NOTABLE AI MOMENTS (1940–1970)

Year	Event
1943	McCulloch & Pitts: Boolean circuit model of the brain
1950	”Alan Turing’s “Computing Machinery and Intelligence
1951	Marvin Minsky develops a neural network machine
1950s	Early AI programs: Samuel’s checkers program, Gelernter’s Geometry Engine, Newell & Simon’s Logic Theorist and General Problem Solver
1956	Dartmouth meeting: “Artificial Intelligence” term adopted
1965	Robinson’s complete algorithm for logical reasoning
1966	Joseph Weizenbaum creates Eliza
1969	Minsky & Papert show limitations of the perceptron; Neural network research declines significantly

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

NOTABLE AI MOMENTS (1970–2000)

Year	Event
1971	Terry Winograd’s Shrdlu dialogue system
1972	Alain Colmerauer invents Prolog programming language
1976	MYCIN, an expert system for disease diagnosis
1980s	Era of expert systems
1990s	Neural networks, probability theory, AI agents
1993	RoboCup initiative to build soccer-playing robots
1997	IBM Deep Blue beats the World Chess Champion

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

NOTABLE AI MOMENTS (2000–2018)

Year	Event
2003	Very large datasets: genomic sequences
2007	Very large datasets: WAC (web as corpus)
2011	IBM Watson wins Jeopardy
2012	US state of Nevada permits driverless cars
2010s	Deep learning takes over: recommendation systems, image analysis, board games, machine translation, pattern recognition
2017	Google AlphaGo beats the world’s best Go player, Ke Jie
2017	AlphaZero learns board games by itself and beats the best programs

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

What is the Forth Wave of AI ? (2018 - 2024)

Year	Event
2018	Generative Adversarial Networks (GANs) gain traction, driving advances in realistic image and video generation.
2019	Transformer architectures dominate natural language processing and generative tasks, influencing many models beyond OpenAI’s developments.
2020	Multimodal AI systems emerge, combining text, image, and audio generation capabilities, showcasing potential in creative and scientific fields.
2021	DALL-E-like models and image generation models see broader adoption, enabling detailed and creative image synthesis from text prompts.
2022	Generative AI for audio advances, with models capable of generating music and realistic speech, expanding AI’s applications in entertainment.
2023	Text-to-3D generation becomes viable, with models generating 3D objects and animations from textual descriptions, aiding industries like gaming and design.
2024	Autonomous AI agents capable of generating text, code, images, and simulations work in tandem, revolutionizing productivity and creativity.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

CLASSICAL AI VS MODERN AI

Classical AI (Before 1990s)

- **The goal of classical AI was to explicitly represent human knowledge using facts and rules.**
- **Facts and rules had to be explicitly specified by people which makes them either limited or not well-defined!**

Key Techniques include:

- **Expert systems, Rule-based systems, Fuzzy systems, and Symbolic Reasoning**

Modern AI (Post-2000s):

- **Modern AI has become more effective due to advancements in data volume, statistical models, and computing power.**
- **It can autonomously infer rules, patterns, and irregularities from data.**

Key Techniques include:

- **Machine learning, computational intelligence (e.g., neural networks,).**

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

Agents & Rationality

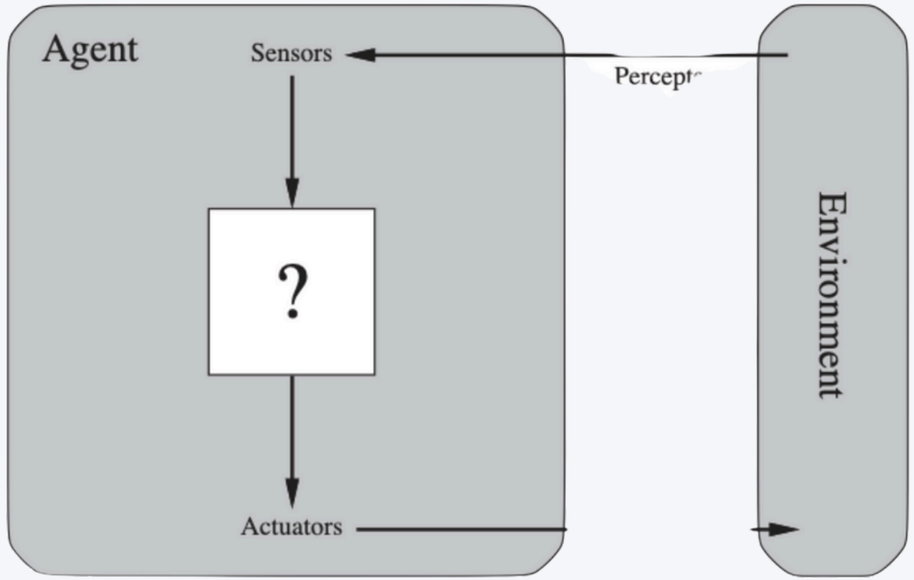
What is an Agent?

An agent **perceives** its environment via sensors and acts upon it via **actuators**.

Examples of Agents:

- **Human Agent:**
- Sensors: Eyes, ears, other organs.
- Actuators: Hands, legs, vocal tract.

- **Robotic Agent:**
- Sensors: Cameras, infrared range finders.
- Actuators: Motors, other mechanical parts.



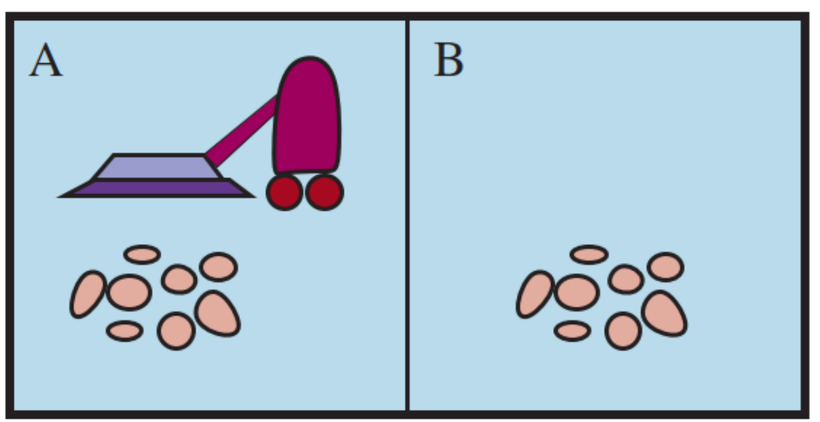
Key Design Elements: PAGE

- **Percepts:** Inputs from the environment.
- **Actions:** Outputs to affect the environment.
- **Goals**: Objectives to achieve.
- **Environments**: Contexts in which the agent operates.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Agent Example: Vacuum-Cleaner Agent

- Percepts: Location, contents (e.g., dirty/clean).
- Actions: Left, Right, Clean, NoOp.



- Agent Function:
  - If square is dirty → Clean.
  - Else → Move to another square.

- Key Question:
- What makes a good agent function? Who decides?

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Rationality & Performance

- **Performance Measure**:
  - Points for squares cleaned per time.
  - Penalties for unnecessary moves.

- **Rational Agent:**
  - Maximizes the performance measure based on percept history and built-in knowledge.

- Rational ≠ Successful:
  - Rational agents lack omniscience and cannot predict all outcomes.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

PEAS Framework

- PEAS (Task Environment Design):
- **Performance Measure**: Success criteria.
- **Environment**: External world.
- **Actuators**: Action mechanisms.
- **Sensors**: Perception mechanisms.

