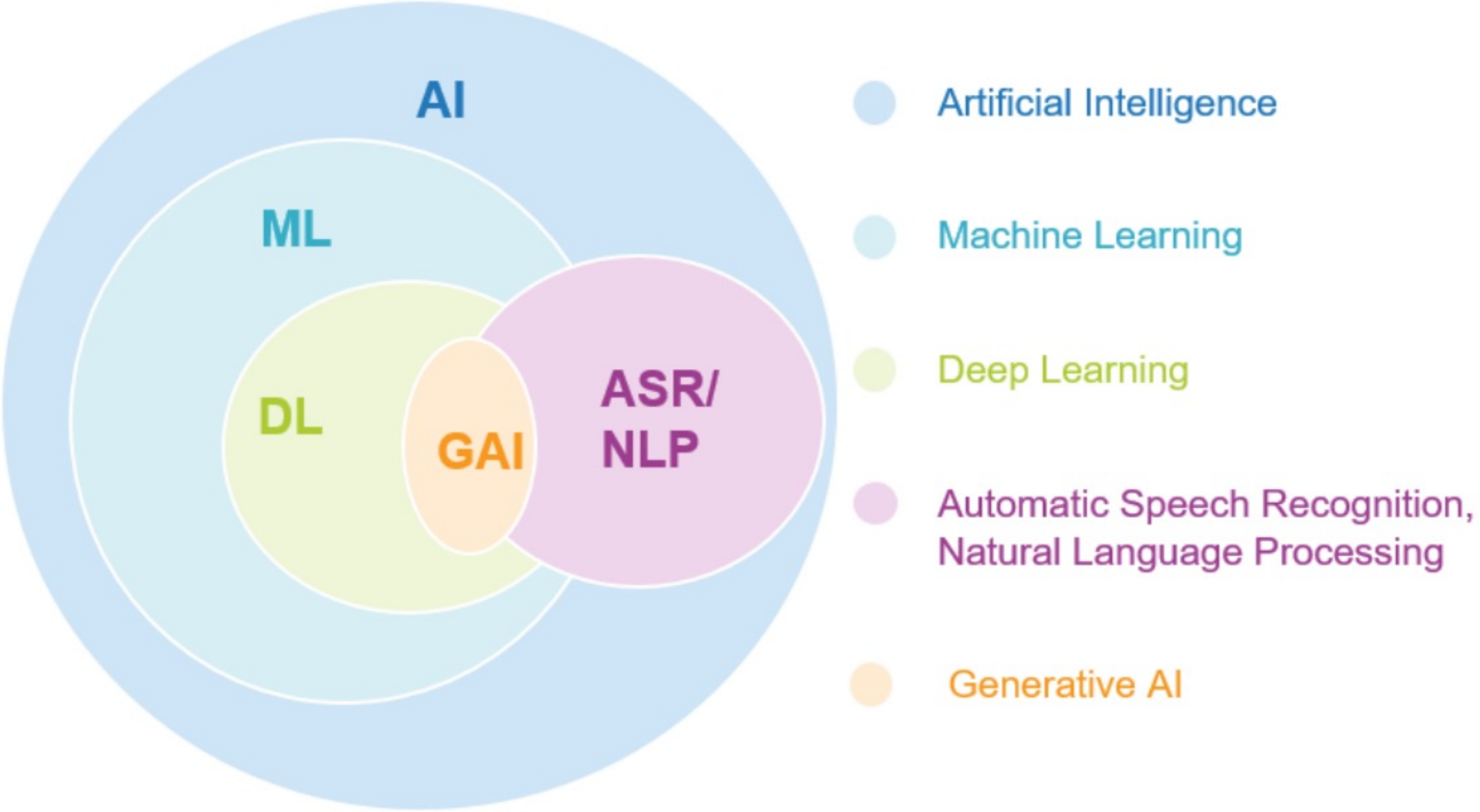
https://developer.nvidia.com/join-nvidia-developer-program

NVIDIA Deep Learning Institute (DLI)

https://learn.nvidia.com/

Artificial Intelligence (AI)

AI, ML, DL, Generative AI



Source: Jeong, Cheonsu. "A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture." arXiv preprint arXiv:2309.01105 (2023).

Generative AI, Agentic AI, Physical AI

Physical AI
Self-driving cars
General robotics

Agentic AI
Coding assistants
Customer service
Patient care

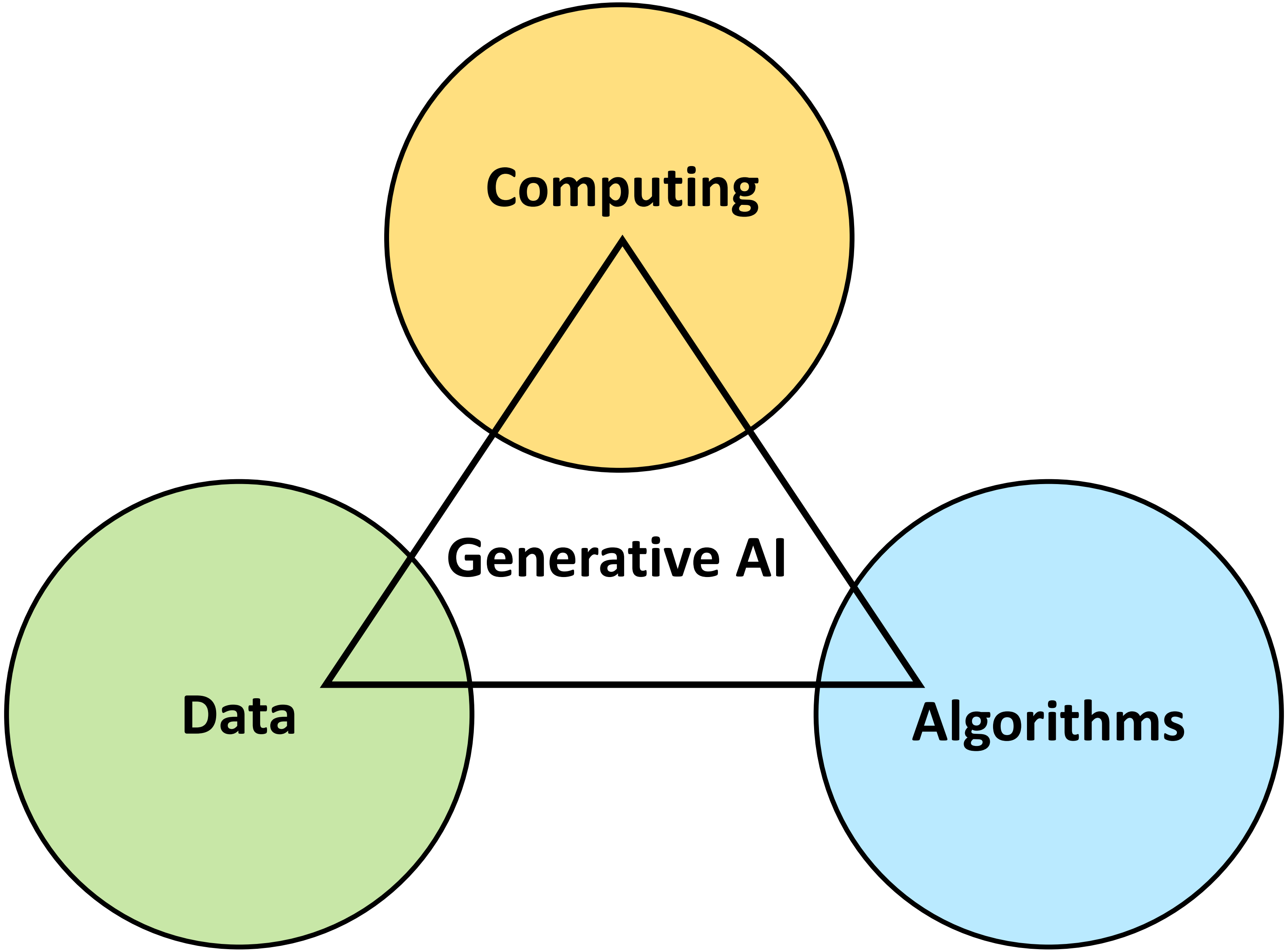
Generative AI
Digital marketing
Content creation

Perception AI
Speech recognition
Deep recommender systems
Medical imaging

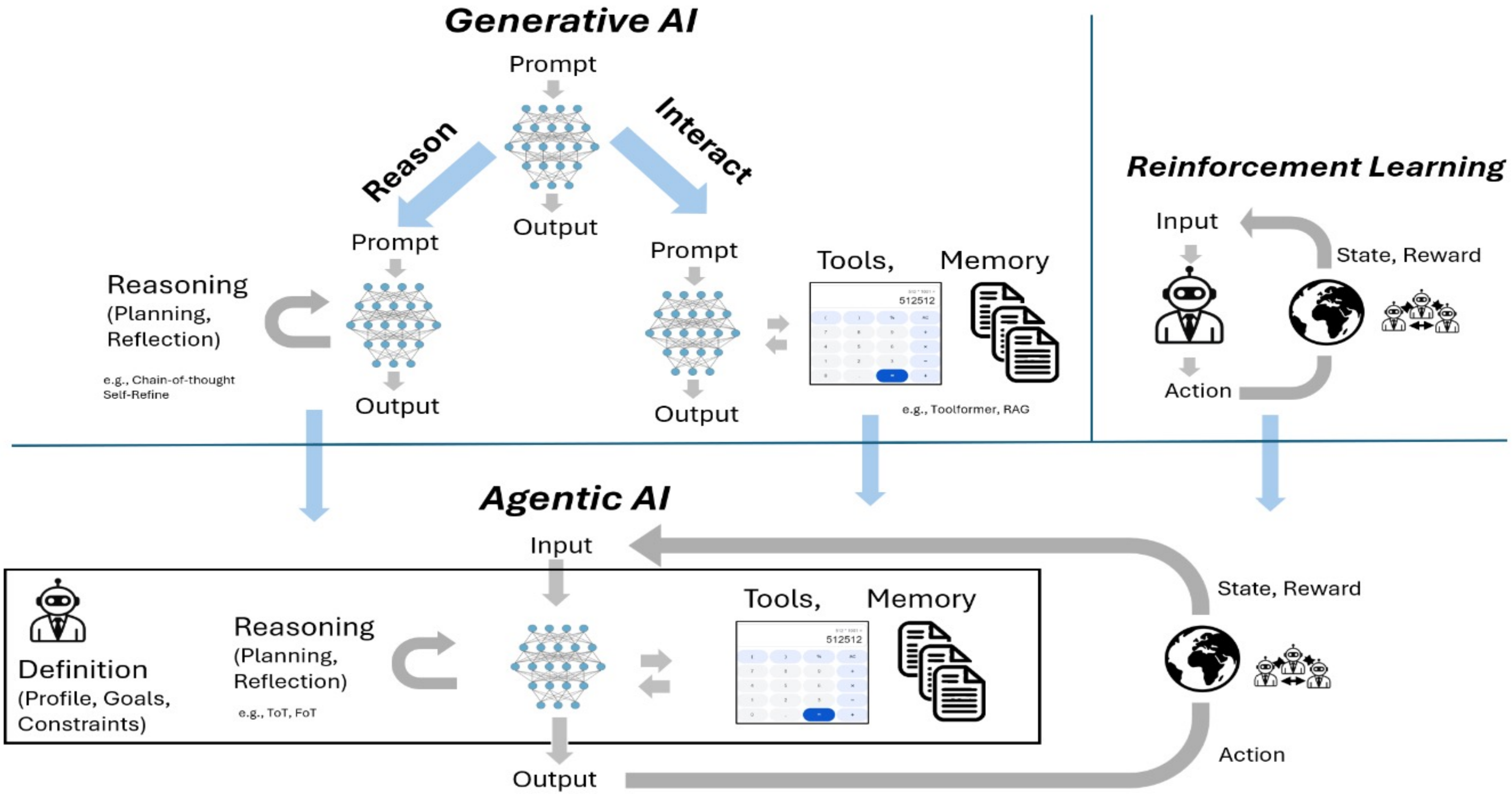
2012 AlexNet
Deep learning breakthrough

Source: NVIDIA (2025), GTC March 2025 Keynote with NVIDIA CEO Jensen Huang, https://www.youtube.com/watch?v=_waPvOwL9Z8

Generative AI



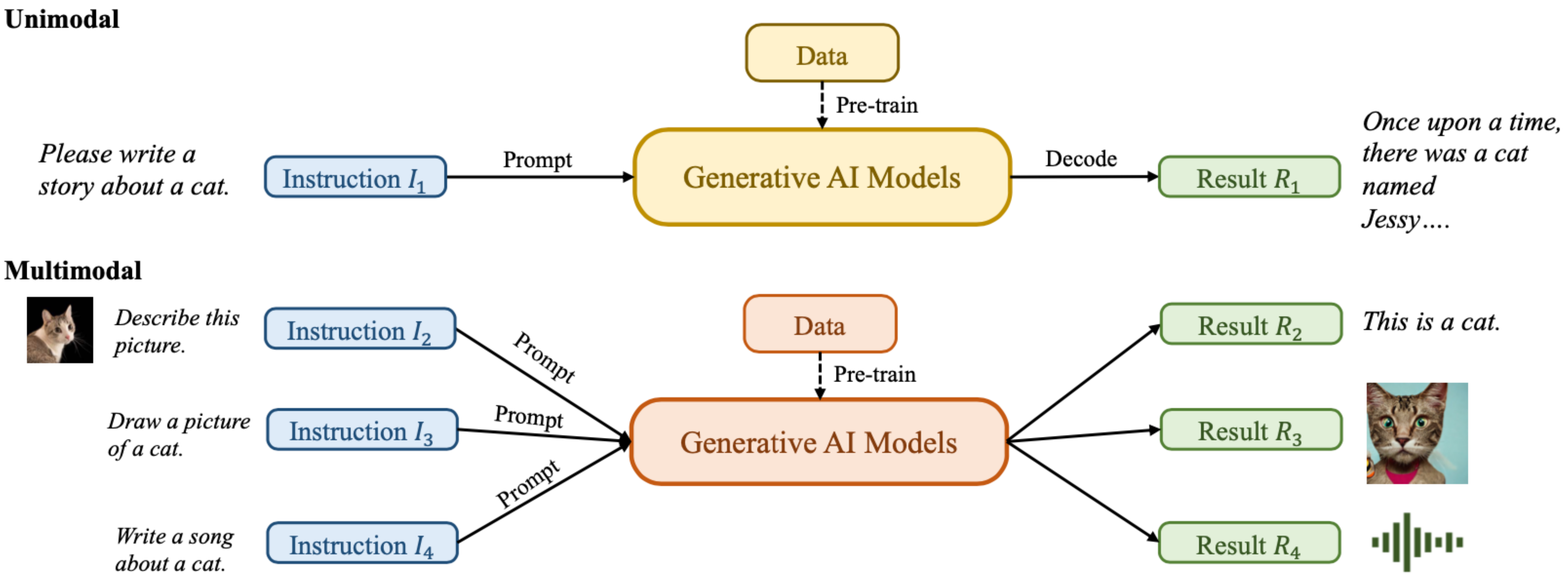
From Generative AI to Agentic AI



Source: Schneider, Johannes. "Generative to Agentic AI: Survey, Conceptualization, and Challenges." arXiv preprint arXiv:2504.18875 (2025).

Generative AI (Gen AI)

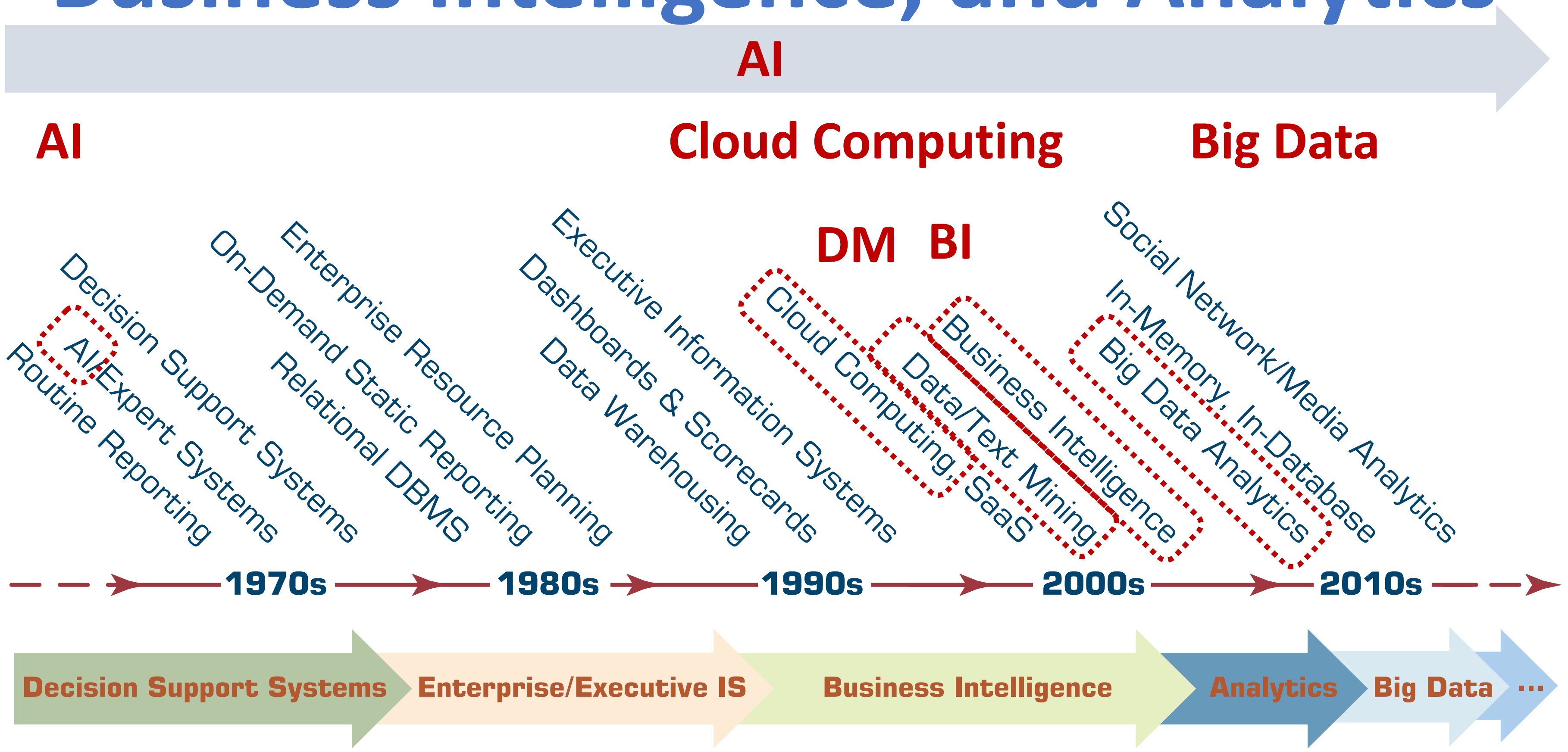
AI Generated Content (AIGC)



Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

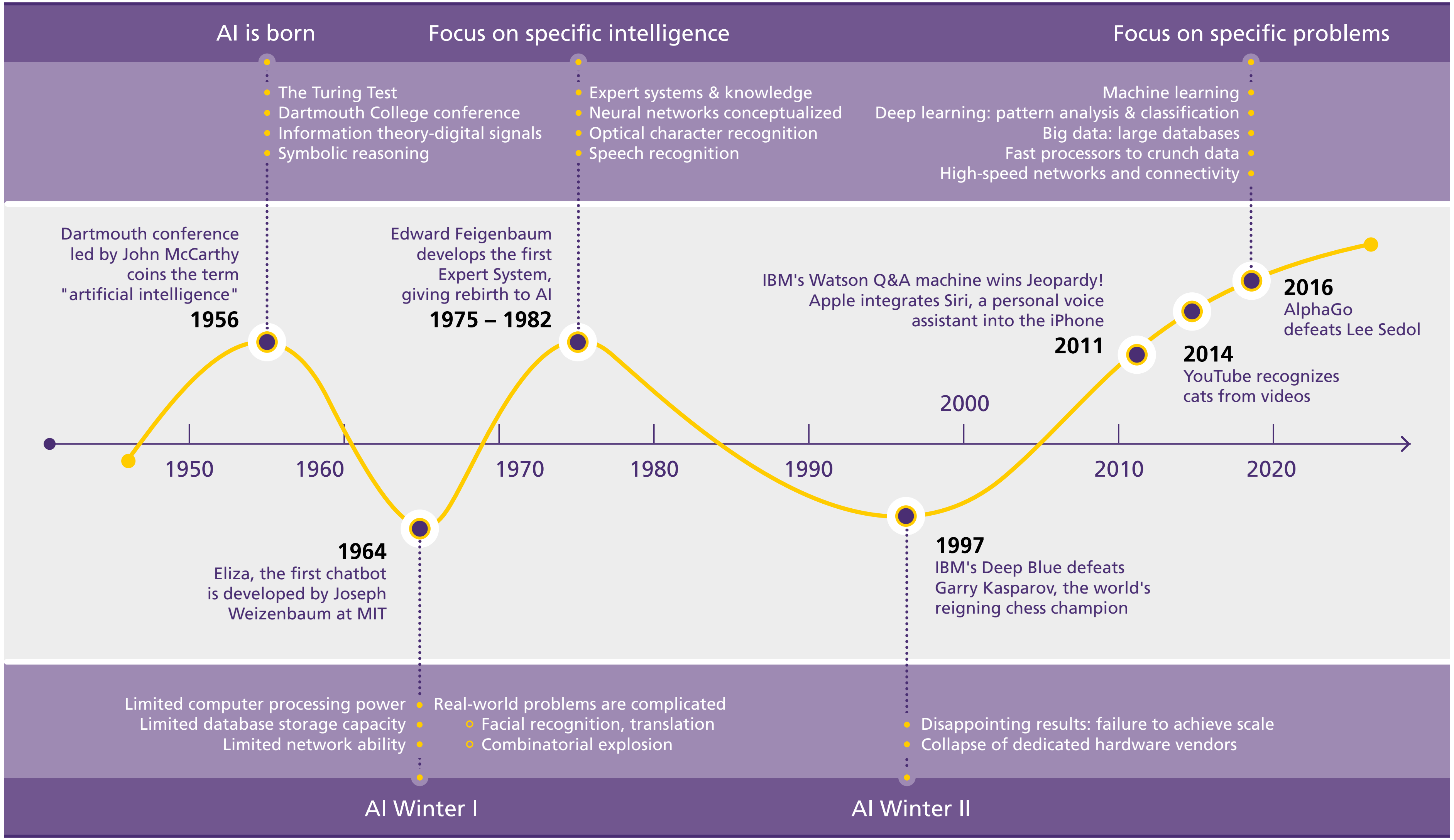
AI, Big Data, Cloud Computing

Evolution of Decision Support, Business Intelligence, and Analytics



Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

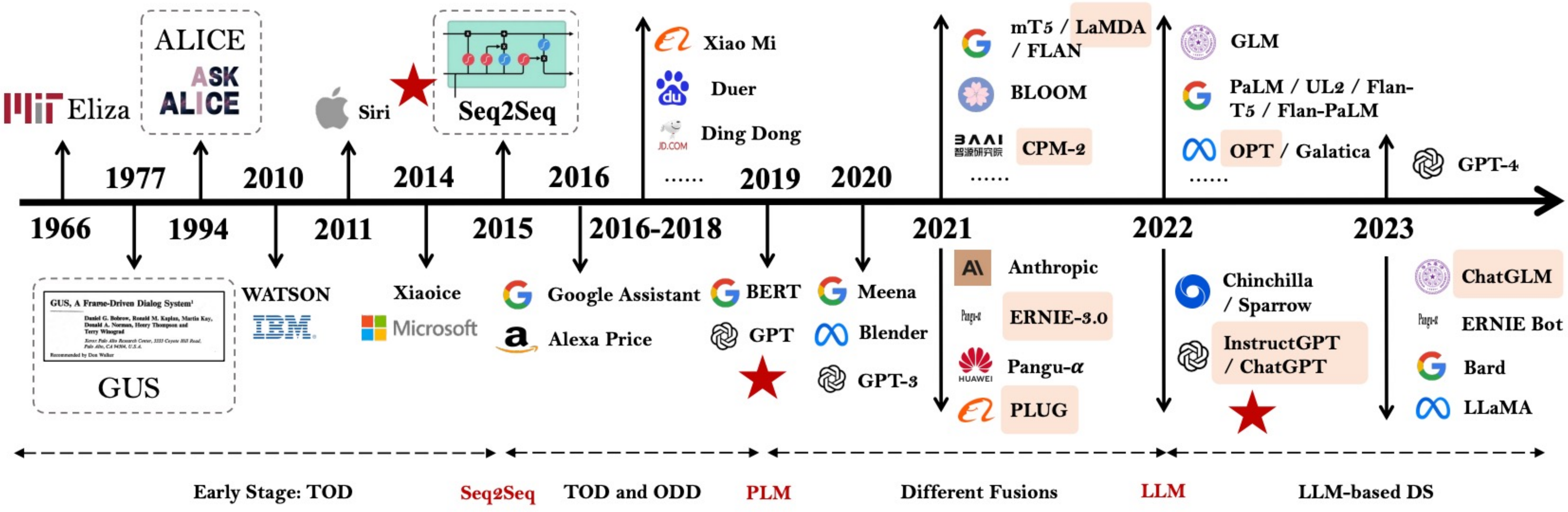
The Rise of AI



Source: DHL (2018), Artificial Intelligence in Logistics, http://www.globalhha.com/doclib/data/upload/doc_con/5e50c53c5bf67.pdf/

The Development of LM-based Dialogue Systems

1) Early Stage (1966 - 2015)
2) The Independent Development of TOD and ODD (2015 - 2019)
3) Fusions of Dialogue Systems (2019 - 2022)
4) LLM-based DS (2022 - Now)



Task-oriented DS (TOD), Open-domain DS (ODD)

Source: Wang, Hongru, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. "A Survey of the Evolution of Language Model-Based Dialogue Systems." arXiv preprint arXiv:2311.16789 (2023).

Definition
of
Artificial Intelligence
(A.I.)

Artificial Intelligence

"… the science and
engineering
of
making
intelligent machines"

(John McCarthy, 1955)

Source: https://digitalintelligencetoday.com/artificial-intelligence-defined-useful-list-of-popular-definitions-from-business-and-science/

Artificial Intelligence

“… technology that thinks and acts like humans”

Source: https://digitalintelligencetoday.com/artificial-intelligence-defined-useful-list-of-popular-definitions-from-business-and-science/

Artificial Intelligence

"… intelligence exhibited by machines or software"

Source: https://digitalintelligencetoday.com/artificial-intelligence-defined-useful-list-of-popular-definitions-from-business-and-science/

4 Approaches of AI

Thinking Humanly	Thinking Rationally
Acting Humanly	Acting Rationally

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

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

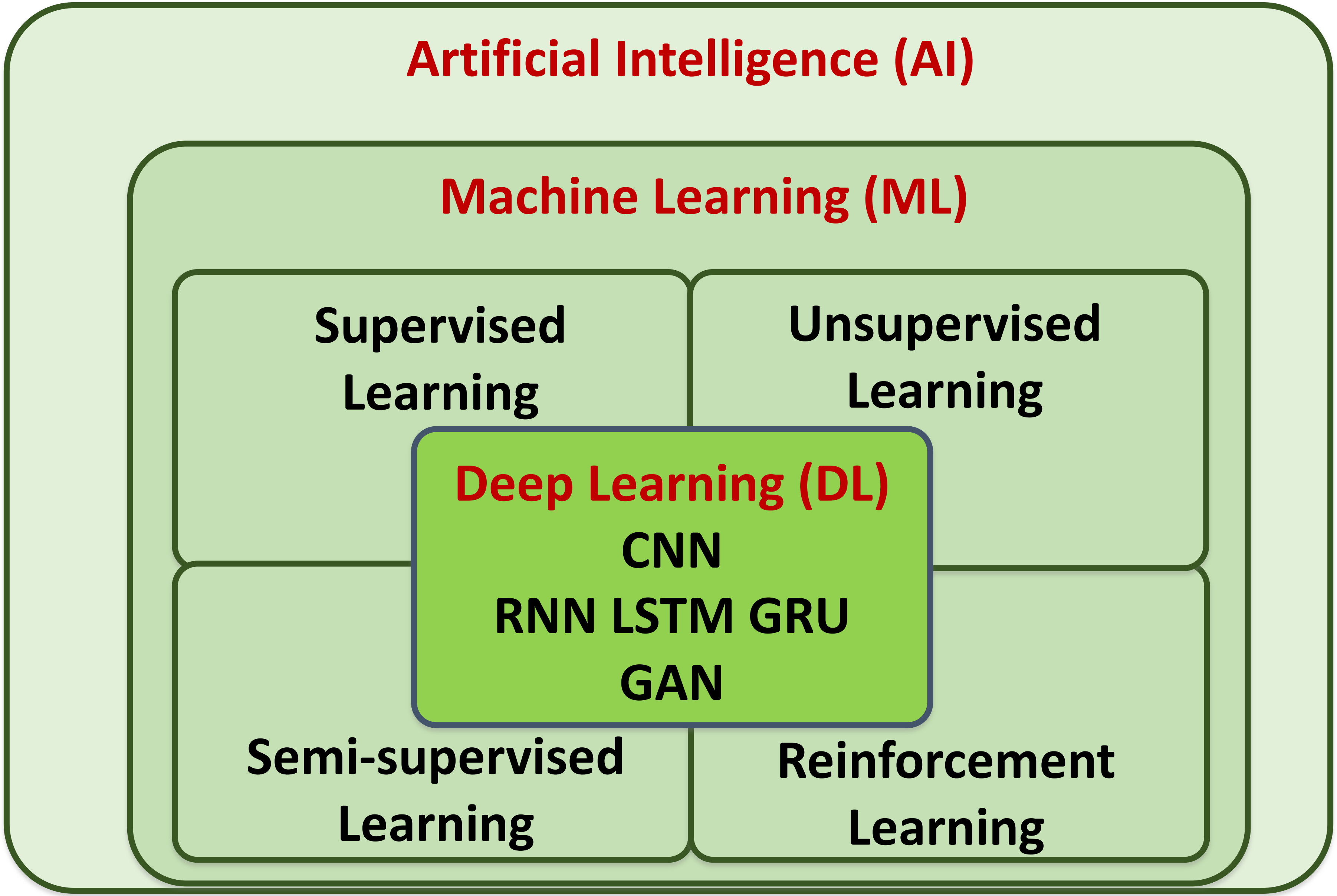
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

AI Acting Humanly: The Turing Test Approach
(Alan Turing, 1950)

- Knowledge Representation
- Automated Reasoning
- Machine Learning (ML)
  - Deep Learning (DL)
- Computer Vision (Image, Video)
- Natural Language Processing (NLP)
- Robotics

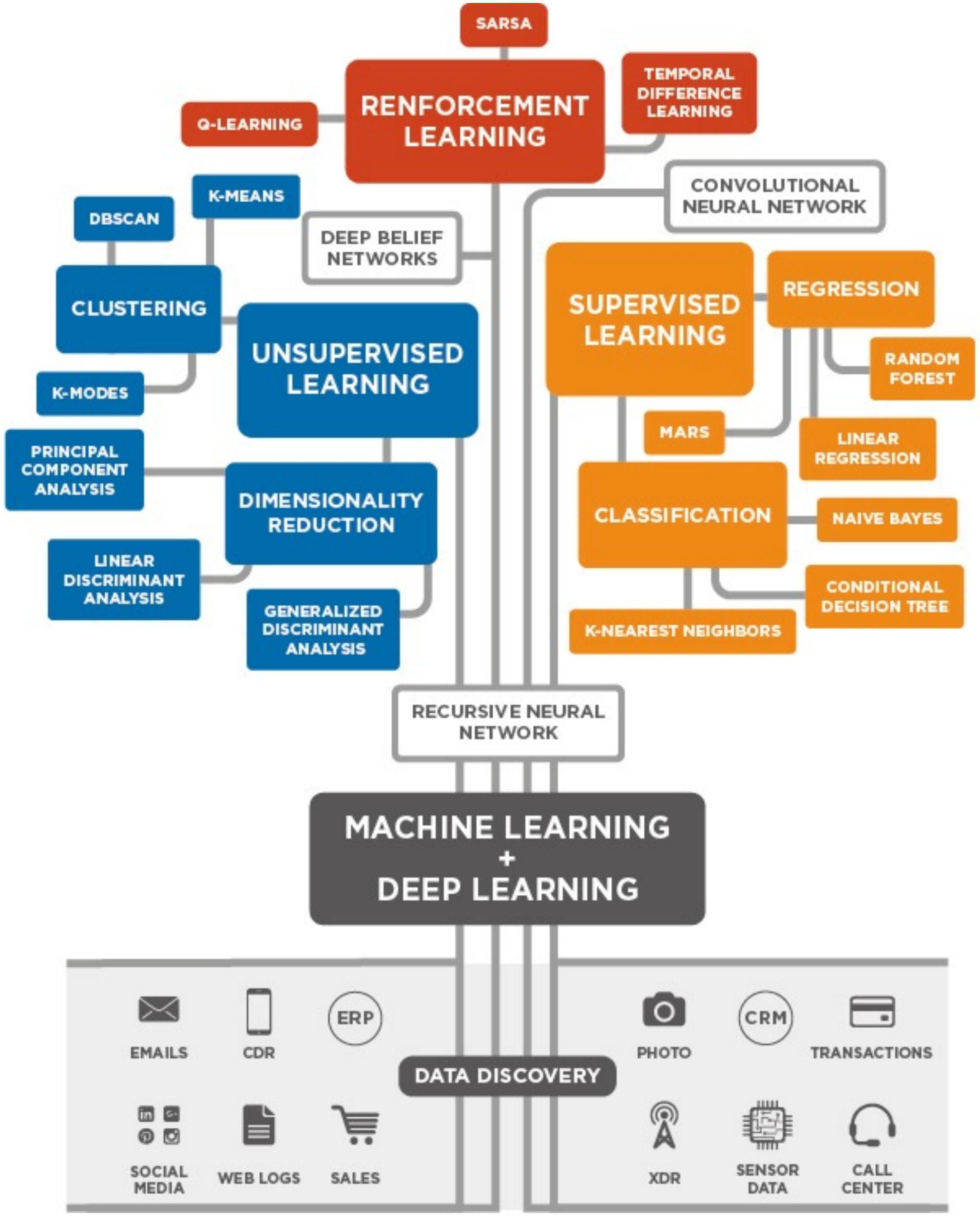
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

AI, ML, DL



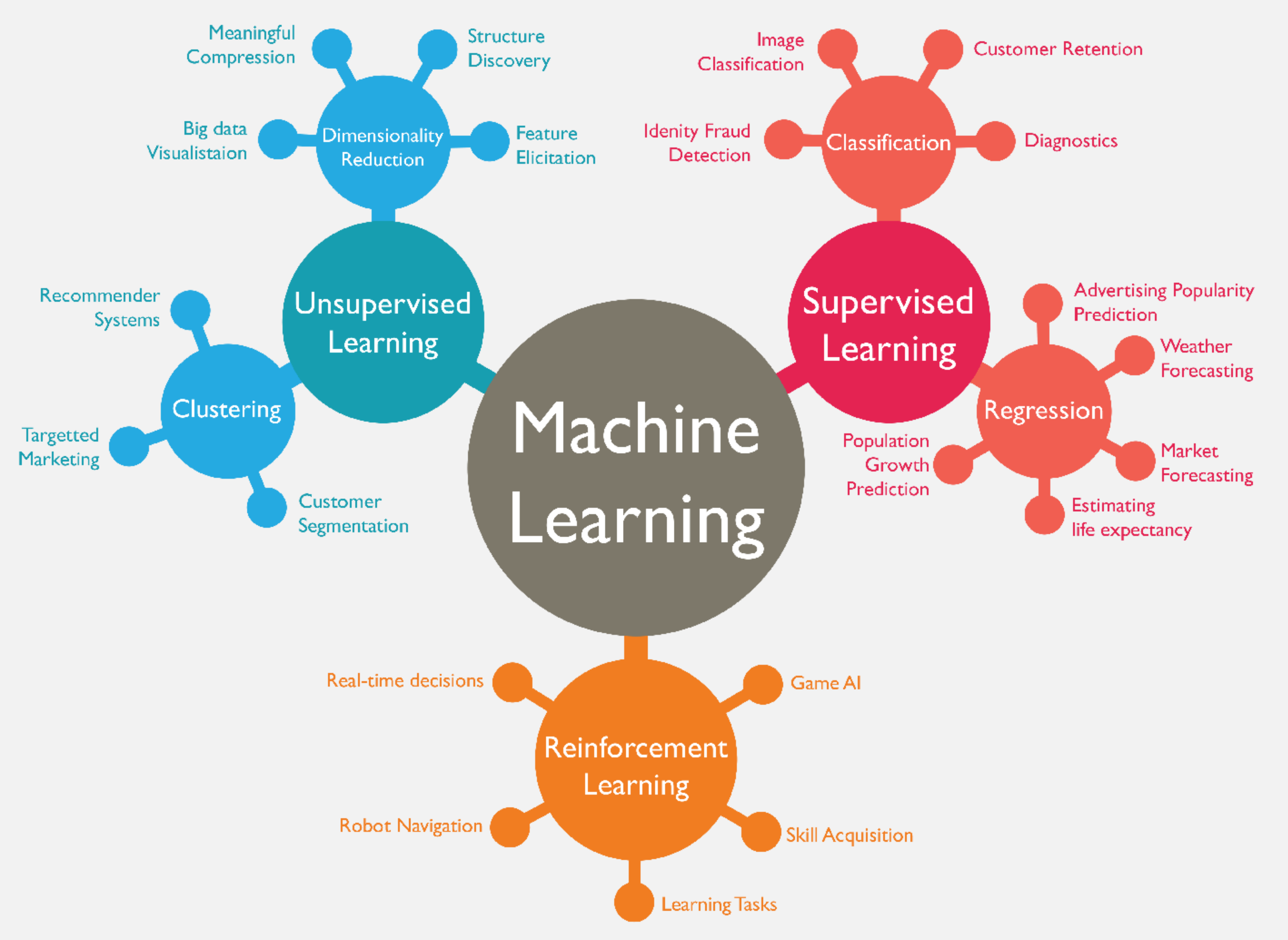
Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/deep_learning.html

3 Machine Learning Algorithms



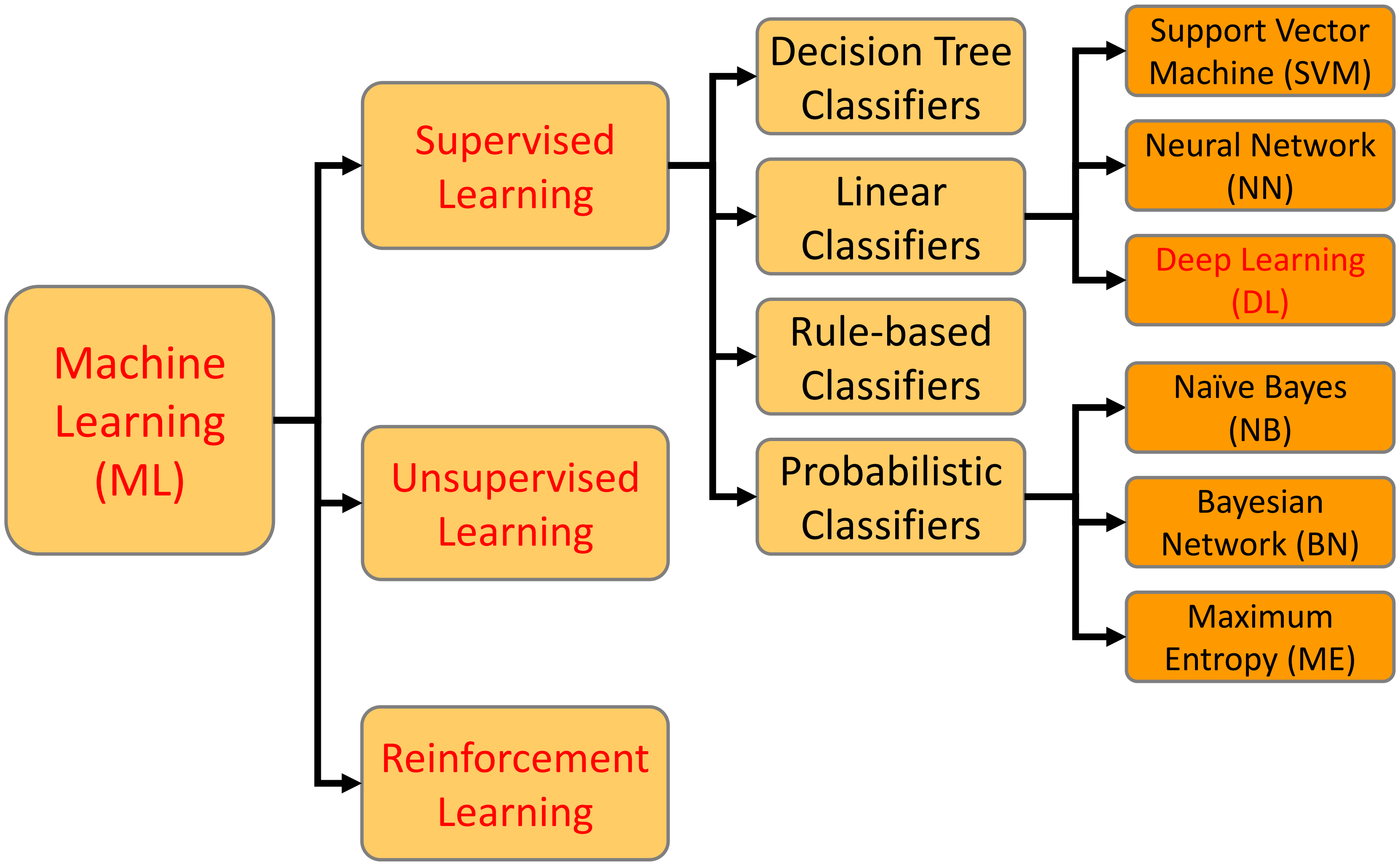
Source: Enrico Galimberti, http://blogs.teradata.com/data-points/tree-machine-learning-algorithms/

Machine Learning (ML)



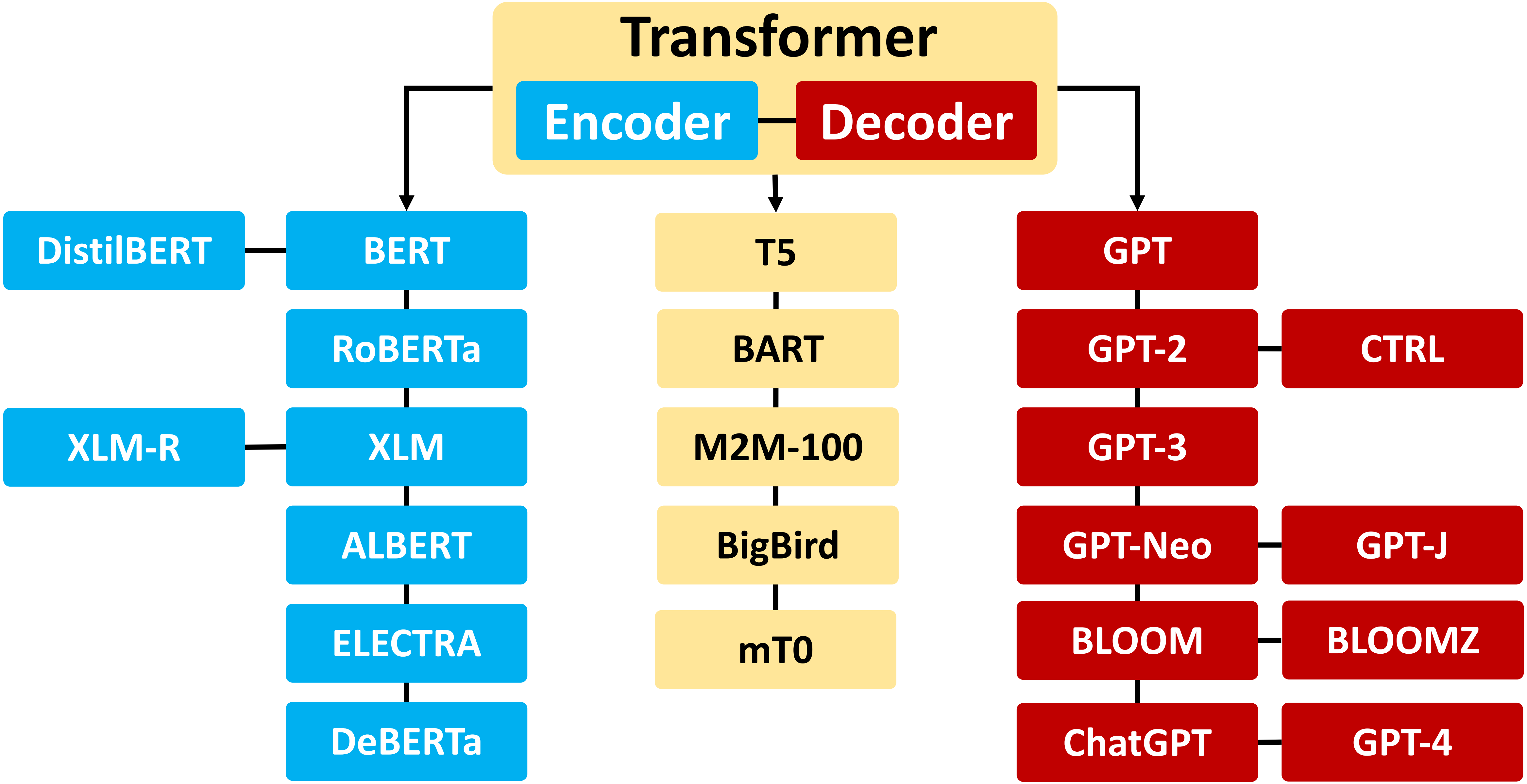
Source: https://www.mactores.com/services/aws-big-data-machine-learning-cognitive-services/

Machine Learning (ML) / Deep Learning (DL)



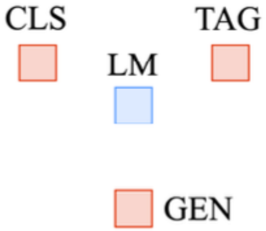
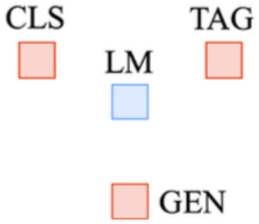
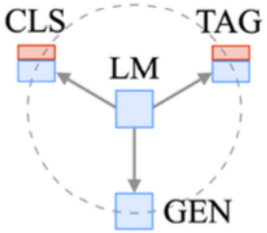
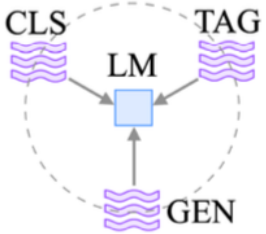
Source: Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma (2015), "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, 311, pp. 18-38.