- **Example**: Autonomous Car:
- **Performance**: Safety, fuel efficiency, time optimization.
- **Environment**: Roads, traffic, pedestrians.
- **Actuators**: Steering, brakes, signals.
- **Sensors**: Cameras, GPS, sonar.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Environment Types

- Environment Types
    - Dimensions of Complexity:
    - Observable: Full vs. Partial.
    - Deterministic: Deterministic vs. Stochastic.
    - Episodic: Episodic vs. Sequential.
    - Static: Static vs. Dynamic.
    - Discrete: Discrete vs. Continuous.
    - Agents: Single vs. Multiple.
- Real World:
    - Partially Observable, Stochastic, Sequential, Dynamic, Continuous, Multi-Agent.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

The Nature of Environments

- **Fully observable vs. partially observable:**
  - Fully Observable Environment: The agent's sensors provide complete information about the environment's state, allowing it to choose actions without uncertainty.
  - Partially Observable Environment: The agent lacks full information due to noisy or inaccurate sensors or missing data, requiring it to make informed guesses about the environment.
- **Single-agent vs. multi-agent :**
  - Single-Agent Environment: The agent operates independently without interactions with other agents.
  - Multi-Agent Environment: Multiple agents interact, which can be competitive (e.g., chess: one agent's gain is another's loss) or cooperative (e.g., taxi-driving: avoiding collisions benefits all agents) or a mix of both.
- **Deterministic vs. stochastic:**
  - Deterministic Environment: The next state is entirely determined by the current state and the agent's actions; otherwise, it is stochastic (e.g., taxi driving with unpredictable events).
  - Uncertain Environment: Either not fully observable or not deterministic; stochastic environments involve probabilistic outcomes, while non-deterministic ones lack defined probabilities for outcomes.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

The Nature of Environments

- **Episodic vs. sequential :**
  - Episodic Environment: Each decision is independent of past and future decisions, simplifying the agent's task (e.g., spotting defective parts on an assembly line).
  - Sequential Environment: Current decisions affect future outcomes, requiring the agent to consider long-term consequences (e.g., chess, taxi driving).
- **Static vs. dynamic:**
  - Static Environment: Does not change while the agent is deliberating, simplifying decision-making as the agent need not monitor changes (e.g., a solved puzzle).
  - Dynamic Environment: Continuously changes, requiring the agent to act quickly as inaction is treated as a decision (e.g., real-time navigation). Semi-Dynamic: The environment stays static, but the agent's performance score changes over time.
- **Discrete vs. continuous:**
  - Static Environment: Remains unchanged during the agent's deliberation, simplifying decision-making as time and environmental updates are not a concern.
  - Dynamic Environment: Changes while the agent deliberates, requiring quick decisions as inaction is treated as doing nothing; Semi-Dynamic: The environment is static, but the agent's performance score changes over time.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Environment Examples

Task	Observable	Deterministic	Static	Episodic	Discrete	Agents
Chess (with clock)	Fully	Deterministic	Semi-dyn.	Sequential	Discrete	Multiple (Comp.)
Poker	Partially	Stochastic	Static	Sequential	Discrete	Multiple (Comp.)
Driving	Partially	Stochastic	Dynamic	Sequential	Continuous	Multiple (Coop.)
Image Recognition	Fully	Deterministic	Static	Episodic	Disc./Cont.	Single

The real world is...

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Solution Quality

- **Defining Solutions:**
    - Solutions must handle unstated assumptions with common-sense reasoning.
- **Types of Solutions:**
    - **Optimal**: Best solution by quality measure.
    - **Satisfying**: Adequate solution.
    - **Approx**. Optimal: Close to the best possible.
    - **Probable**: Likely a valid solution.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Types of Agents

- **Simple Reflex Agent:**
  - Acts based on current percept; ignores history.
- **Model-Based Reflex Agent:**
  - Maintains internal state based on percept history.
- **Goal-Based Agent:**
  - Actions aim to achieve defined goals.
- **Utility-Based Agent:**
  - Measures performance through utility functions.
- **Learning Agent:**
  - Adapts and improves through experience (online/offline).

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Simple reflex agents

- This type of agent is usually made of a general-purpose interpreter for condition-action rules to create rule sets for specific task environments.

    - For example: the vacuum world agent is a simple reflex agent (because its decision is based only on the current location and on whether that location contains dirt).

By ignoring the percept history, the agent program become very small compared to the corresponding table (from 4T to 4 possibilities).

This type relies on condition-action rules (Also called situation- action rules, productions, or if-then rules):

- if car-in-front-is-braking then initiate-braking.

Advantage:

- Simple enough!

Disadvantage:

- Limited intelligence ~ Actions depend only on the current information provided by their sensors!
- Works only if the environment is fully observable!
- Can lead to Infinite loops.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Model-based reflex agents

The most effective way to handle partial observability is to keep track of the part of the world that the agent can’t see now.

The agent maintains an **internal state** (i.e., memory) that depends on the percept history.

- Hence, has an access to information that is not currently available to their sensors.

The internal state can contain information about the state of the external environment.

This knowledge about “how the world works” is called a model of the world. An agent that uses such a **model** is called a model-based agent.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Goal-based agents

- Knowing something about the current state of the environment is not always enough to decide what to do.
  - For example: at a road intersection, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to go to.

The appropriate action for the agent will often depend on what its **goals** are, and so it must be provided with some goal information.

If a **long** sequence of actions is required to reach the goal, then search and planning must be implemented.

The goal-based agent is more flexible:

- We can simply specify a new goal, rather than re-programming all the rules.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Utility-based agents

Goals alone are not enough to generate high-quality behavior in most environments.

This is because there are often many sequences of actions that can result in the same goal being achieved.

- For example: many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.

This type of agent programs appear when we design decision making agents that must handle the **uncertainty** inherent in stochastic or partially observable environments.

This type of agent makes its decisions based on the **maximum utility of its choices**. Its focus is not only on achieving goals but to find the best alternative/way to reach that particular goal.

To do so, we utilize a **utility function** of choice, which maps a state or sequence of states to a value, to rate each possible solution against the general performance measure.

A ***rational*** utility-based agent always chooses the action that maximizes the expected utility of the action outcomes.

Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Learning agents

his type of agent operates in initially unknown environments and become more knowledgeable over time so as to improve performance.

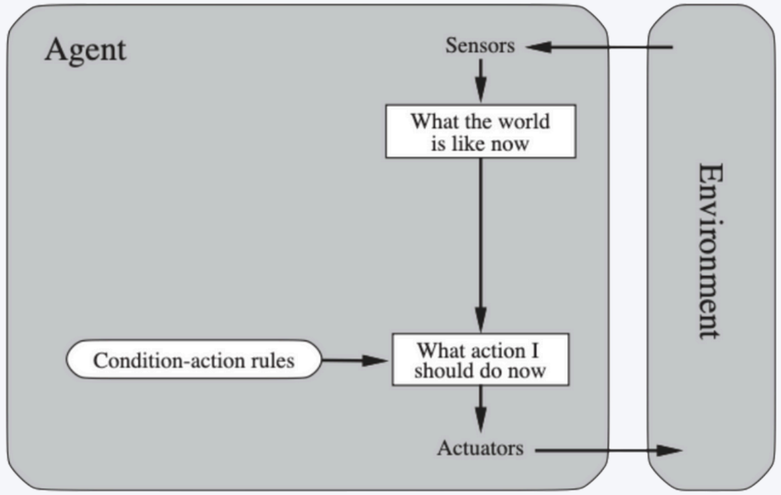
Consists of four main components:

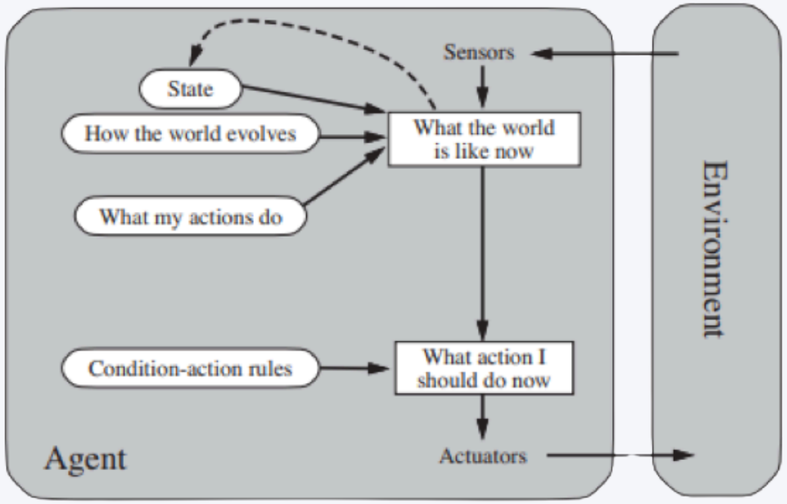
- **Critic**: Evaluates how well the agent is doing wrt the external performance standard.
- **Learning element::** Makes improvements
- **Performance element:** Contains the knowledge about the environment and selects the actions.
- **Problem generator:** Suggests actions that will lead to new and informative experiences.

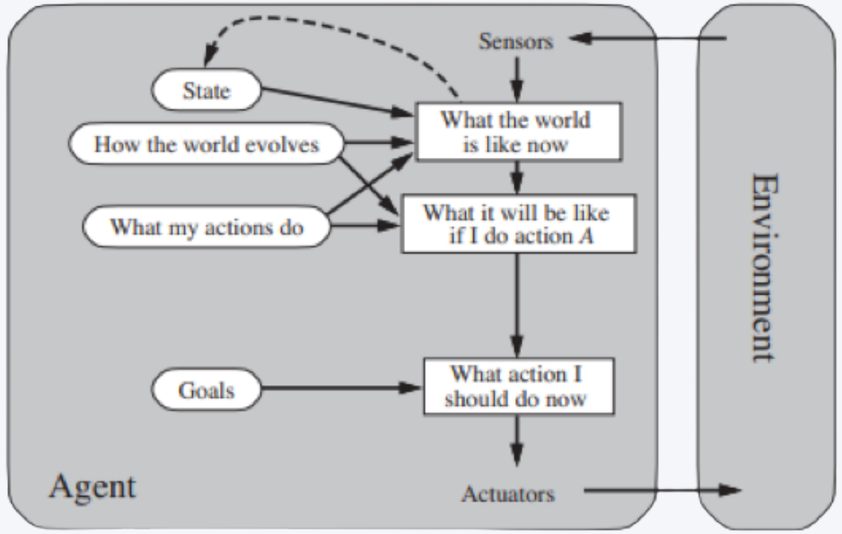
For example, from Google Assistant to other predictive searches, all use learning agents to adapt and learn about the user and make accurate suggestions, and recommendations, and deliver appropriate ads.