Transformer Models



Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Four Paradigms in NLP (LM)

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Feature (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
Transfer Learning: Pre-training, Fine-Tuning (FT) c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
GAI: Pre-train, Prompt, and Predict (Prompting) d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Generative AI

Text, Image, Video, Audio Applications

Comparison of Generative AI and Traditional AI

Feature	**Generative AI**	**Traditional AI**
Output type	New content	Classification/Prediction
Creativity	High	Low
Interactivity	Usually more natural	Limited

Generative AI

- **Generative AI: The Art of Creation**
- **Definition: AI systems capable of creating new content**
- **Characteristics: Creativity, interactivity**

LMArena Leaderboard

Rank (UB)	Model	Score	95% CI (±)	Votes	Organization	License
1	gemini-2.5-pro	1455	±5	41,731	Google	Proprietary
1	claude-opus-4-1-20250805-thinking-16k	1451	±6	11,750	Anthropic	Proprietary
2	o3-2025-04-16	1444	±4	43,898	OpenAI	Proprietary
2	gpt-5-high	1442	±6	15,076	OpenAI	Proprietary
2	chatgpt-4o-latest-20250326	1441	±4	36,426	OpenAI	Proprietary
3	gpt-4.5-preview-2025-02-27	1439	±6	15,271	OpenAI	Proprietary
3	claude-opus-4-1-20250805	1438	±6	18,341	Anthropic	Proprietary
5	gpt-5-chat	1430	±6	11,808	OpenAI	Proprietary
6	qwen3-max-preview	1428	±7	8,781	Alibaba	Proprietary
8	grok-4-0709	1422	±5	21,446	xAI	Proprietary

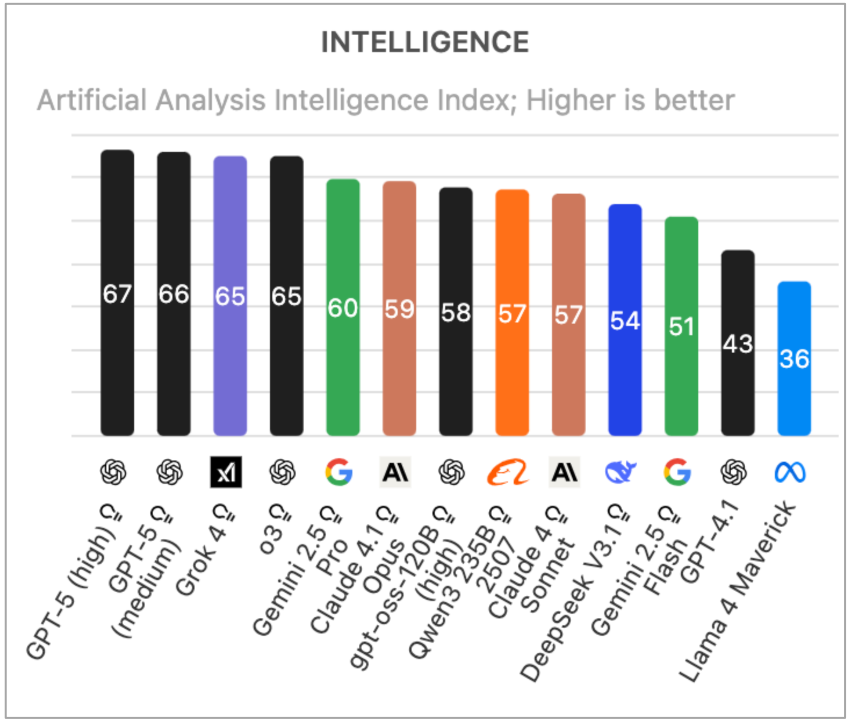
https://lmarena.ai/

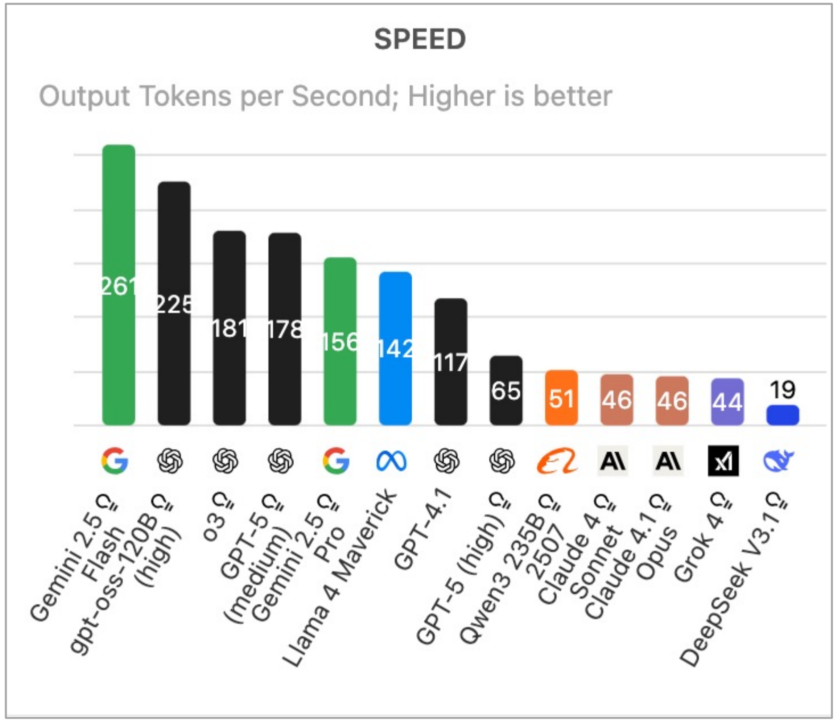
LMArena Leaderboard

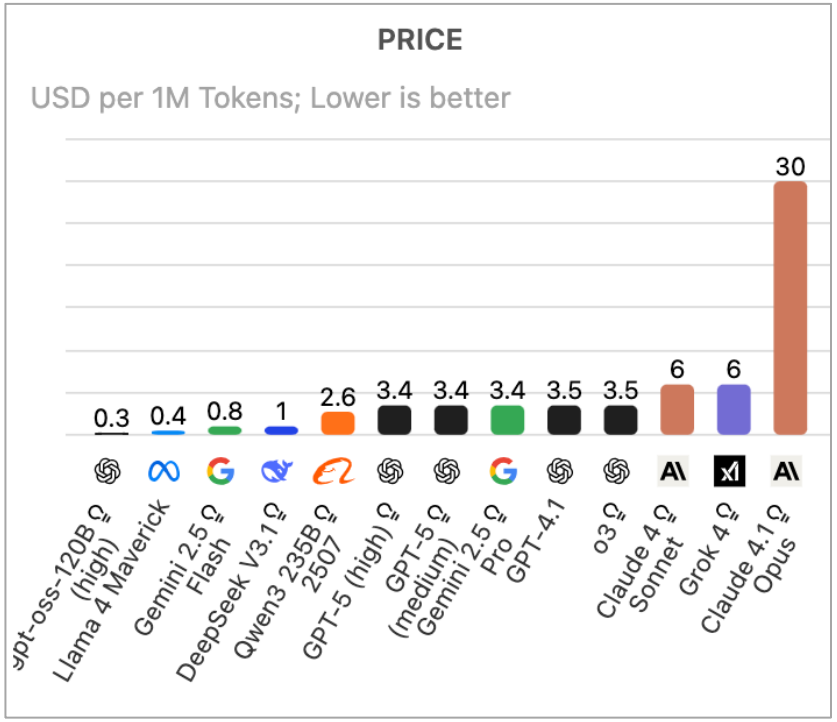
Model 239 / 239	Overall	Hard Prompts	Coding	Math	Creative Writing	Instruction Following	Longer Query	Multi-Turn
claude-opus-4-1-…	1	1	1	1	1	1	1	1
gemini-2.5-pro	1	2	3	1	1	1	1	1
chatgpt-4o-lates…	2	4	3	13	2	5	4	1
gpt-5-high	2	2	3	1	7	5	11	6
o3-2025-04-16	2	4	3	1	8	6	13	7
claude-opus-4-1-…	3	2	1	1	1	1	1	1
gpt-4.5-preview-…	3	5	4	8	1	4	3	1
gpt-5-chat	5	3	3	8	3	5	3	1
qwen3-max-preview	6	4	2	1	7	4	4	3
claude-opus-4-20…	8	4	3	6	2	2	2	7
deepseek-r1-0528	8	8	4	10	8	15	13	14
deepseek-v3.1	8	6	4	1	7	6	5	9
deepseek-v3.1-th…	8	4	3	1	2	4	1	7
grok-4-0709	8	10	12	1	4	6	8	7
kimi-k2-0711-pre…	8	10	7	13	16	24	22	7
kimi-k2-0905-pre…	8	5	3	-	6	16	12	7
qwen3-235b-a22b-…	8	4	3	2	9	6	4	7
glm-4.5	10	7	4	7	14	7	8	10

https://huggingface.co/spaces/lmarena-ai/lmarena-leaderboard

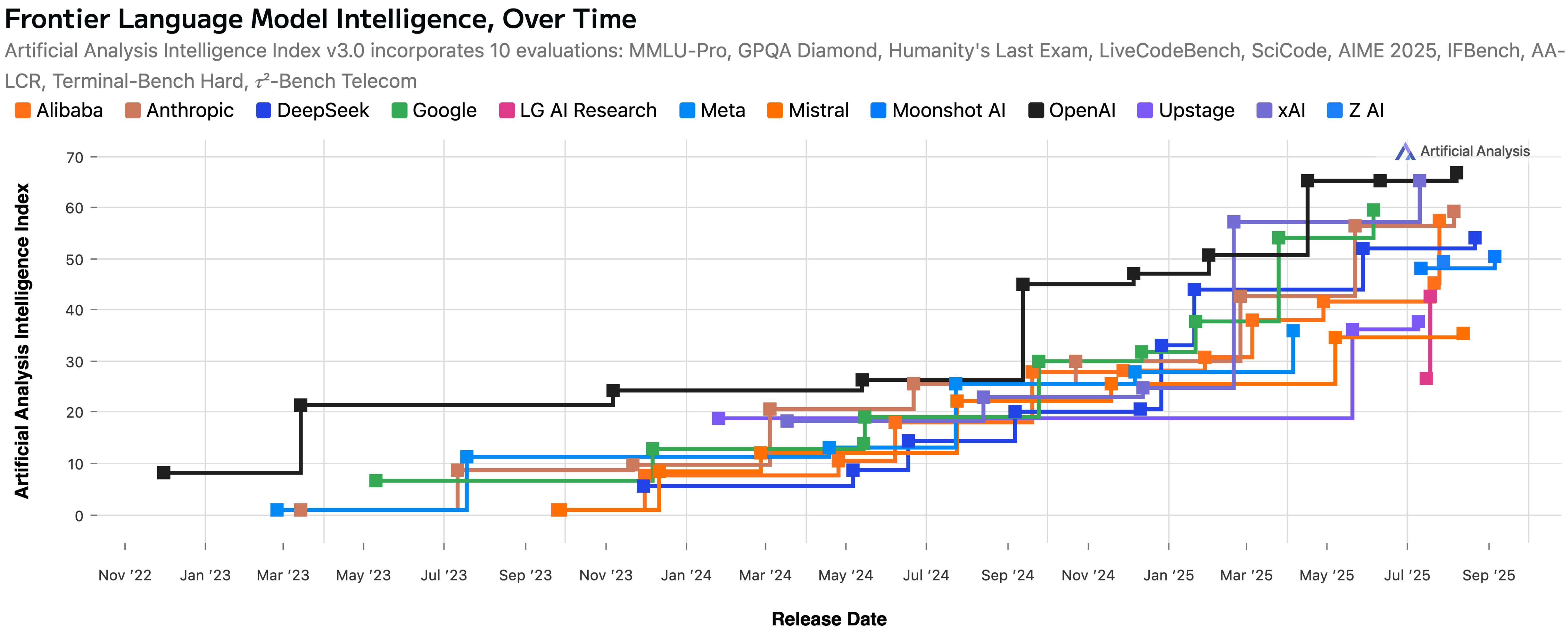
Artificial Analysis Intelligence Index
Intelligence, Speed, Price







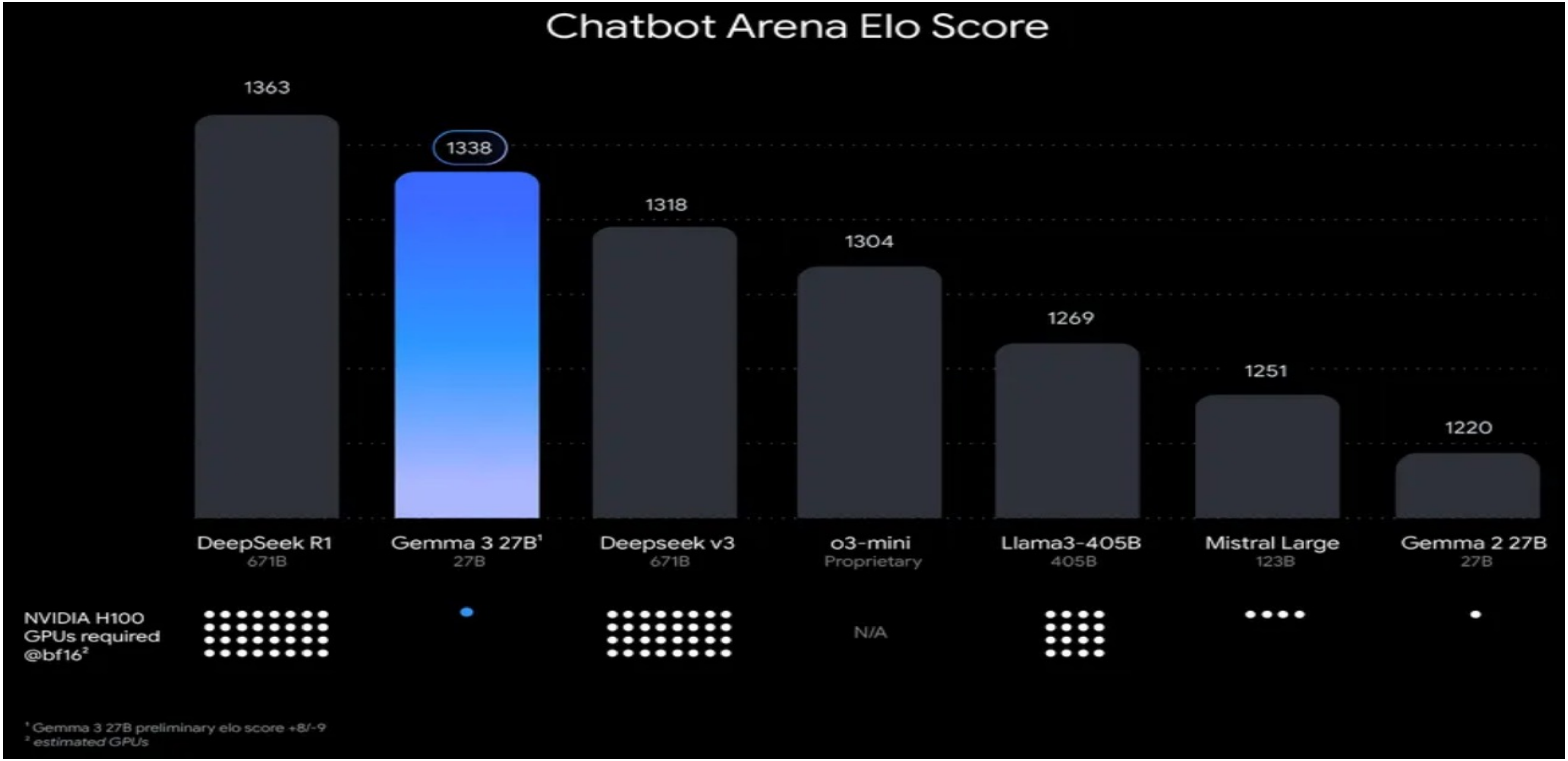
Artificial Analysis Intelligence Index 2022-2025



Source: https://artificialanalysis.ai/

Google Gemma 3 27B

The most capable model you can run on a single GPU or TPU



Source: https://blog.google/technology/developers/gemma-3/

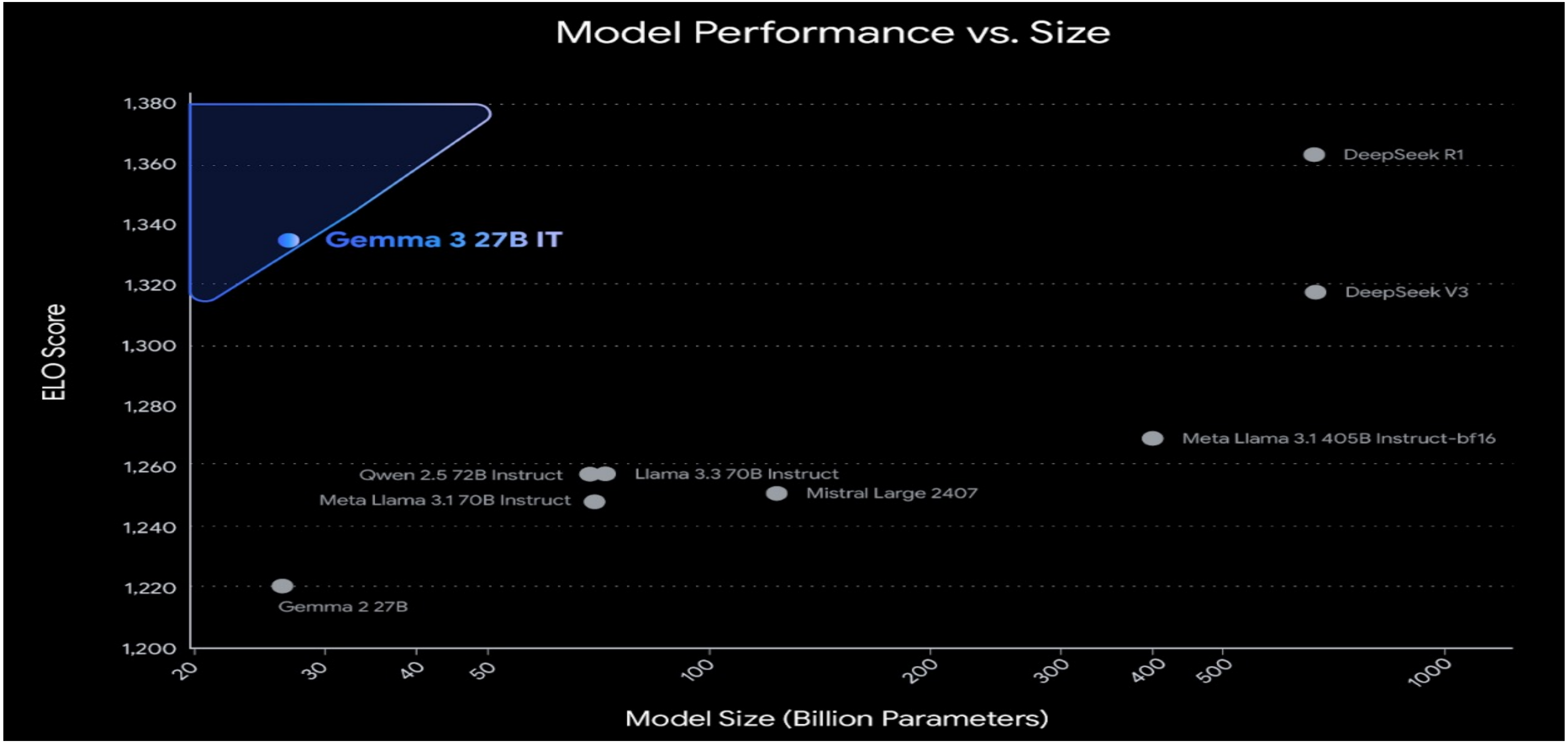
Google Gemma 3 Multimodality (vision-language input and text outputs)

MODEL	SIZE (in billion parameter)	CONTEXT LENGTH	LANGUAGES	INPUT MODALITIES
Gemma 3 1B (IT)	1B	32k	English	Input: Text Output: Text
Gemma 3 4B (IT)	4B	128k	+140 Languages	Input: Text, Image Output: Text
Gemma 3 12B (IT)	12B	128k	+140 Languages	Input: Text, Image Output: Text
Gemma 3 27B (IT)	27B	128k	+140 Languages	Input: Text, Image Output: Text
Shield Gemma 2	4B	8k	+140 Languages	Input: Text, Image Output: Text

Source: https://developers.googleblog.com/en/introducing-gemma3/

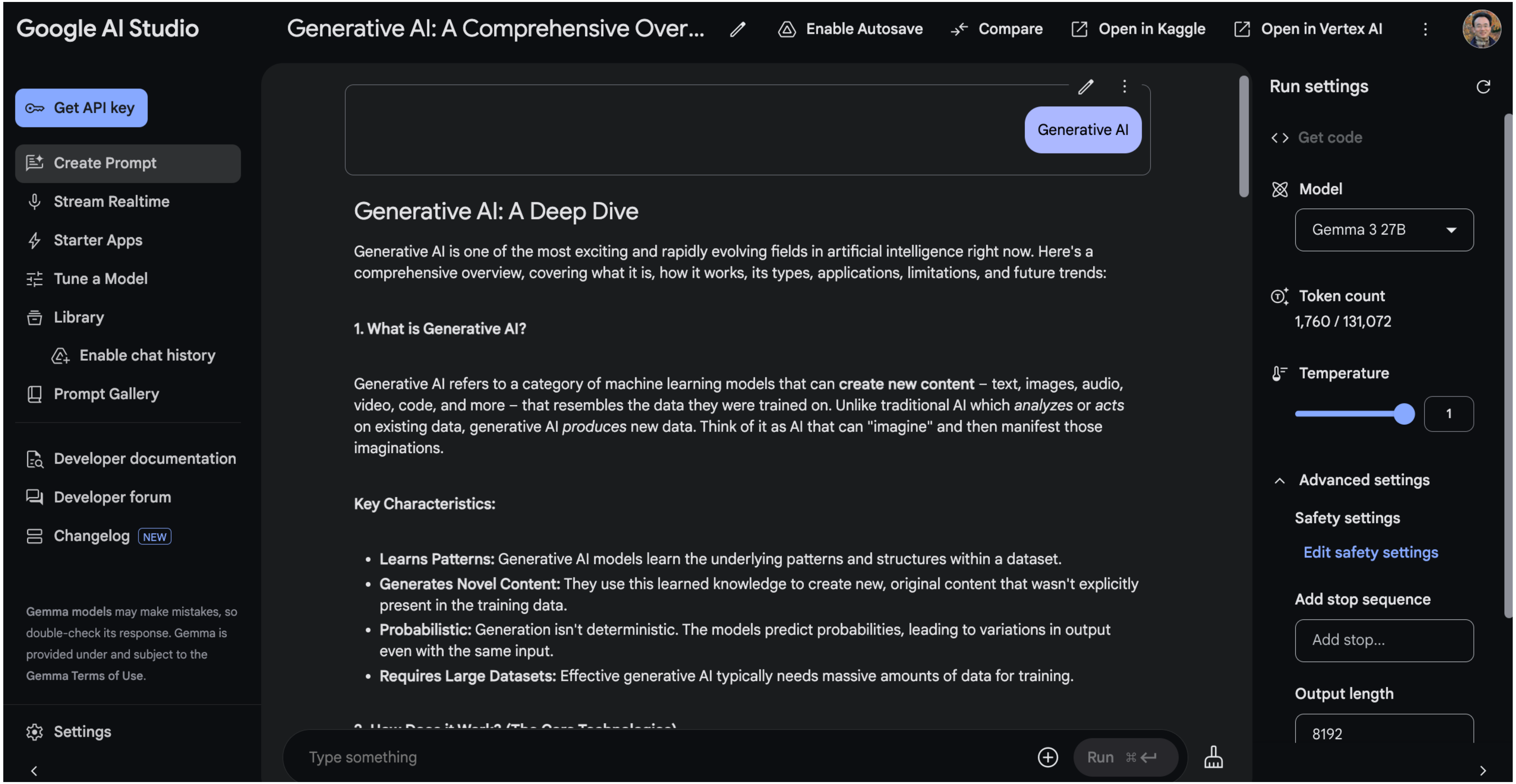
Google Gemma 3: Pre-training and Post-training

(distillation, reinforcement learning, and model merging)



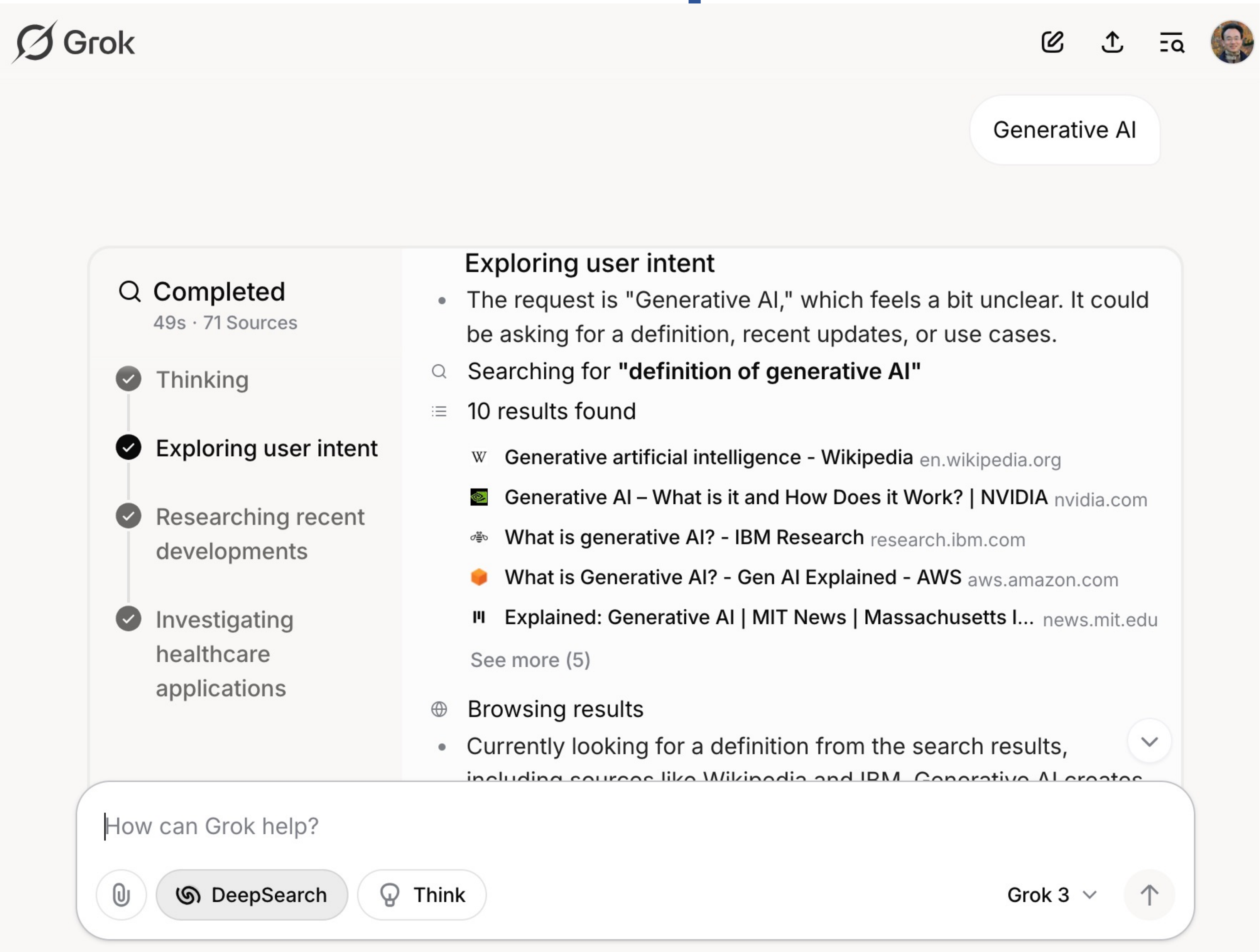
Source: https://developers.googleblog.com/en/introducing-gemma3/

Google AI Studio (Gemma 3 27B)



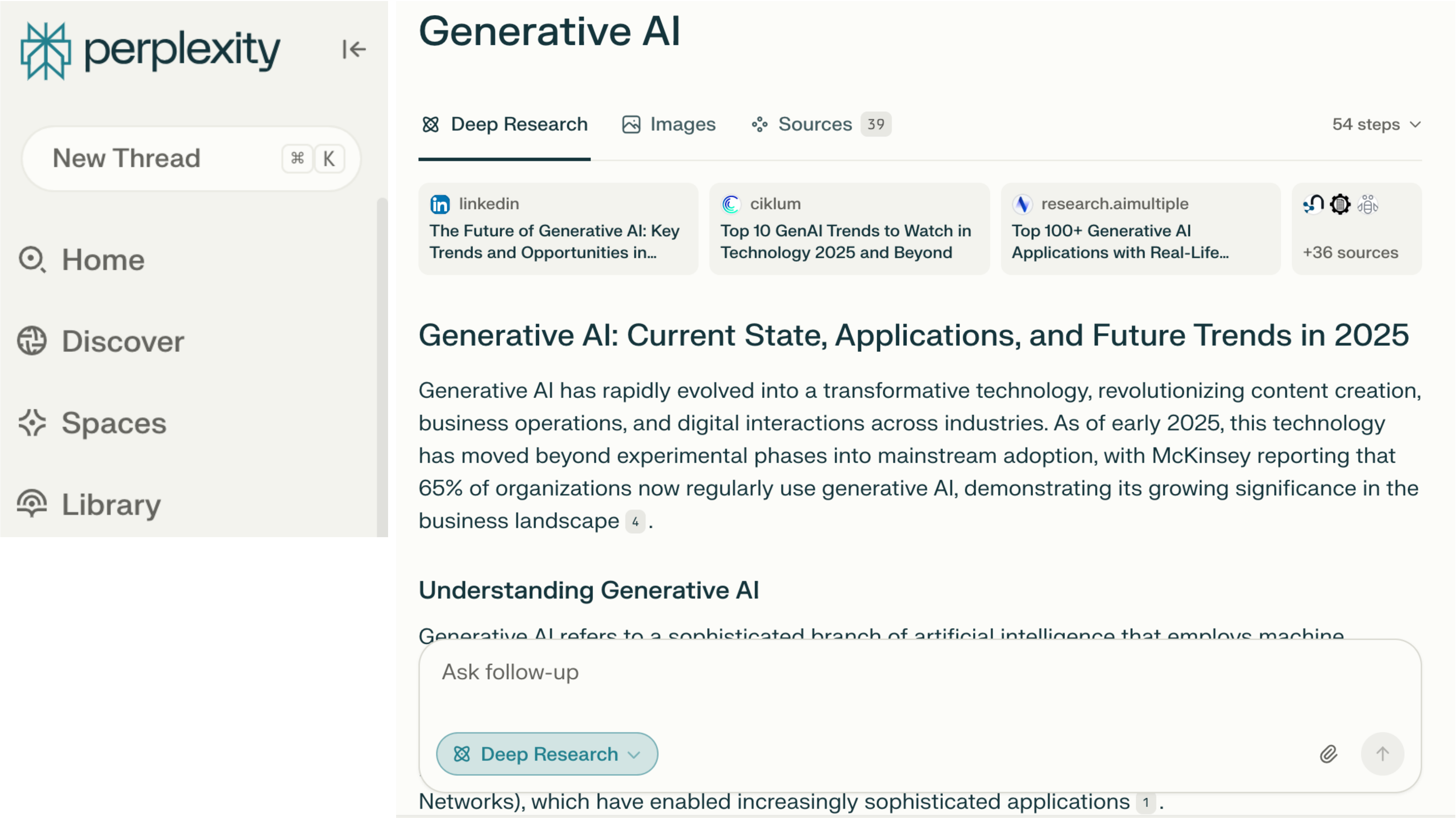
Source: https://aistudio.google.com/

Grok 3 Deep Search



Source: https://grok.com/chat

Perplexity.ai Deep Research



Source: https://www.perplexity.ai/

Token

Tiktokenizer

gpt-4o

System | You are a helpful assistant

User | Content

Add message

```
<|im_start|>system<|im_sep|>You are a helpful
assistant<|im_end|><|im_start|>user<|im_sep|><|im_end|>
<|im_start|>assistant<|im_sep|>
```

Token count

16

```
<|im_start|>system<|im_sep|>You are a helpful assistan
t<|im_end|><|im_start|>user<|im_sep|><|im_end|><|im_st
art|>assistant<|im_sep|>
```

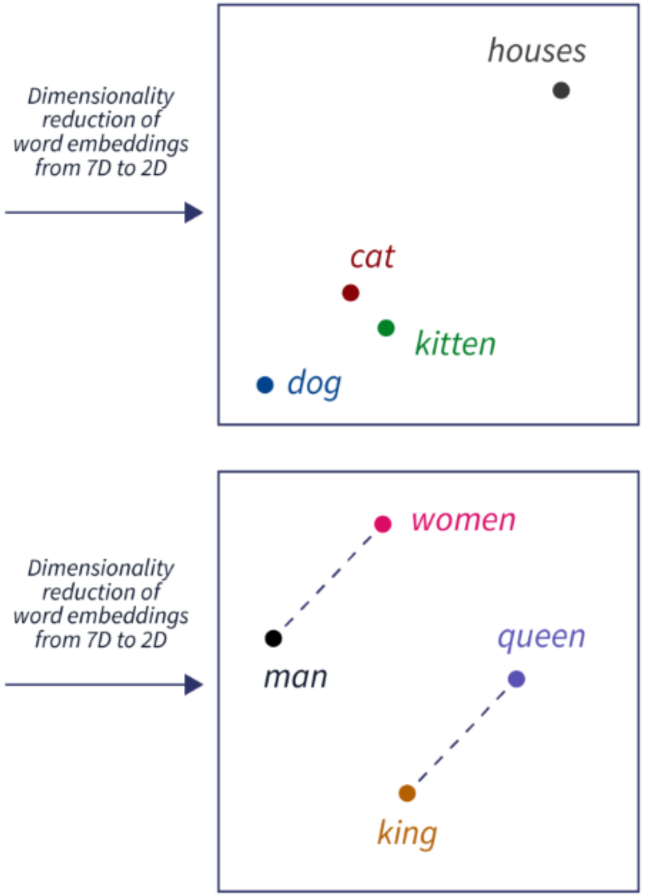
```
200264, 17360, 200266, 3575, 553, 261, 10297, 29186, 2
00265, 200264, 1428, 200266, 200265, 200264, 173781, 2
00266
```

Show whitespace

https://tiktokenizer.vercel.app/

Word Embeddings

Word	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
kitten →	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
dog →	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
houses →	0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8
man →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
women →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
king →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
queen →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

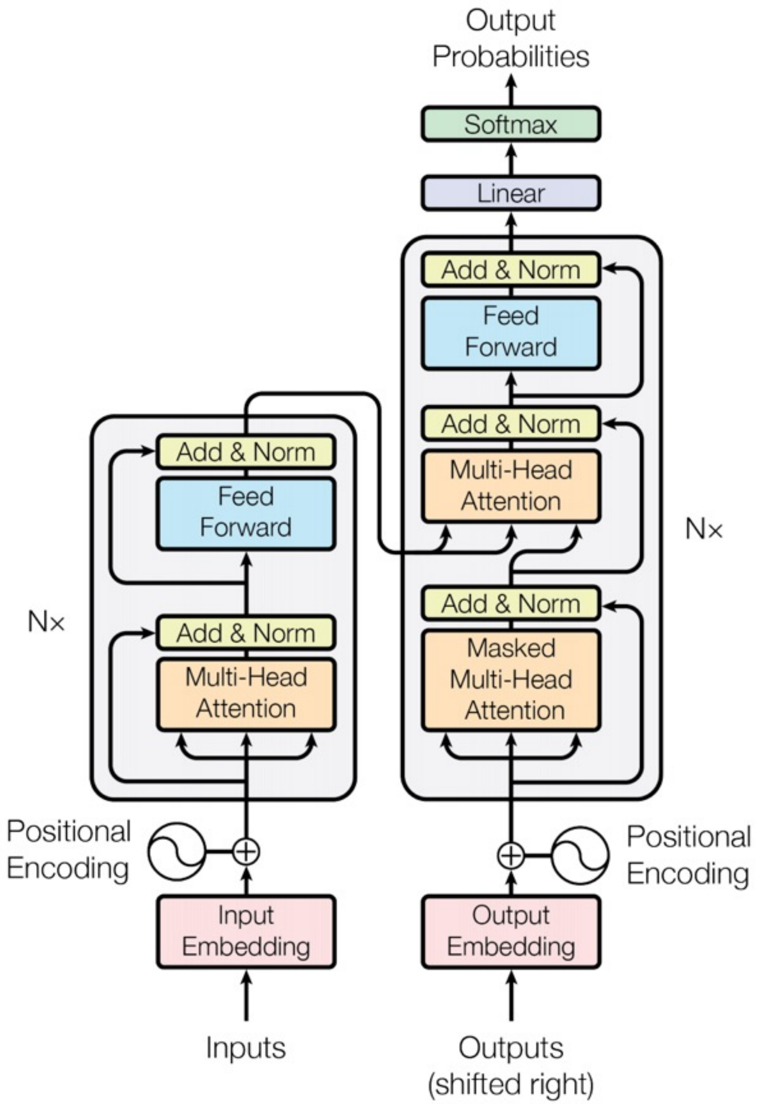


Word | Word Embedding | Dimensionality reduction | Visualization of Word embeddings in 2D

Source: https://www.scaler.com/topics/tensorflow/tensorflow-word-embeddings/

Transformer (Attention is All You Need)

(Vaswani et al., 2017)

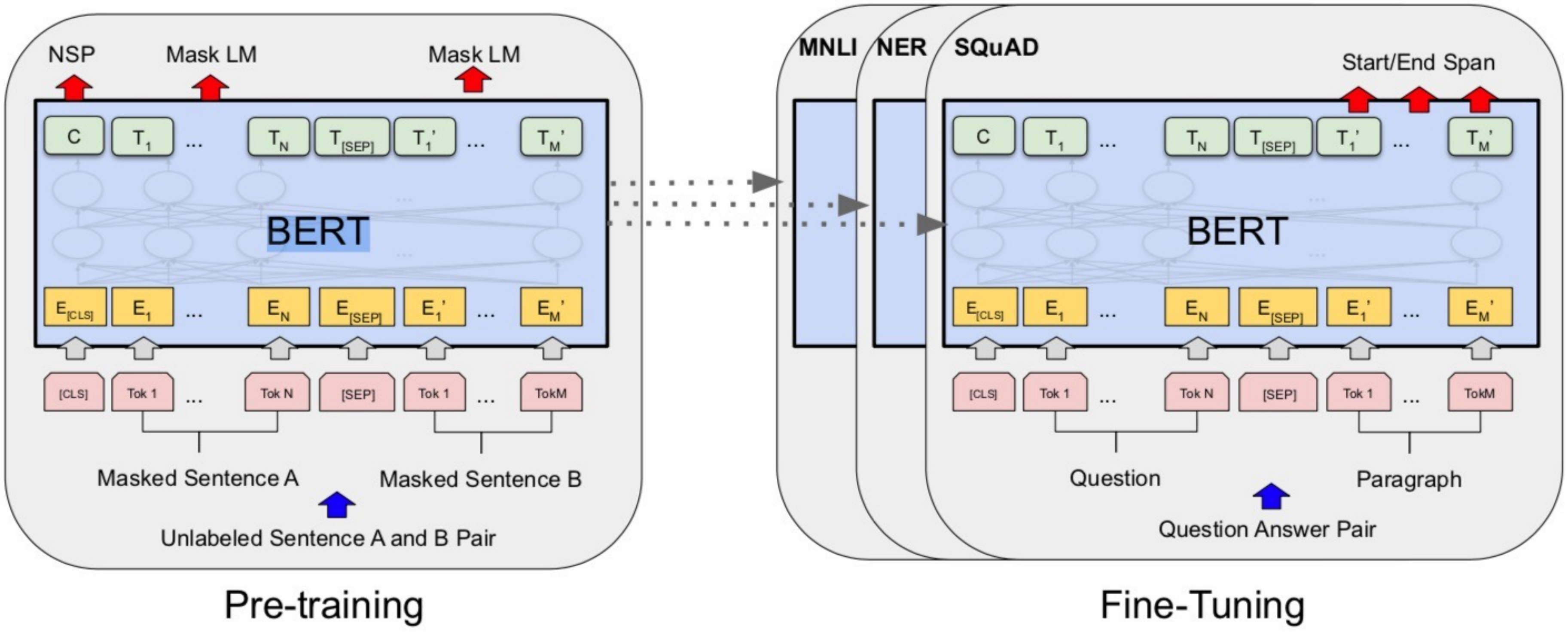


Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

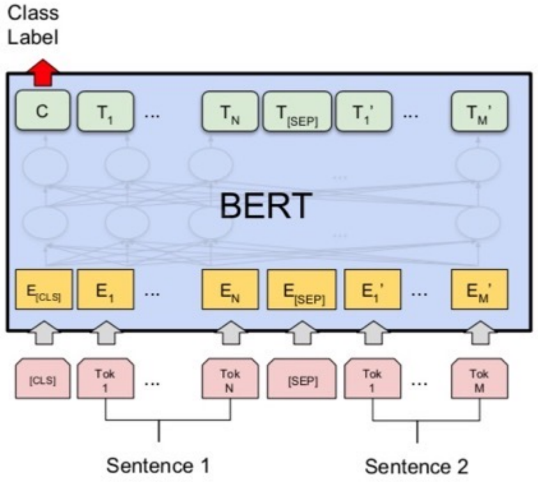
BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT

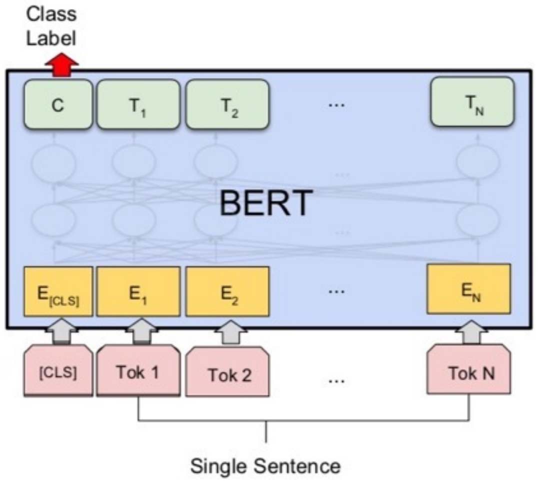


Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

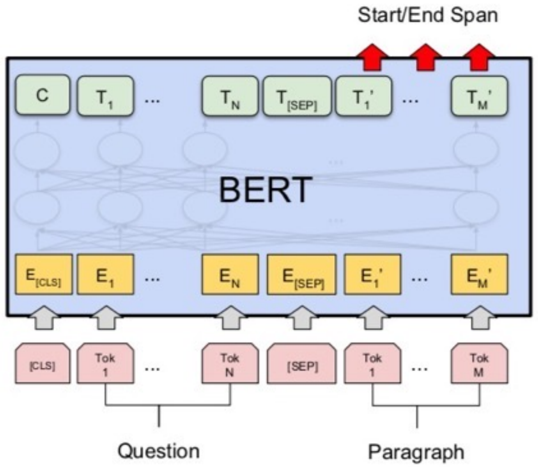
Fine-tuning BERT on Different Tasks



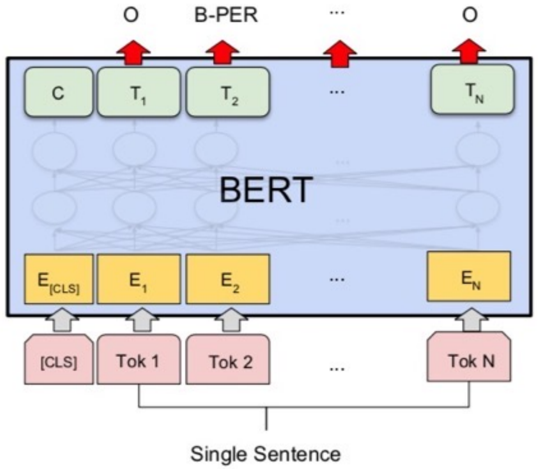
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



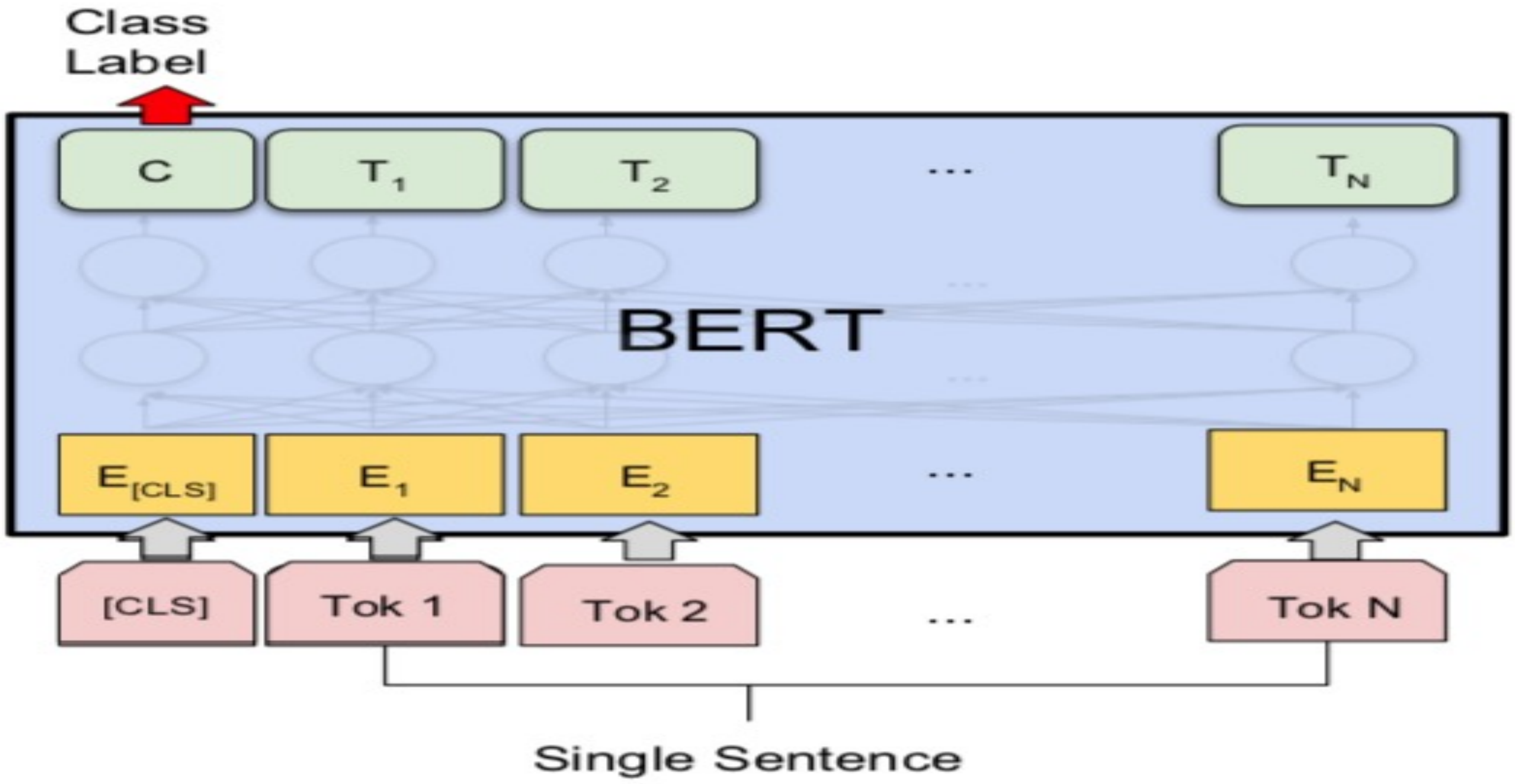
(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

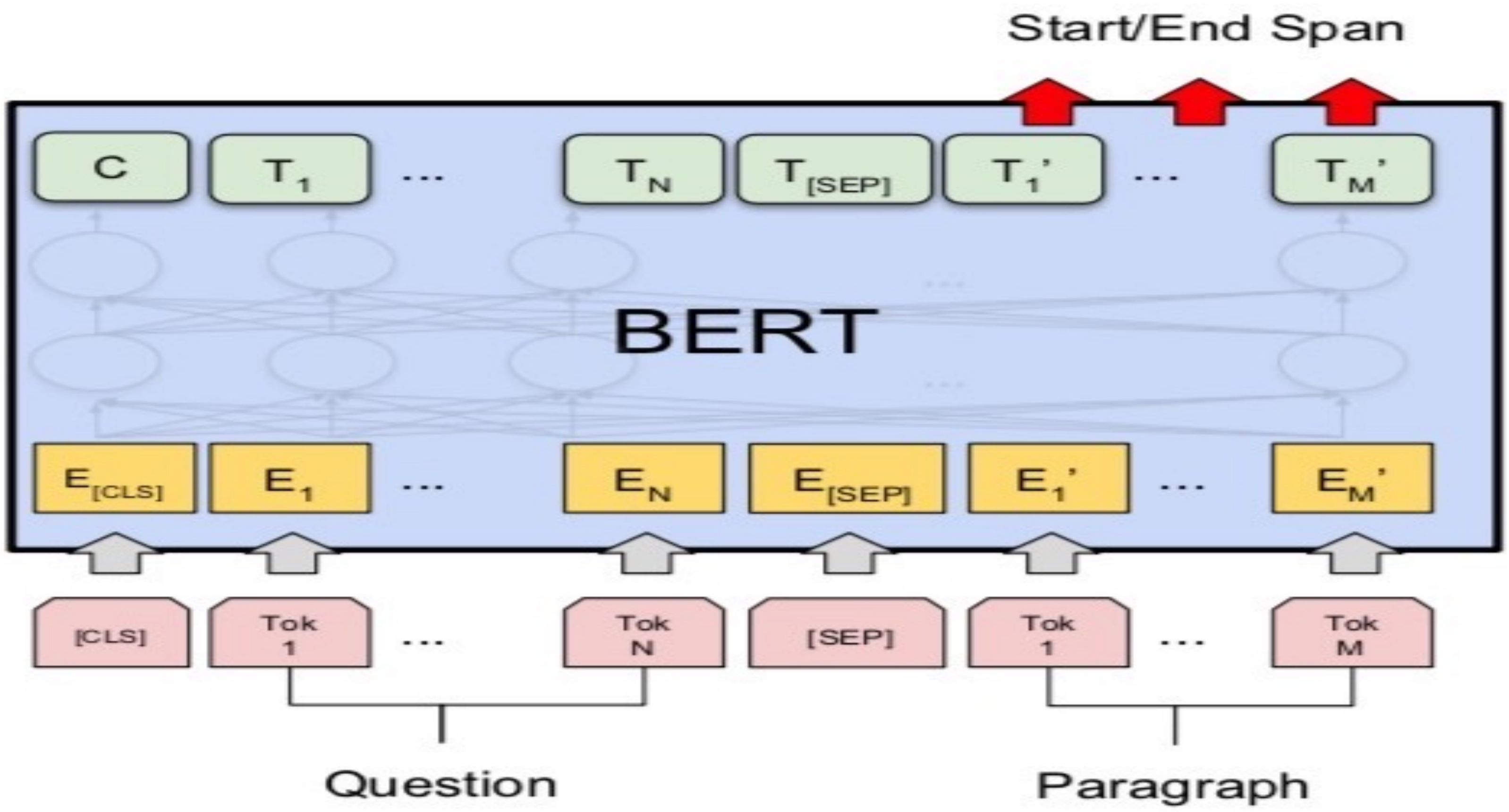
Sentiment Analysis: Single Sentence Classification



(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

Fine-tuning BERT on Question Answering (QA)

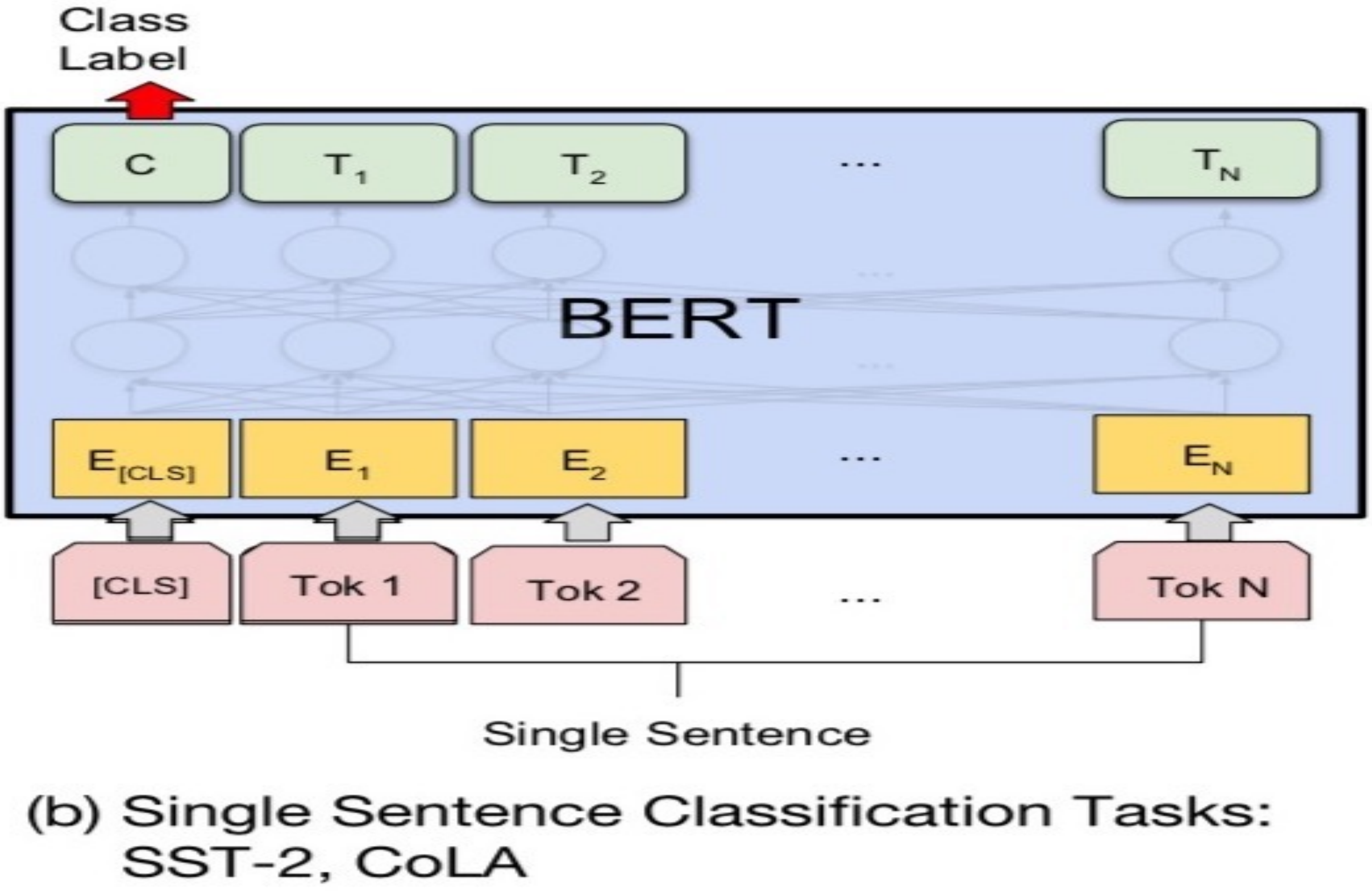


(c) Question Answering Tasks: SQuAD v1.1

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

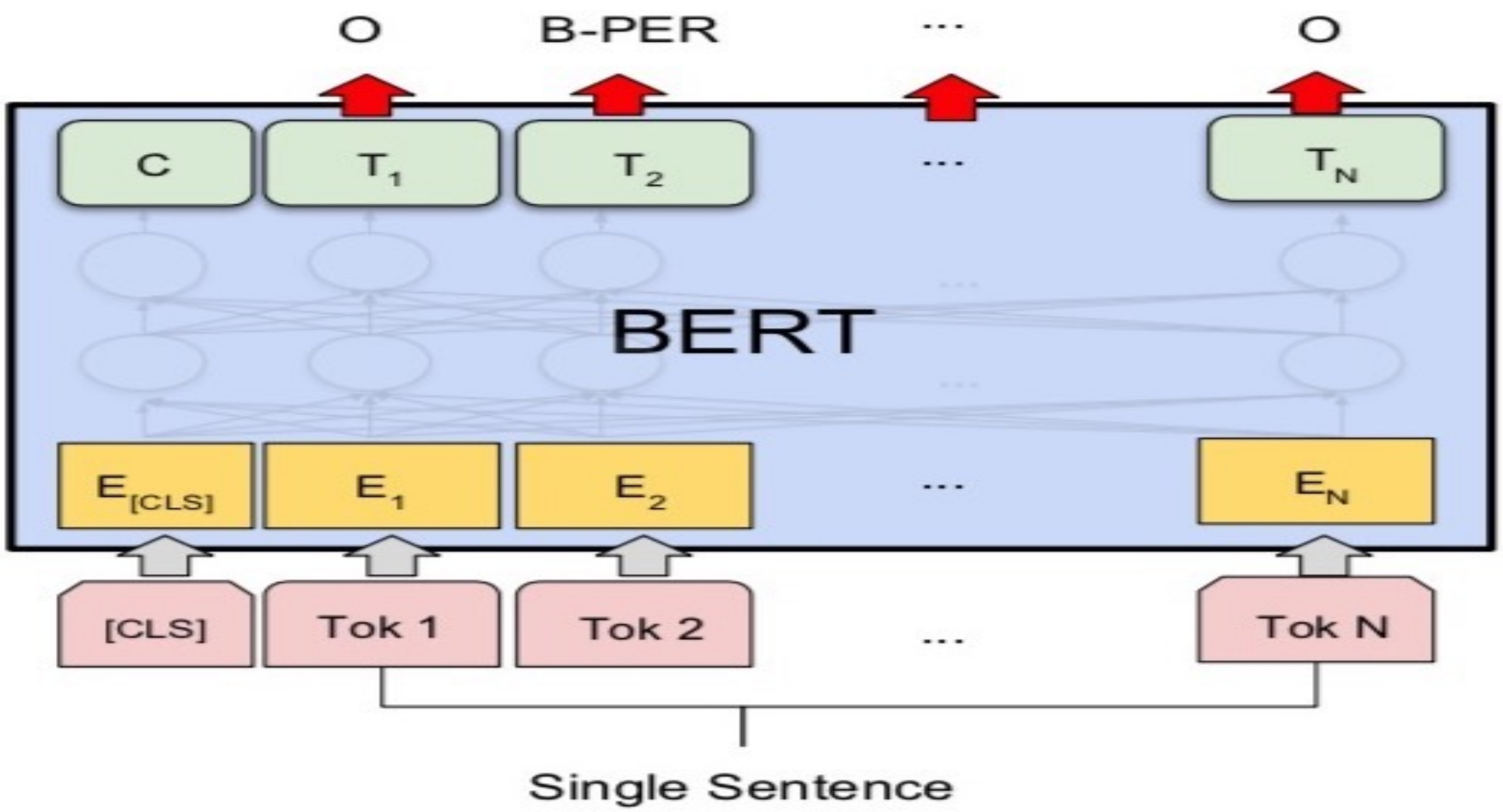
Intent Detection (ID; Classification)



Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Slot Filling (SF)

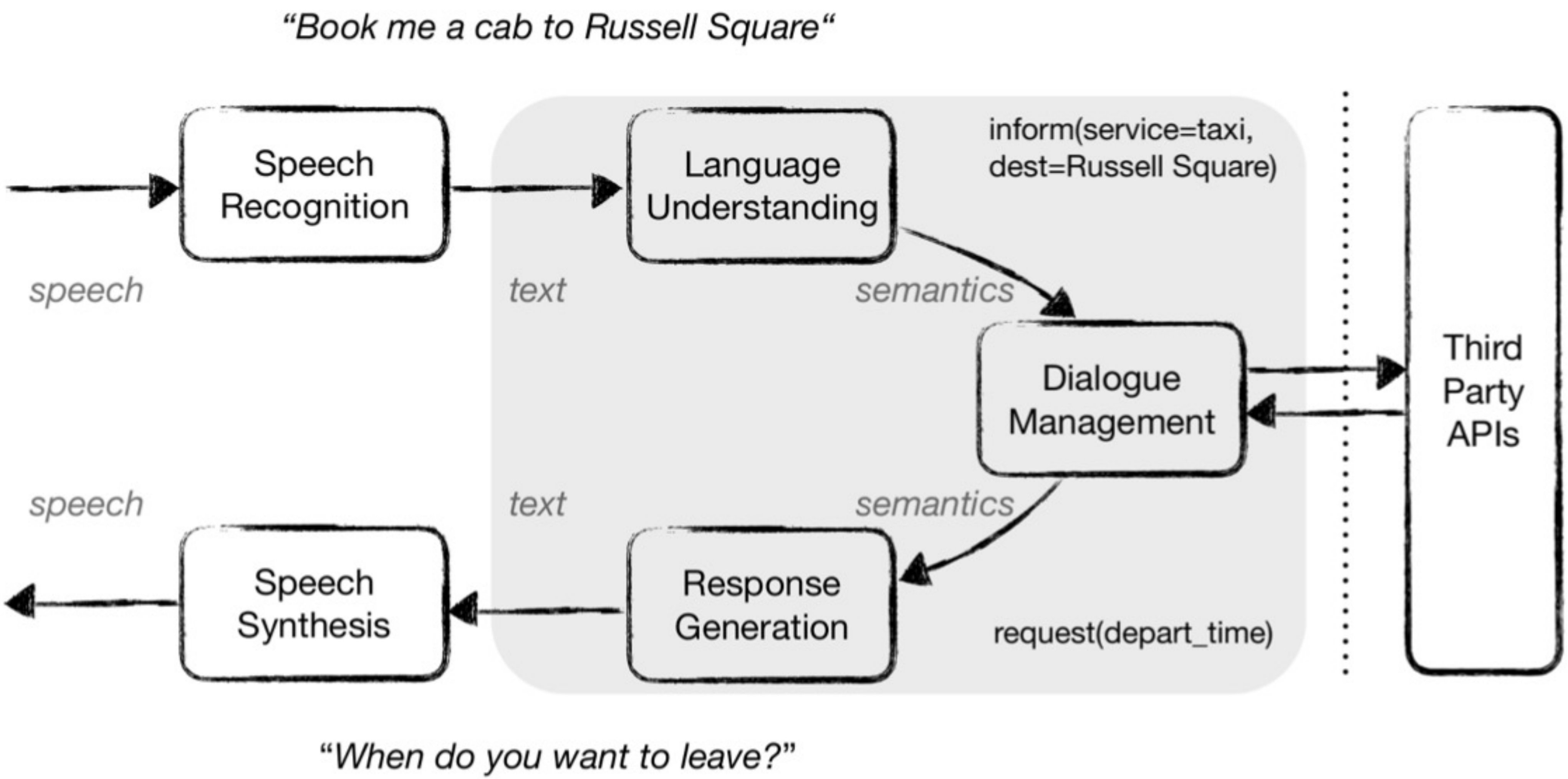


(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Task-Oriented Dialogue (ToD) System

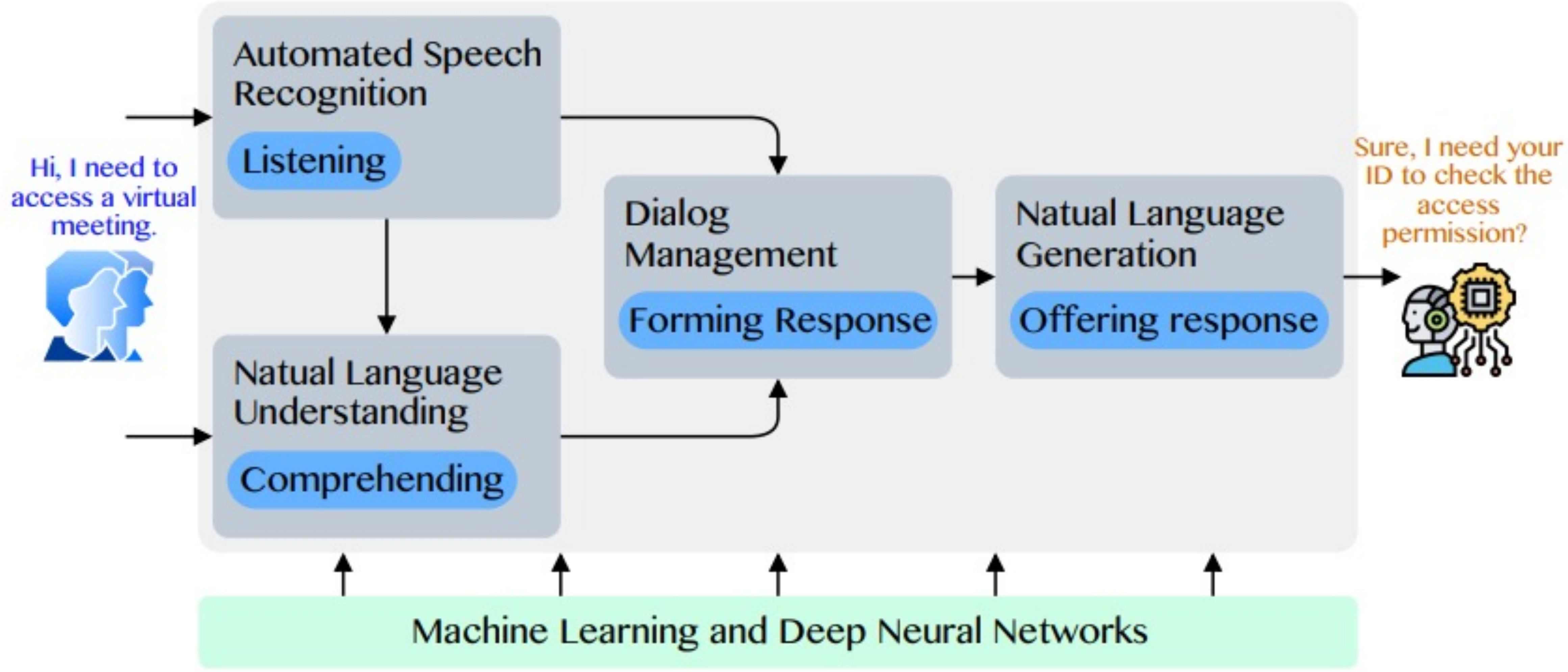
Speech, Text, NLP



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.

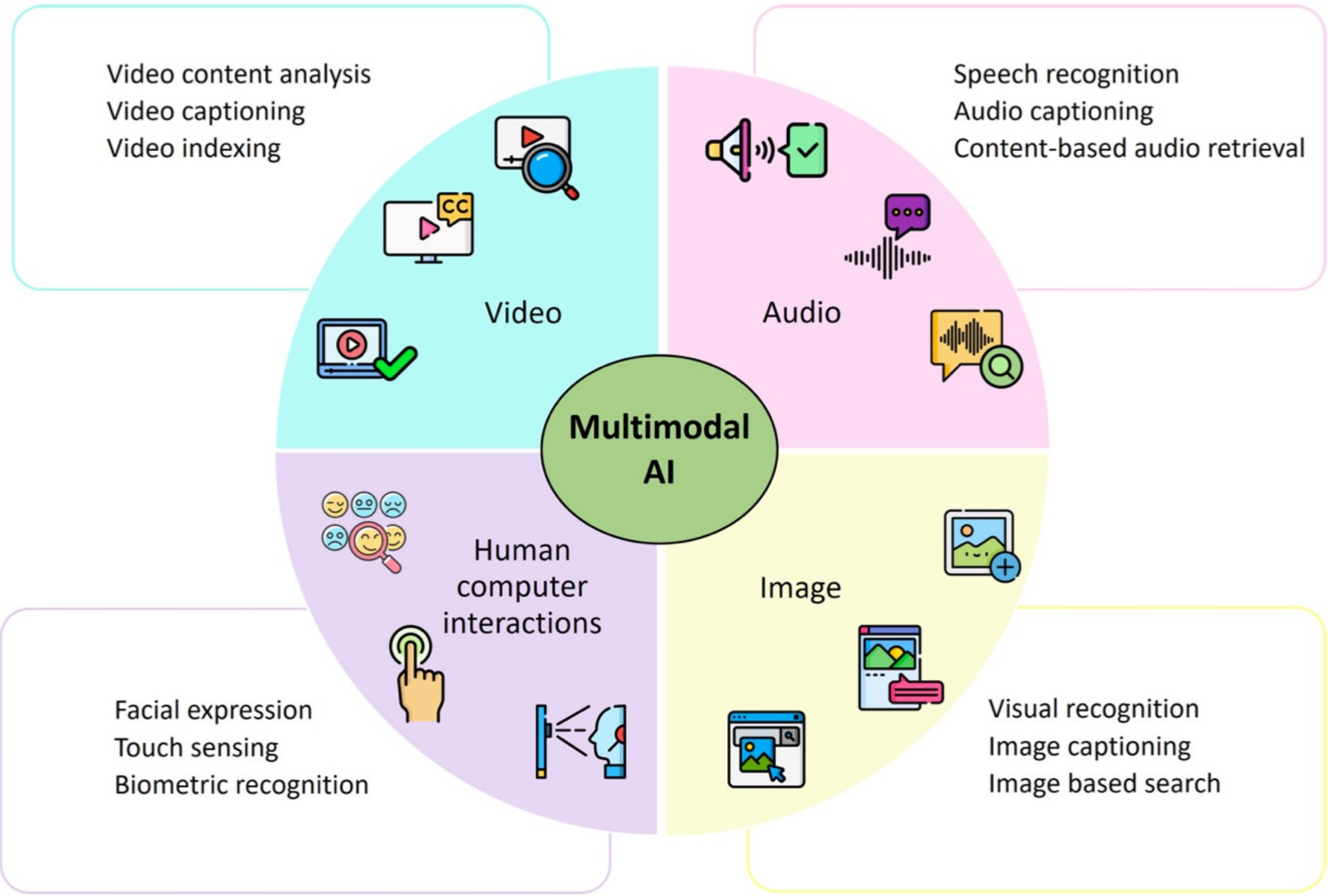
Conversational AI

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.

Technological Integration for Multimodal AI



Source: Sohail, Shahab Saquib, Faiza Farhat, Yassine Himeur, Mohammad Nadeem, Dag Øivind Madsen, Yashbir Singh, Shadi Atalla, and Wathiq Mansoor. "The Future of GPT: A Taxonomy of Existing ChatGPT Research, Current Challenges, and Possible Future Directions." Current Challenges, and Possible Future Directions (April 8, 2023) (2023).

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

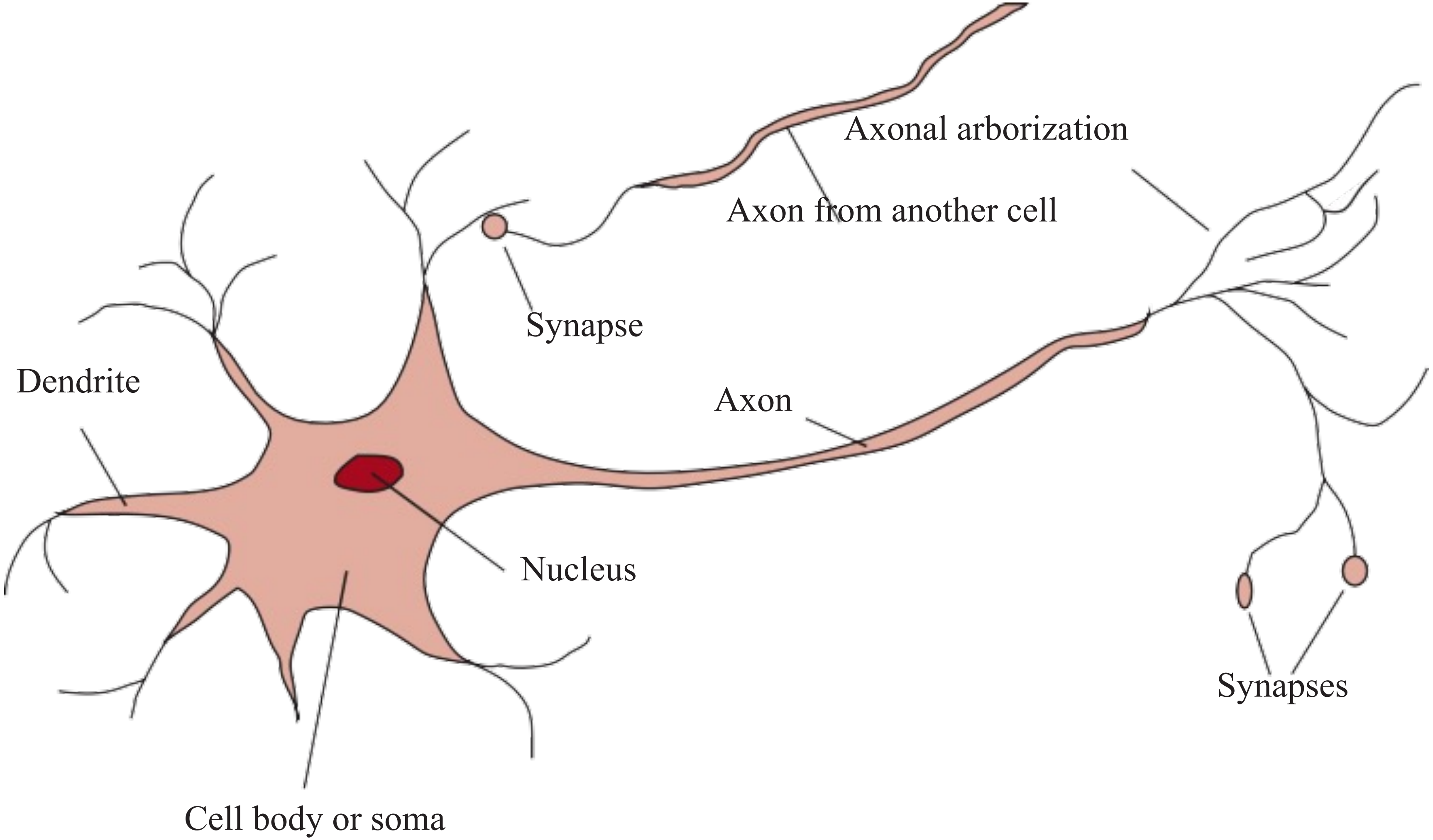
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Acting Rationally: The Rational Agent Approach

- AI has focused on the study and construction of agents that **do the right thing.**
- **Standard model**

Neuroscience

The parts of a nerve cell or neuron



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

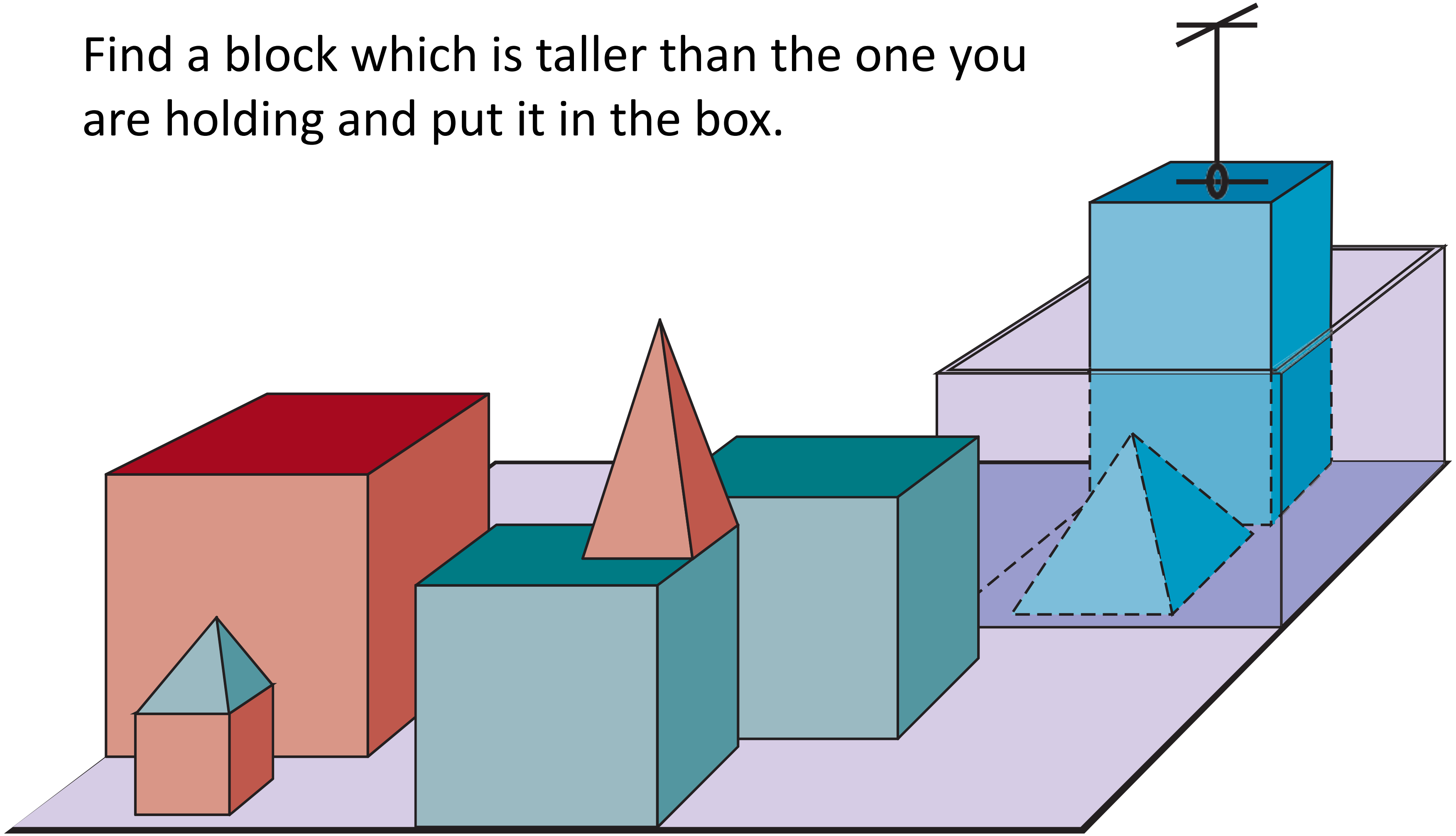
Comparison of Computer and Human Brain

	Supercomputer	Personal Computer	Human Brain
Computational units	10^6 GPUs + CPUs	8 CPU cores	10^6 columns
	10^{15} transistors	10^{10} transistors	10^{11} neurons
Storage units	10^{16} bytes RAM	10^{10} bytes RAM	10^{11} neurons
	10^{17} bytes disk	10^{12} bytes disk	10^{14} synapses
Cycle time	10^{-9} sec	10^{-9} sec	10^{-3} sec
Operations/sec	10^{18}	10^{10}	10^{17}

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

A scene from the blocks world

Find a block which is taller than the one you are holding and put it in the box.



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

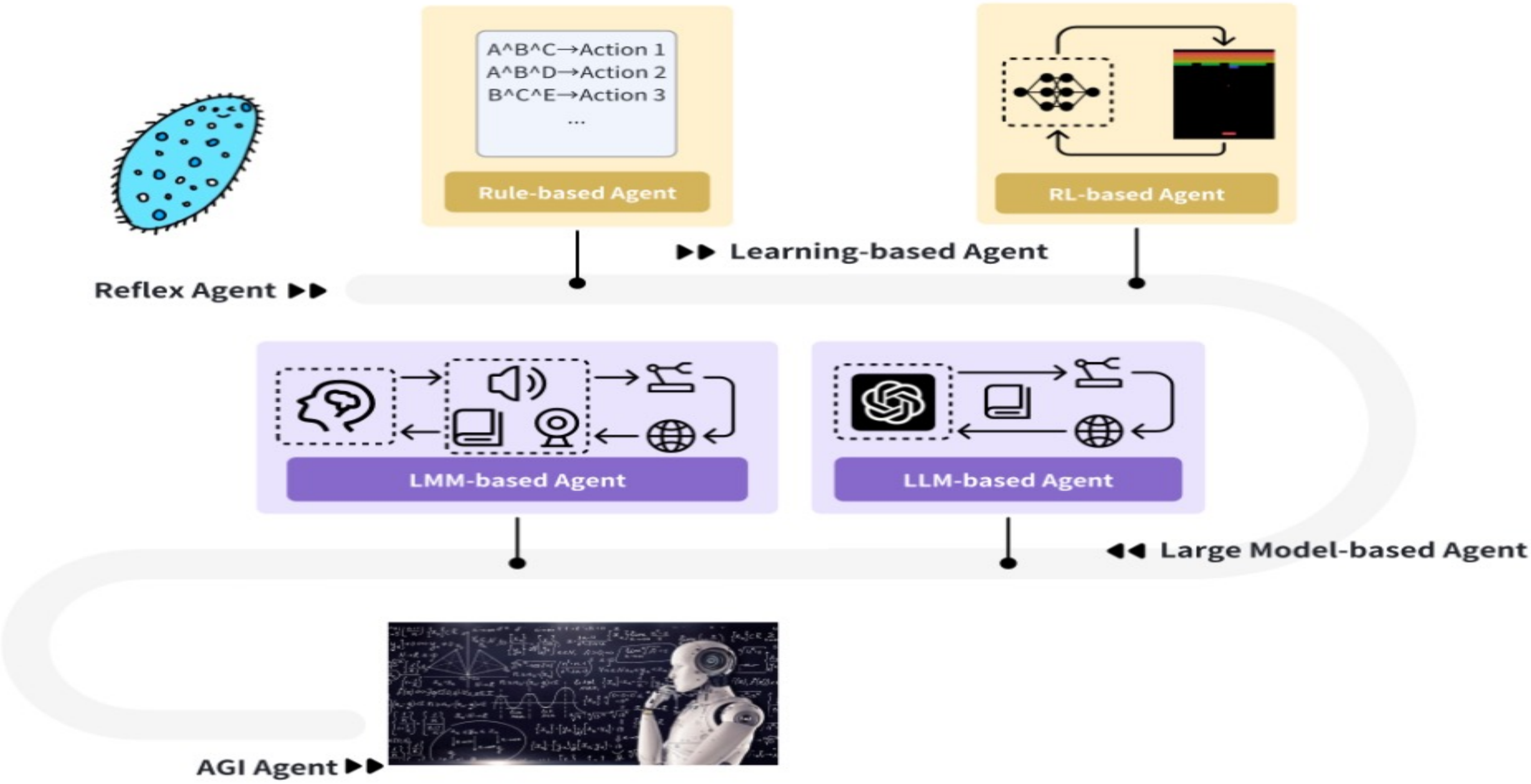
Intelligent Agents

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

Intelligent Agents Roadmap



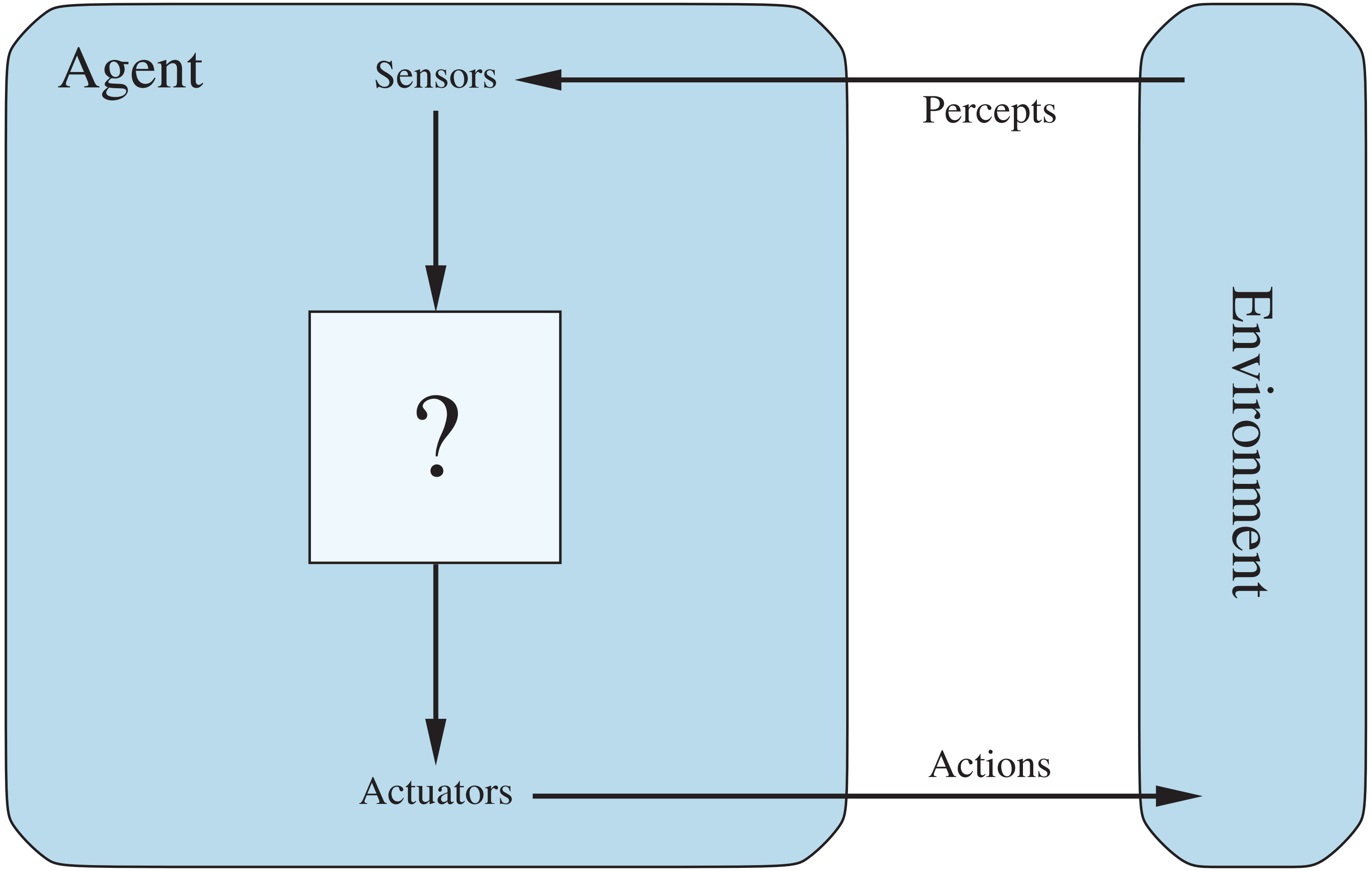
Source: Cheng, Yuheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang et al. "Exploring large language model based intelligent agents: Definitions, methods, and prospects." arXiv preprint arXiv:2401.03428 (2024).