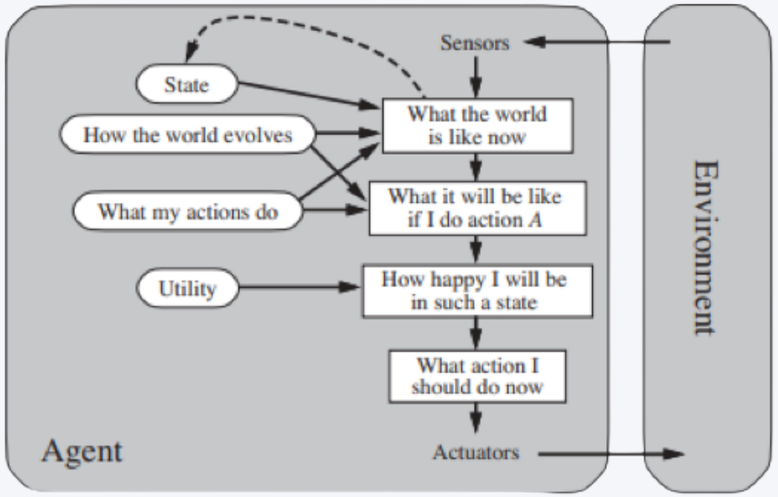
Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

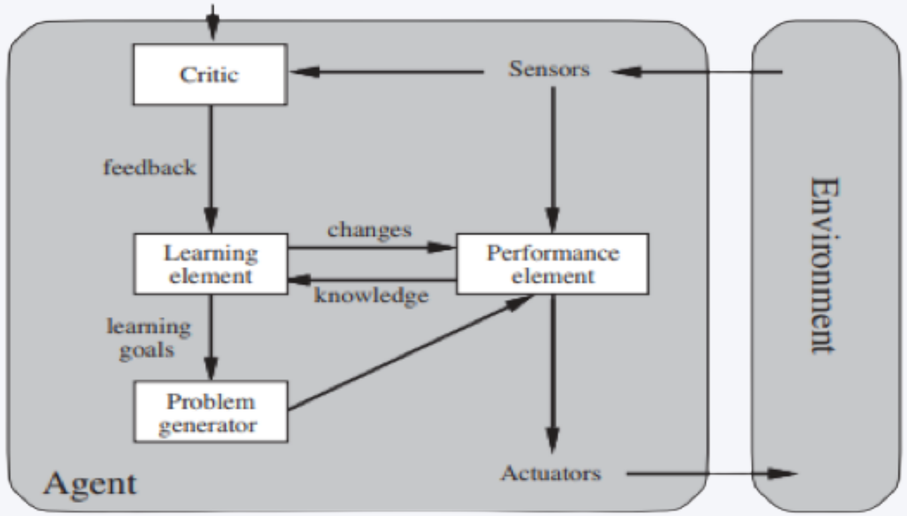
Agent Logic & Types









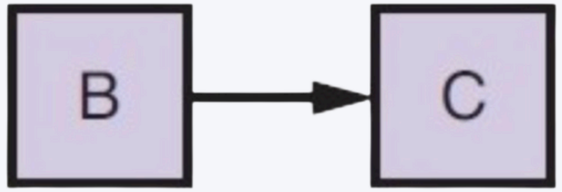


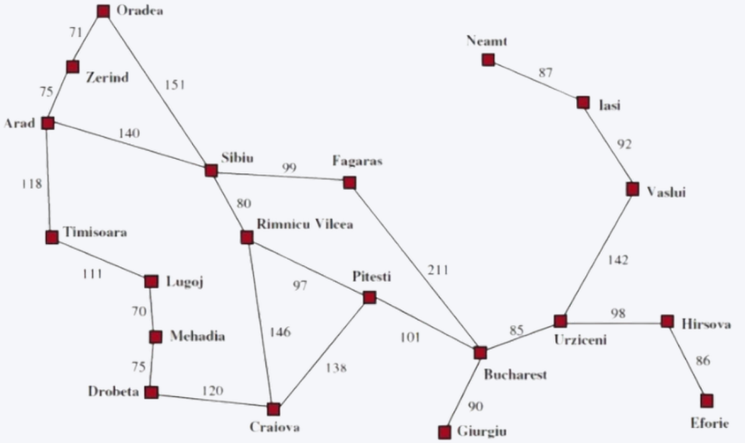
Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Agent Representations

Atomic representation

- Each state of the world is indivisible (it has no internal structure).
  - For example: the problem of finding a driving route from one end of a country to the other via some sequence of cities.





For the purposes of solving this problem, it may suffice to reduce the state of world to just the name of the city we are in— a single atom of knowledge.

The algorithms underlying search and game-playing (see later this semester) all work mostly with atomic representations.

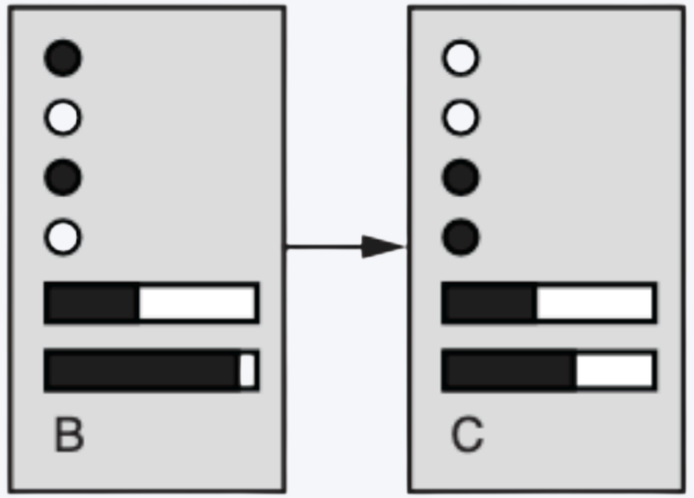
Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

Agent Representations

Factored representation

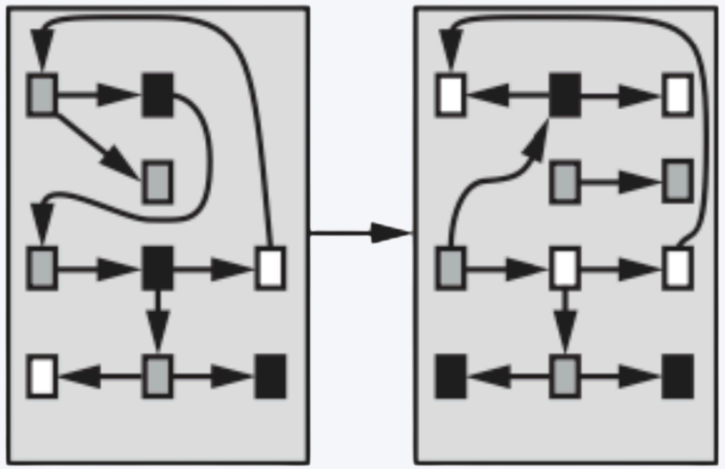
- A factored representation splits up each state into a fixed set of variables or attributes, each of which can have a value.

- Two different factored states can share some attributes such as being at some particular GPS location (see back circles in right figure).



Structured representation

- Structured representation describes the world using variables while capturing knowledge and reasoning.



Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

PHILOSOPHY OF AI

IS AI **POSSIBLE**?

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

Is AI Possible ?

There are different opinions...

...some are slightly **positive:**

- “every feature of intelligence can be so precisely described that a machine can be made to simulate it” *(McCarthy et al, 1955)* ...

and some lean towards the **negative**:

- “AI stands not even a ghost of a chance of producing durable results” *(Sayre, 1993)*

It’s all in the definitions:

- what do we mean by “**thinking**” and “**intelligence**”?

Turing's Key Contribution

Paper: "Computing Machinery and Intelligence" – Alan Turing (1950).

Key Contributions:

- Introduced the "imitation game" (Turing Test) to define intelligence.
- Discussed objections to AI, including nearly all objections raised since.
- A must-read foundational work in AI and philosophy!

TURING’s Key Objections

01

The **Theological Objection:**

Claim: Thinking requires an immortal soul, which only humans have.

02

The **"Heads in the Sand" Objection:**

Claim: The idea of machines thinking is too dreadful; we hope it’s not true.

03

The **Mathematical Objection:**

Claim: Based on Gödel’s incompleteness theorem, there are limits to what machines can do logically.

04

The **Argument from Consciousness:**

Claim: Machines cannot feel emotions like pleasure, grief, or anger.

05

Arguments from Various Disabilities:

Claim: Machines will never do certain human things like:

- Be kind, resourceful, beautiful, friendly.
- Have a sense of humor or tell right from wrong.
- Fall in love or enjoy strawberries and cream.

06

Lady Lovelace’s Objection:

Claim: Machines cannot originate anything; they only do what we program them to do.

07

Argument **from Continuity in the Nervous System:**

Claim: The nervous system’s continuous behavior cannot be mimicked by discrete-state machines.

08

Argument **from Informality of Behavior:**

Claim: Human behavior lacks strict rules; machines with fixed rules cannot replicate this.

09

The Argument from Extrasensory Perception (ESP):

Scenario: A man with telepathic abilities guesses the suit of cards more accurately than a machine.
Claim: Machines cannot replicate such abilities.

Turing’s Comment: This was the strongest argument in his view, given the statistical evidence for ESP at the time.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

Are Turing’s objections still relevant, and can machines truly "**think**" or "**feel**"?

أكاديمية طويق
Tuwaiq Academy

Assignment: Reflection on Turing’s Objections.

“**Topic**: Are Turing’s objections still relevant, and can machines truly "**think**" or "**feel**"?”

Write a one-page research responding this question giving your perspective and what you have witnessed in today”s AI !

Strong AI

THE BRAIN REPLACEMENT EXPERIMENT

Done by Searle (1980) and Moravec (1988)..

- Suppose we gradually replace each neuron in your head with an electronic copy...

...what will happen to your mind, your **consciousness**?

- Searle argues that you ***will gradually feel dislocated*** from your body
- Moravec argues you **won't** notice anything

THE TECHNOLOGICAL SINGULARITY

Will AI lead to superintelligence?

“...ever accelerating progress of technology and changes in the mode of human life, which gives the appearance of approaching some essential singularity in the history of the race beyond which human affairs, as we know them, could not continue” **(von Neumann, mid-1950s)**

“We will successfully reverse-engineer the human brain by the mid-2020s. By the end of that decade, computers will be capable of human-level intelligence.” **(Kurzweil, 2011)**

“There is not the slightest reason to believe in a coming singularity.” **(Pinker, 2008)**

CLASSICAL SEARCH ALGORITHMS

WHAT IS SEARCH IN AI?

- Often we are not given an algorithm to solve a problem, but only a specification of a solution — **we have to search for it.**

- A **typical problem** is when the agent is in one state, it has a set of deterministic actions it can carry out, and wants to get to a goal state.

- Many **AI** problems can be abstracted into the problem of finding a path in a directed graph.

- Often there is more than one way to represent a problem as **a graph**.

State-Space Search:

- The agent explores possible states and actions to find a solution.
- The goal is to find a path from the start state to the goal state.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

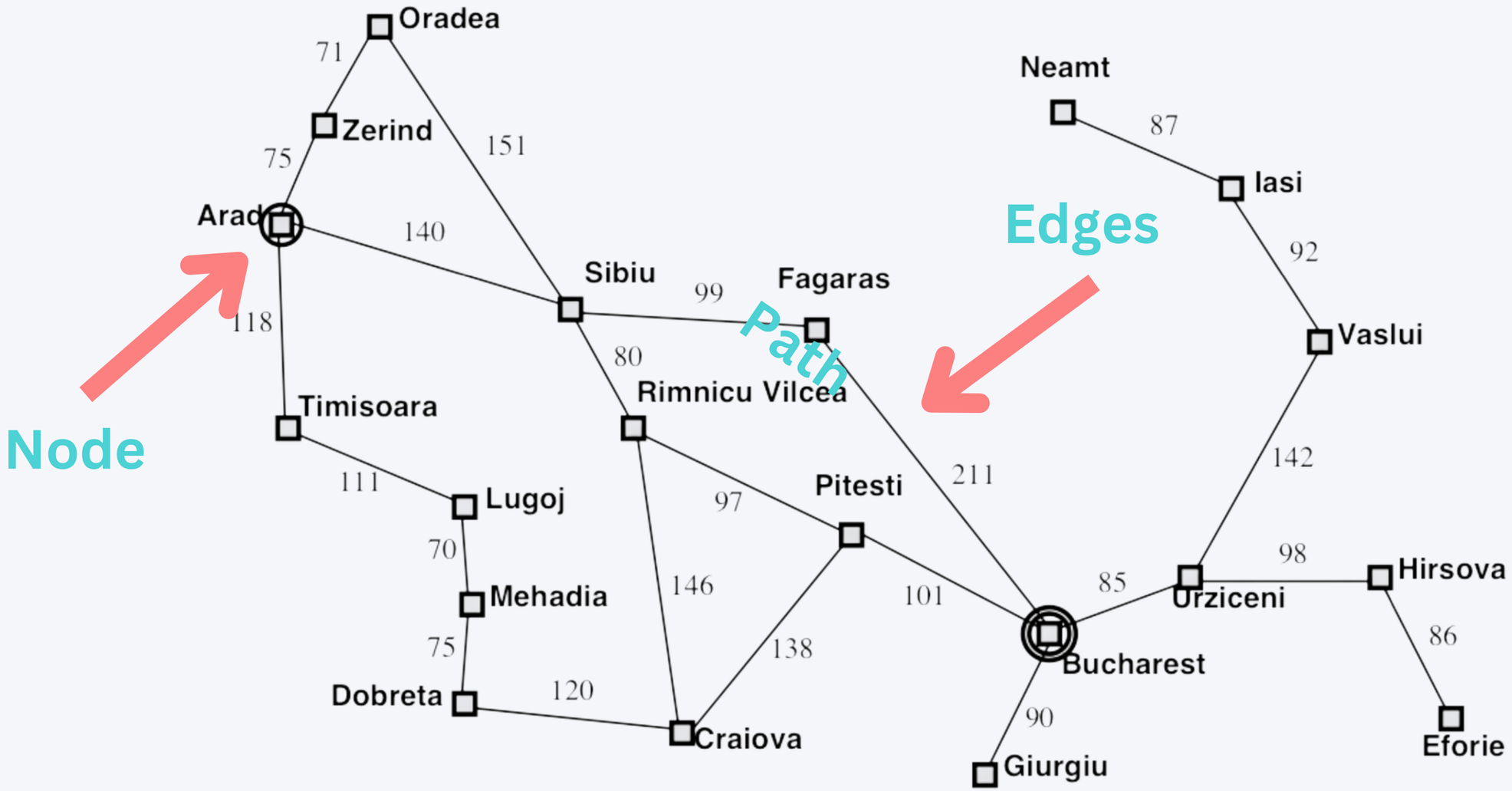
GRAPHS AND STATES:

Problems are often represented as graphs.

- For example, consider cities as **nodes** and roads connecting them as **edges**. A **path** represents a series of roads leading from one city to another.
- **Nodes (States)**: Represent the points or conditions in the problem.
- **Edges (Actions)**: Transitions or steps that connect nodes.
- **Path:** A sequence of edges leading from the start node to the goal node. Problems are often represented as graphs.