AI Agents

- Traditional AI Agents
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents
  - Learning agents

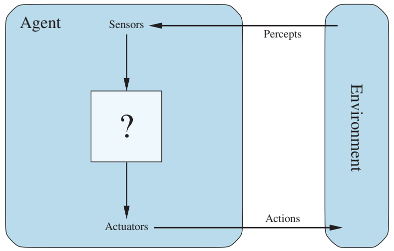
- Evolution of AI Agents
  - LLM-based Agents
  - Multi-modal agents
  - Embodied AI agents in virtual environments
  - Collaborative AI agents

Agents interact with environments through sensors and actuators



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

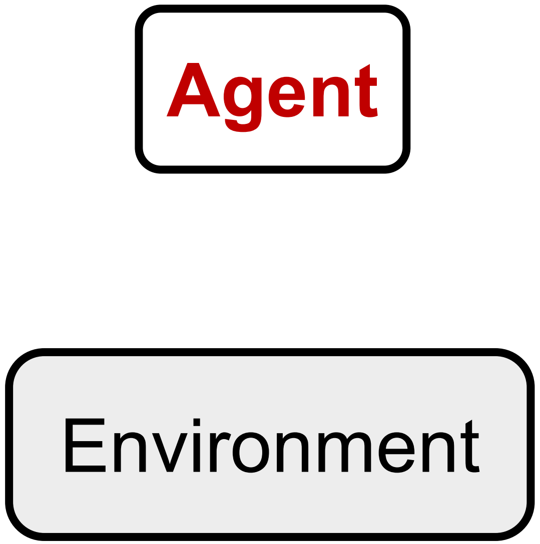
AI Agents



- Definition: An **AI agent** is an **entity** that **perceives** its **environment** and takes **actions** to achieve **goals**
- Components:
  1. **Sensors**: Perceive the environment
  2. **Actuators**: Act upon the environment
  3. **Decision-making** mechanism: Process inputs and decide on actions

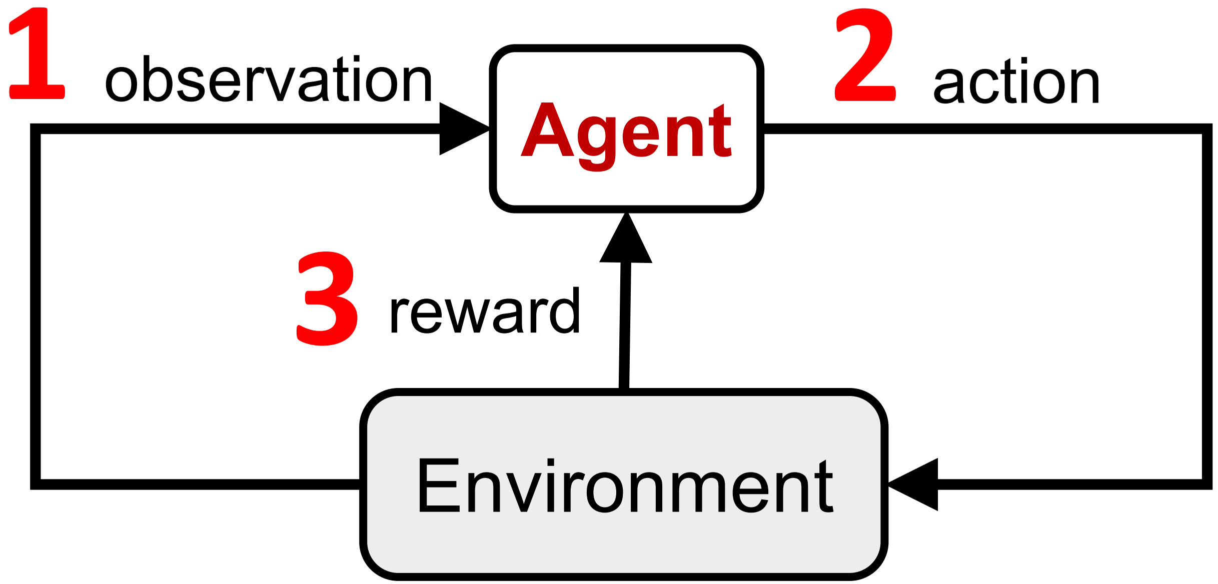
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Reinforcement Learning (DL)



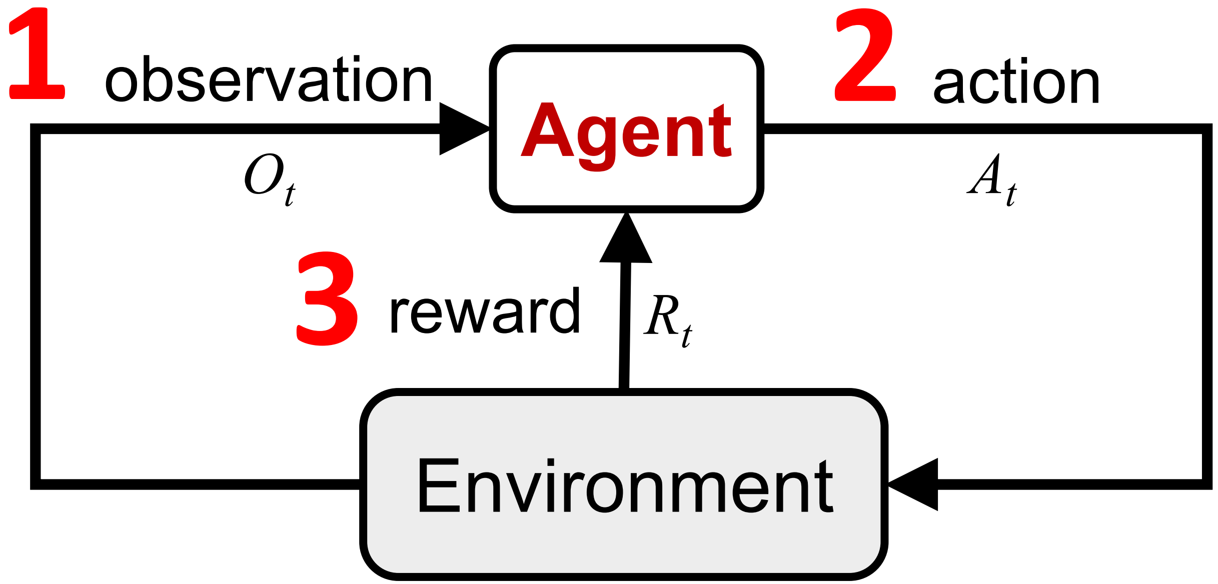
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



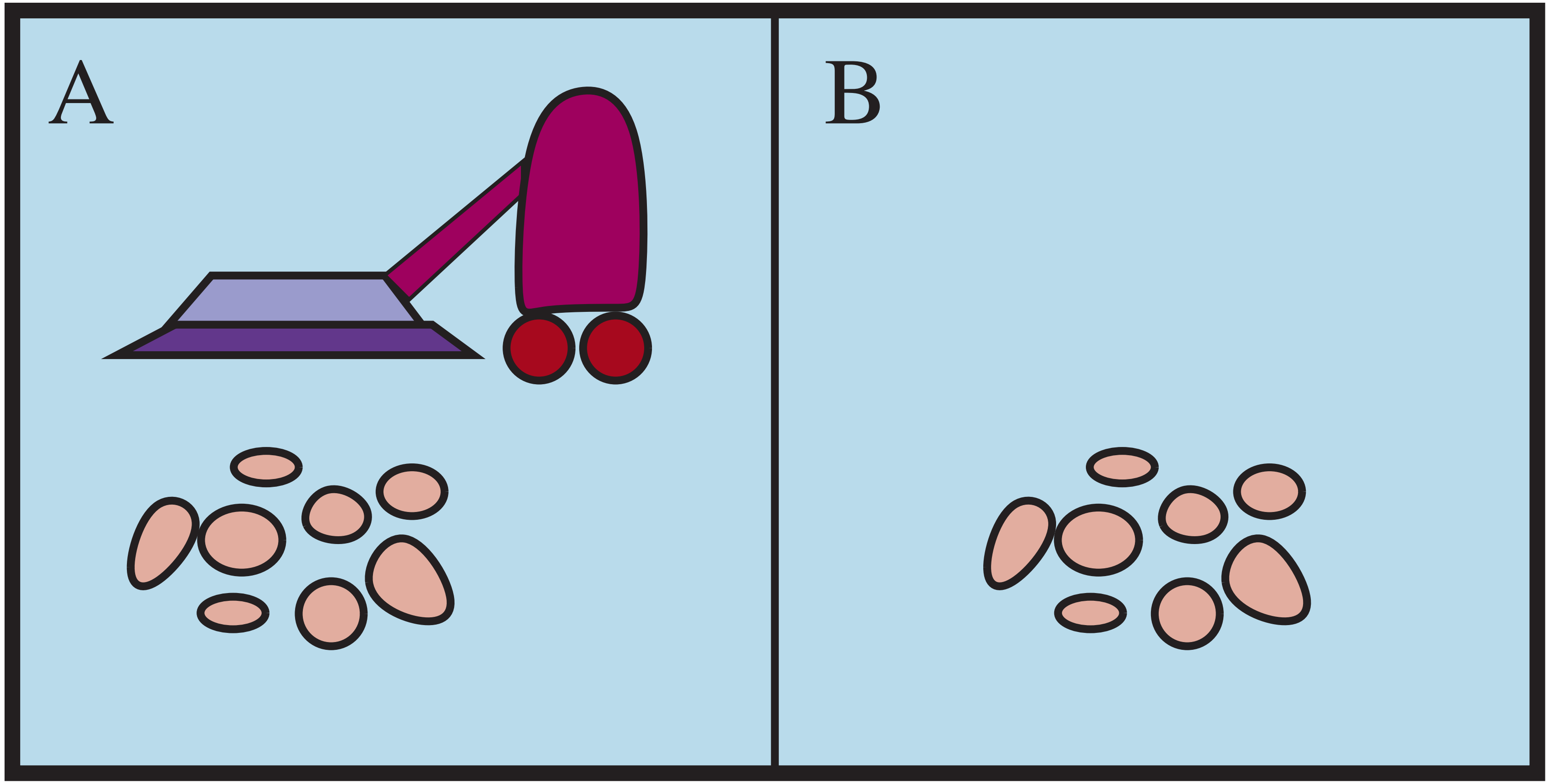
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



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

A vacuum-cleaner world with just two locations



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

Partial tabulation of a simple agent function for the vacuum-cleaner world

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$
$[A, Clean], [A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$

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

PEAS description of the task environment for an automated taxi driver

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

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

Examples of Agent Types and their PEAS descriptions

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student’s score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice

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

Examples of Task Environments and their Characteristics

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

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

The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time.

It retains the complete percept sequence in memory.

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified

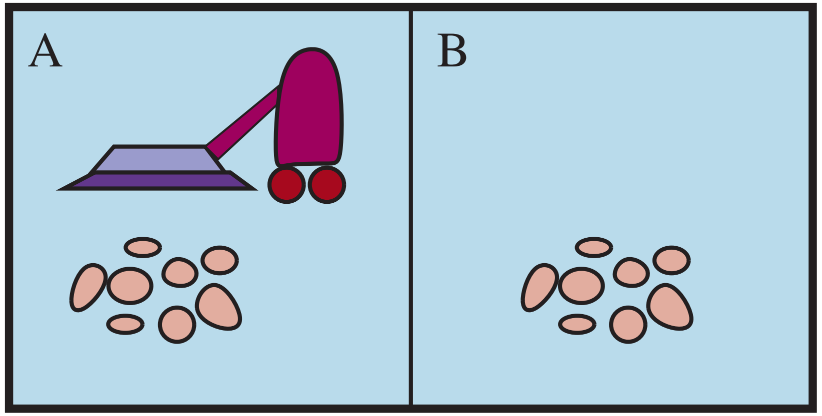
  append *percept* to the end of *percepts*
  action ← LOOKUP(*percepts*, *table*)
  return *action*

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

The agent program for a simple reflex agent in the two-location vacuum environment.

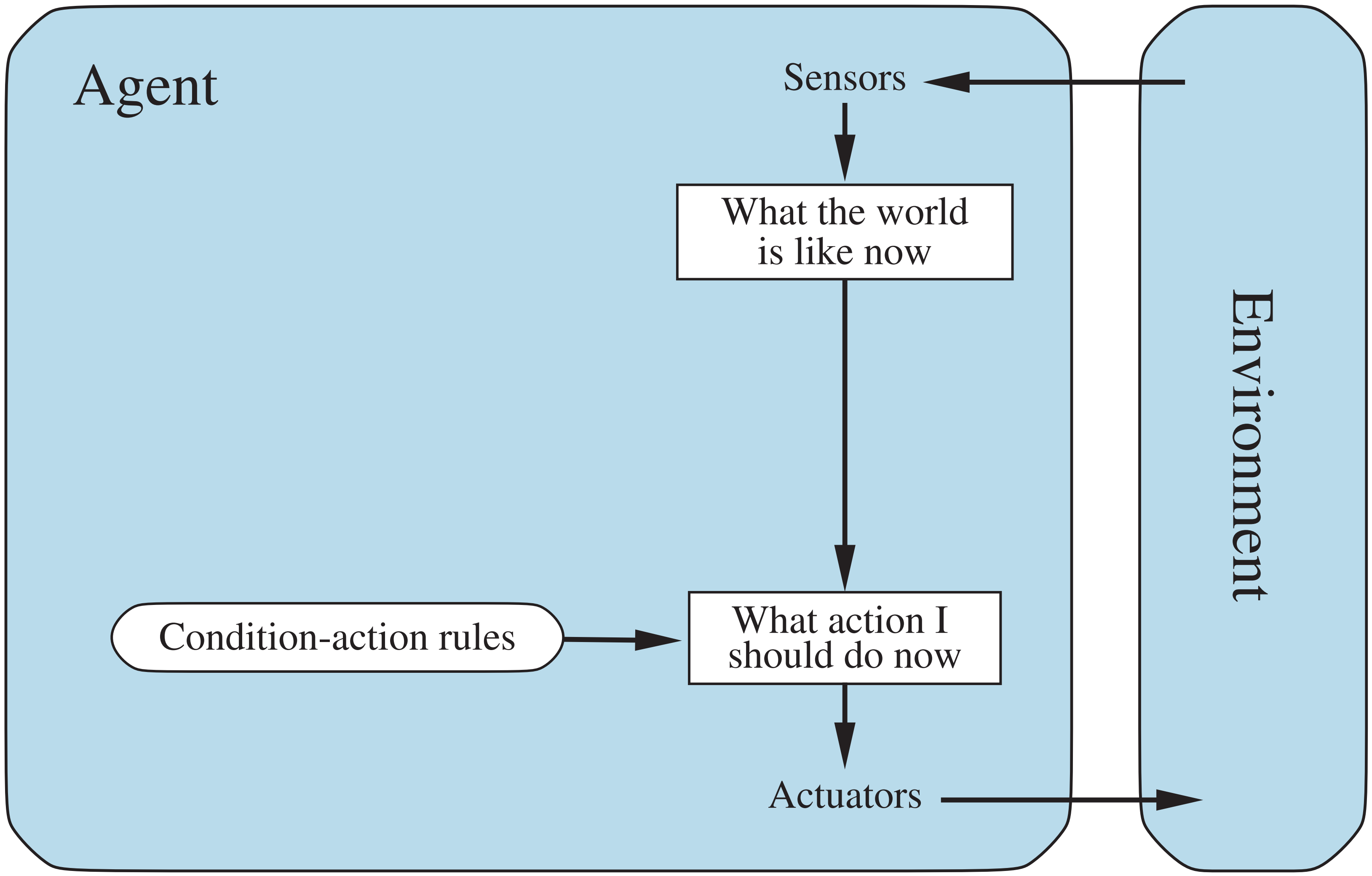
function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*



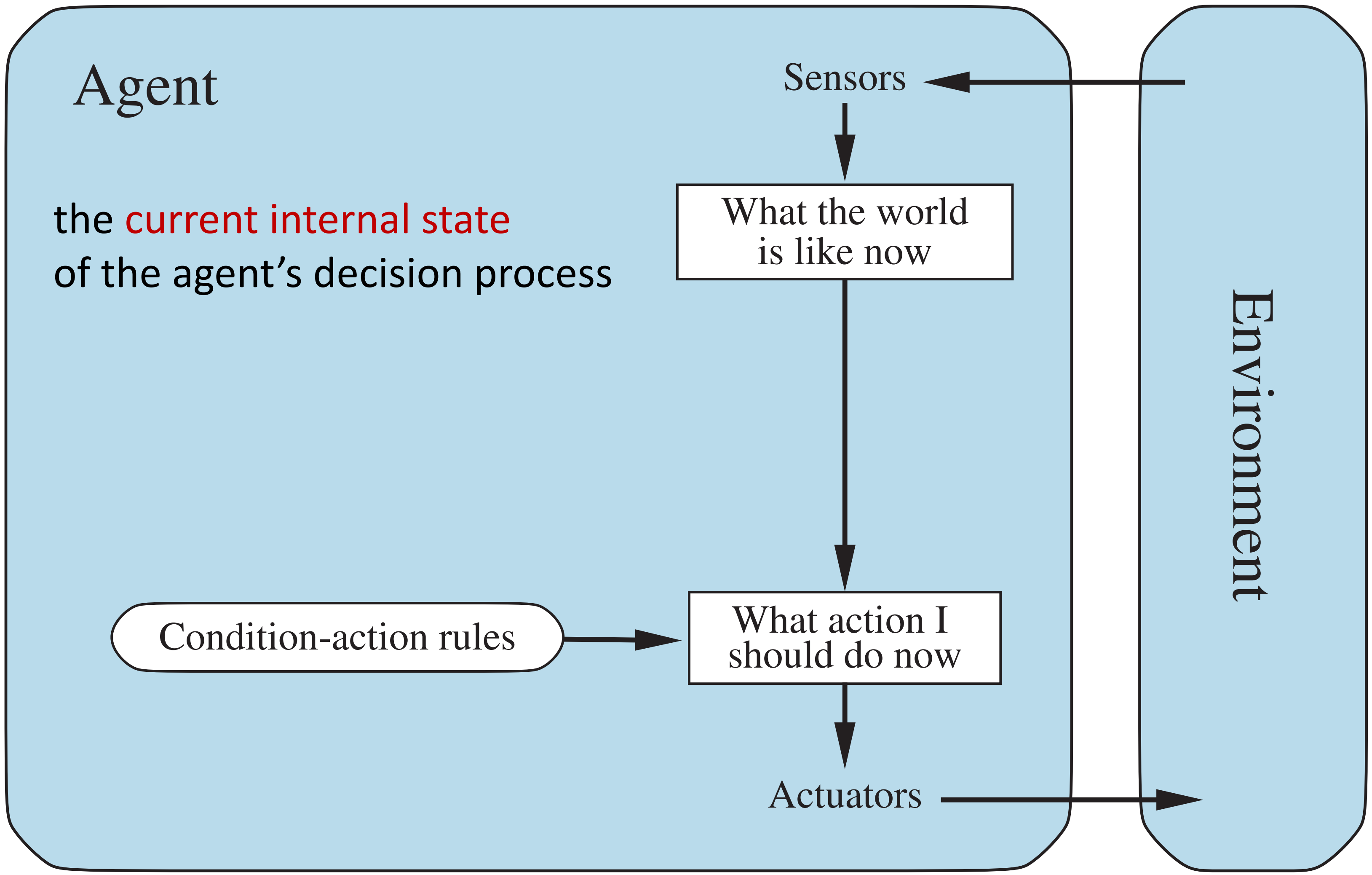
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Schematic Diagram of a Simple Reflex Agent



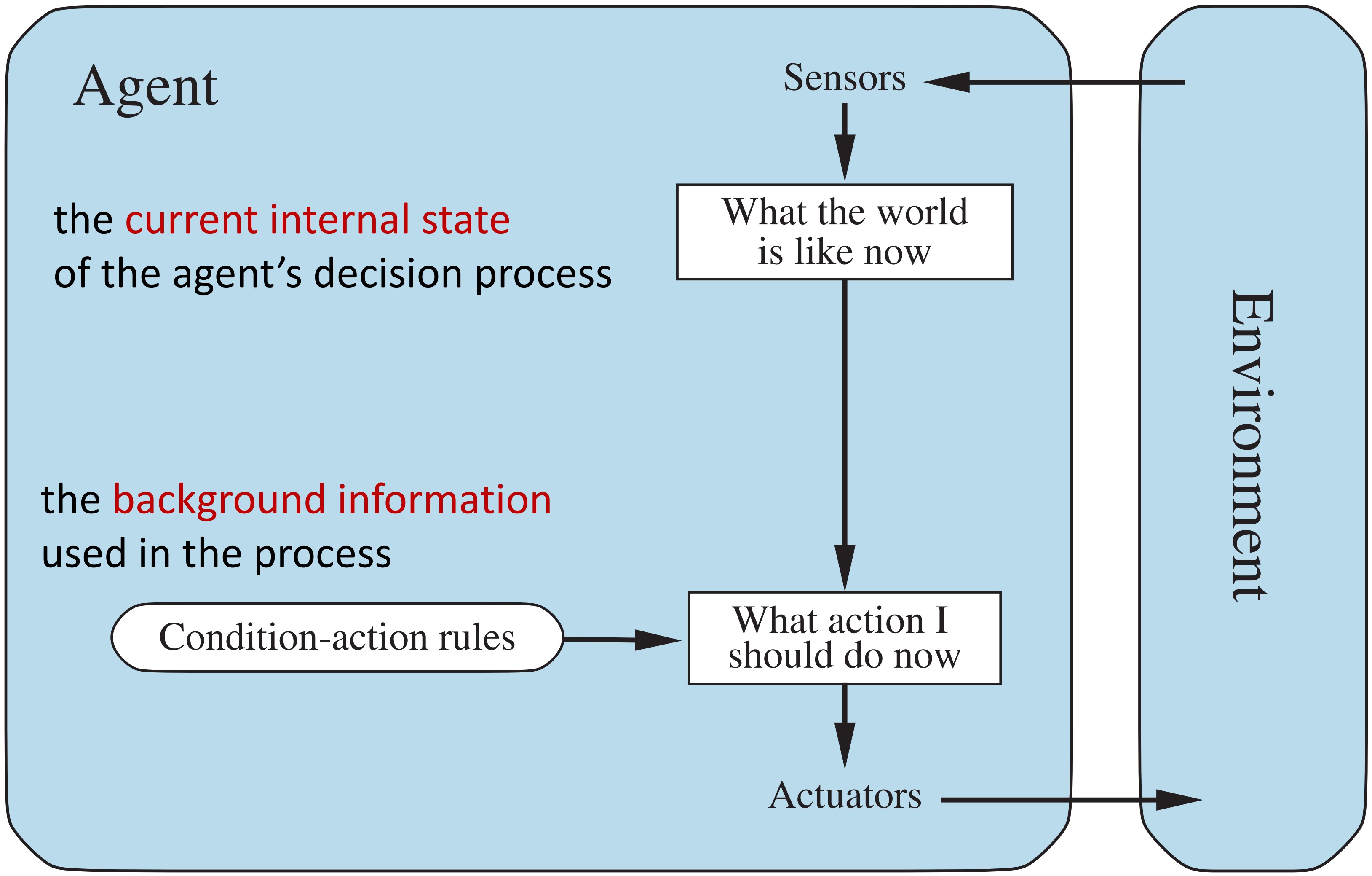
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Schematic diagram of a simple reflex agent



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

Schematic diagram of a simple reflex agent



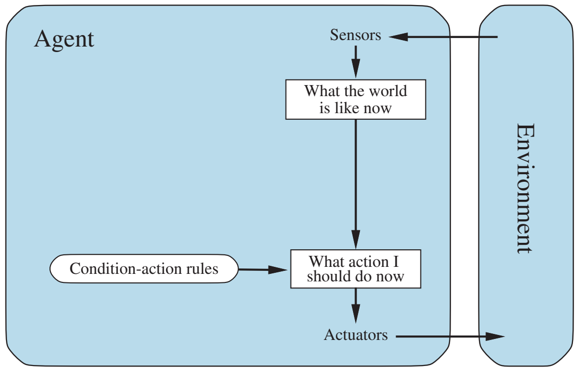
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A Simple Reflex Agent

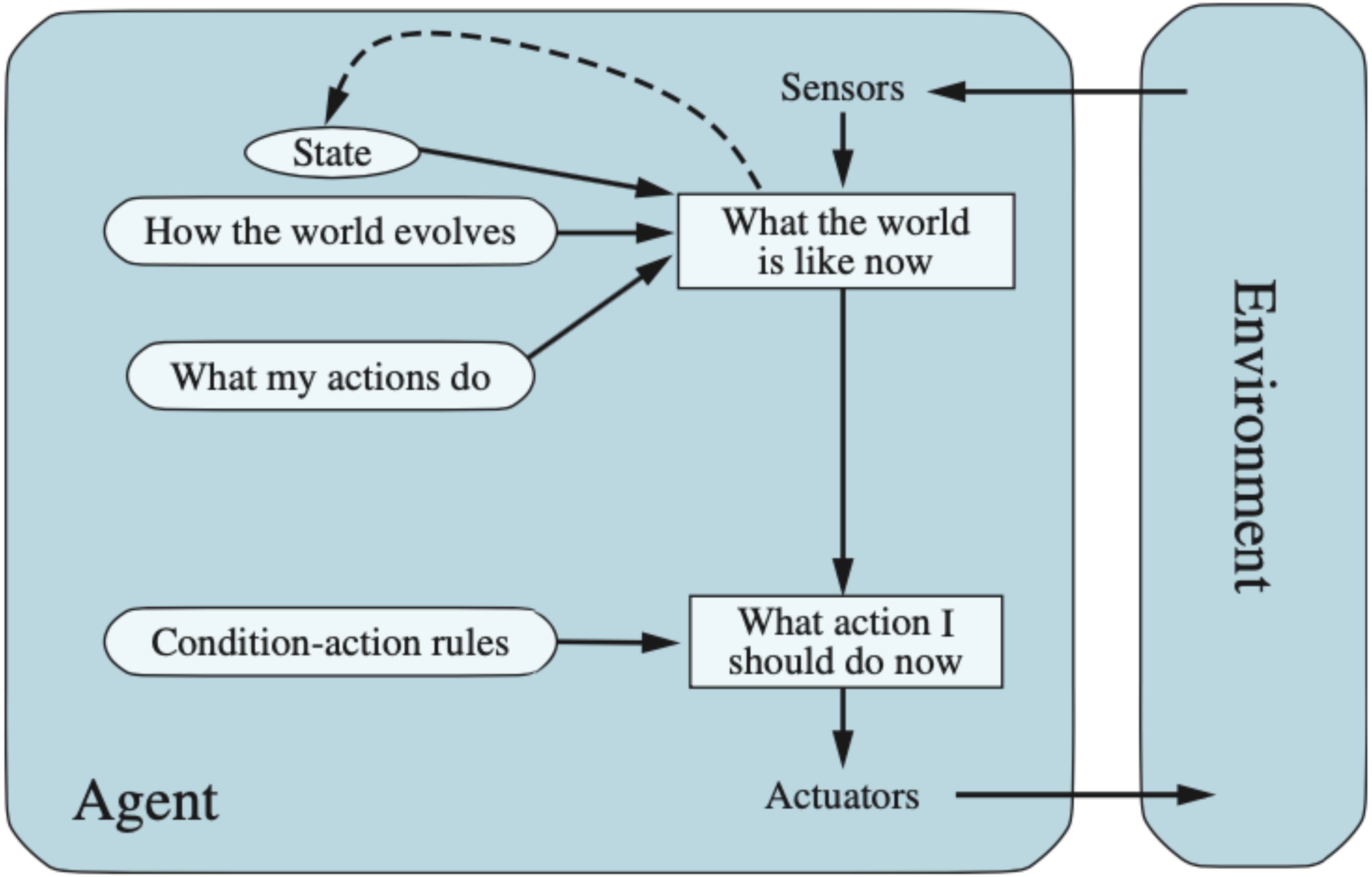
It acts according to a rule whose condition matches the current state, as defined by the percept.

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  $state \leftarrow$ INTERPRET-INPUT($percept$)
  $rule \leftarrow$ RULE-MATCH($state, rules$)
  $action \leftarrow rule.$ACTION
  return *action*



A Model-based Reflex Agent



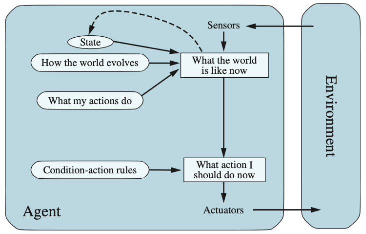
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A model-based reflex agent

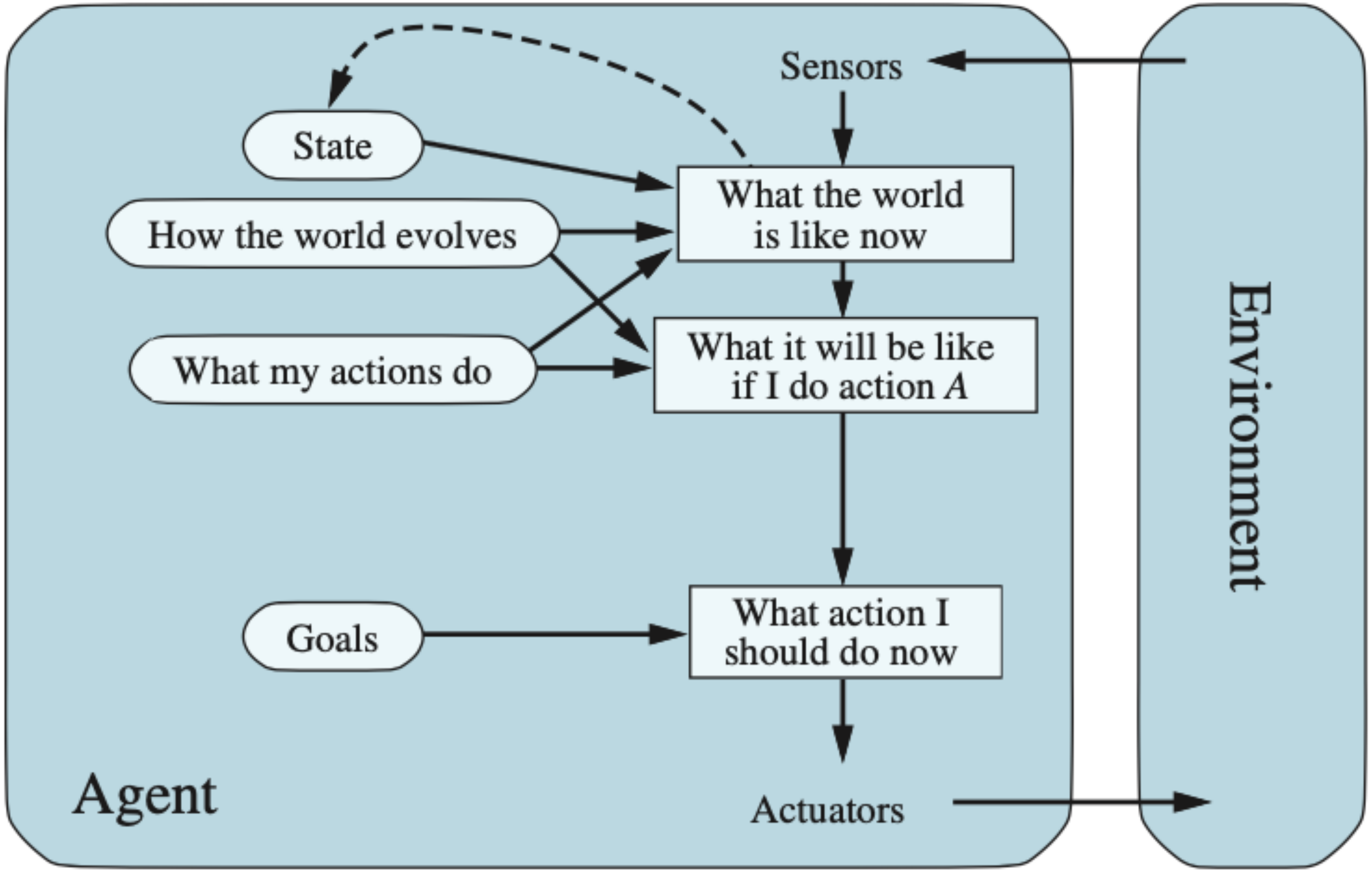
It keeps track of the current state of the world,
using an internal model.
It then chooses an action in the same way as the reflex agent.

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    transition_model, a description of how the next state depends on
      the current state and action
    sensor_model, a description of how the current world state is reflected
      in the agent's percepts
    rules, a set of condition–action rules
    action, the most recent action, initially none

  $state \leftarrow$ UPDATE-STATE($state, action, percept, transition\_model, sensor\_model$)
  $rule \leftarrow$ RULE-MATCH($state, rules$)
  $action \leftarrow rule$.ACTION
  return *action*

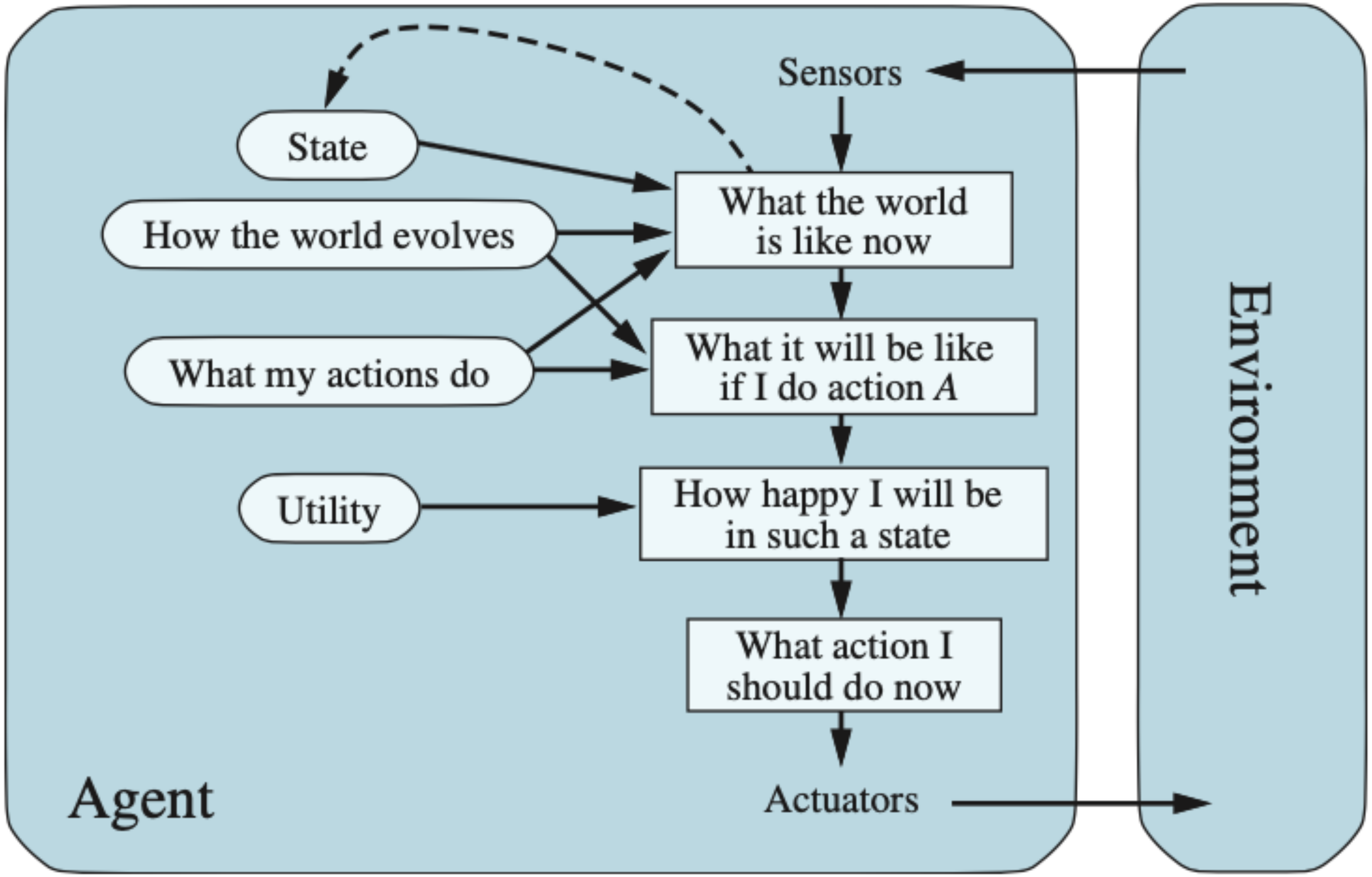


A model-based, goal-based agent



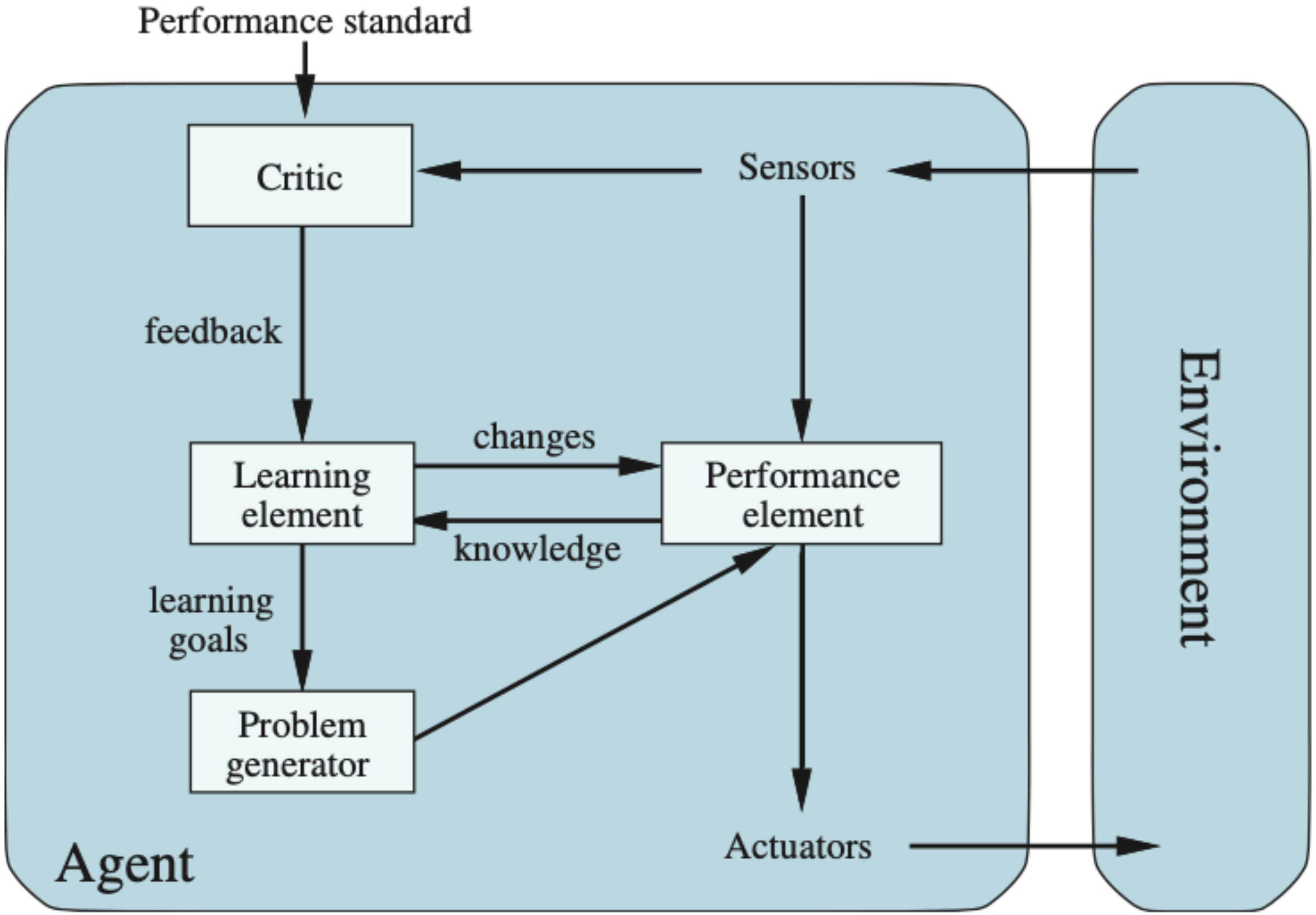
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A model-based, utility-based agent



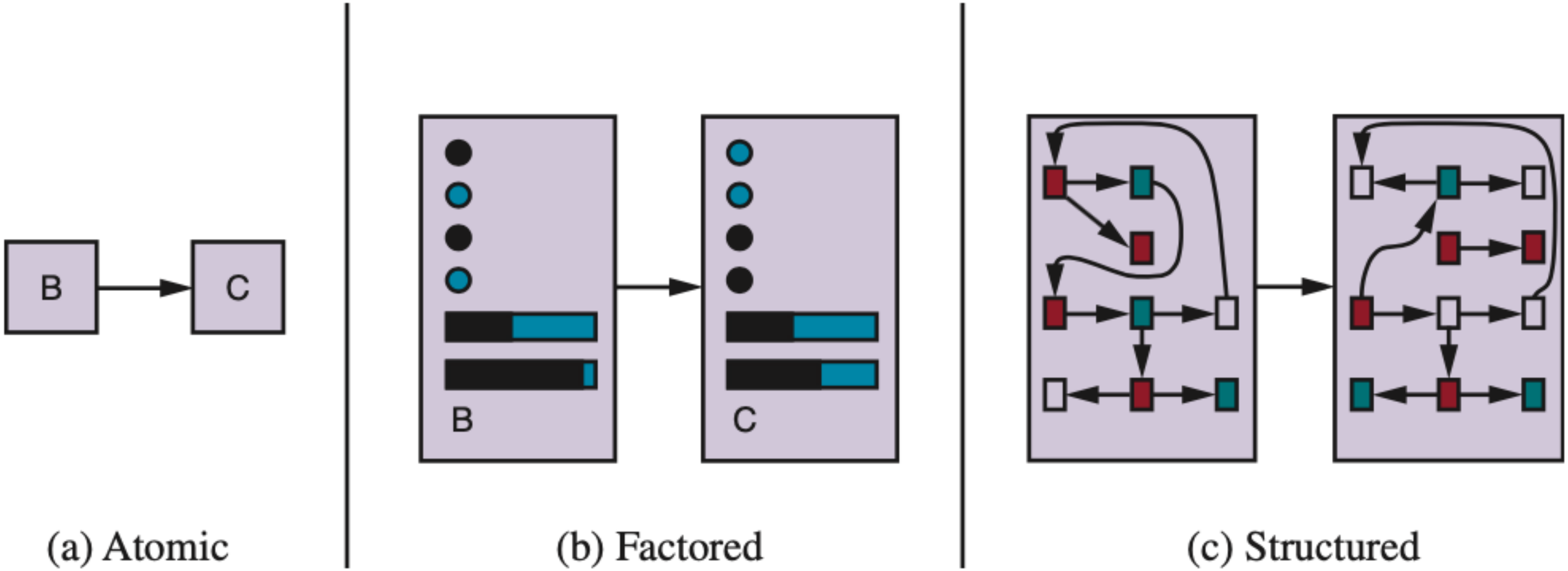
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A general learning agent



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

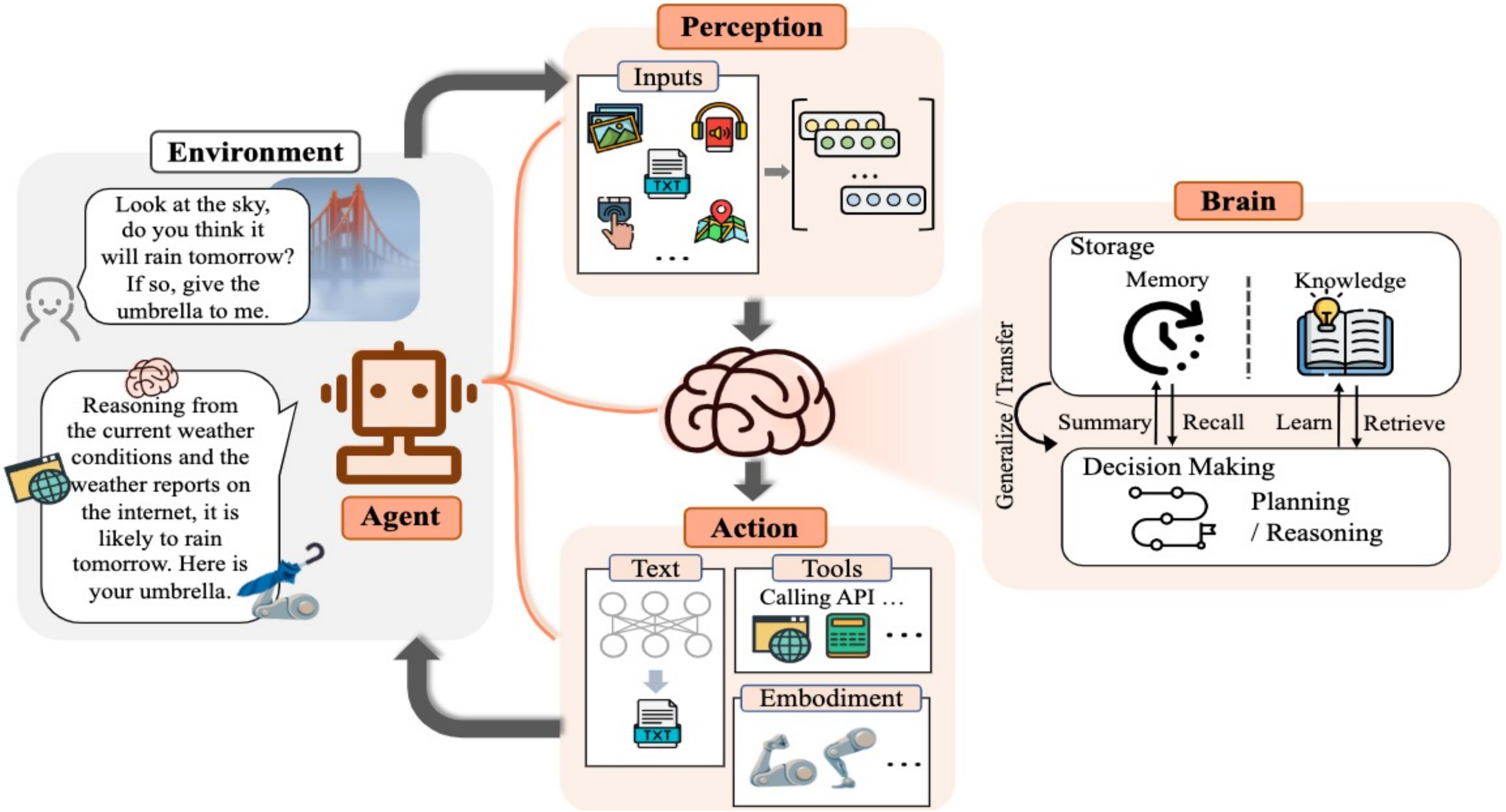
Three ways to represent states and the transitions between them



(a) Atomic (b) Factored (c) Structured

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

Large Language Model (LLM) based Agents

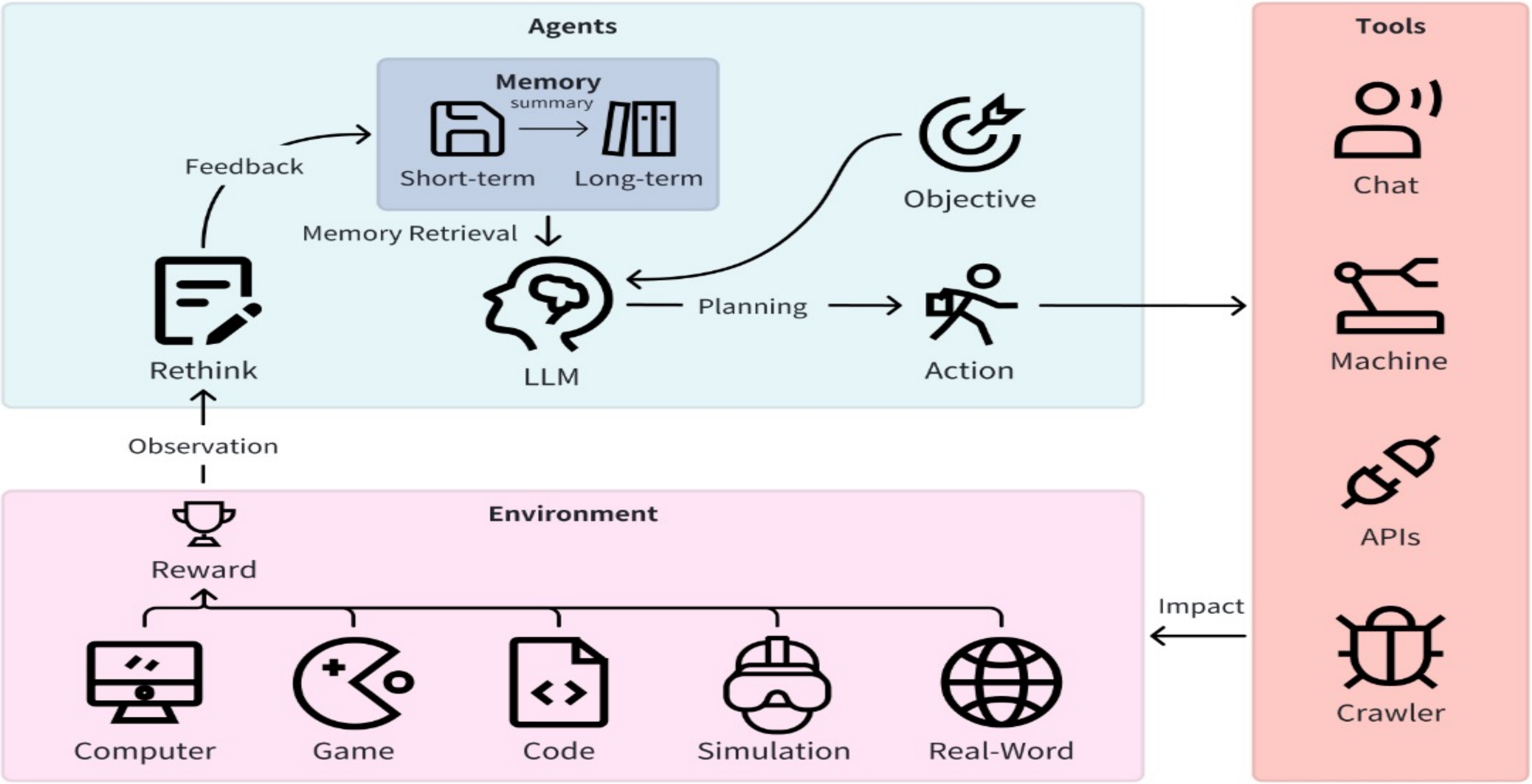


Source: Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., ... & Gui, T. (2023). The rise and potential of large language model based agents: A survey. arXiv preprint arXiv:2309.07864.