EXAMPLE: TRAVEL IN ROMANIA

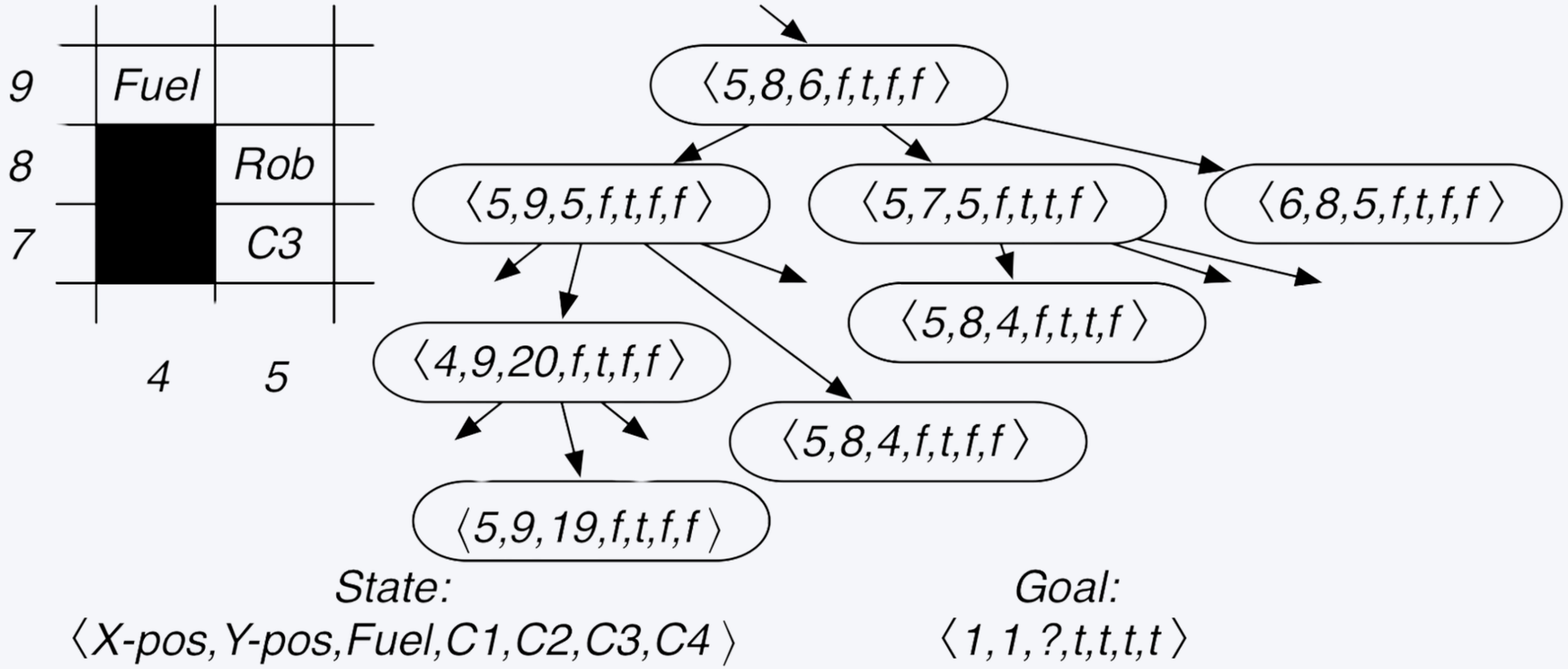
We want to drive from Arad to Bucharest in Arad



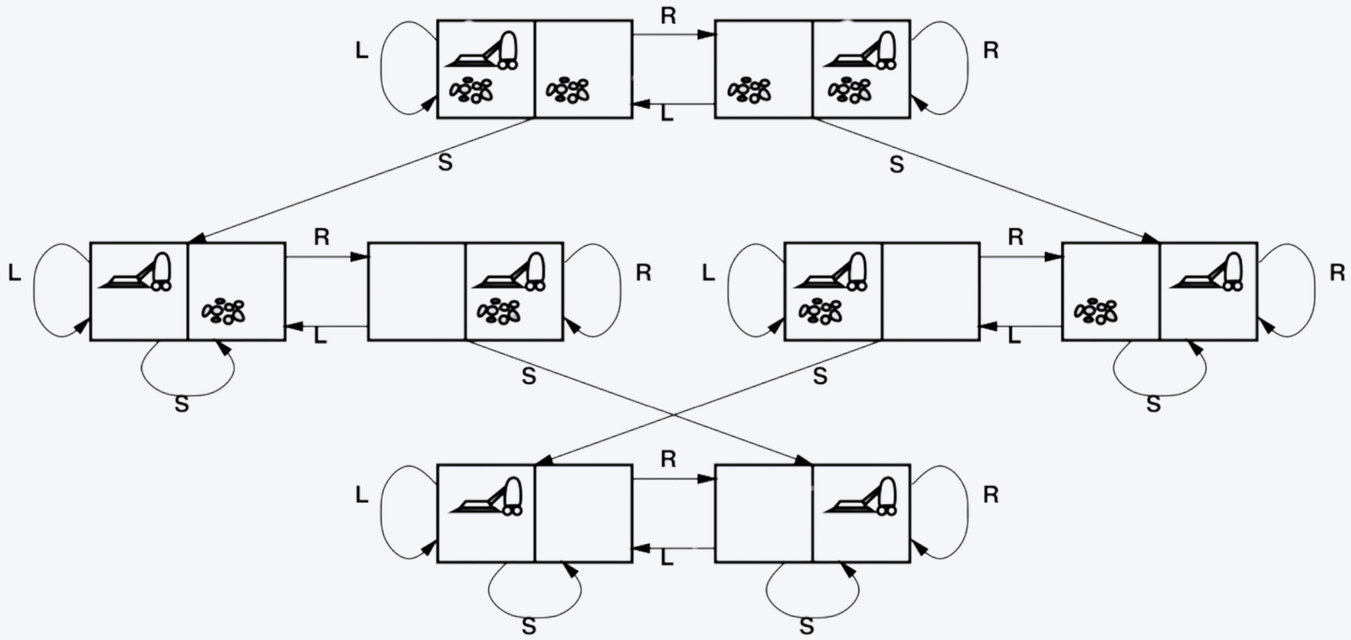
Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

EXAMPLE: GRID GAME

Grid game: Rob needs to collect coins C1, C2, C3, C4 ,without running out of fuel, and end up at location (1,1):



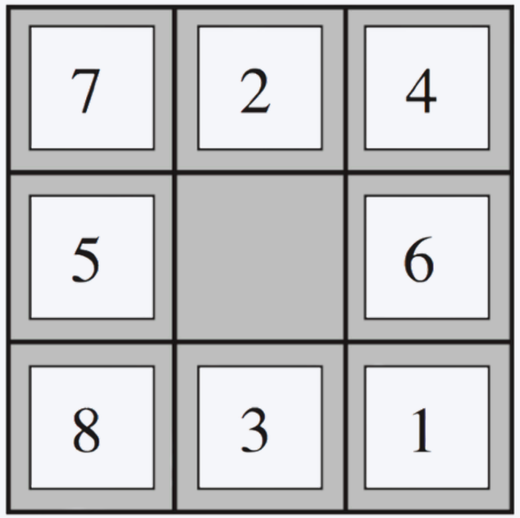
EXAMPLE: VACUUM-CLEANING AGENT



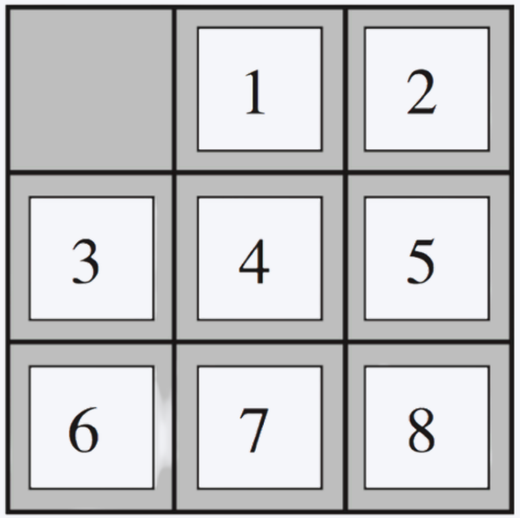
States	*[room A dirty?, room B dirty?, robot location]*
Initial state	*any state*
Actions	*left, right, suck, do-nothing*
Goal test	*[false, false, –]*

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

EXAMPLE: THE 8-PUZZLE



Start State

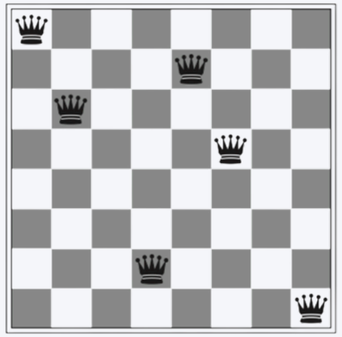


Goal State

States	*a 3 x 3 matrix of integers*
Initial state	*any state*
Actions	*move the blank space: left, right, up, down*
Goal test	*equal to the goal state*

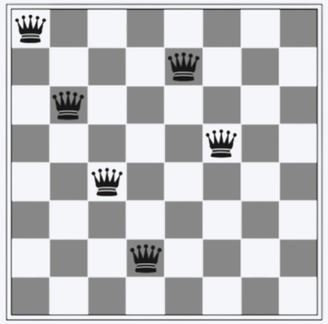
Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

EXAMPLE: THE 8-QUEENS PROBLEM



States	*any arrangement of 0 to 8 queens on the board*
Initial state	*no queens on the board*
Actions	*add a queen to any empty square*
Goal test	*8 queens on the board, none attacked*

This gives us $64 \times 63 \times \cdots \times 57 \approx 1.8 \times 10^{14}$ possible paths to explore!



States	*one queen per column in leftmost columns, none attacked*
Initial state	*no queens on the board*
Actions	*add a queen to a square in the leftmost empty column, make sure that no queen is attacked*
Goal test	*8 queens on the board, none attacked*

Using this formulation, we have only 2,057 paths!

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

TYPES OF SEARCH ALGORITHMS:

Uninformed Search:

These algorithms do not use additional information about the goal.

Depth-First Search (DFS):

- Imagine exploring a maze by always turning left until you hit a dead end, then backtracking to try a new path. This is how DFS operates. Explores one path deeply before backtracking.

Breadth-First Search (BFS):

- Explores all nodes at a given depth before going deeper.

Uniform-Cost Search (UCS):

- Explores paths with the lowest cost first.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

TYPES OF SEARCH ALGORITHMS:

Heuristic Search:

- These algorithms use a **heuristic (a guiding function)** to estimate the closeness to the goal.

What is a Heuristic?

- A heuristic is a function that estimates how close a node is to the goal.
-

Examples of Heuristics:

- **Straight-Line Distance (SLD):** The shortest distance between two points in navigation.
- **Manhattan Distance:** Used in grid-based problems, sums the horizontal and vertical steps to the goal.
- **Misplaced Tiles**: For the 8-puzzle, counts how many tiles are not in their goal positions.

Greedy Best-First Search:

- Chooses the node that appears closest to the goal.

A*Search:

- Combines UCS and heuristic information for optimal pathfinding.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

HOW DO WE SEARCH IN A GRAPH?

A generic search algorithm:

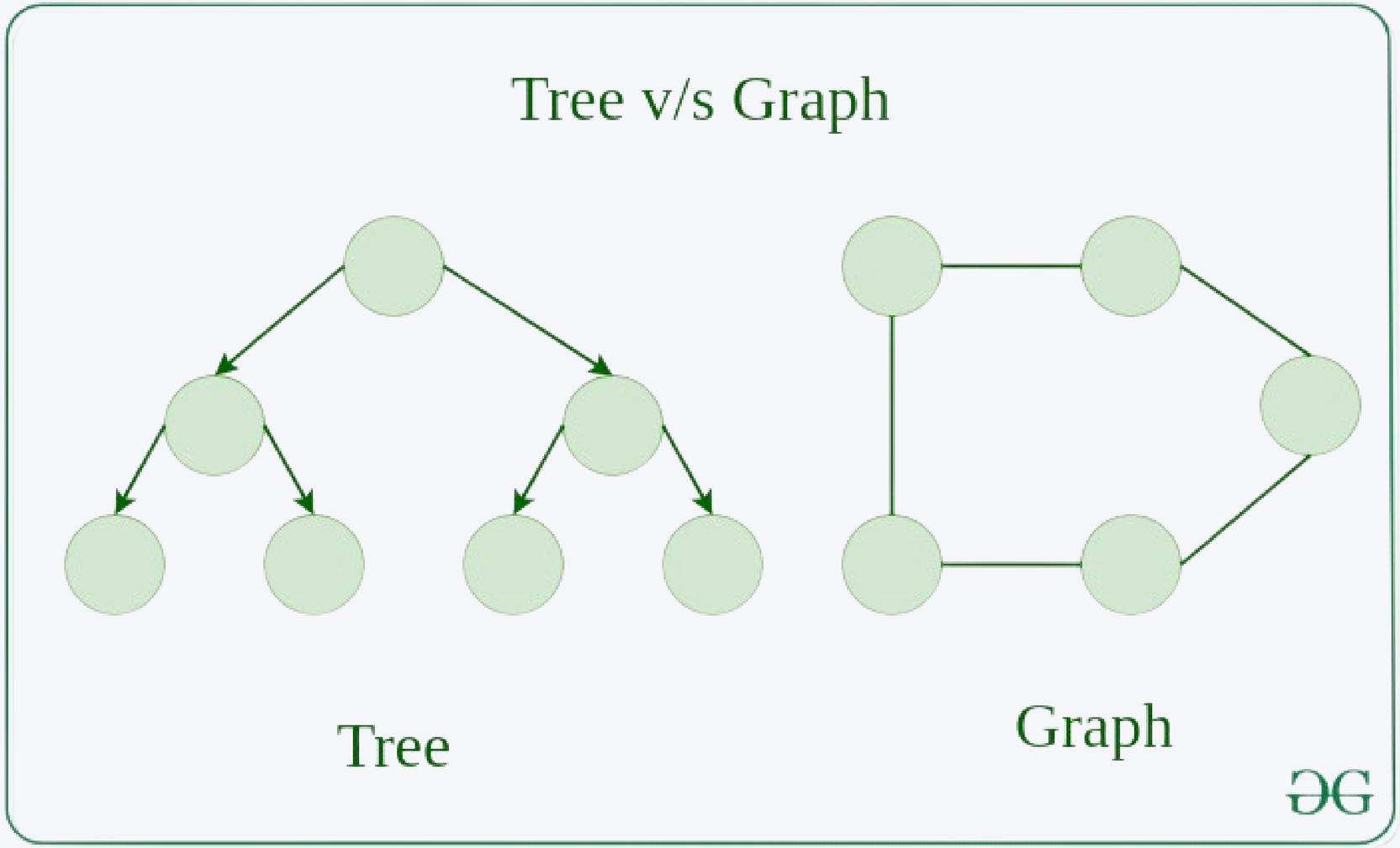
- Given a graph, start nodes, and a goal description, incrementally explore paths from the start nodes.
- Maintain a **frontier** of nodes that are to be explored.
- As search proceeds, the frontier expands into the unexplored nodes
- until a goal node is encountered.
- The way in which the frontier is expanded defines the search strategy.

Tree Search vs. Graph Search:

- Tree Search: Does not check if a node has been visited, which might lead to revisiting states.
- Graph Search: Keeps track of visited nodes to avoid duplicating work and loops.

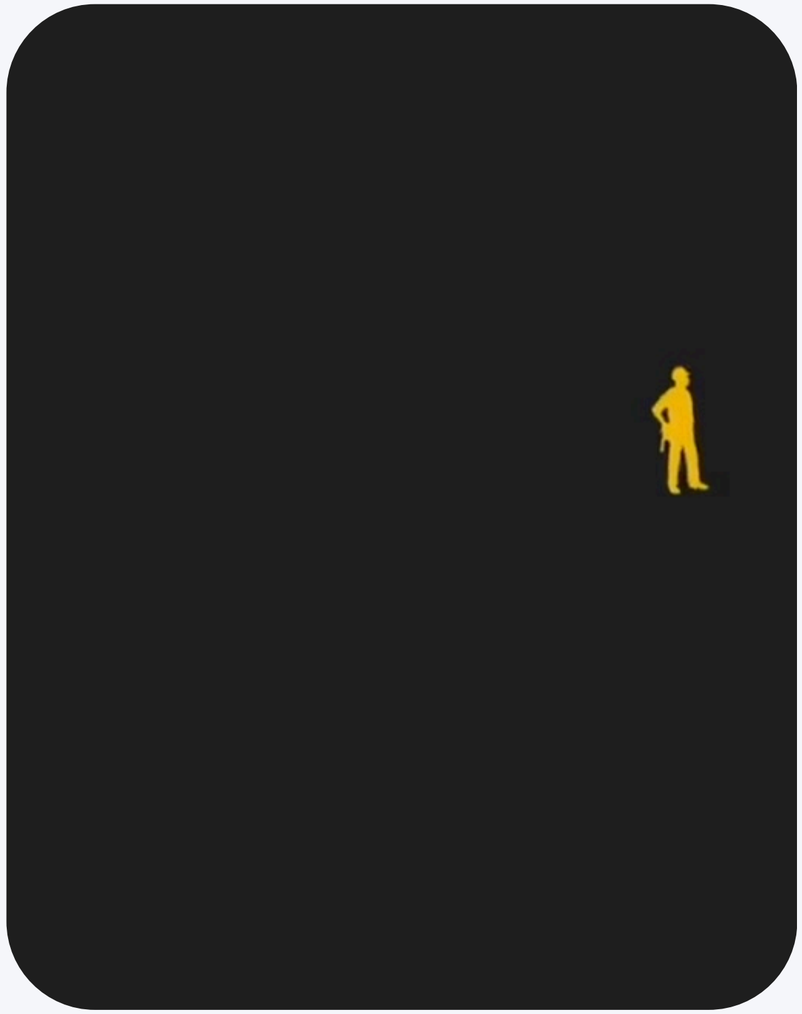
Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