LLM-based Agents

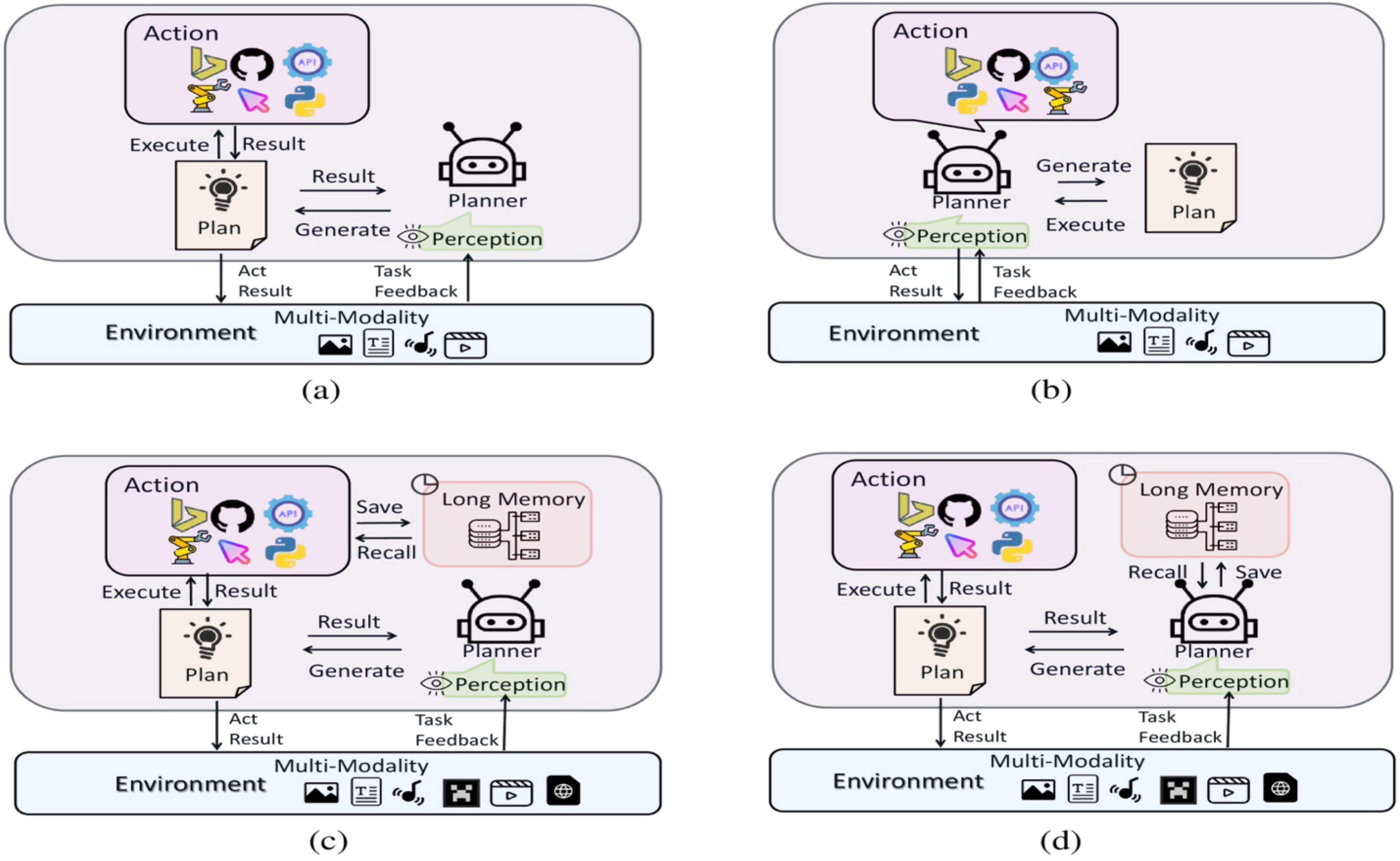
- **Definition: AI agents that use Large Language Models as their core decision-making mechanism**
- **Key Features:**
  - **Natural language interface**
  - **Vast knowledge base**
  - **Ability to understand context and nuance**
  - **Generalize to new tasks with minimal additional training**

LLM-based Agents



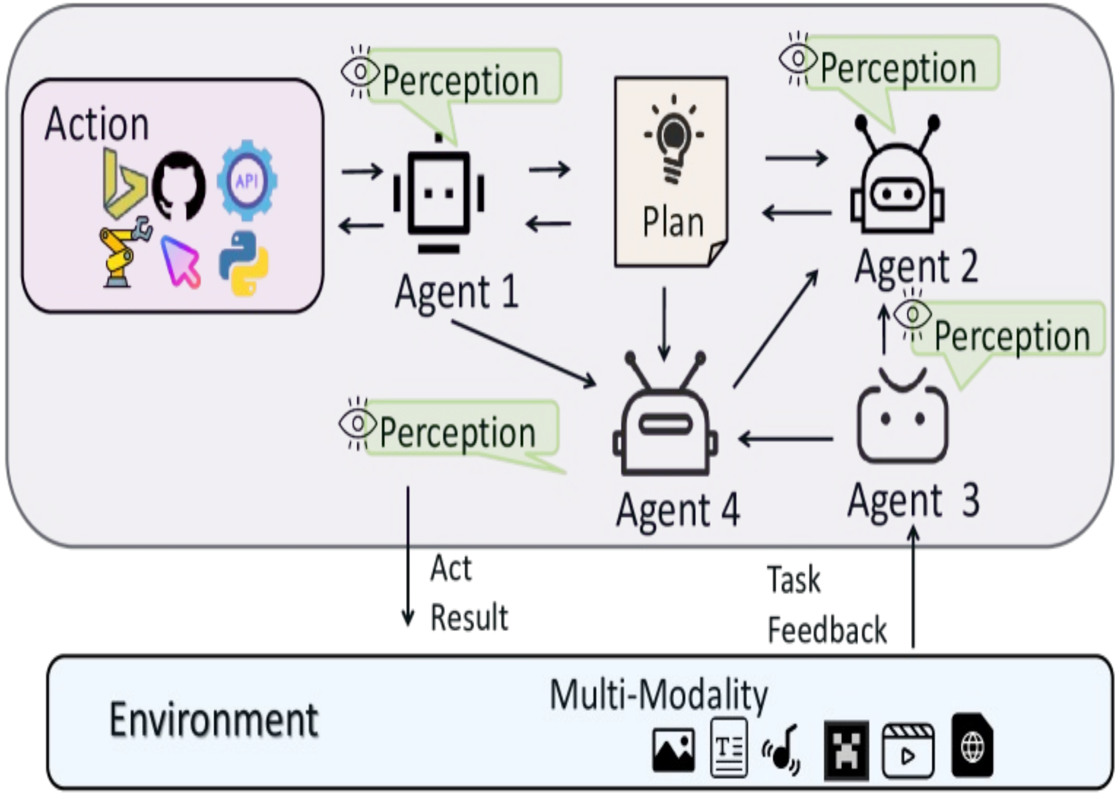
Source: Cheng, Yuheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang et al. "Exploring large language model based intelligent agents: Definitions, methods, and prospects." arXiv preprint arXiv:2401.03428 (2024).

Large Multimodal Agents (LMA)

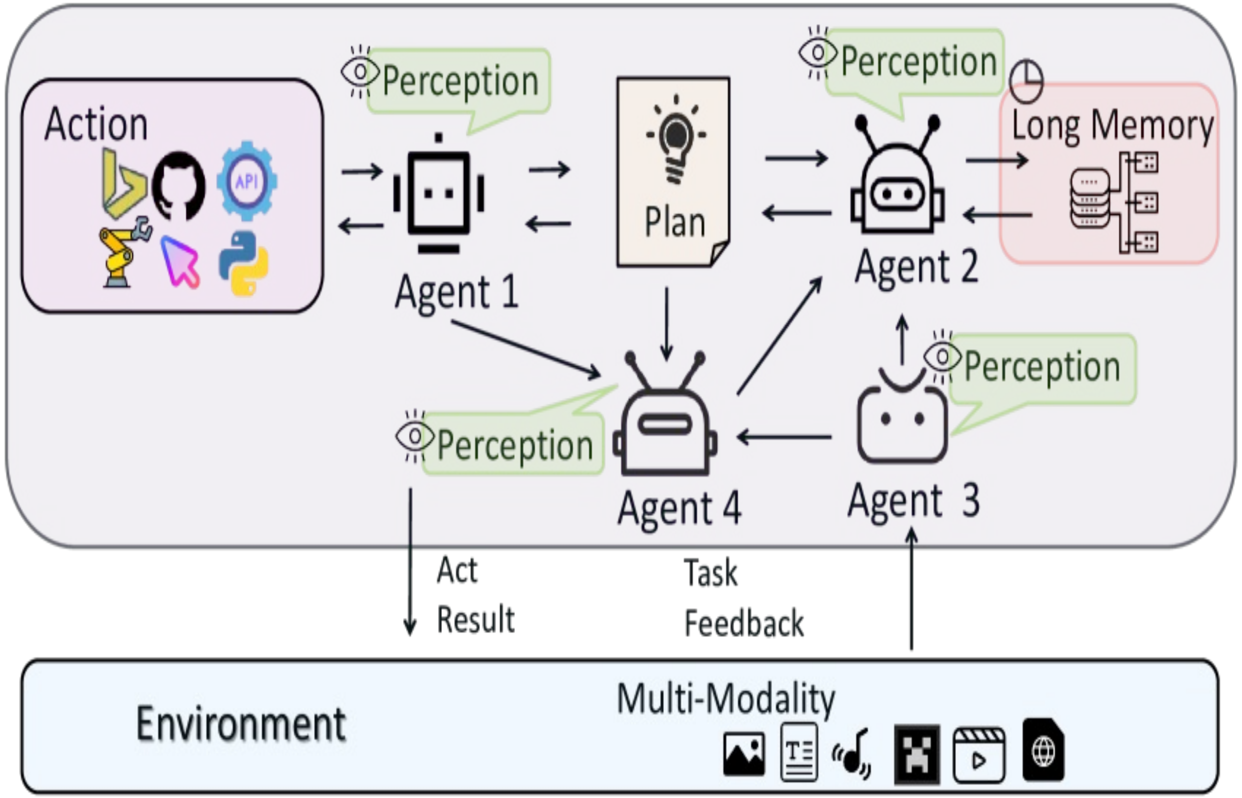


Source: Xie, J., Chen, Z., Zhang, R., Wan, X., & Li, G. (2024). Large Multimodal Agents: A Survey. ArXiv, abs/2402.15116.

Large Multimodal Agents (LMA)



(a)



(b)

Source: Xie, J., Chen, Z., Zhang, R., Wan, X., & Li, G. (2024). Large Multimodal Agents: A Survey. ArXiv, abs/2402.15116.

A2A

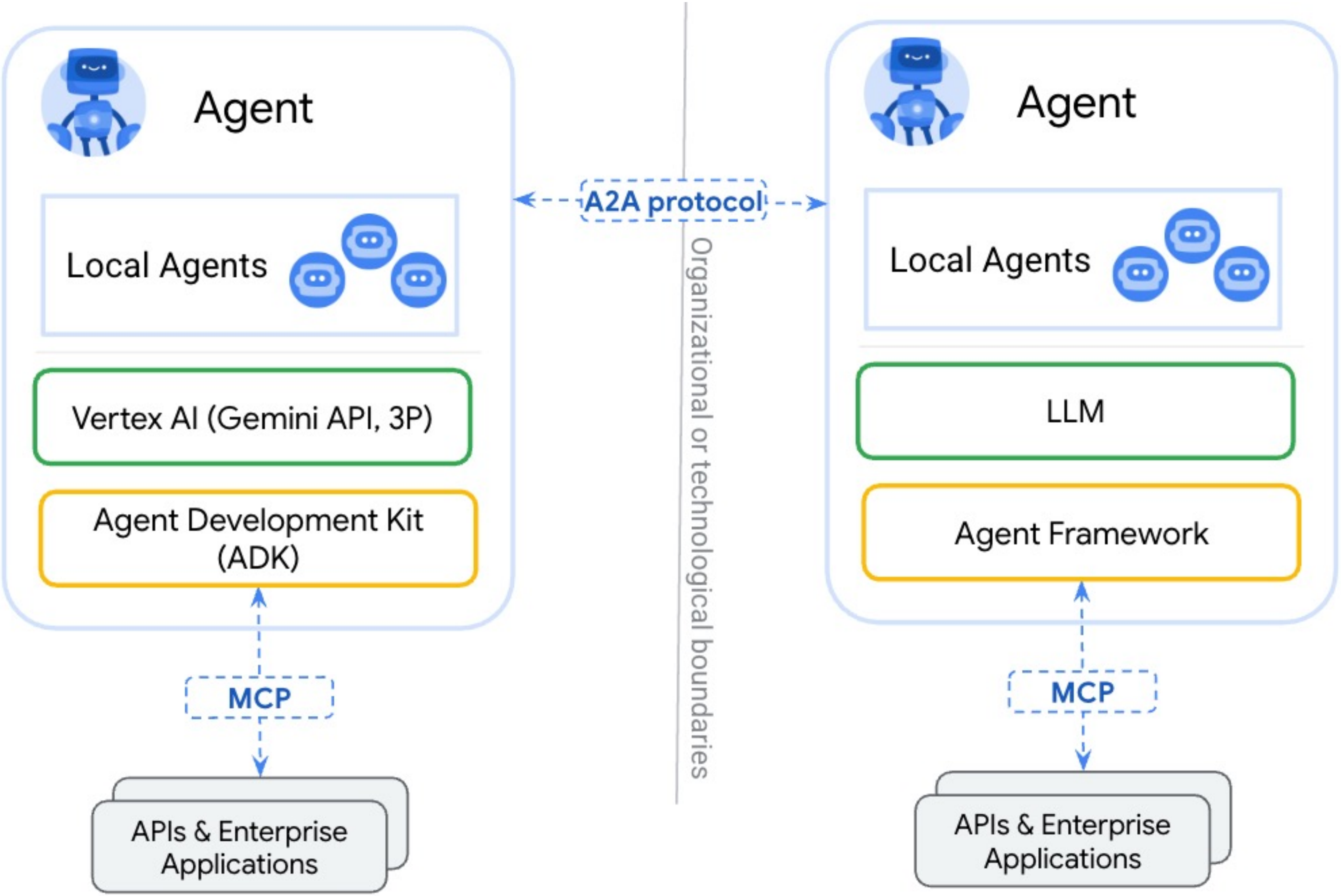
(Agent2Agent Protocol)

for agent-agent collaboration

MCP

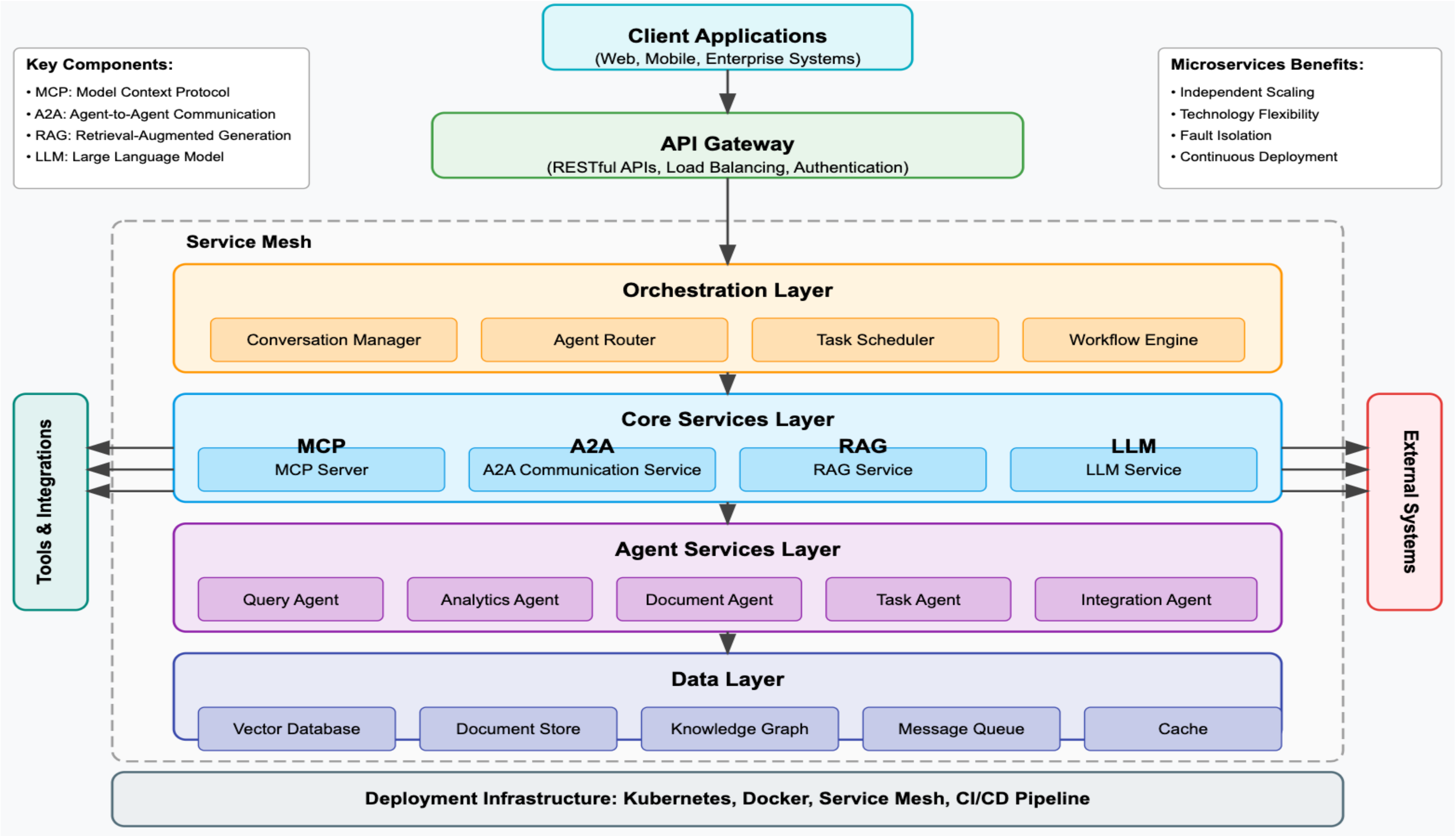
(Model Context Protocol)

for tools and resources

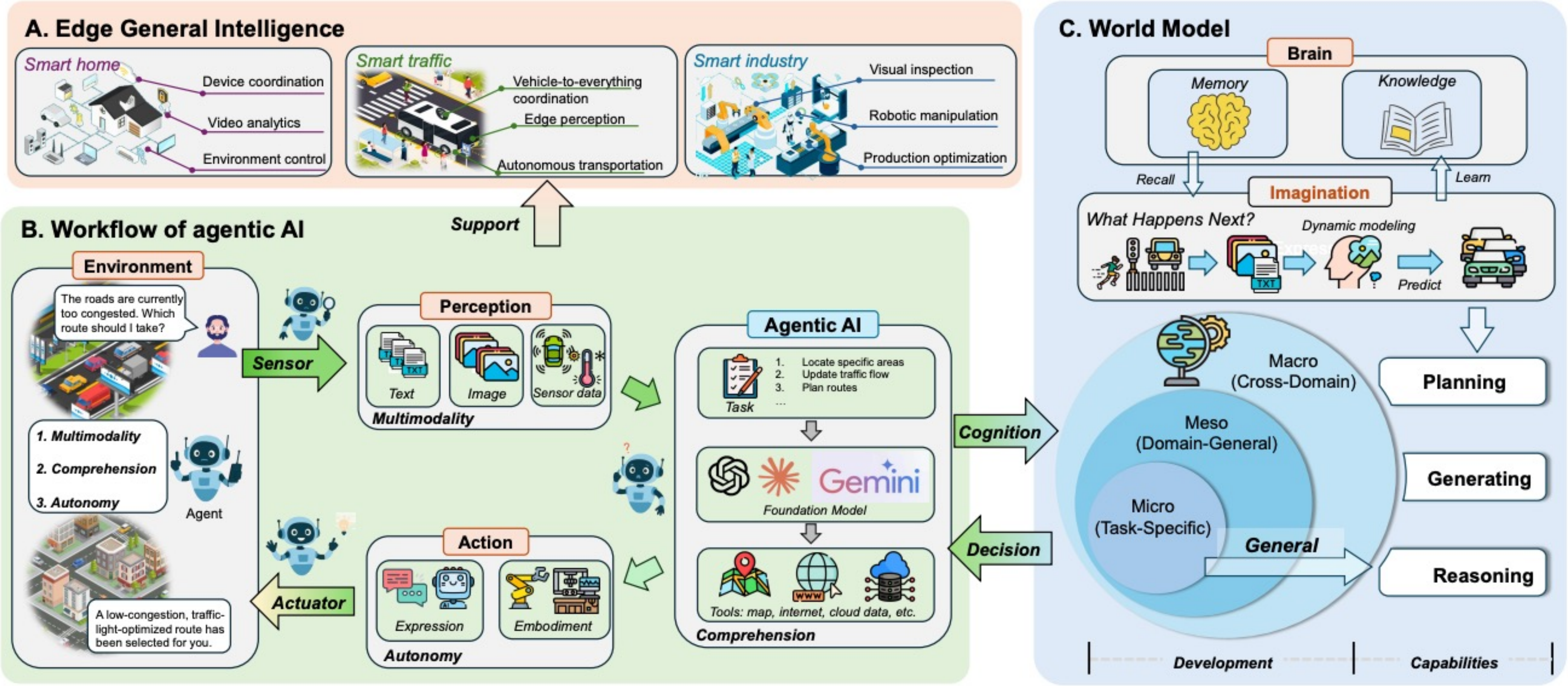


Source: https://google.github.io/A2A/

Agentic AI System with Microservices Architecture



Agentic AI and World Model for Edge General Intelligence



Source: Changyuan Zhao, Guangyuan Liu, Ruichen Zhang, Yinqiu Liu, Jiacheng Wang, Jiawen Kang, Dusit Niyato et al (2025), "Edge general intelligence through world models and agentic AI: Fundamentals, solutions, and challenges." arXiv preprint arXiv:2508.09561.

Artificial Intelligence

Problem Solving

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:
2. Problem Solving

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

LLM-enhanced Problem Solving

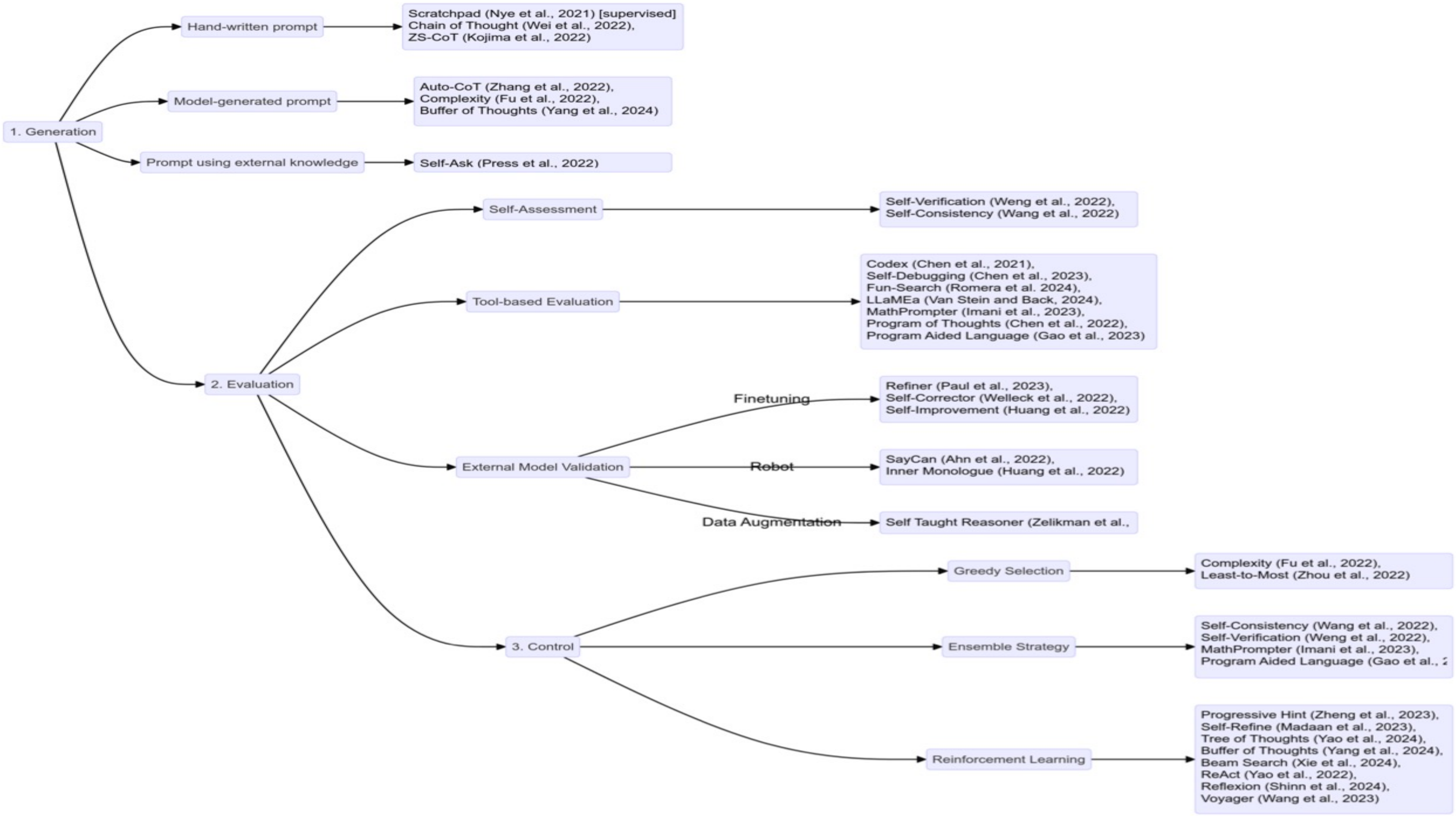
Problem Solving and LLM-enhanced Techniques

- Traditional Search Algorithms
  - Breadth-first Search (BFS)
  - Depth-first Search (DFS)
  - A* Search
- LLM-enhanced Problem Solving
  - Chain-of-thought (CoT) prompting
  - Few-shot learning for problem decomposition
  - Integration with external tools and APIs

LLM-based Agents for Complex Problem Solving

- **ReAct: Reasoning and Acting in Language Models**
- **MRKL (Modular Reasoning, Knowledge and Language) systems**
- **LLM-powered planning and decision making**
- **Chain-of-thought Prompting**
    - **Solving a complex problem using chain-of-thought prompting**

LLM-Reasoning Approaches: Prompt Generation, Evaluation, and Control



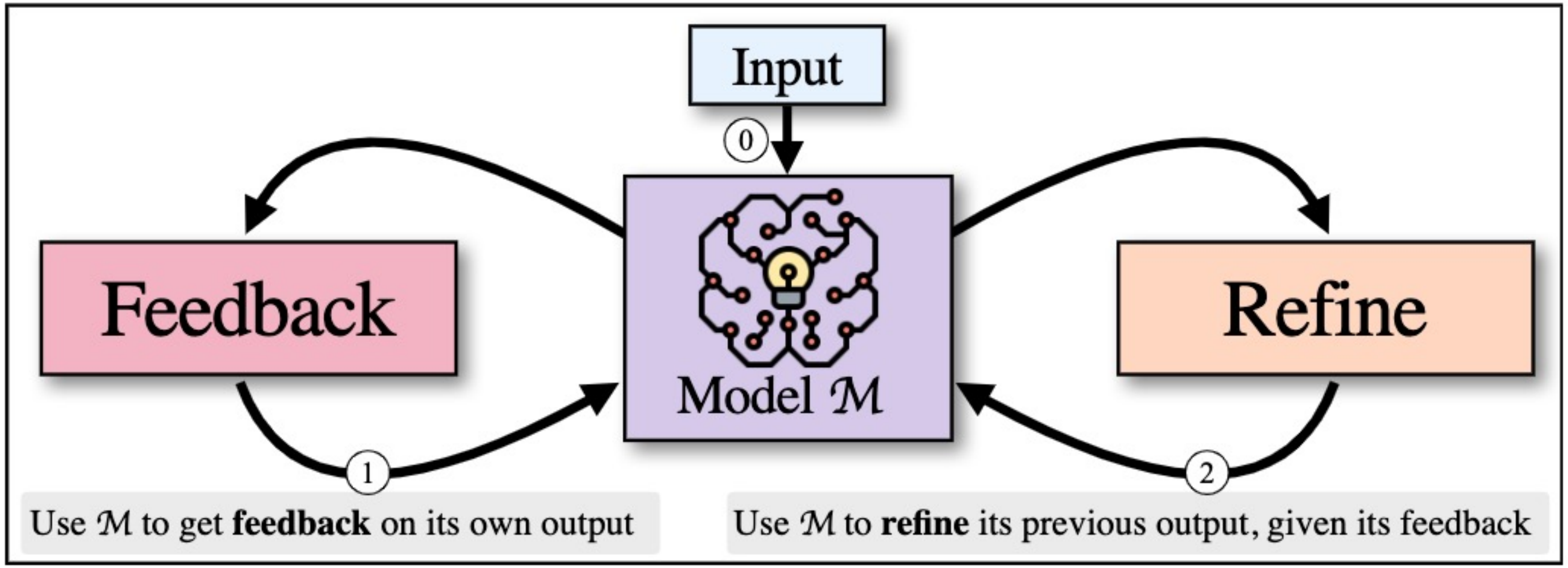
Source: Plaat, A., Wong, A., Verberne, S., Broekens, J., van Stein, N., & Back, T. (2024). Reasoning with Large Language Models, a Survey. arXiv preprint arXiv:2407.11511.

LLM-Reasoning Approaches: Prompt Generation, Evaluation, and Control

Approach	Domain	Step generation	Step evaluation	Step control
Scratchpad [Nye et al., 2021]	math word	**hand-wr/supervised**	-	greedy/1prompt
Chain-of-thought [Wei et al., 2022b]	math word	**hand-written**	-	greedy/1prompt
ZS-CoT [Kojima et al., 2022]	math word	**hand-written**	-	greedy/1prompt
Auto-CoT [Zhang et al., 2022]	math word	**model-generated**	-	clustering
Complexity [Fu et al., 2022]	math word	**hand-written**	self-consistency	greedy/1prompt
Self-ask [Press et al., 2022]	math word	**external knowledge**	LLM	multi-hop questions
Self-verification [Weng et al., 2022]	math word	hand-written	**back-verify**	ensemble
Self-consistency [Wang et al., 2022b]	math word	hand-written	**majority**	ensemble
Codex [Chen et al., 2021]	code	-	**tool-based**	-
Self-debugging [Chen et al., 2023]	code	hand-written	**tool-based**	greedy
Fun-search [Romera-Paredes et al., 2024]	code	hand-written	**tool-based**	evolutionary algorithm
LLaMEa [van Stein and Bäck, 2024]	code	hand-written	**tool-based**	evolutionary algorithm
MathPrompter [Imani et al., 2023]	math	hand-written	**tool-based**	ensemble
Program-of-thoughts [Chen et al., 2022]	math word	hand-written, Codex	**Python+Consist.**	decouple reason/compute
Program-aided-language [Gao et al., 2023]	math word	hand-written, Codex	**NLP/Python**	ensemble
Refiner [Paul et al., 2023]	math word	finetune	**critic model**	gen/crit feedback
Self-corrector [Welleck et al., 2022]	math word	finetune	**corrector model**	gen/corr feedback
Self-improvement [Huang et al., 2022a]	math word	finetune	**self-assessment**	CoT/consistency
Say-can [Ahn et al., 2022]	robot	model-generated	**external model**	greedy
Inner-monologue [Huang et al., 2022b]	robot	hand-written	**various**	greedy
Self-taught-reasoner [Zelikman et al., 2022]	math word	finetune	**augmentation**	greedy/feedback
Least-to-most [Zhou et al., 2022]	math word	hand-written	self-assessment	**curriculum**
Progressive-hint [Zheng et al., 2023]	math word	model-generated	self-assessment	**stable prompt**
Self-refine [Madaan et al., 2023]	math word	model-generated	self-assessment	**greedy/feedback**
Tree-of-thoughts [Yao et al., 2024]	puzzles	model-generated	self-assessment	**BFS/DFS**
Buffer-of-thoughts [Yang et al., 2024]	math word	thought template	self-assessment	**buffer manager**
Beam-search [Xie et al., 2024]	math word	model-generated	self-assessment	**Beam Search**
ReAct [Yao et al., 2022]	action	external knowledge	self-assessment	**reinforcement learning**
Reflexion [Shinn et al., 2024]	decision	model-generated	ext model	**reinforcement learning**
Voyager [Wang et al., 2023]	Minecraft	model-generated	Minecraft	**reinforcement learning**

Source: Plaat, A., Wong, A., Verberne, S., Broekens, J., van Stein, N., & Back, T. (2024). Reasoning with Large Language Models, a Survey. arXiv preprint arXiv:2407.11511.

Self-Refine: Iterative refinement with self-feedback



(a) **Dialogue:** x, $\mathbf{y_t}$	(b) FEEDBACK fb	(c) REFINE $\mathbf{y_{t+1}}$

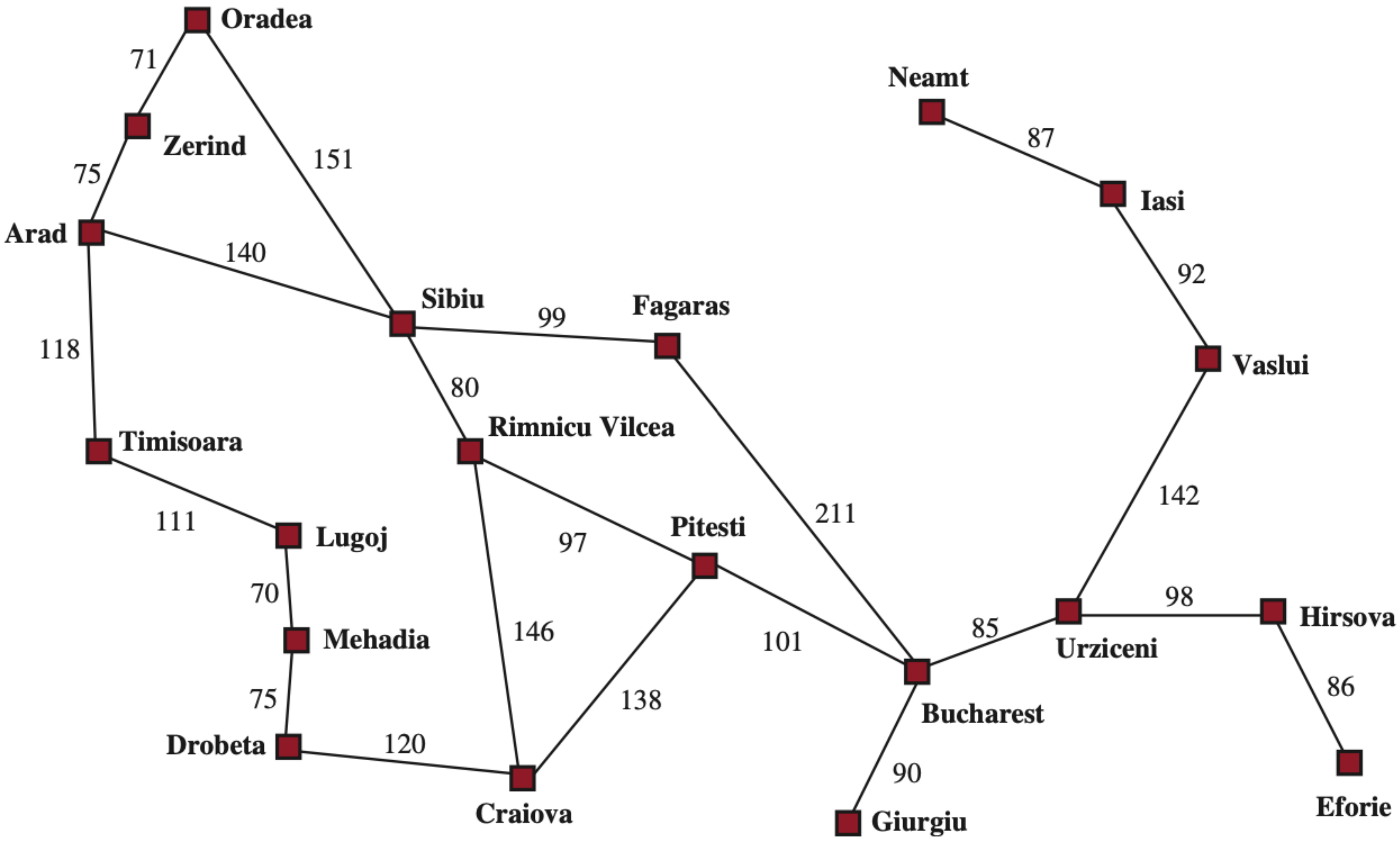
`User: I am interested in playing Table tennis.`
 	
`Response: I'm sure it's a great way to socialize, stay active`
	`Engaging: Provides no information about table tennis or how to play it.`
	
	`User understanding: Lacks understanding of user's needs and state of mind.`
		`Response (refined): That's great to hear (...) ! It's a fun sport requiring quick reflexes and good hand-eye coordination. Have you played before, or are you looking to learn?`

Source: Madaan, Aman, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon et al. "Self-refine: Iterative refinement with self-feedback." Advances in Neural Information Processing Systems 36 (2024).

Solving Problems by Searching

AI: Solving Problems by Searching

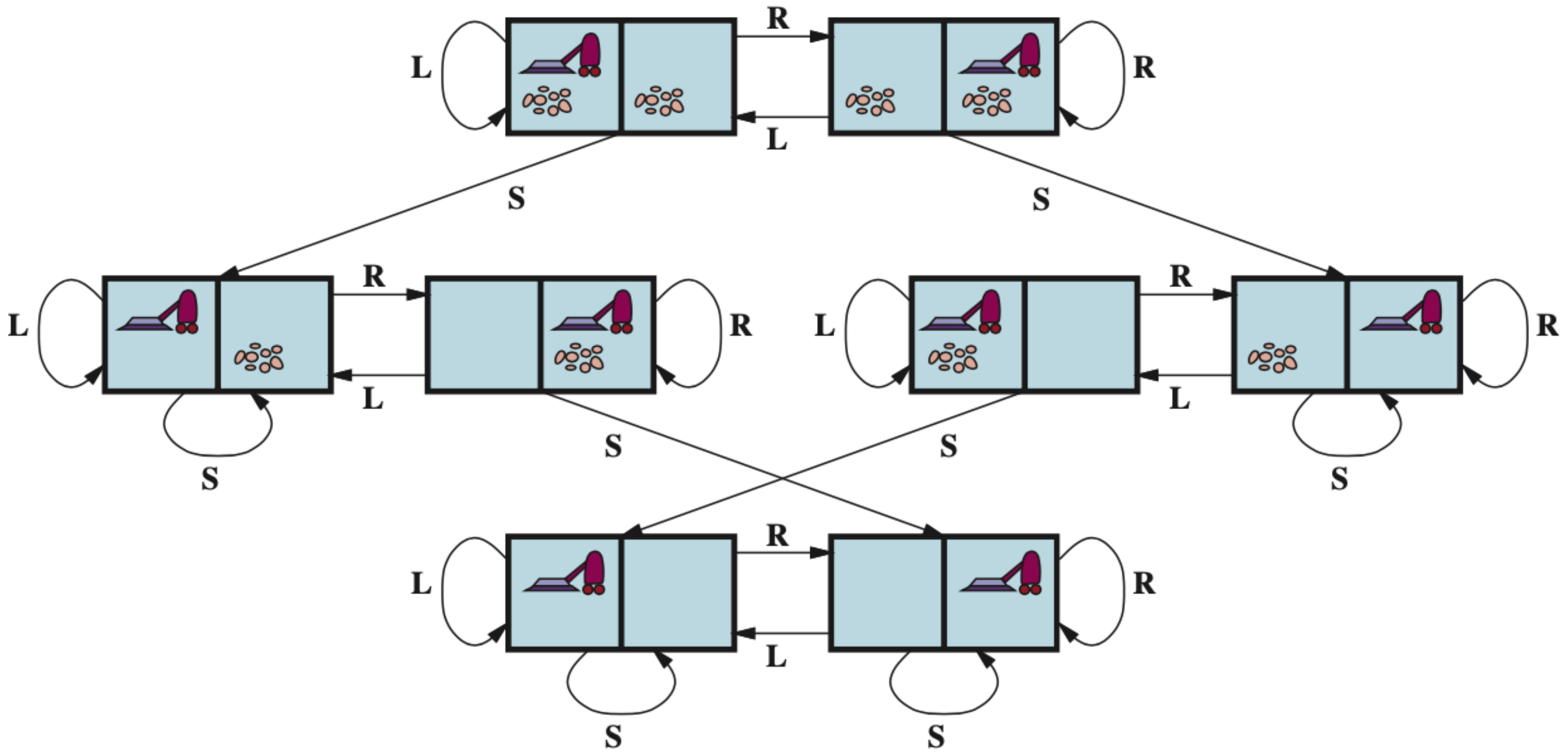
A simplified road map of part of Romania, with road distances in miles.