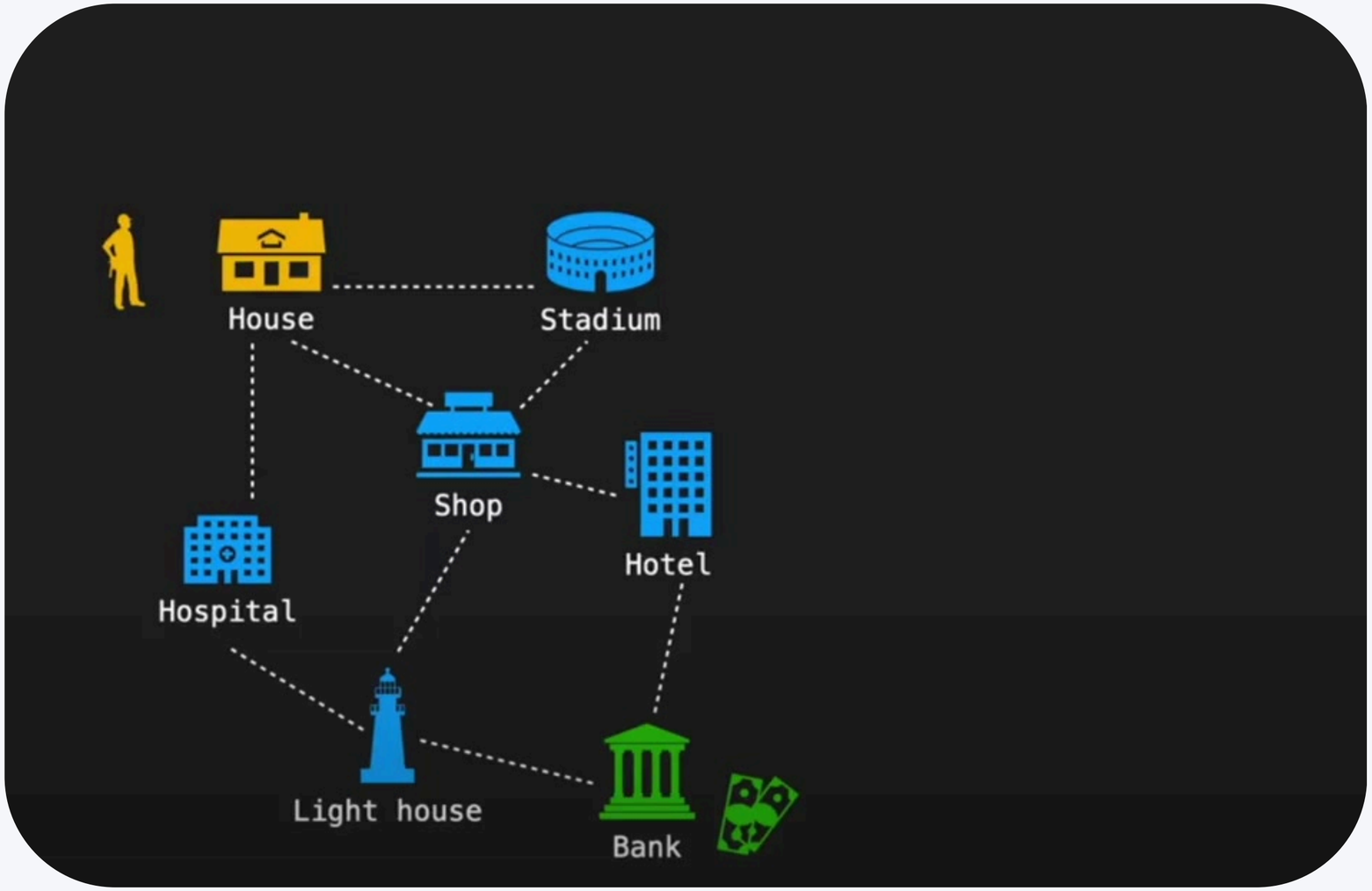
TREE SEARCH VS. GRAPH SEARCH:



Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

GREEY BEST FIRST SEARCH | REAL WORLD EXAMPLE



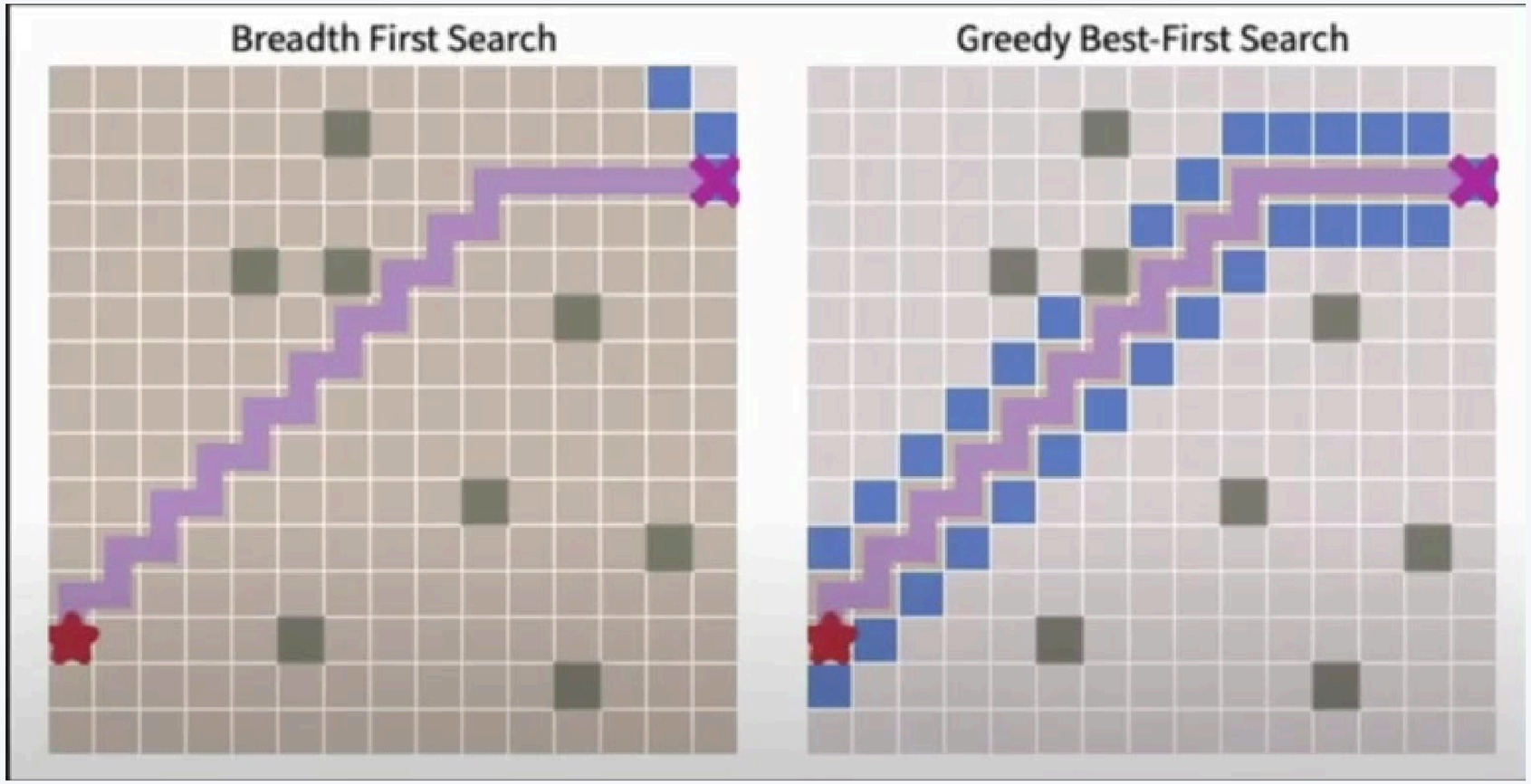


Uninformed Search
No info about search space
Depth , Breadth First Search

Heuristic of Informed Search
No heuristic function helps he search " how far we are from the goal?" and "how to reach the goal?"
Greedy Best First Search