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

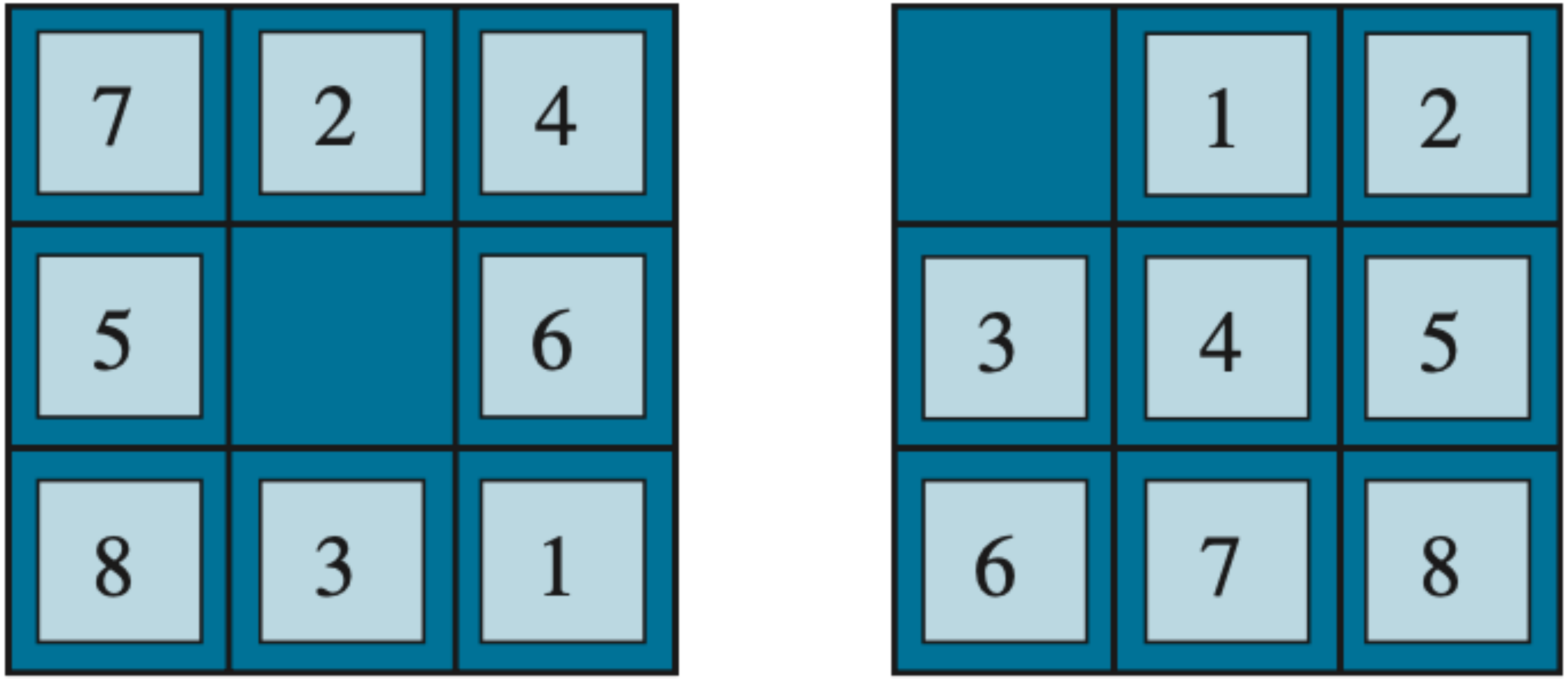
The state-space graph for the two-cell vacuum world

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



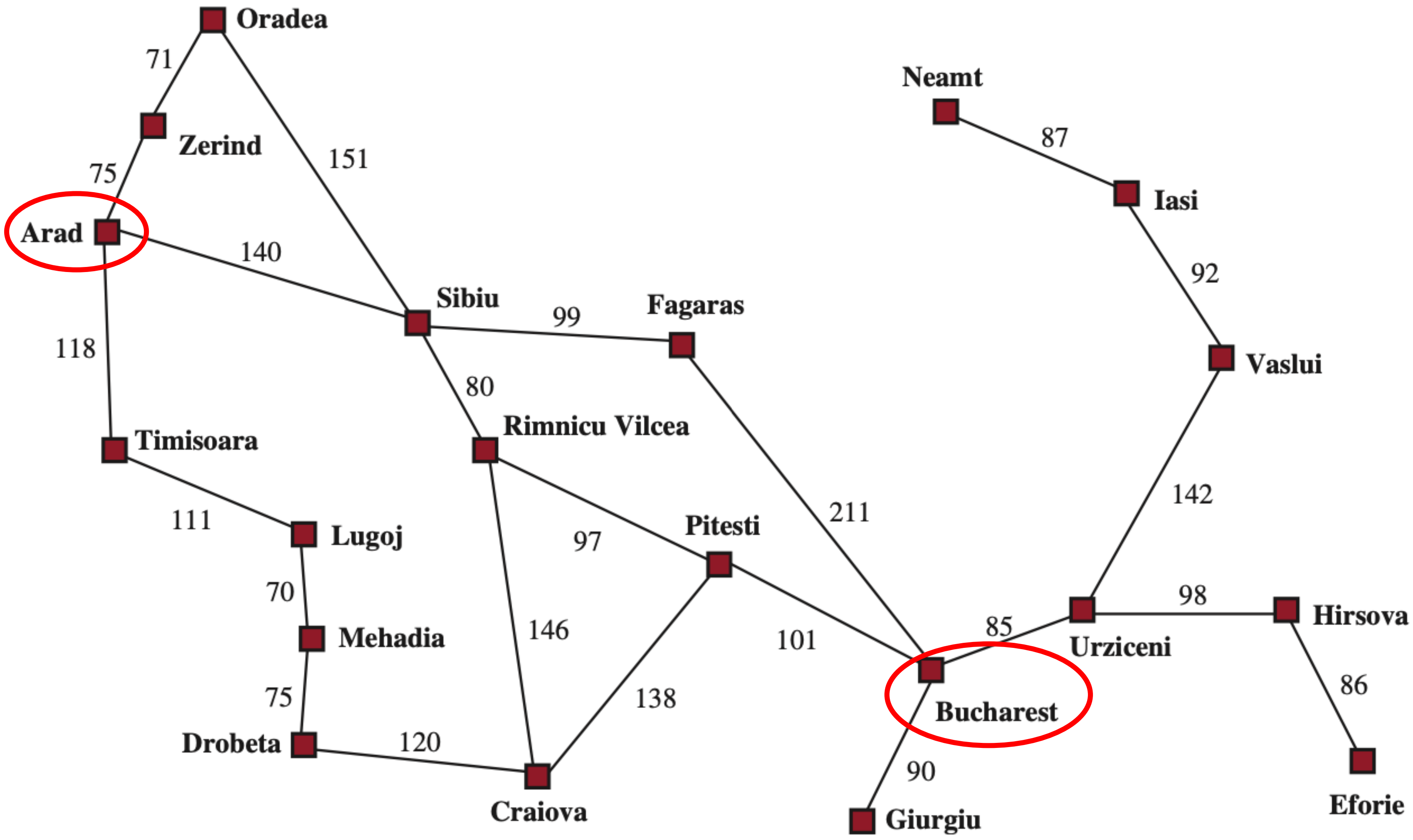
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A typical instance of the 8-puzzle



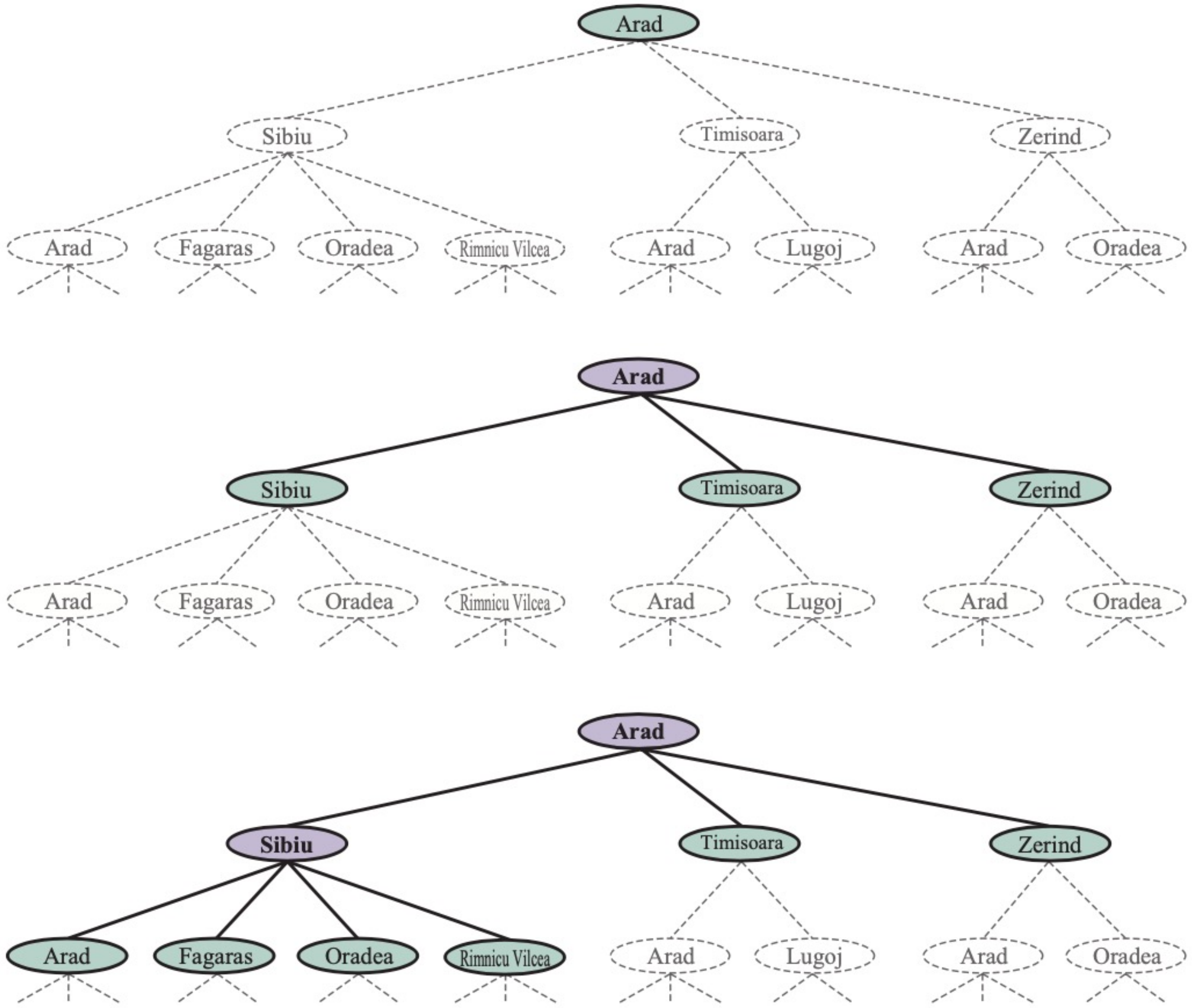
Start State Goal State

Arad to Bucharest



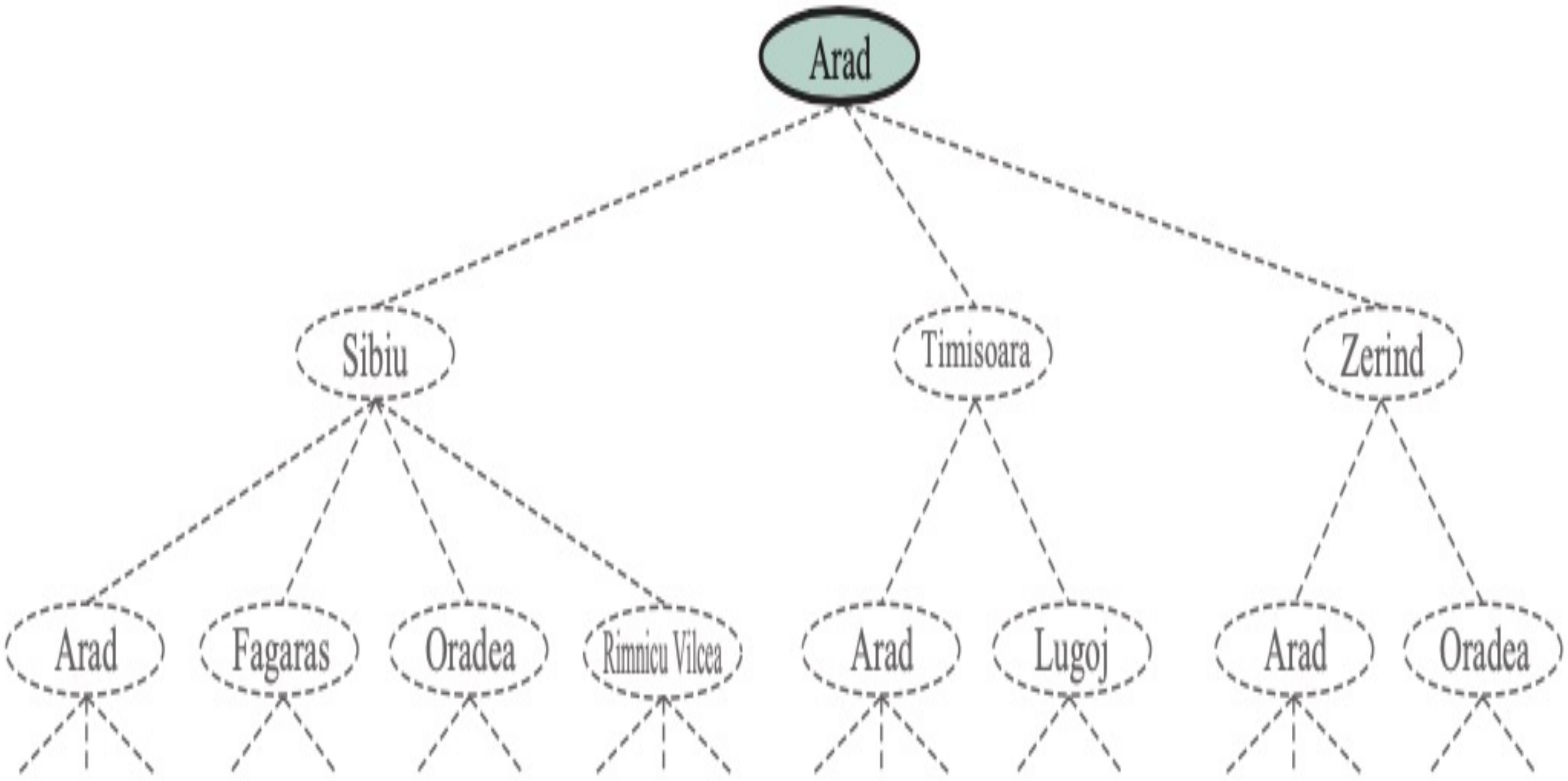
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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



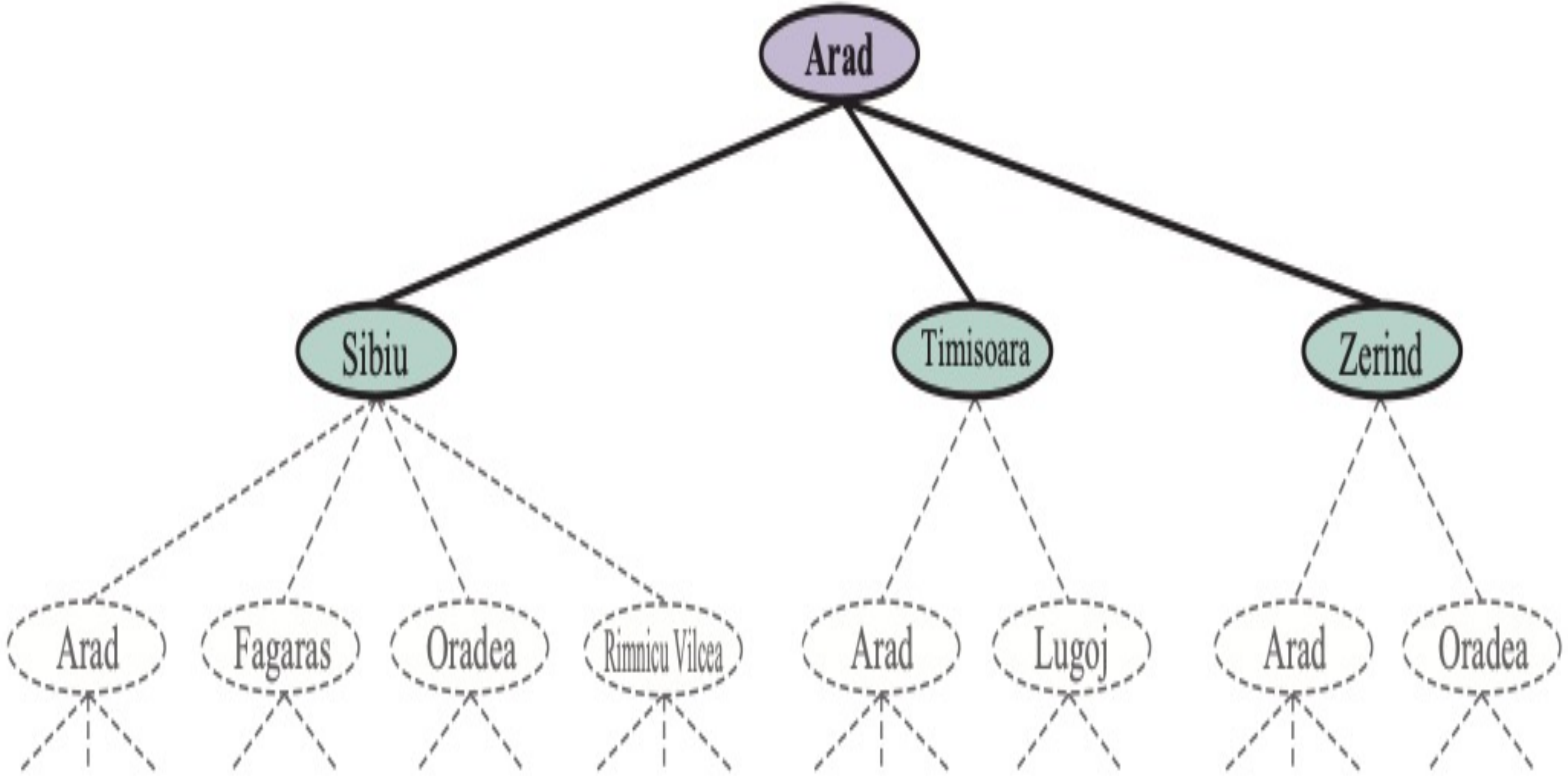
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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



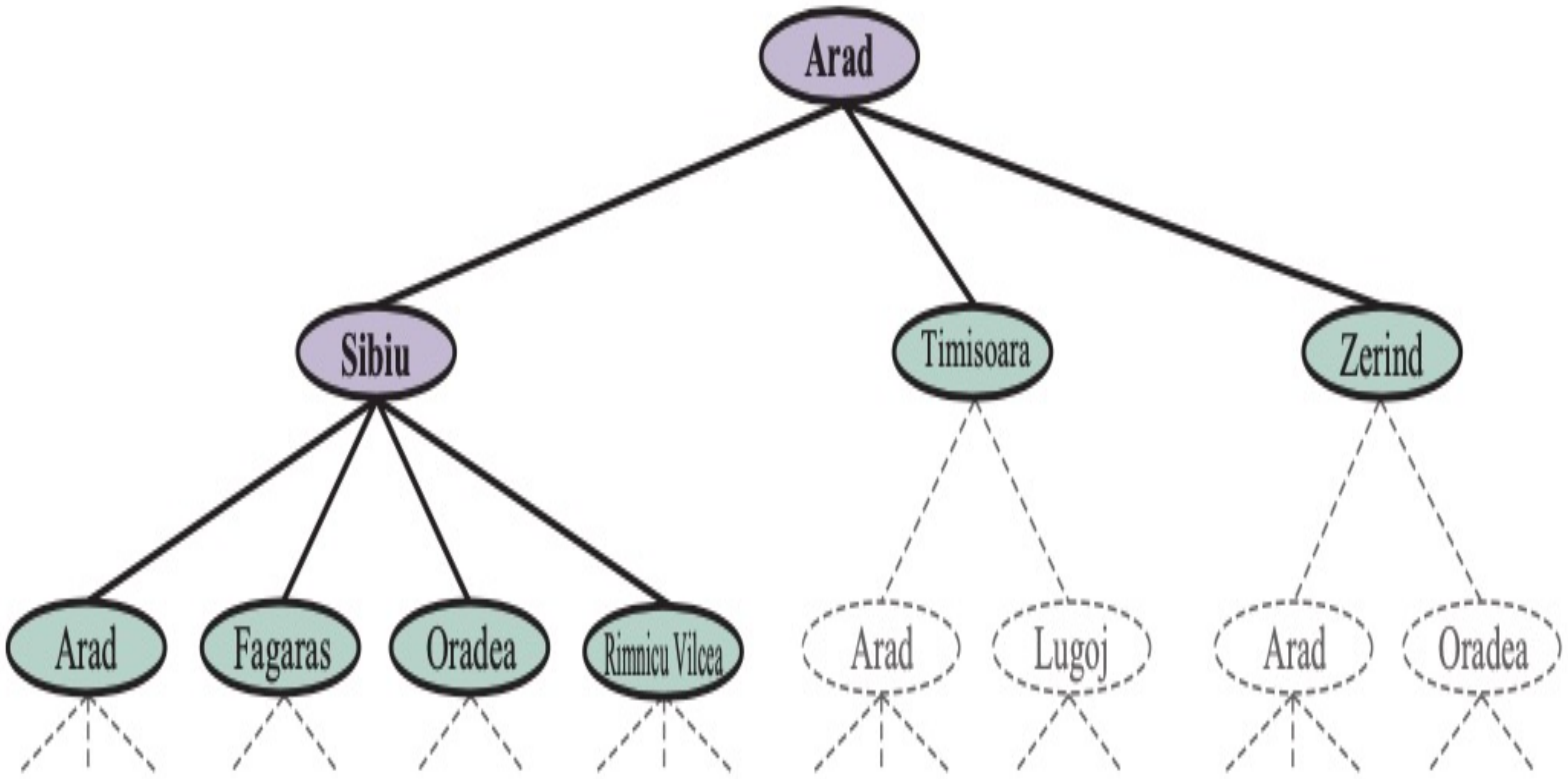
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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



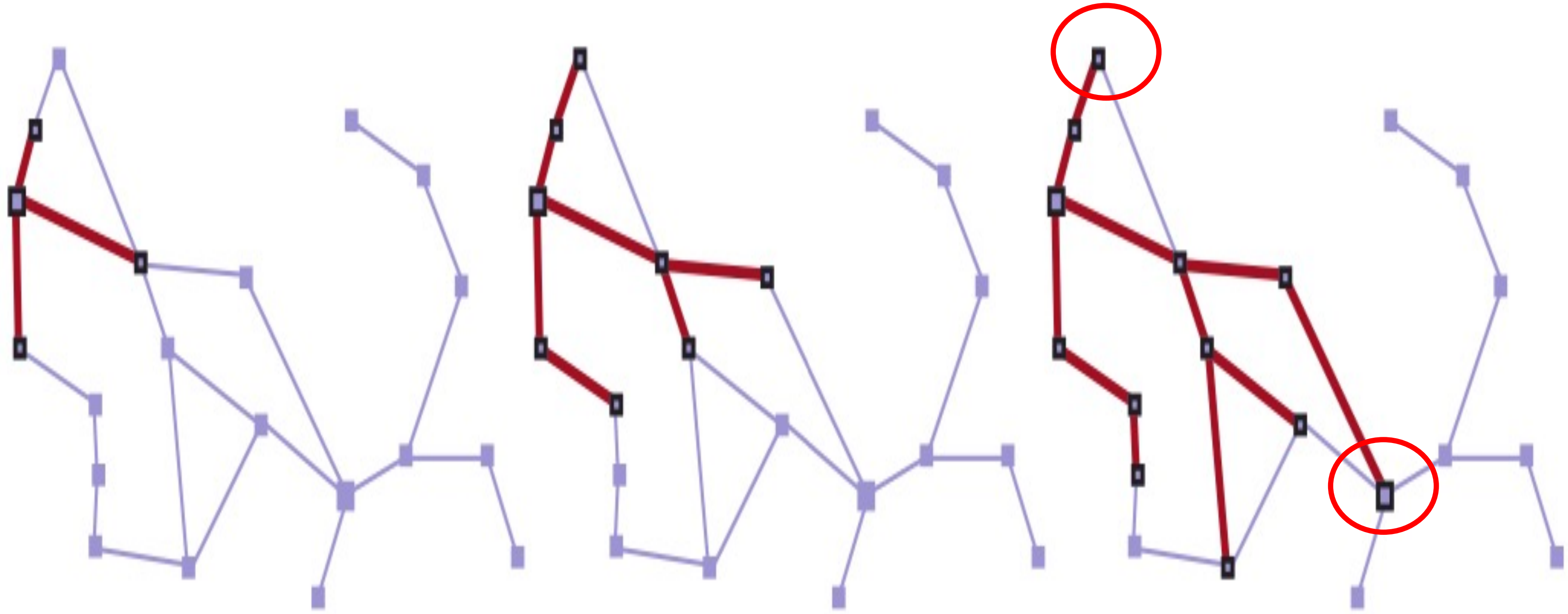
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

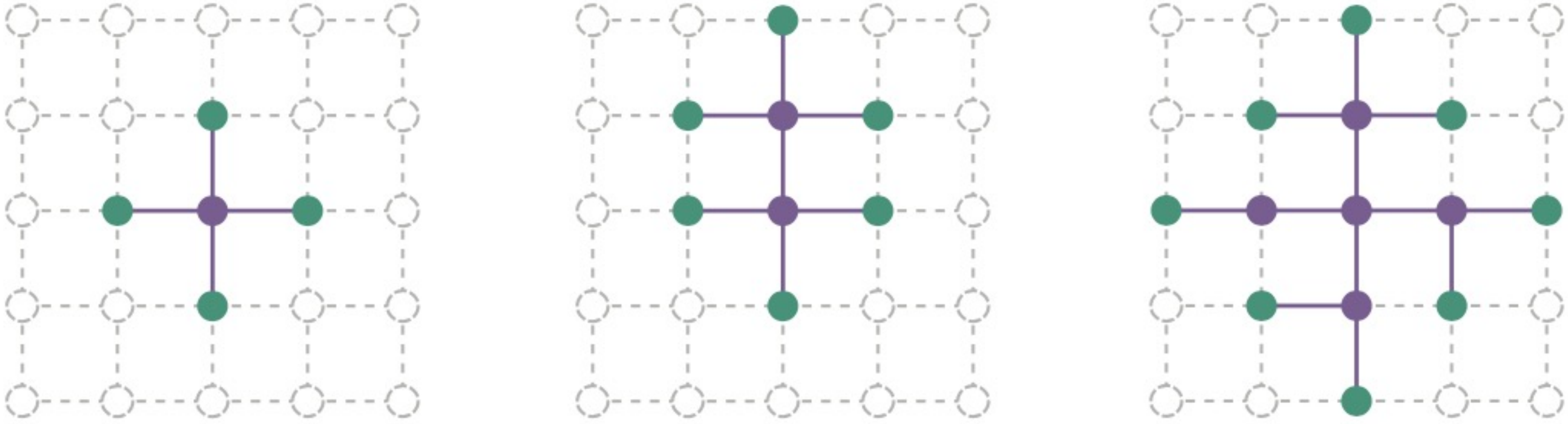


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

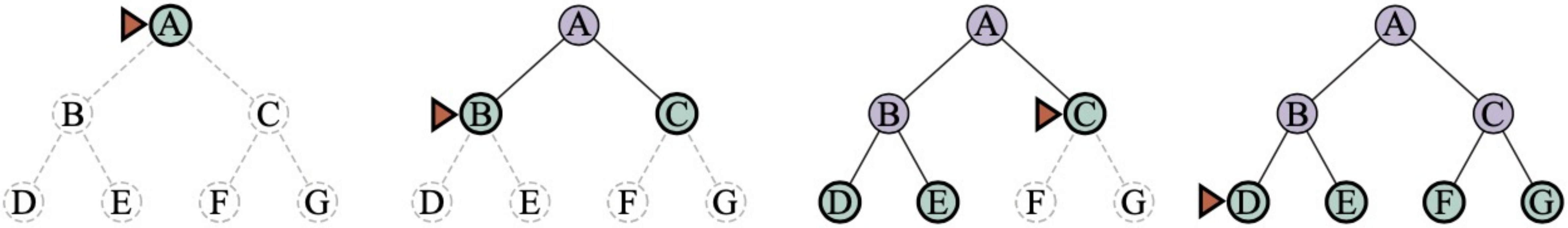
The Best-First Search (BFS) Algorithm

```
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)
```

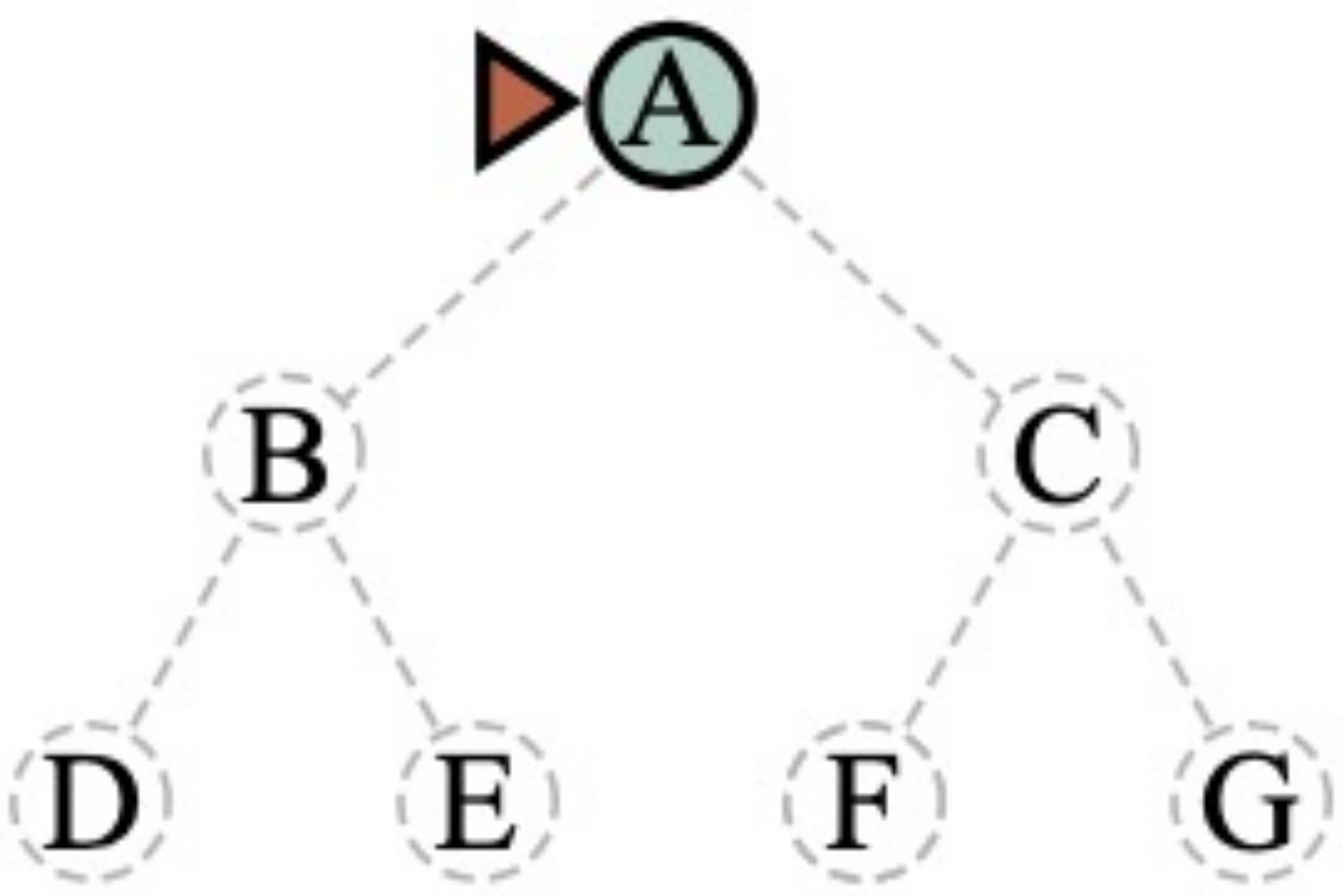
Breadth-First Search on a Simple Binary Tree

Bread-First Search (BFS)



Breadth-First Search on a Simple Binary Tree

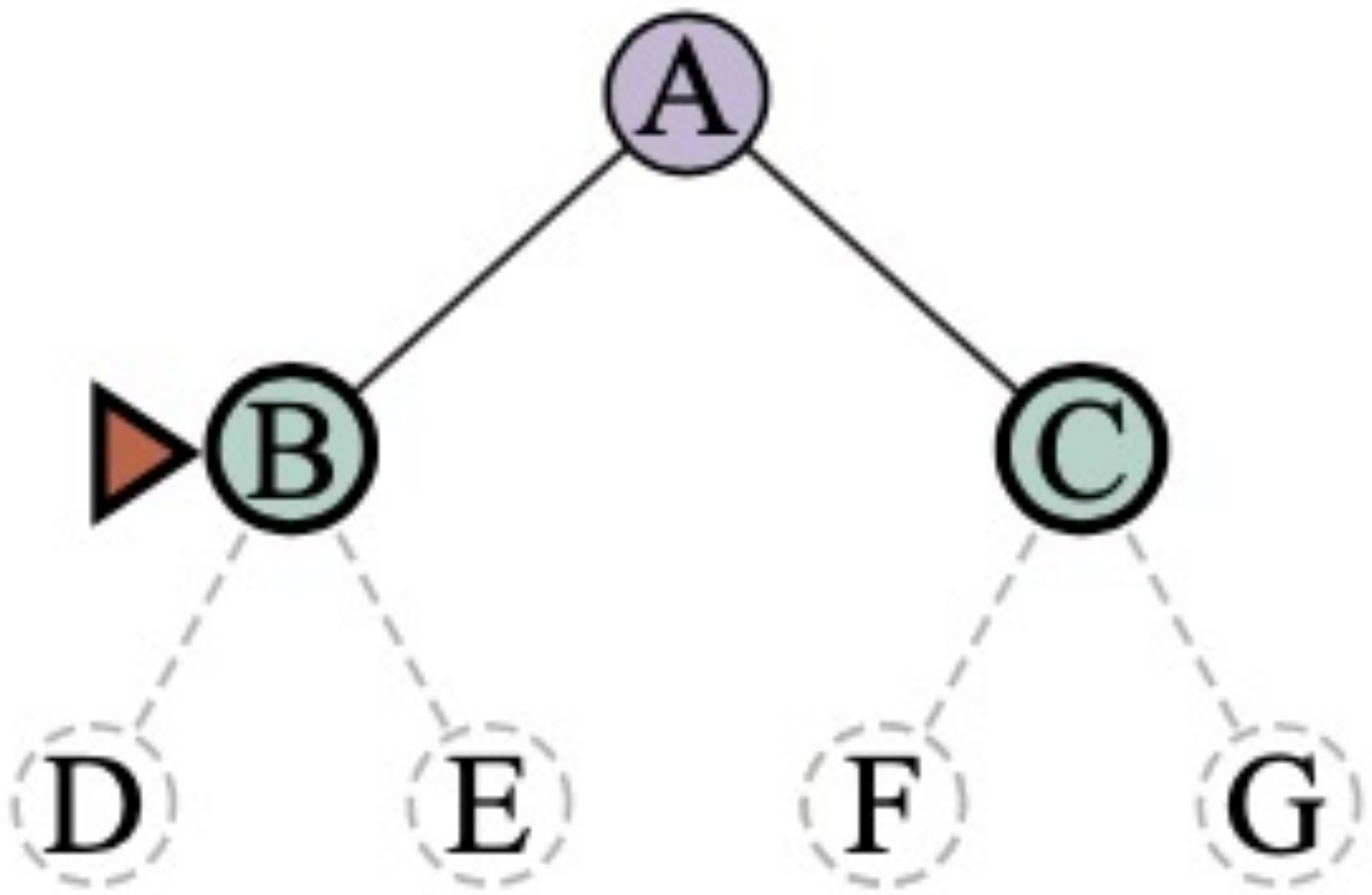
Bread-First Search (BFS)



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

Breadth-First Search on a Simple Binary Tree

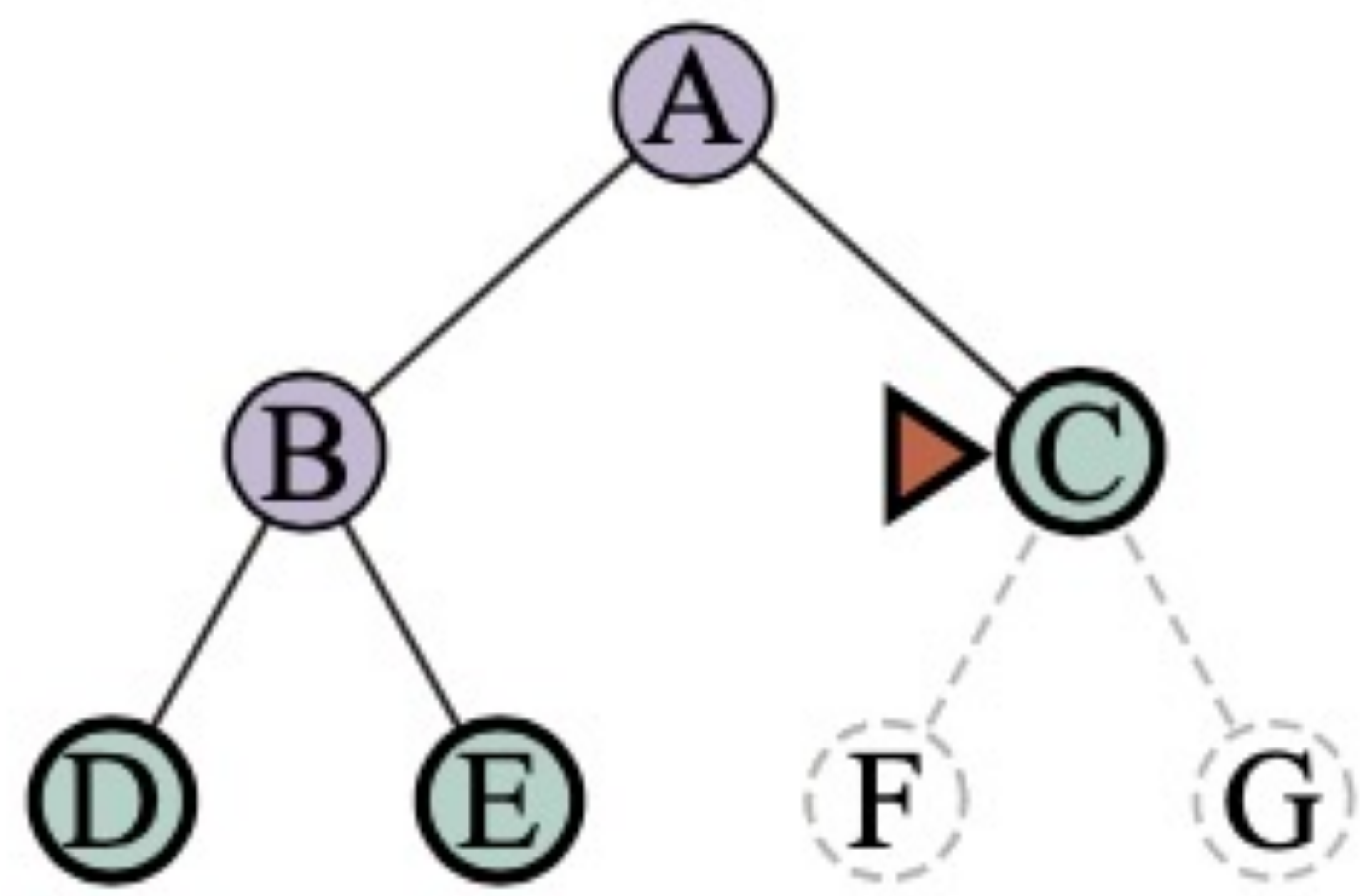
Bread-First Search (BFS)



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

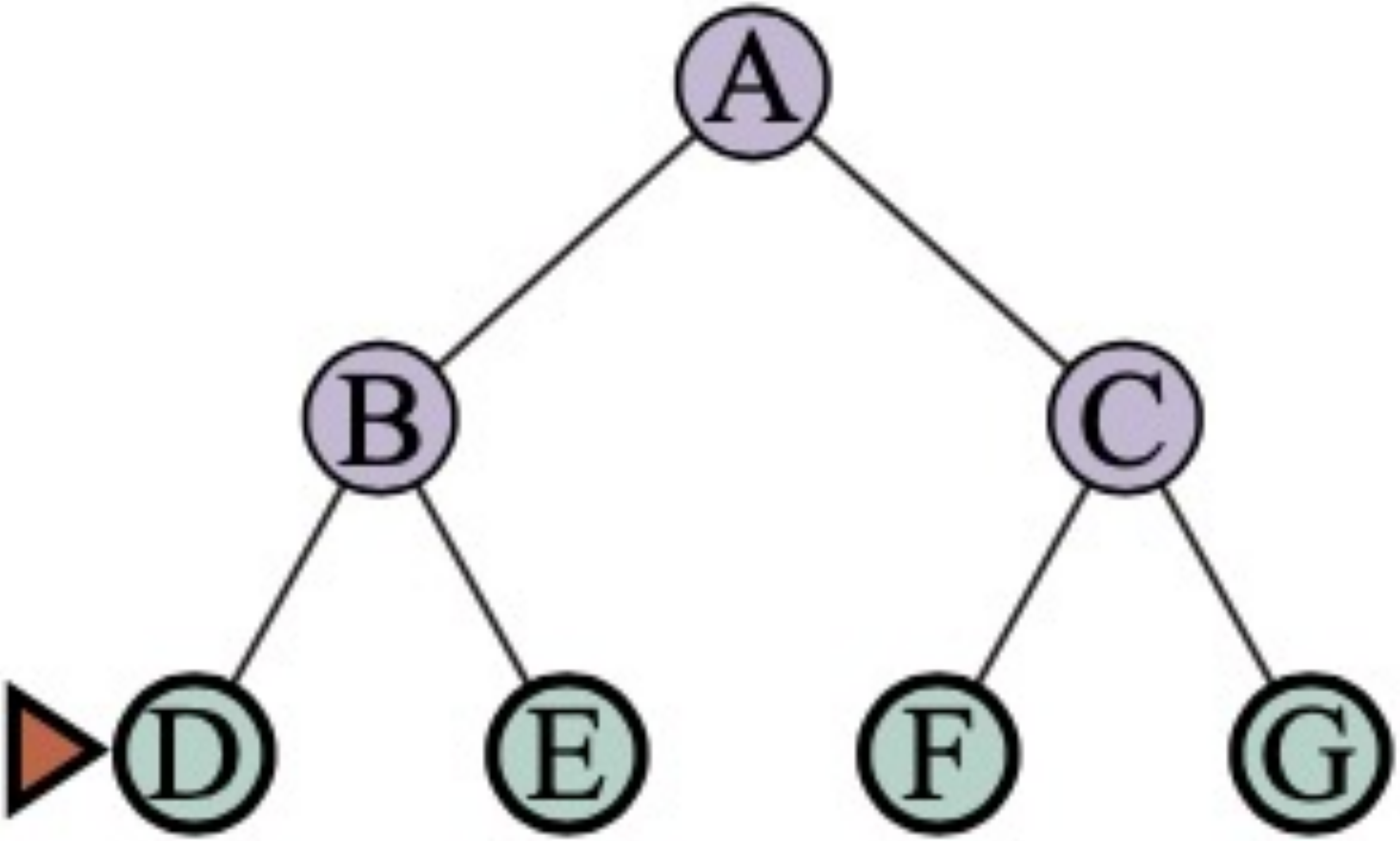
Breadth-First Search on a Simple Binary Tree

Bread-First Search (BFS)



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

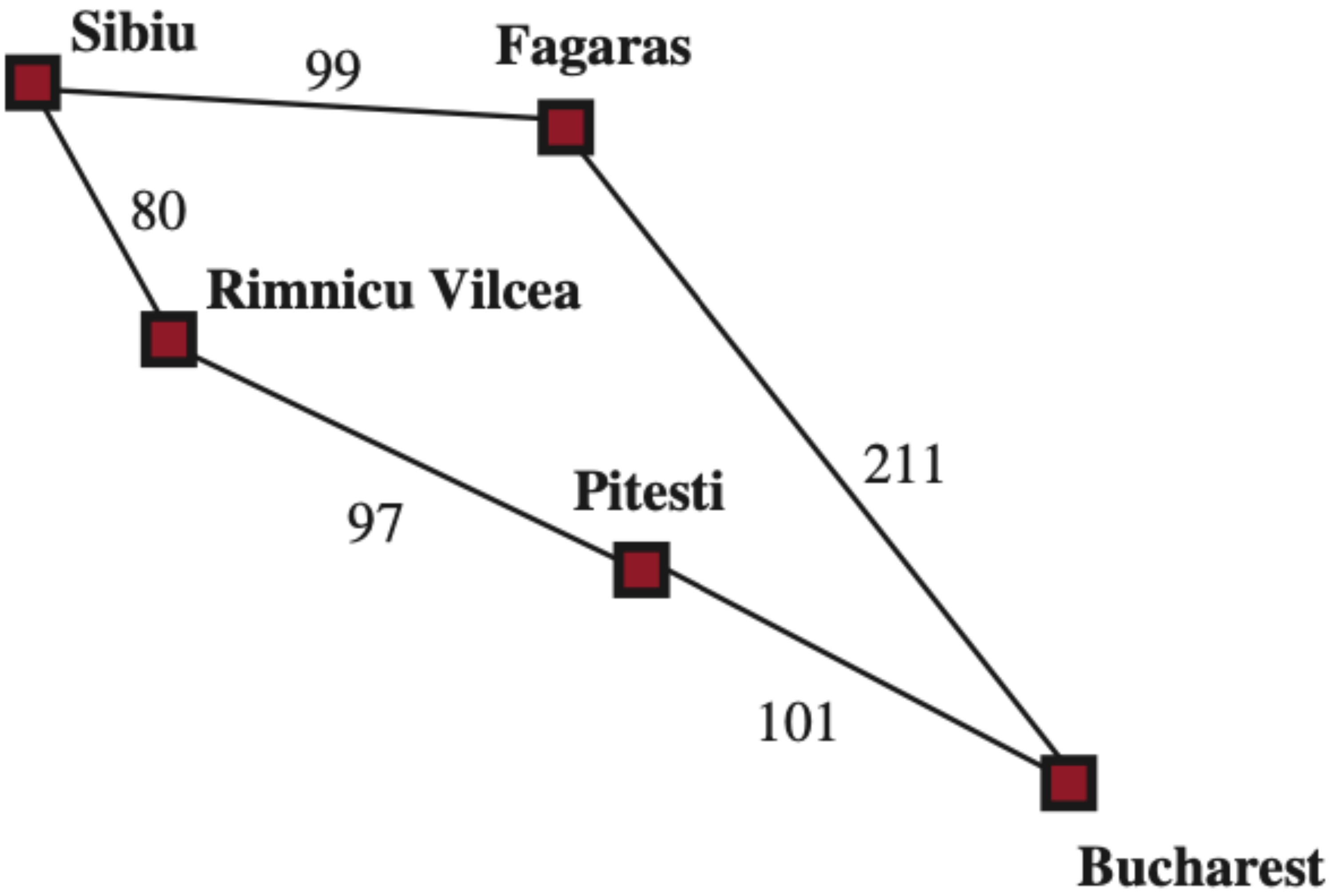
Breadth-First Search and Uniform-Cost Search Algorithms

```
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

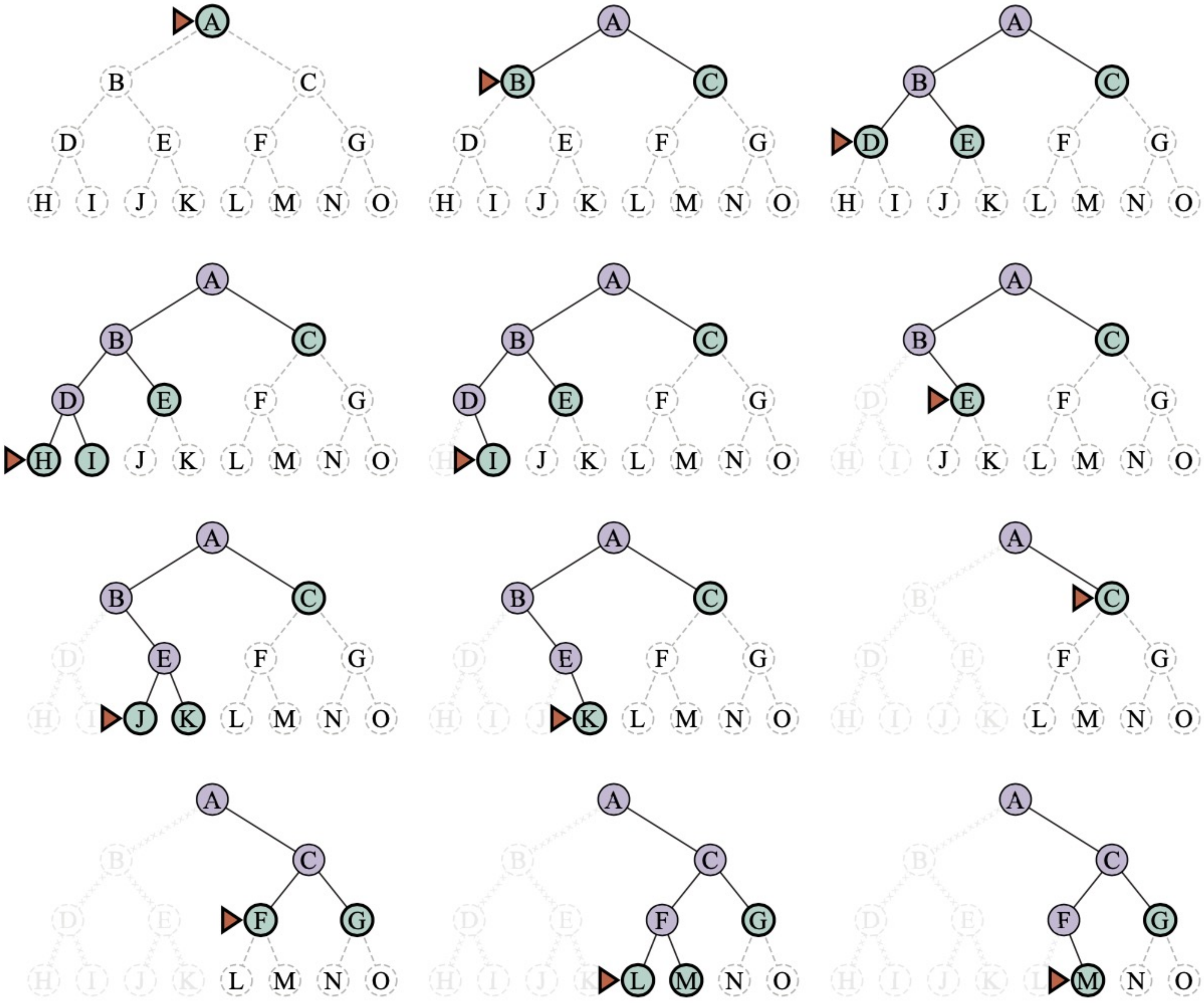
Part of the Romania State Space
Uniform-Cost Search



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

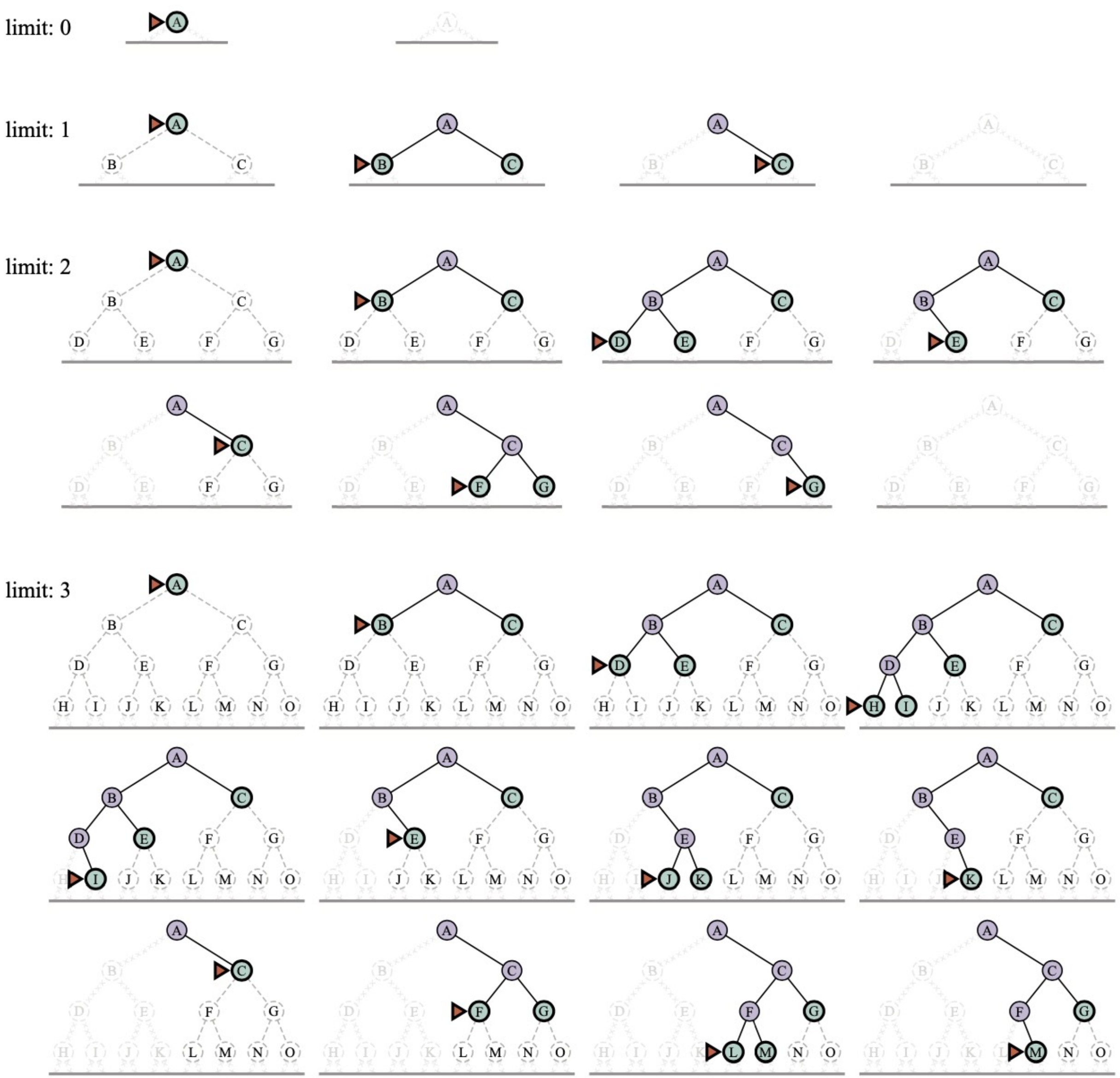
Iterative deepening and depth-limited tree-like search