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

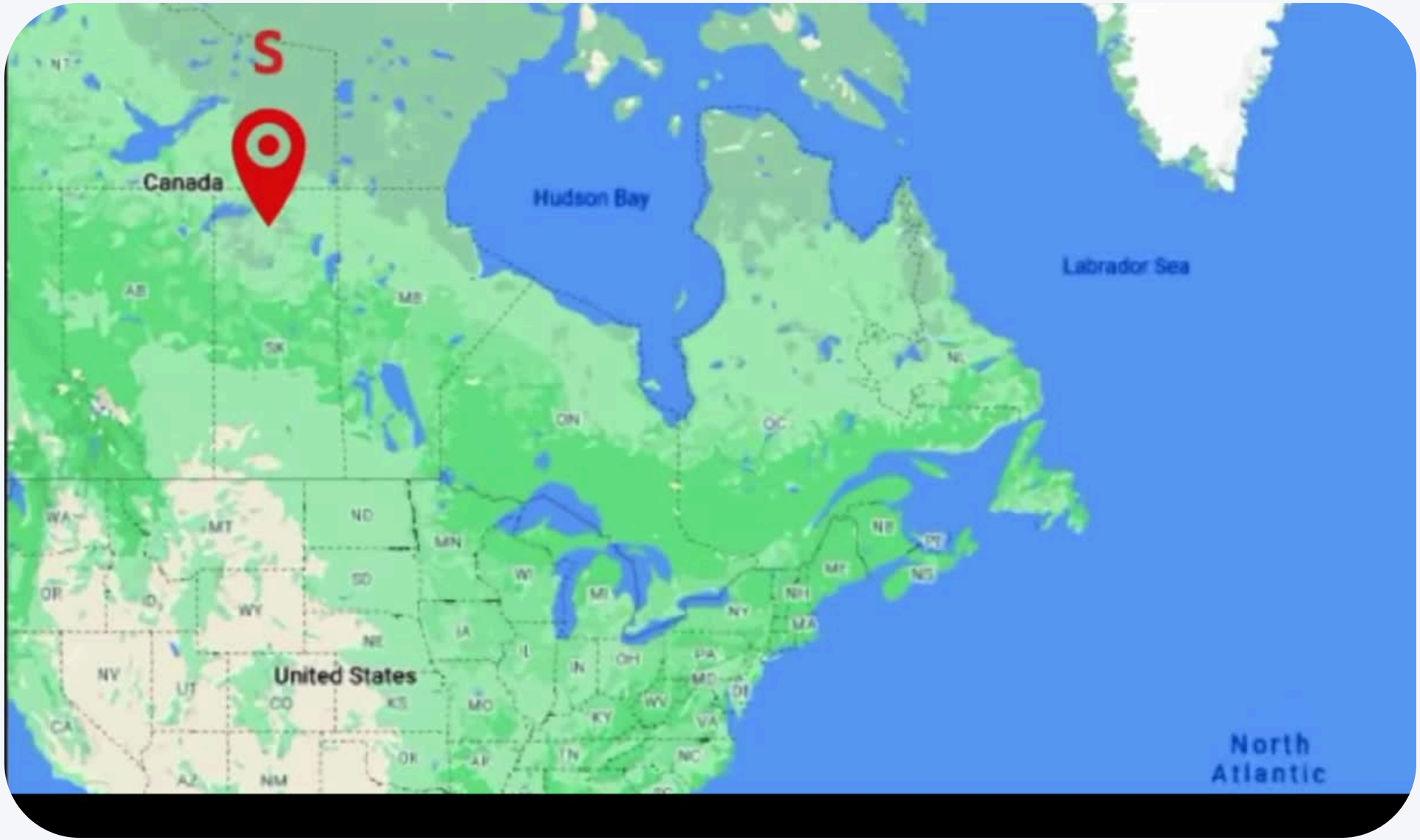
GREEY BEST FIRST SEARCH | REAL WORLD EXAMPLE



Faster and much efficient

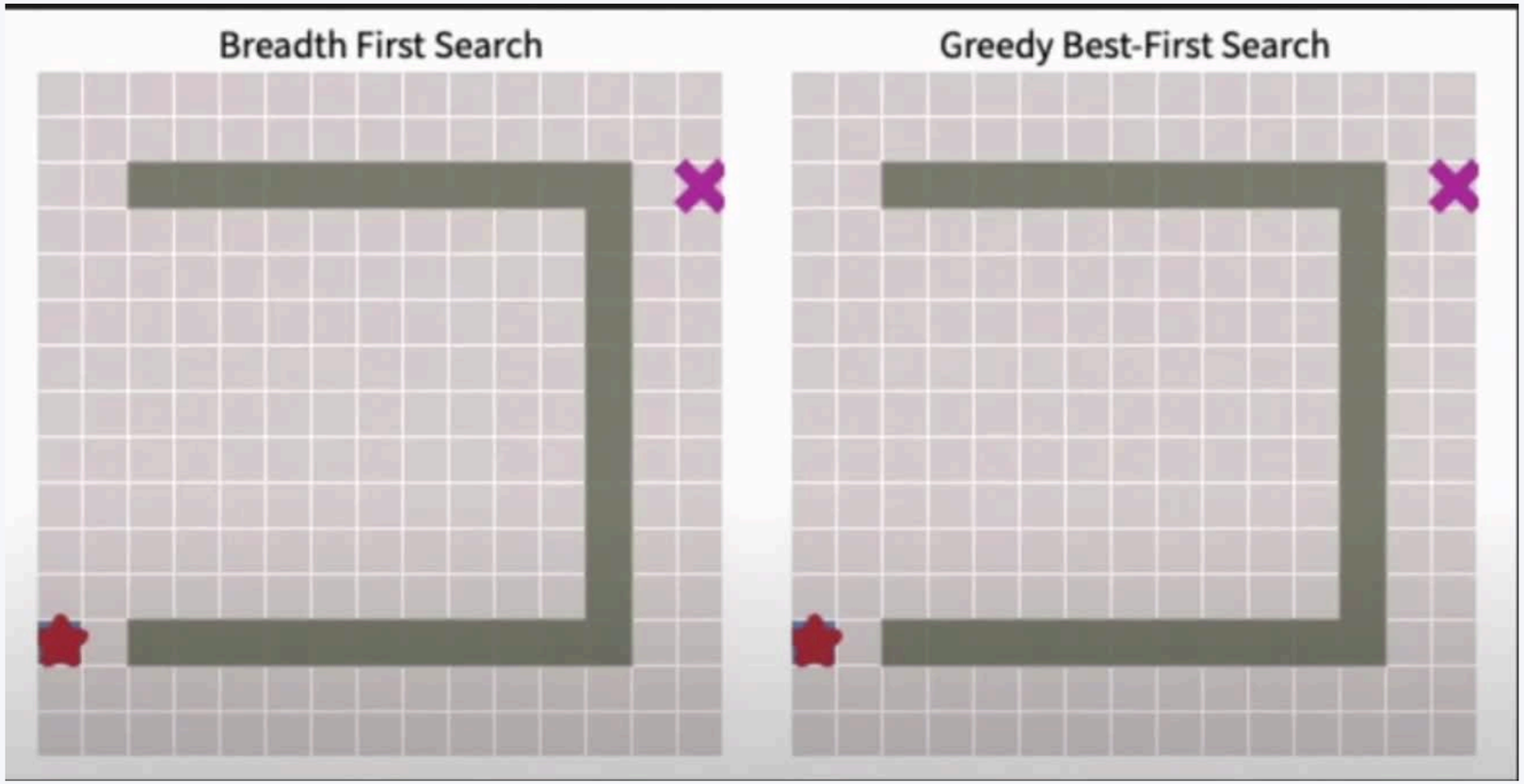
Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

GREEY BEST FIRST SEARCH | REAL WORLD EXAMPLE



Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

GREEY BEST FIRST SEARCH | DOES IT FAIL ?



Not Optimal

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

INTERACTIVE DEMO

Explore Search Algorithms:

- Use the interactive demo tool to visualize different search strategies: **Pathfinding.js Demo**.
- **Experiment** with the algorithms and heuristics on a grid-based search problem.
- Observe how changes in **heuristic** affect the performance and result.

Live Demonstration:

- Consider demonstrating the tool live during the lecture to show real-time performance of each algorithm.

Prepared Examples:

- Prepare a few pre-set scenarios to highlight key differences between algorithms, such as:
  - Comparing DFS, BFS, and A* on a maze with multiple solutions.
  - Illustrating the impact of a good vs. poor heuristic in A* search.

Activity:

- Encourage students to test different scenarios to see how search algorithms behave.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

CONCLUSION

Search as the Foundation of AI:

- **Search** is a fundamental concept in AI because it models decision-making and problem-solving.
- Many real-world problems—from navigation systems to robotics—rely on search algorithms to find efficient solutions.

- These methods serve as a bridge to more complex AI tasks such as learning, optimization, and planning.

Applications of Search:

- **Robotics**: Planning paths for autonomous robots.
- **Gaming**: Designing AI opponents that plan moves intelligently.
- **Optimization**: Finding optimal resource allocations in logistics.

Why Search Matters:

- Search algorithms are the starting point for understanding broader AI techniques. By mastering them, you build a strong foundation for more advanced topics like machine learning and neural networks.

Poole & Mackworth, 2010 (http://artint.info/slides/), and Russell & Norvig, 2005 (http://aima.cs.berkeley.edu/)

أكاديمية طويق
Tuwaiq Academy

Any Questions ?