```
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
```

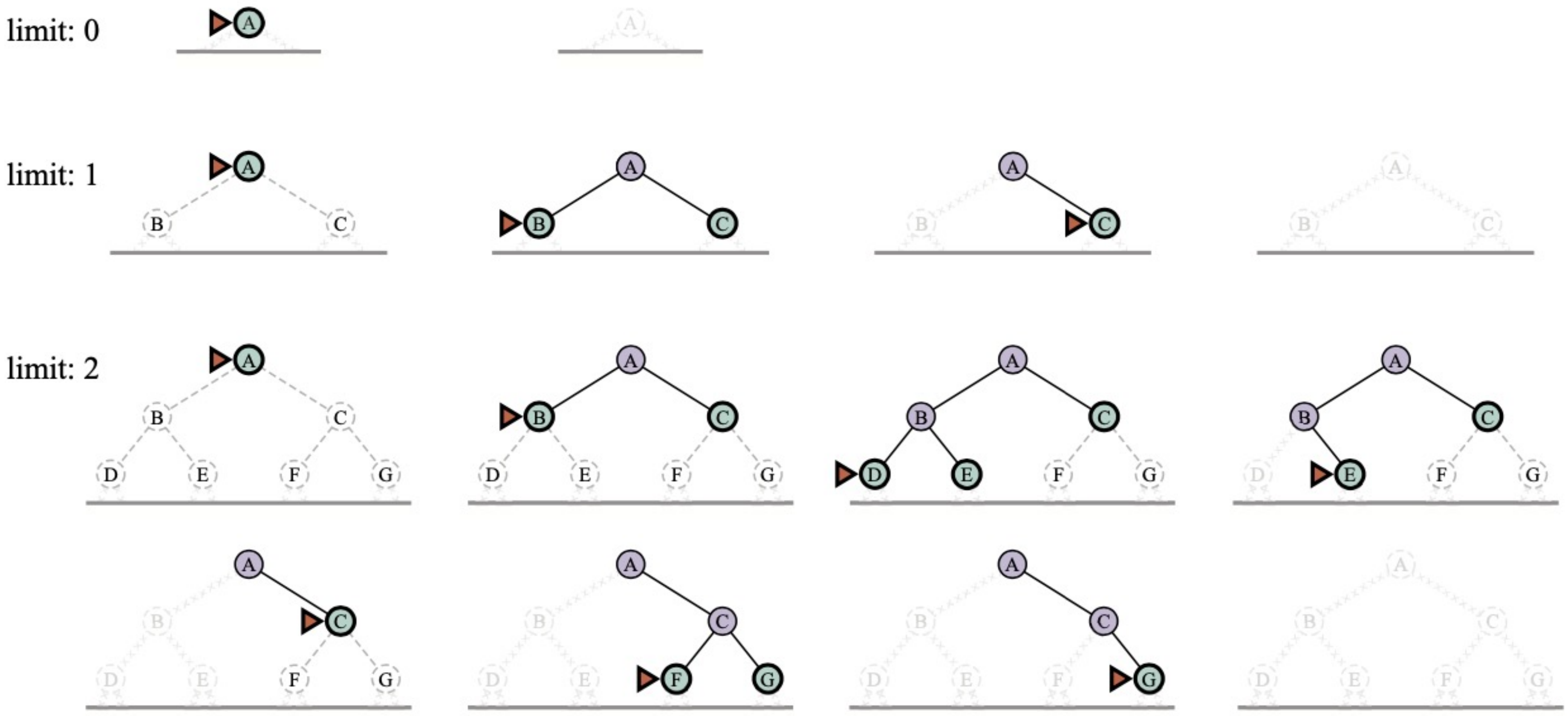
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Four iterations of iterative deepening search



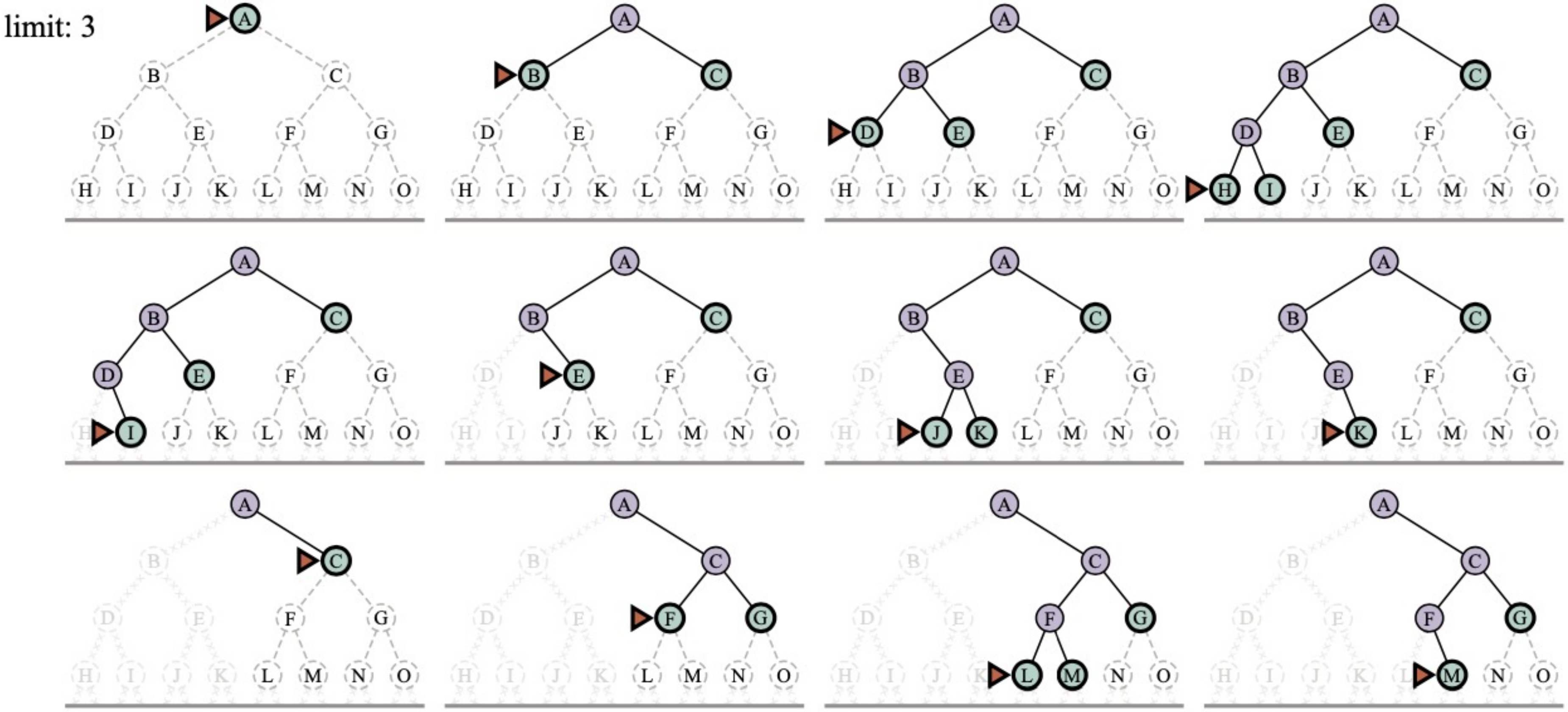
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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



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 $solution$

Bidirectional Best-First Search

keeps two frontiers and two tables of reached states

function PROCEED(*dir*, *problem*, *frontier*, *reached*, $reached_2$, *solution*) **returns** a solution
    // *Expand node on frontier; check against the other frontier in* $reached_2$.
    // *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 $reached_2$ **then**
        $solution_2$ ← JOIN-NODES(*dir*, *child*, $reached_2$[s]))
        if PATH-COST($solution_2$) < PATH-COST(*solution*) **then**
          solution ← $solution_2$
  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

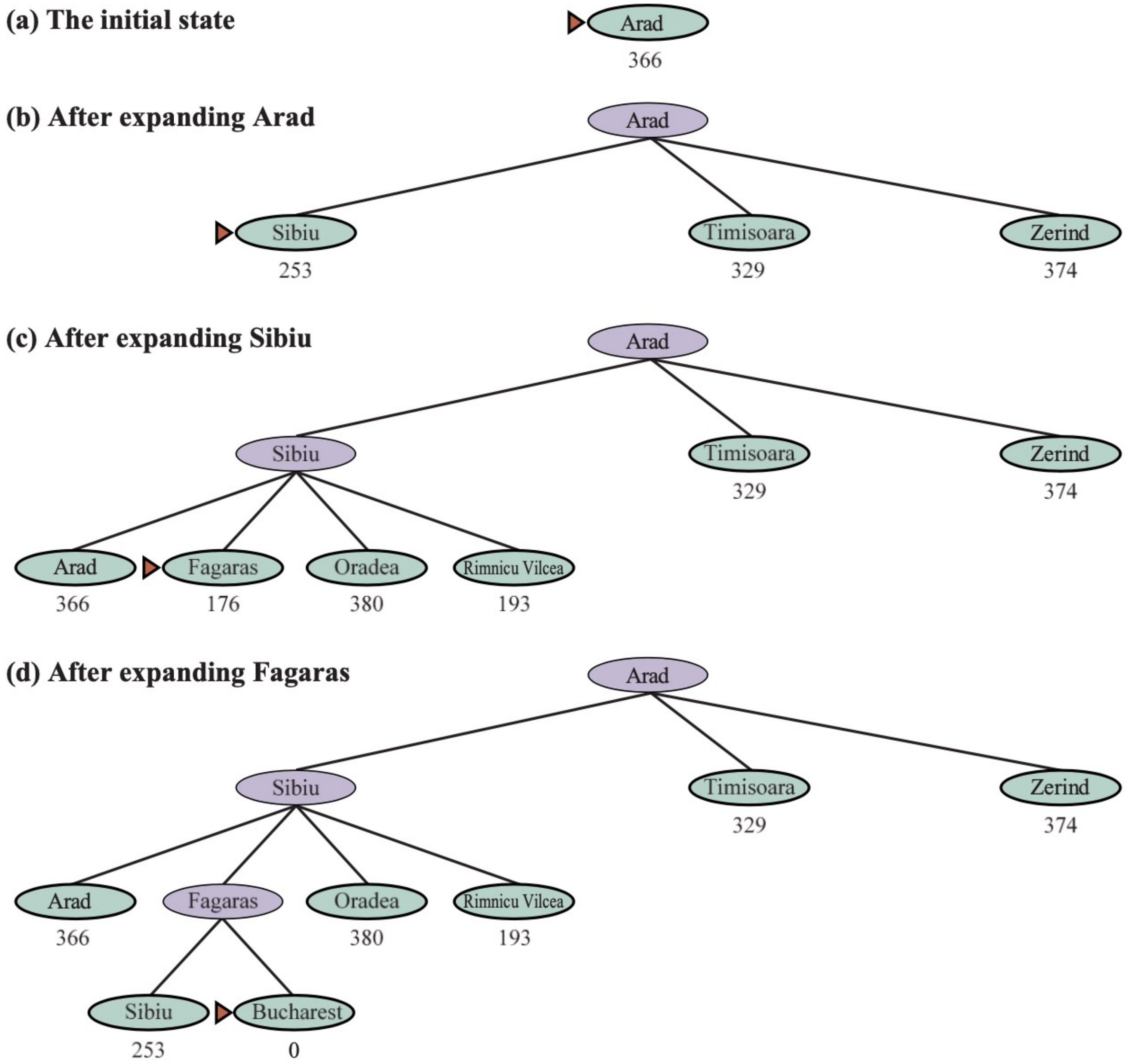
Values of *hSLD*

—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

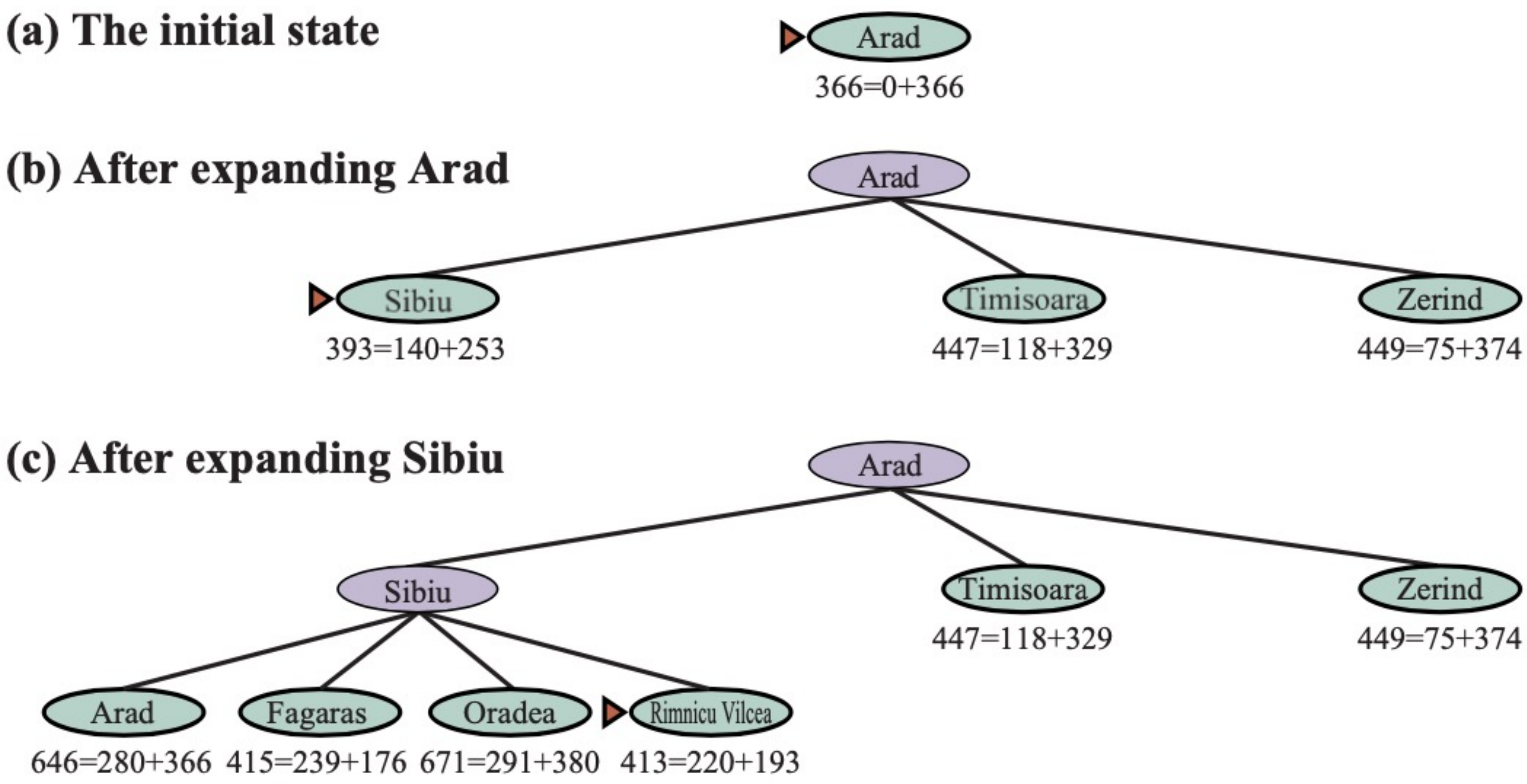
A* search



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}

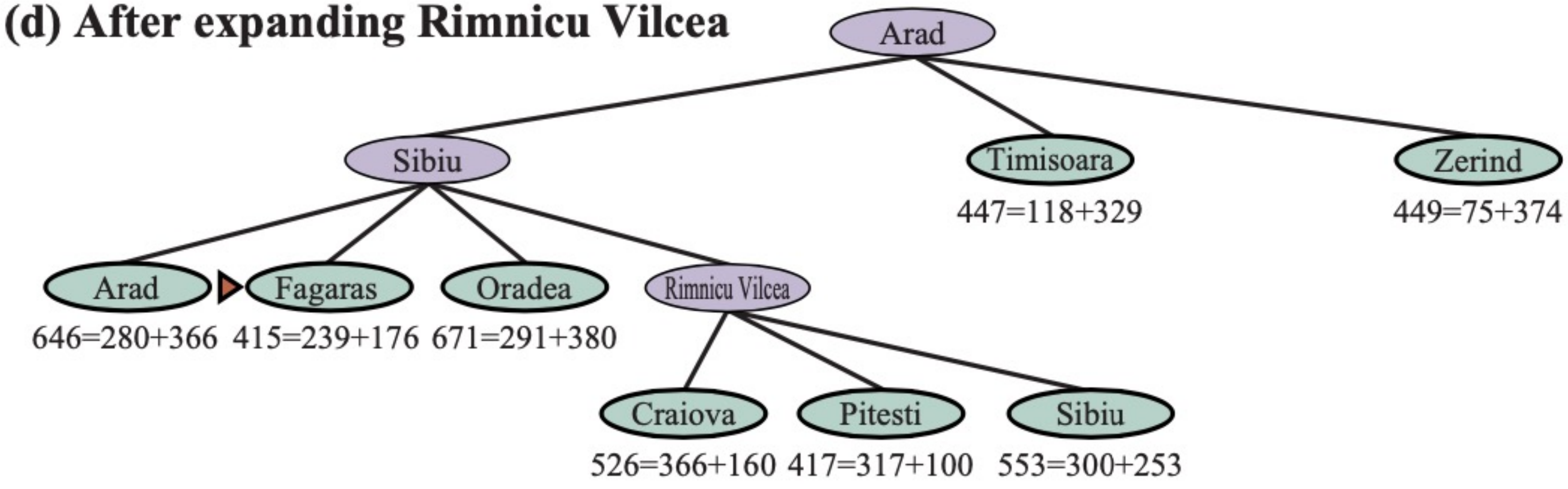


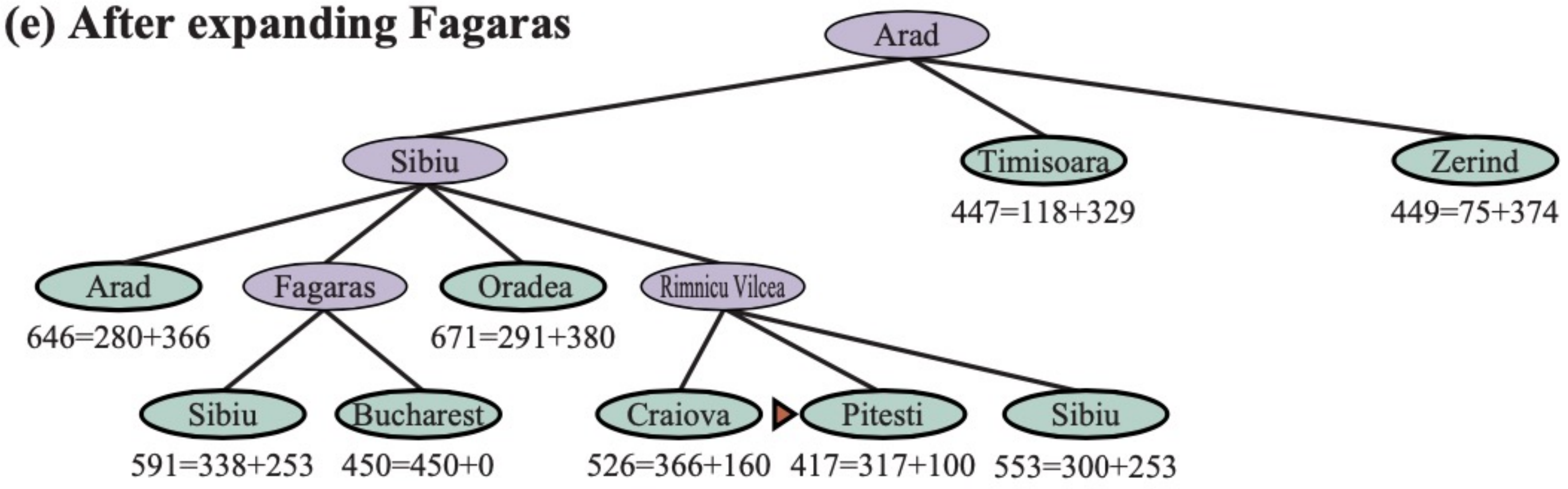
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A∗ search

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

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

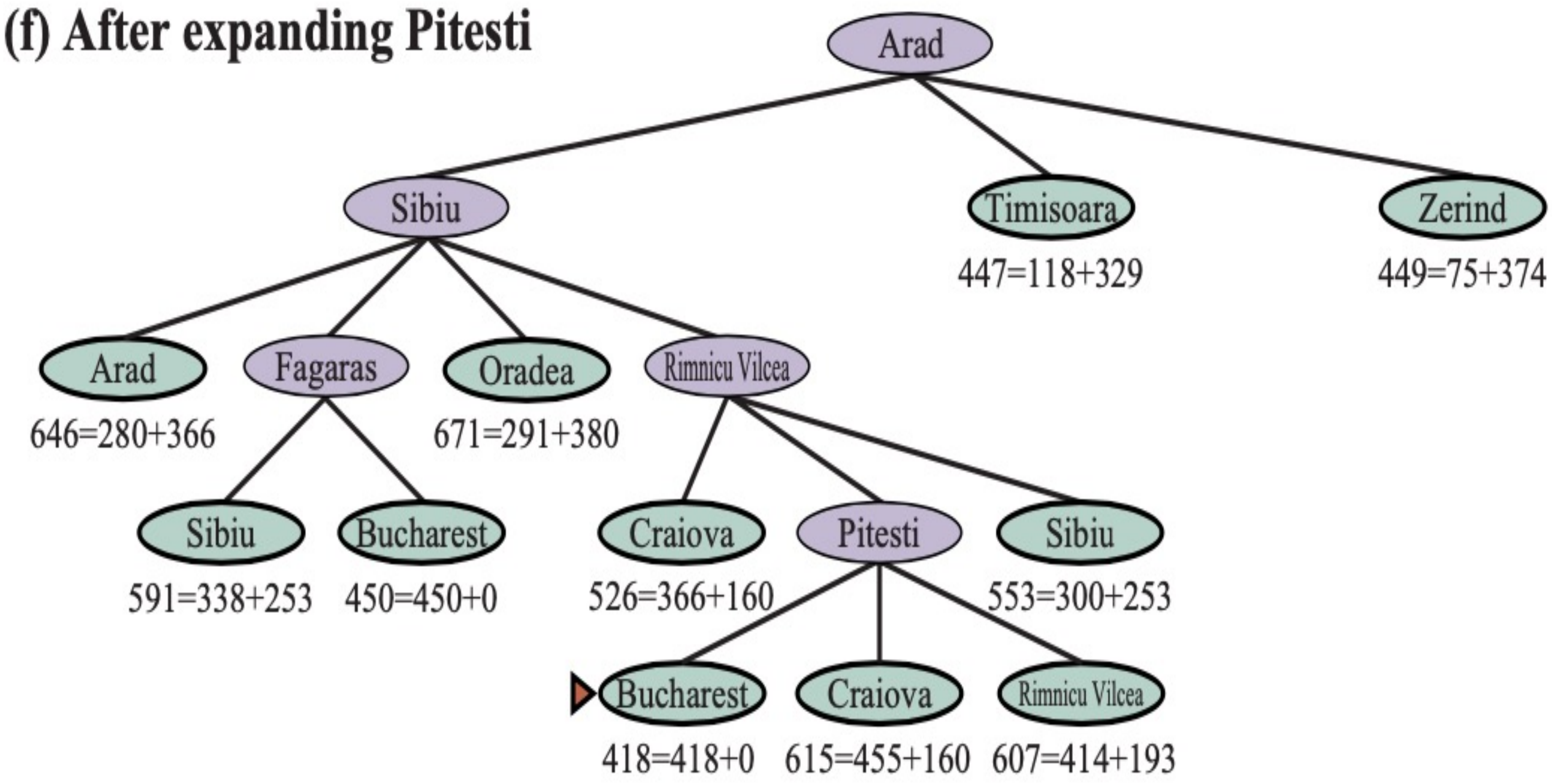




Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

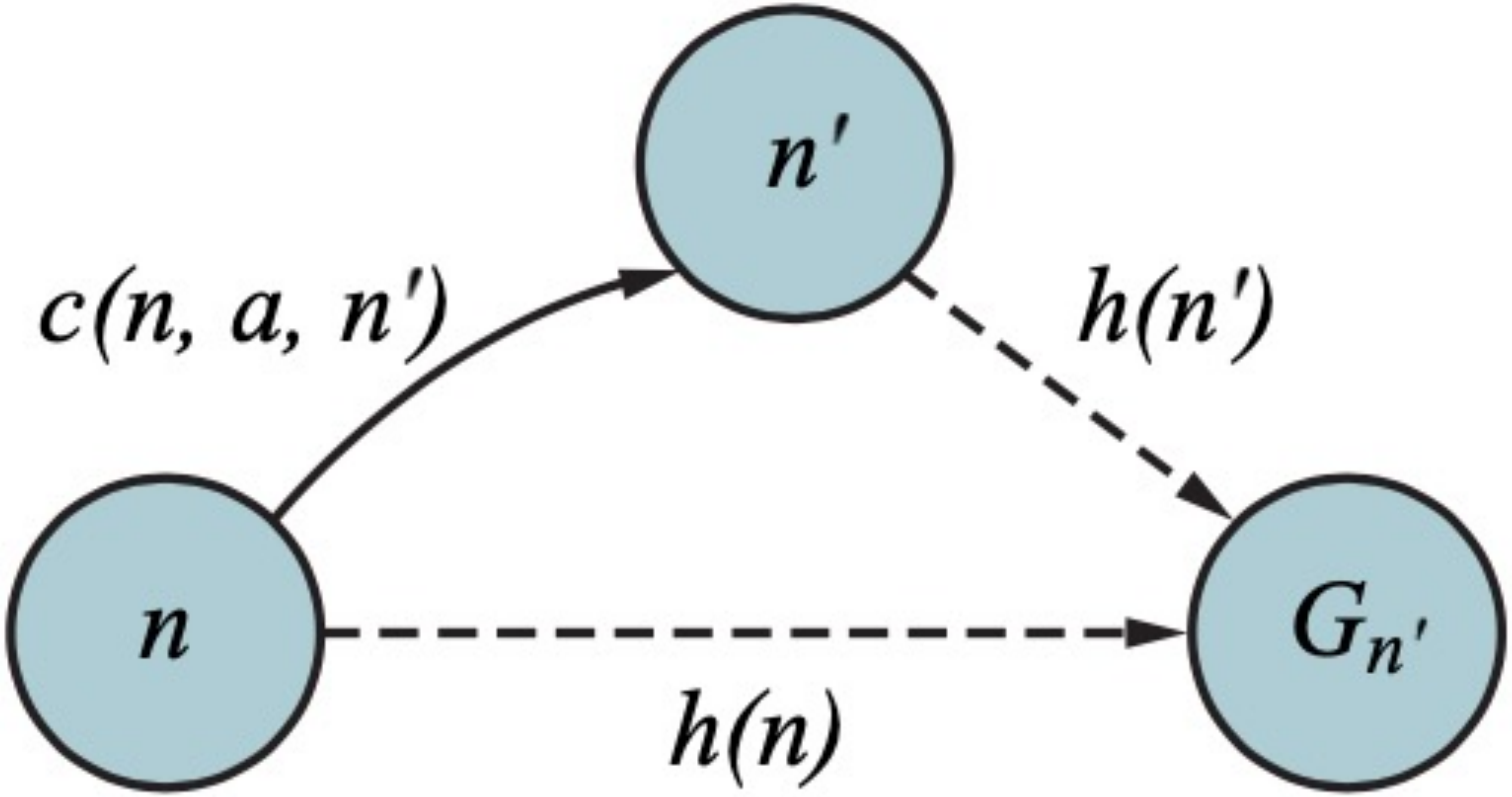
A∗ search

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

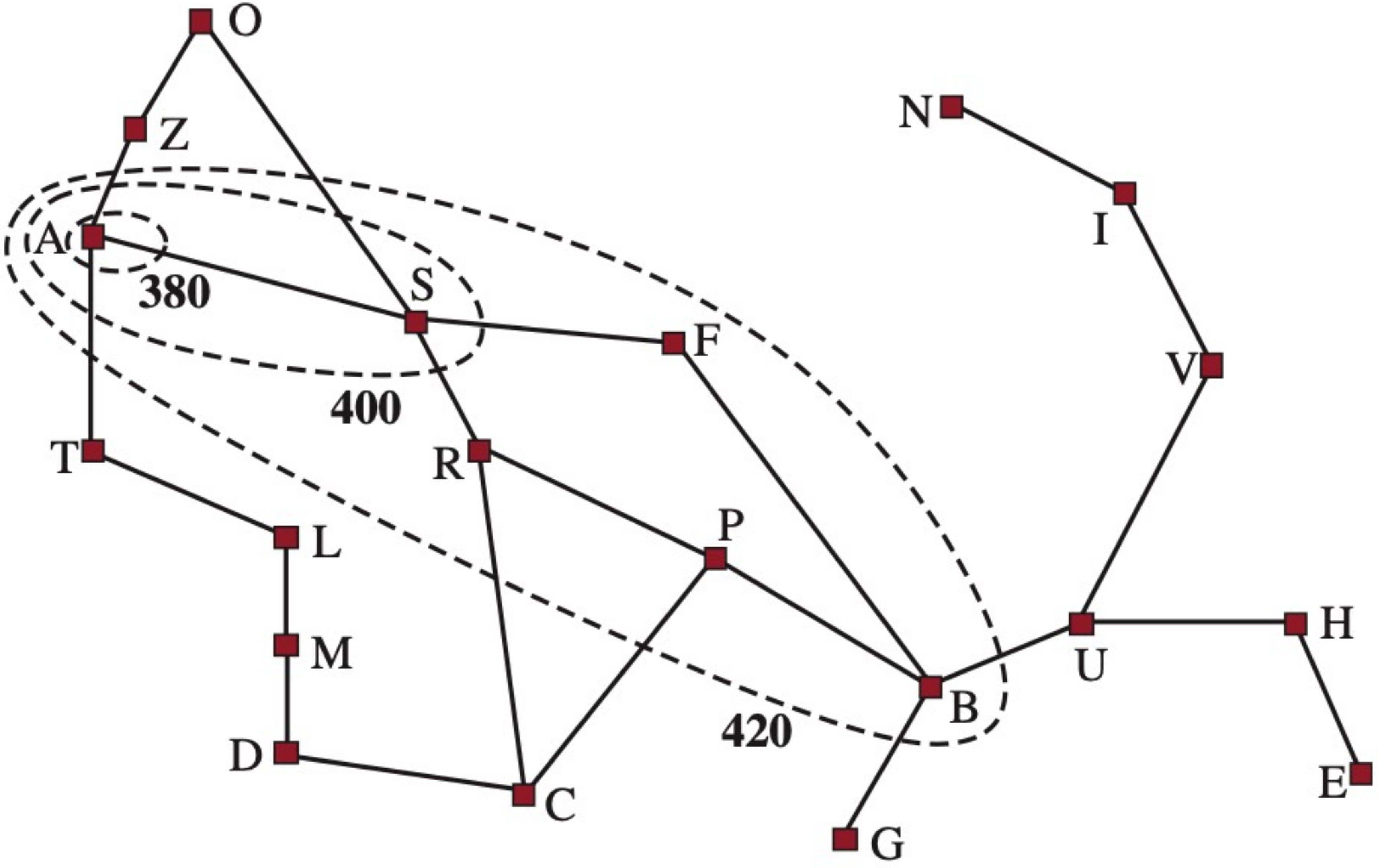
Triangle Inequality



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′).

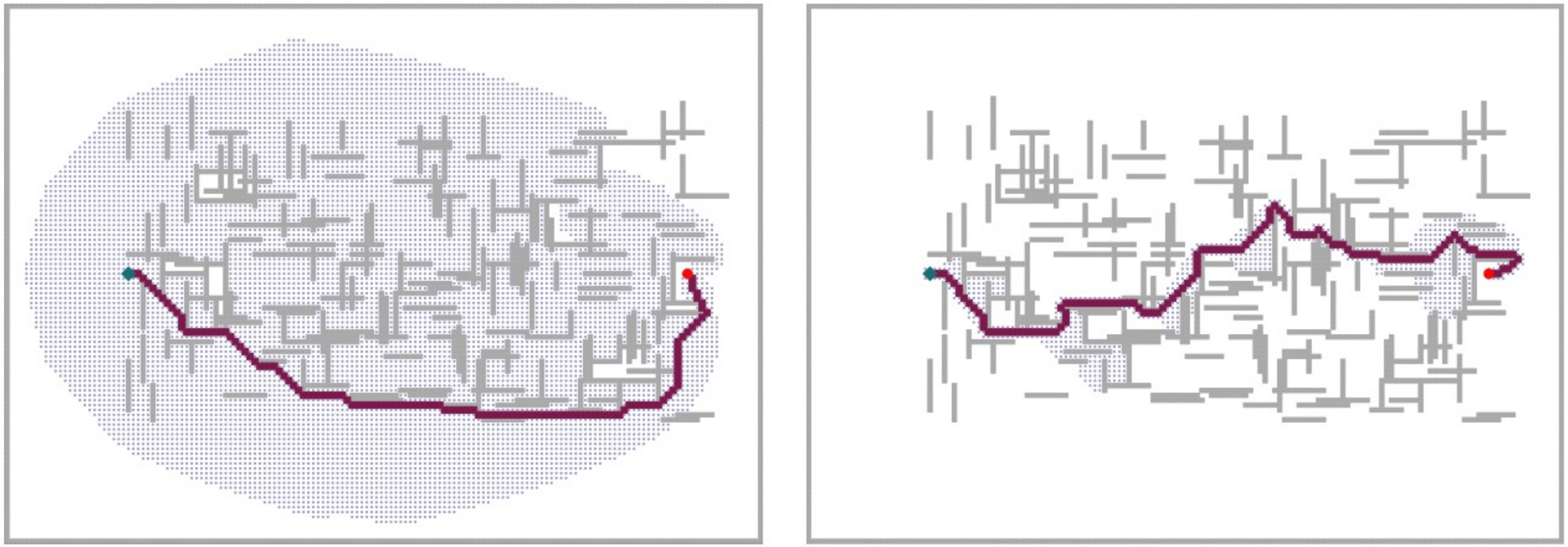
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

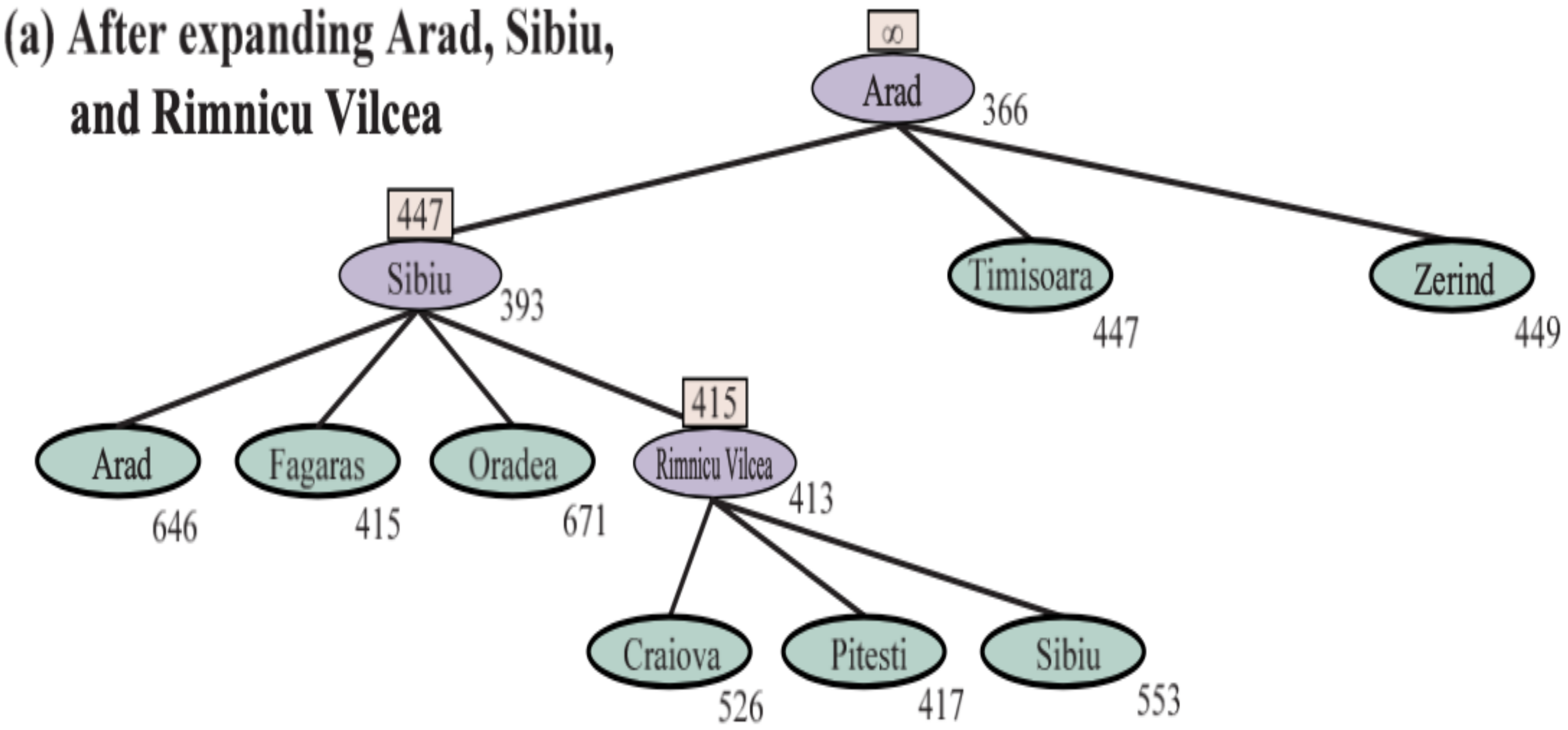
Recursive Best-First Search (RBFS) Algorithm

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*
        $s.f \leftarrow \max(s.\text{PATH-COST} + h(s),\ node.f)$
    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*, $best.f$

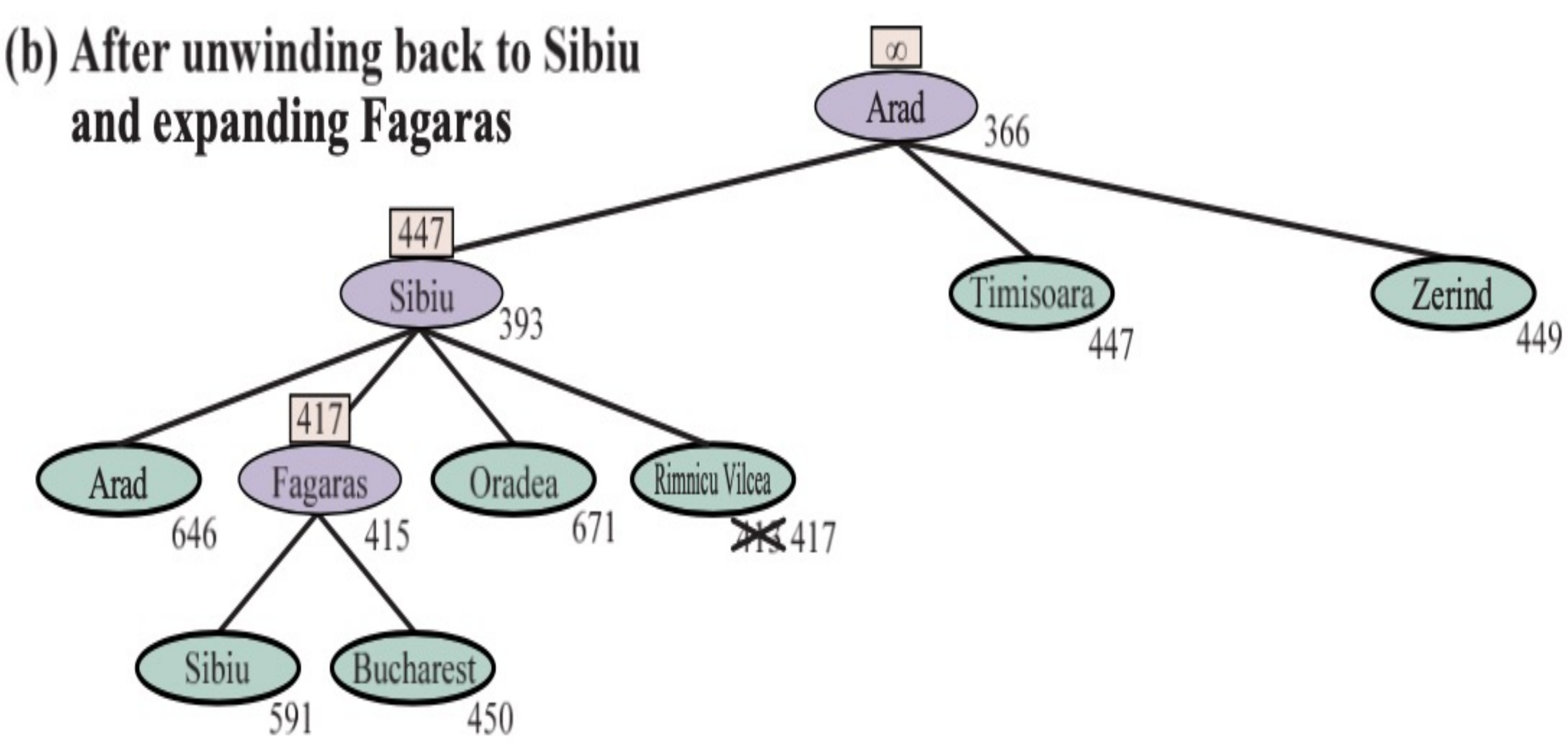
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Recursive Best-First Search (RBFS)



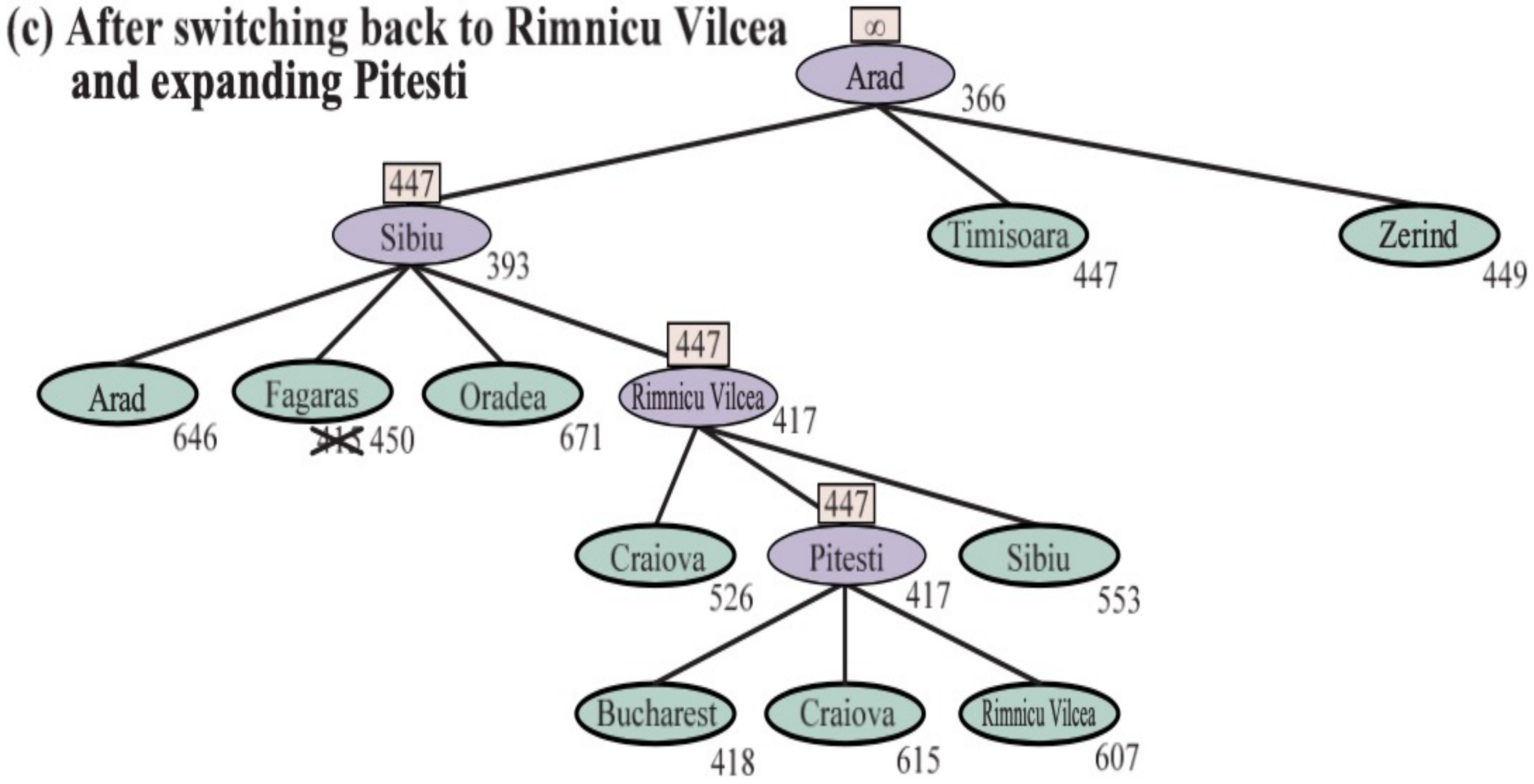
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Recursive Best-First Search (RBFS)



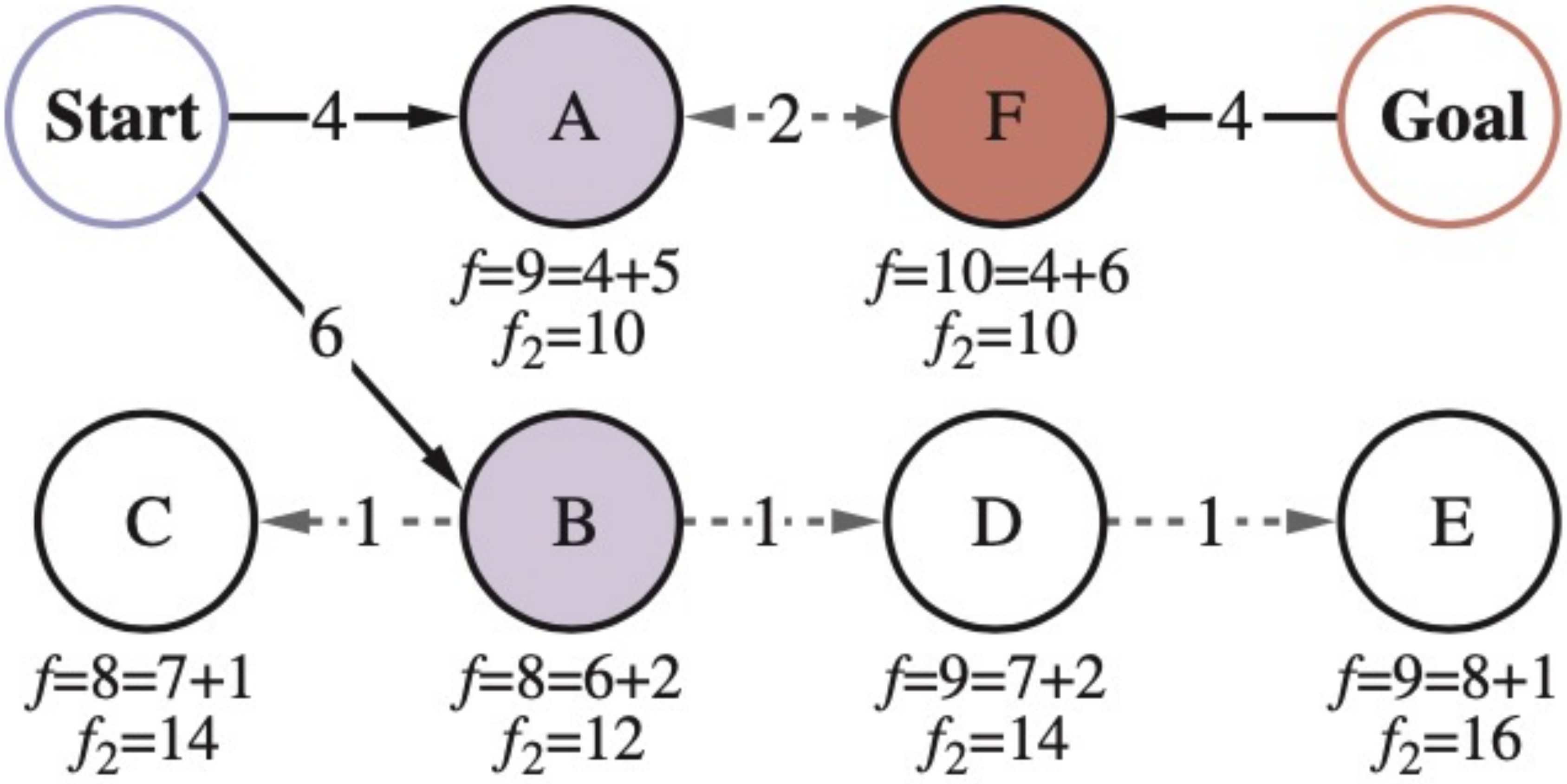
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Recursive Best-First Search (RBFS)



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

Bidirectional Search maintains two frontiers

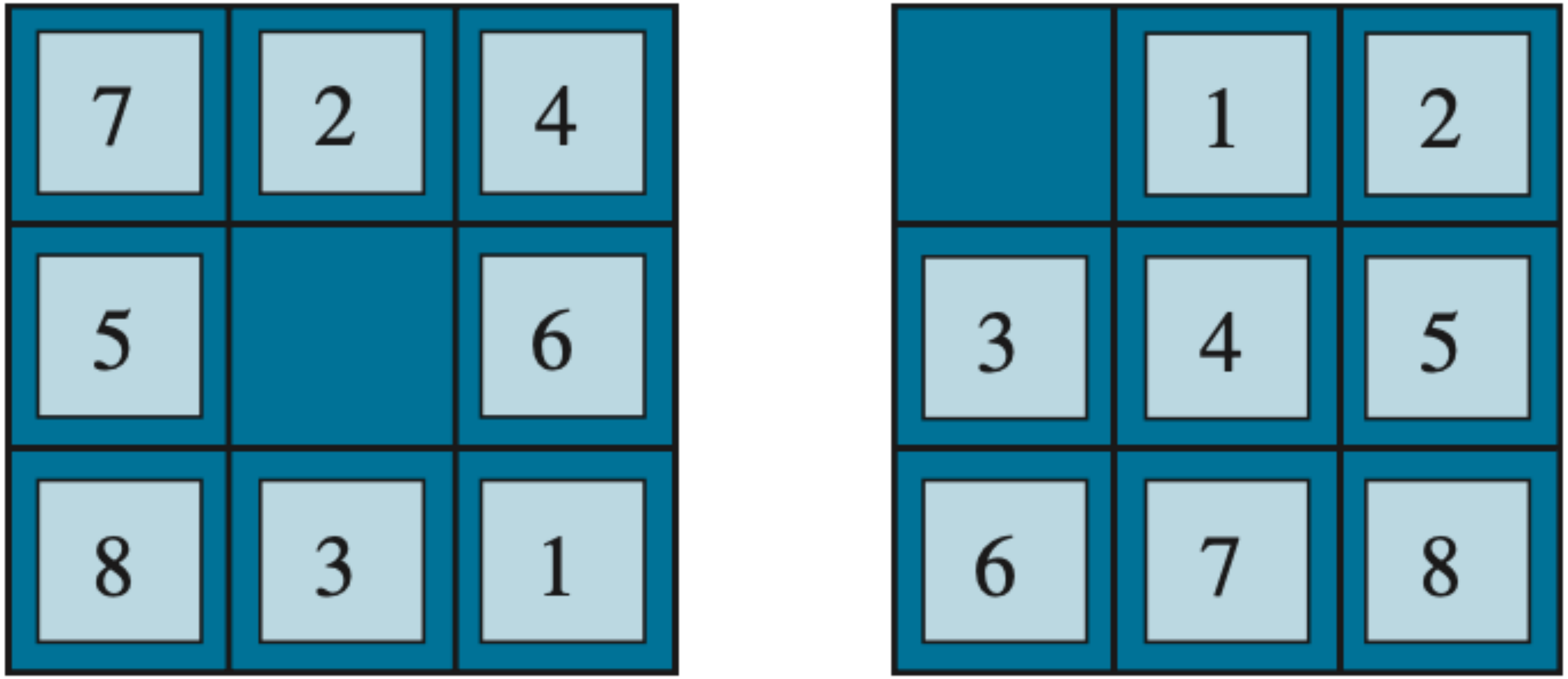


On the left, nodes A and B are successors of Start; on the right, node F is an inverse successor of Goal

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



Start State

Goal State

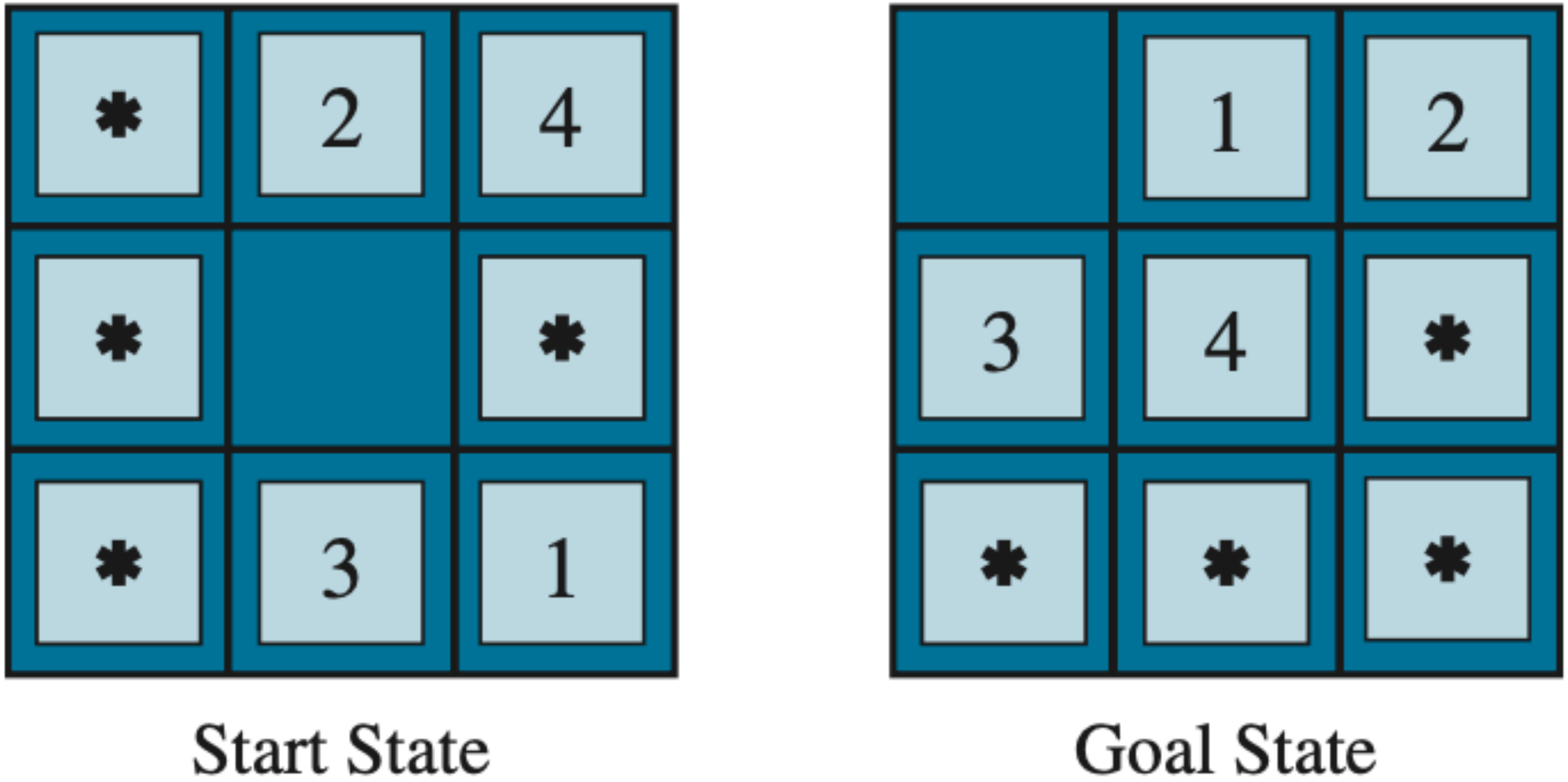
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

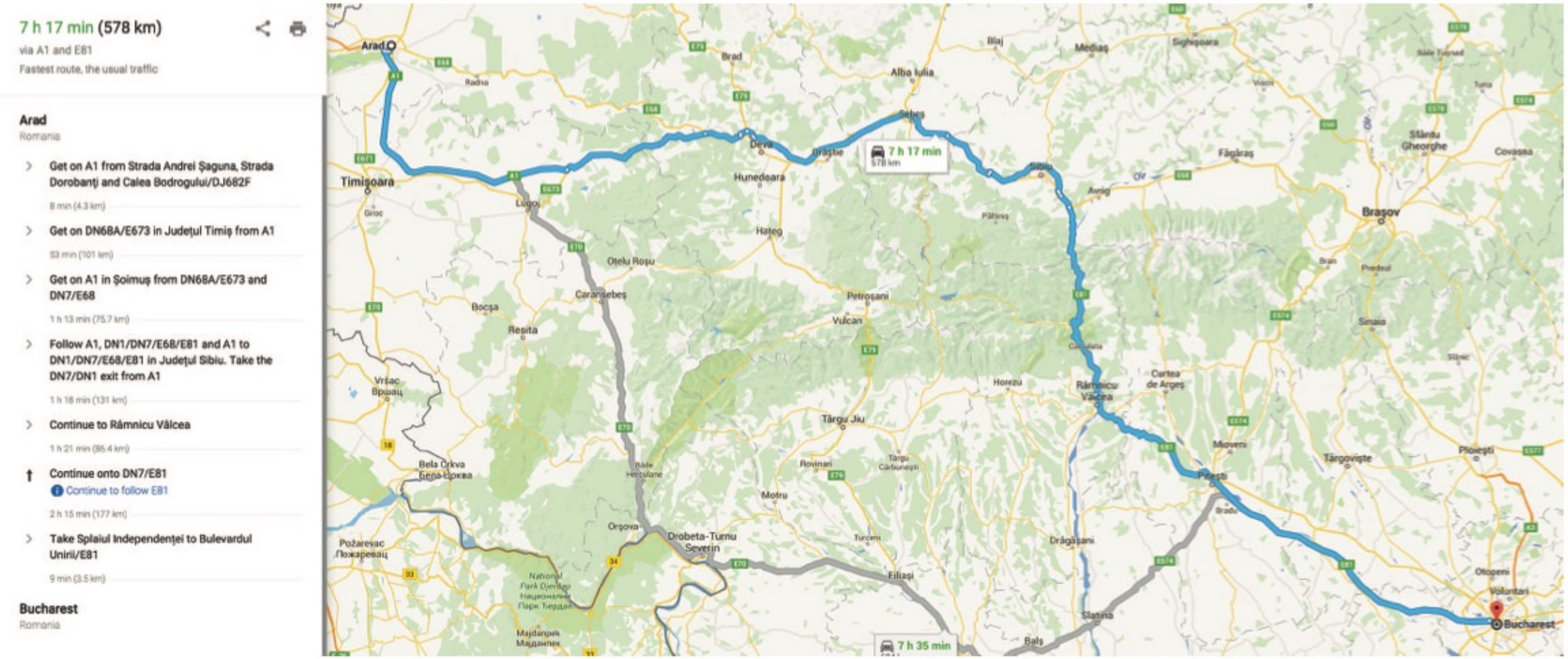
A subproblem of the 8-puzzle



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

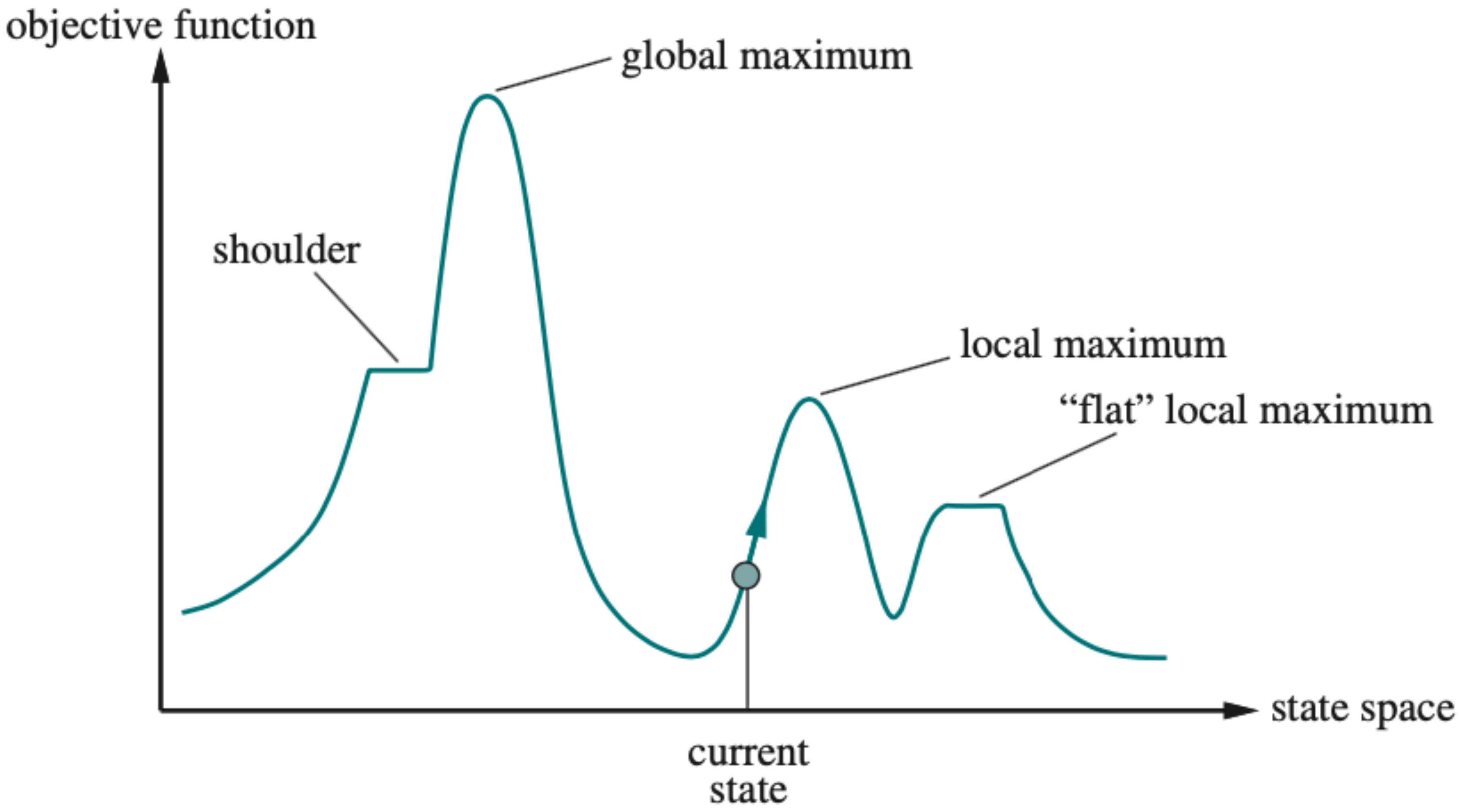
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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



Search in Complex Environments

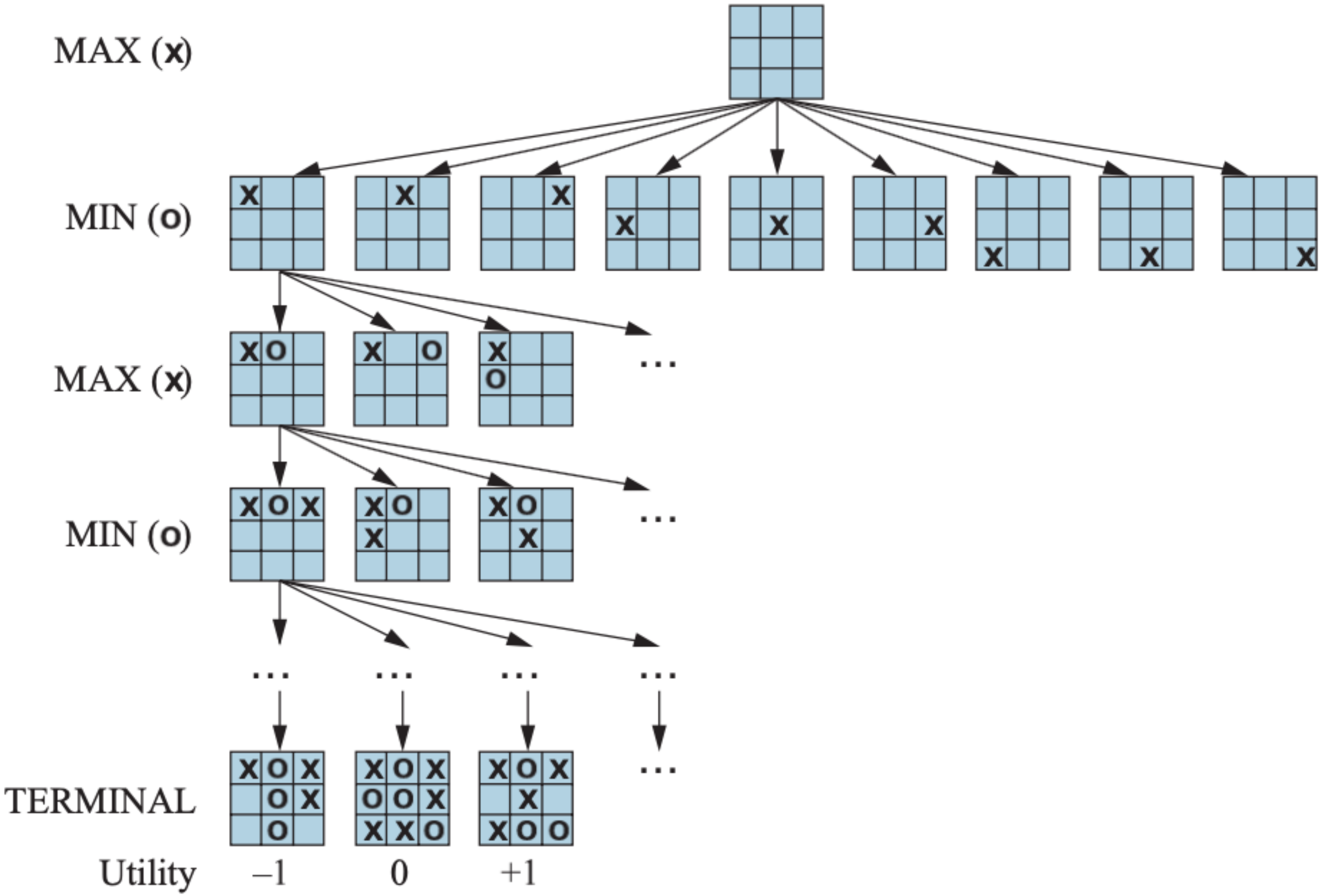
A one-dimensional state-space landscape



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

Adversarial Search and Games

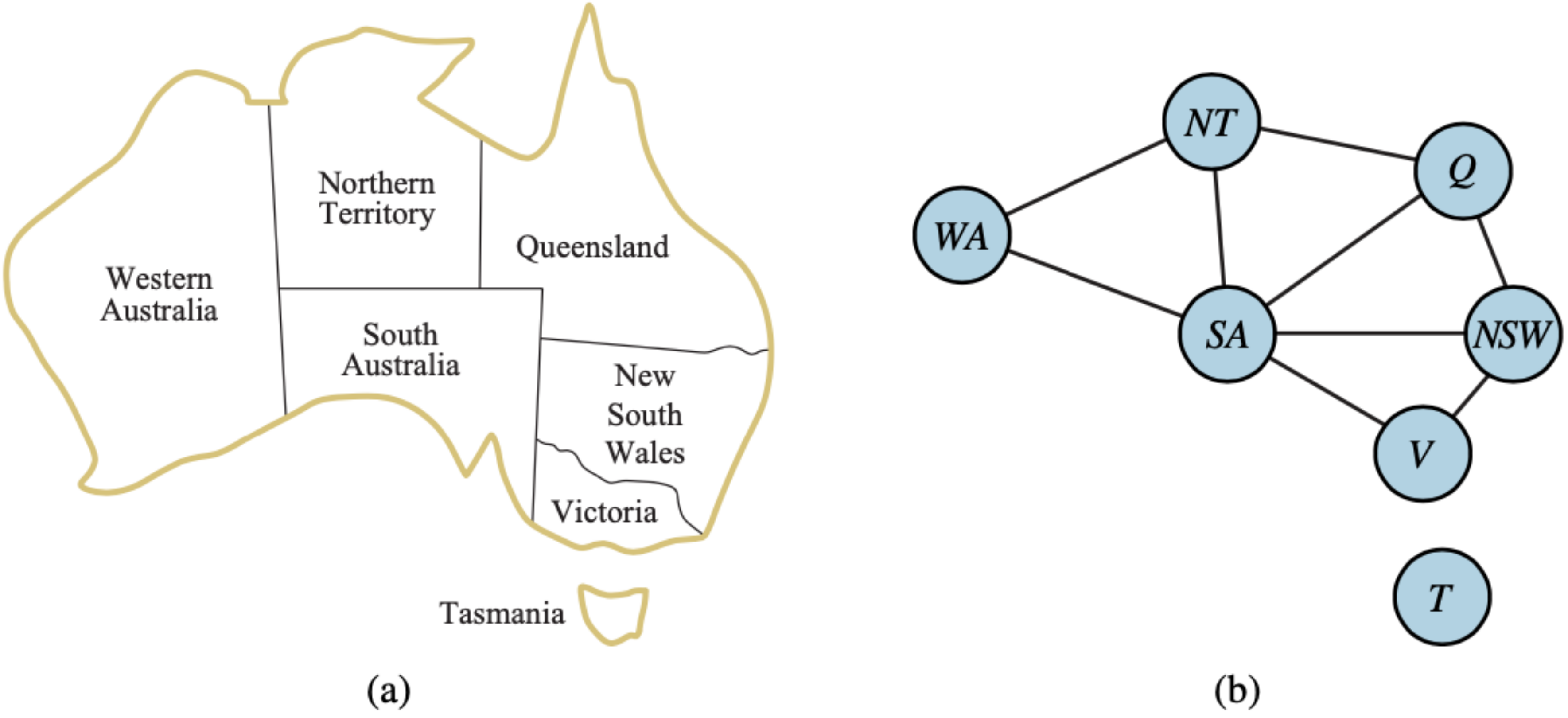
Game Tree for the Game of Tic-tac-toe



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

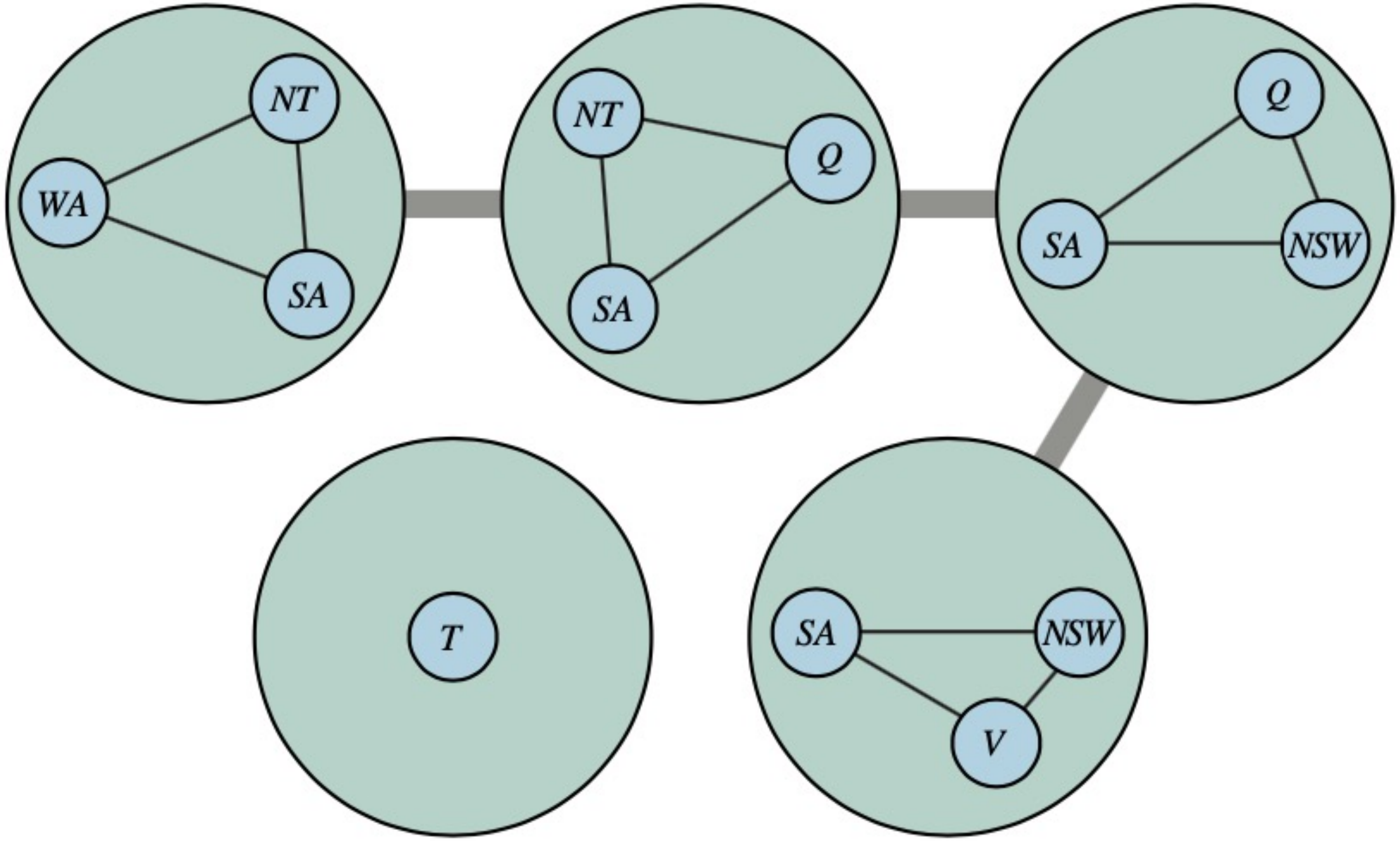
Constraint Satisfaction Problems

The Map-Coloring Problem Represented as a Constraint Graph



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)

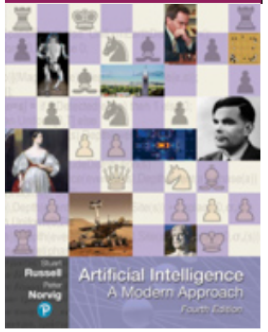
- **Artificial Intelligence: A Modern Approach (AIMA)**
  - http://aima.cs.berkeley.edu/
- **AIMA Python**
  - http://aima.cs.berkeley.edu/python/readme.html
  - https://github.com/aimacode/aima-python
- **Search**
  - http://aima.cs.berkeley.edu/python/search.html
- **Games: Adversarial Search**

  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.

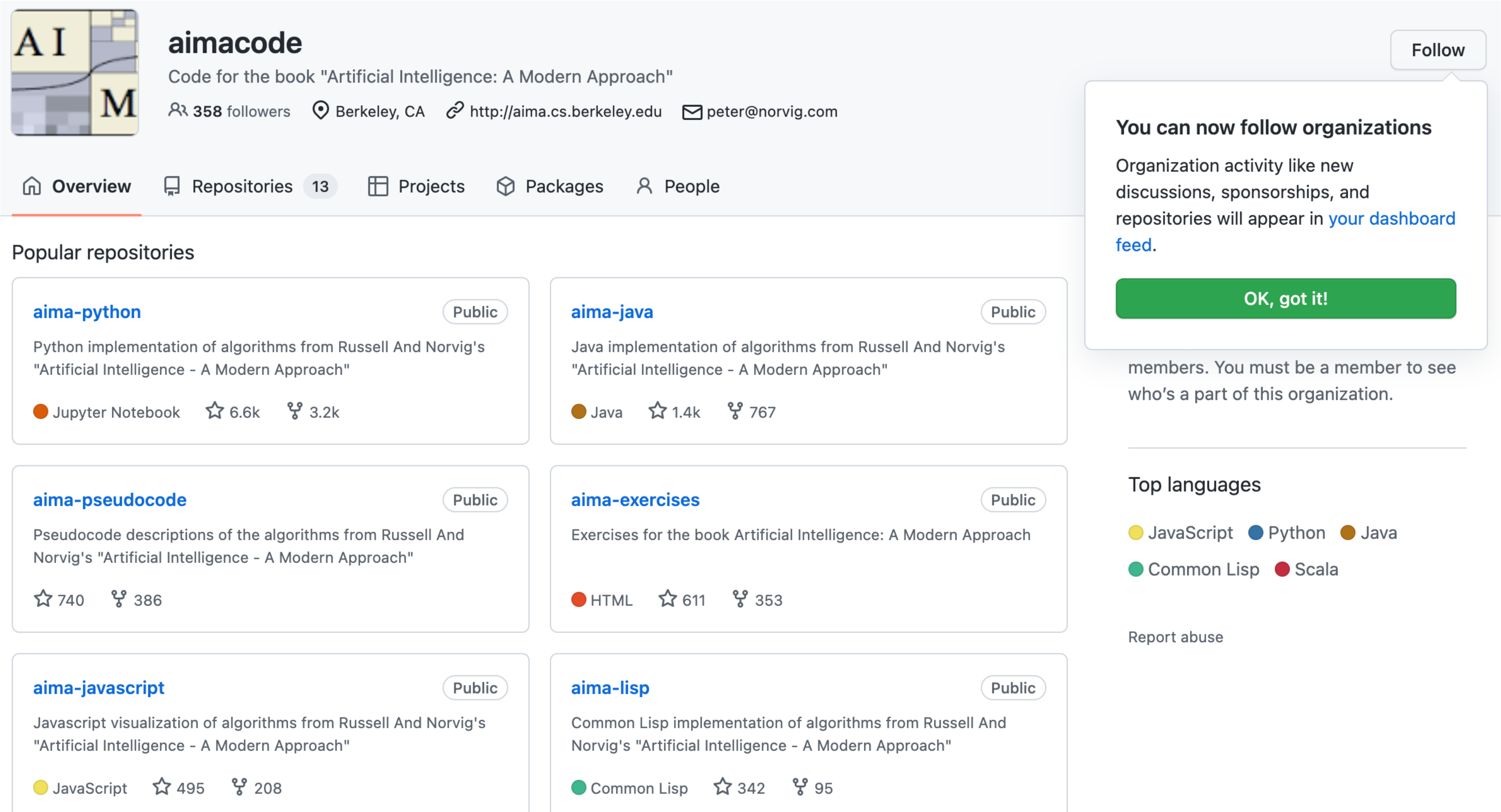
by Stuart Russell and Peter Norvig

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

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

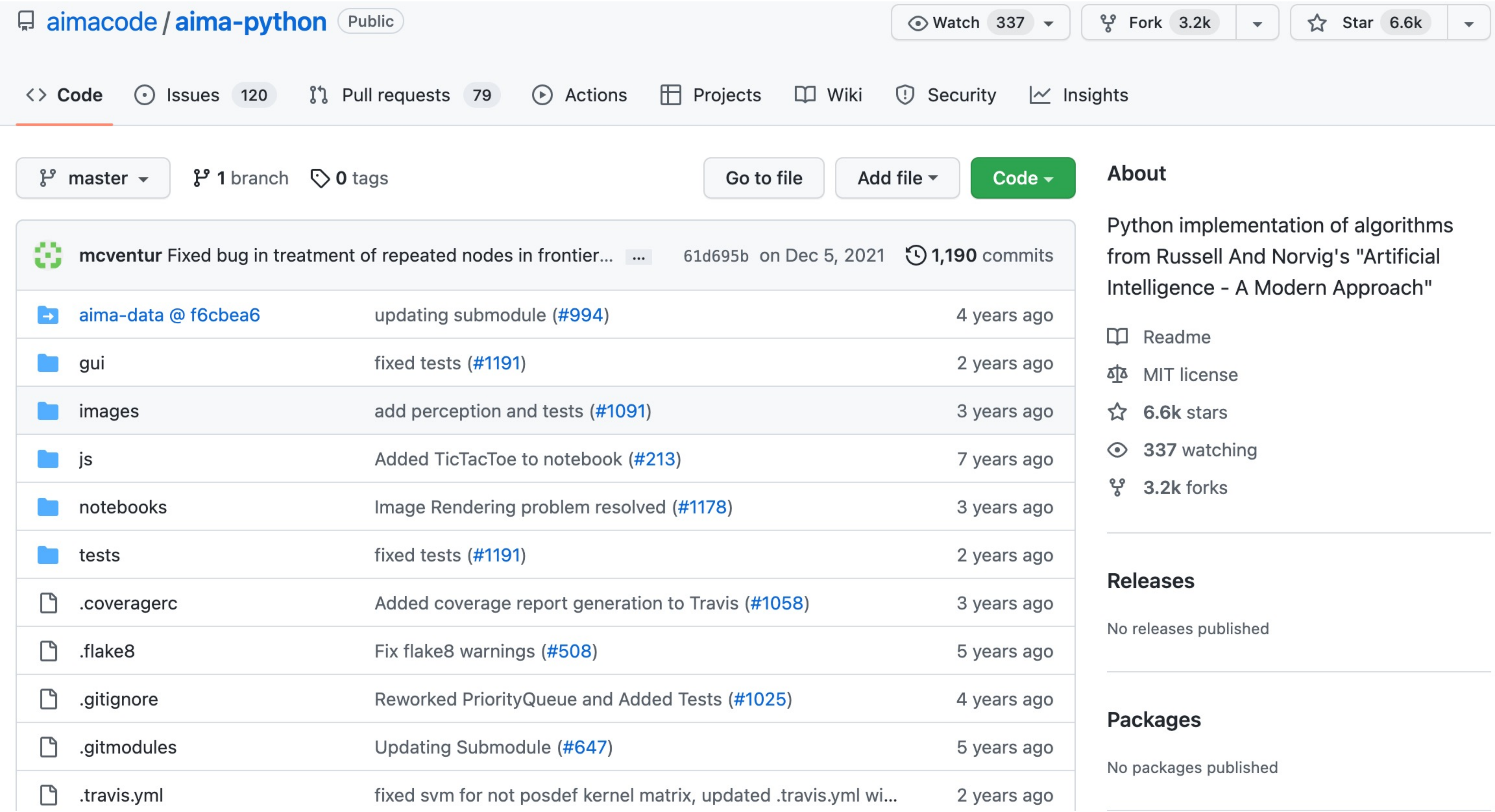
http://aima.cs.berkeley.edu/

AIMA Code



https://github.com/aimacode

AIMA Python



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

Papers with Code
State-of-the-Art (SOTA)

Search for papers, code and tasks | Browse State-of-the-Art | Follow | Discuss | Trends | About | Log In/Register

Browse State-of-the-Art

1509 leaderboards • 1327 tasks • 1347 datasets • 17810 papers with code

Follow on Twitter for updates

Computer Vision

Task	Leaderboards	Papers with code
Semantic Segmentation	33 leaderboards	667 papers with code
Image Classification	52 leaderboards	564 papers with code
Object Detection	54 leaderboards	467 papers with code
Image Generation	51 leaderboards	231 papers with code
Pose Estimation	40 leaderboards	231 papers with code

▸ See all 707 tasks

Natural Language Processing

Machine Translation | Language Modelling | Question Answering | Sentiment Analysis | Text Generation

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

- **Artificial Intelligence**
- **Intelligent Agents**
- **Problem Solving**

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Thomas R. Caldwell (2025), The Agentic AI Bible: The Complete and Up-to-Date Guide to Design, Build, and Scale Goal-Driven, LLM-Powered Agents that Think, Execute and Evolve, Independently published
- Numa Dhamani and Maggie Engler (2024), Introduction to Generative AI, Manning
- Denis Rothman (2024), Transformers for Natural Language Processing and Computer Vision - Third Edition: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3, 3rd ed. Edition, Packt Publishing
- Ben Auffarth (2023), Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT and other LLMs, Packt Publishing.
- Aurélien Géron (2022), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd 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.
- Khaled Bayoudh, 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.
- Abdelrahman Mohamed, 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.
- Longbing Cao (2022). "Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci." IEEE Intelligent Systems 37, no. 3: 6-19.
- Thien Huynh-The, 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.
- Thippa Reddy Gadekallu, Thien Huynh-The, Weizheng Wang, Gokul Yenduri, Pasika Ranaweera, Quoc-Viet Pham, Daniel Benevides da Costa, and Madhusanka Liyanage (2022). "Blockchain for the Metaverse: A Review." arXiv preprint arXiv:2203.09738.
- Alec Radford, 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.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. (2024) "Reasoning with Large Language Models, a Survey." arXiv preprint arXiv:2407.11511.
- Madaan, Aman, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon et al. (2024) "Self-refine: Iterative refinement with self-feedback." Advances in Neural Information Processing Systems 36.