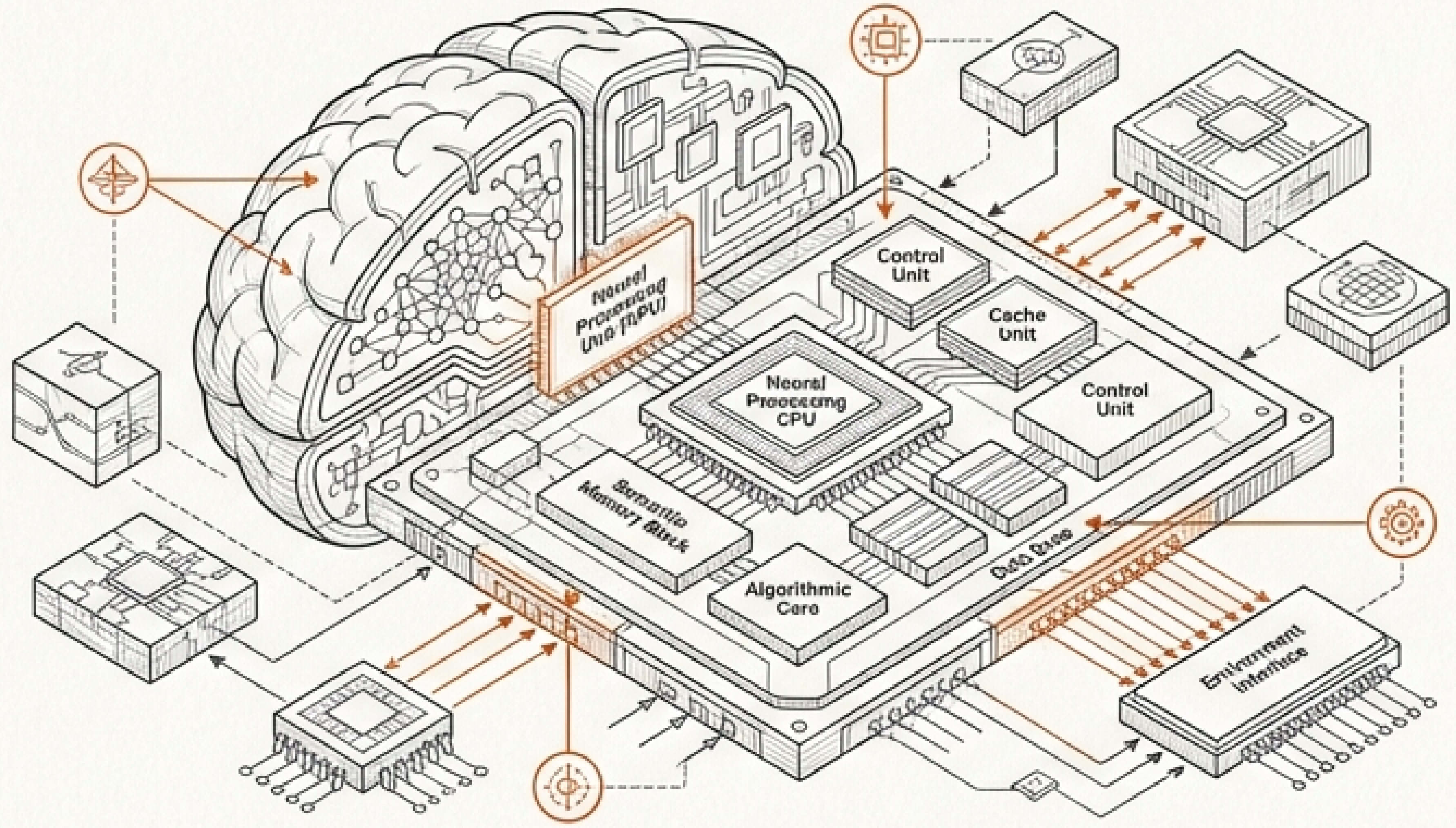
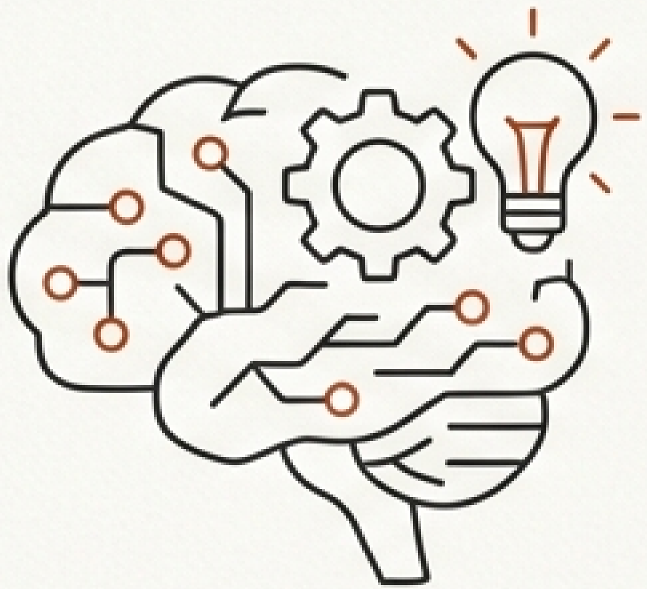
Deconstructing Intelligence: The Core Principles of AI



A foundational overview of the agents, environments, and search strategies that define modern Artificial Intelligence.

An introductory guide for students of computer science.

Artificial Intelligence is Man-Made Thinking Power



What is AI?

A branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and are able to make decisions.

AI enables a machine to have human-based skills such as **learning**, **reasoning**, and **problem-solving** without being explicitly pre-programmed for every task.

Why is it important?

To create software and devices that solve real-world problems with speed and accuracy.

Key Applications

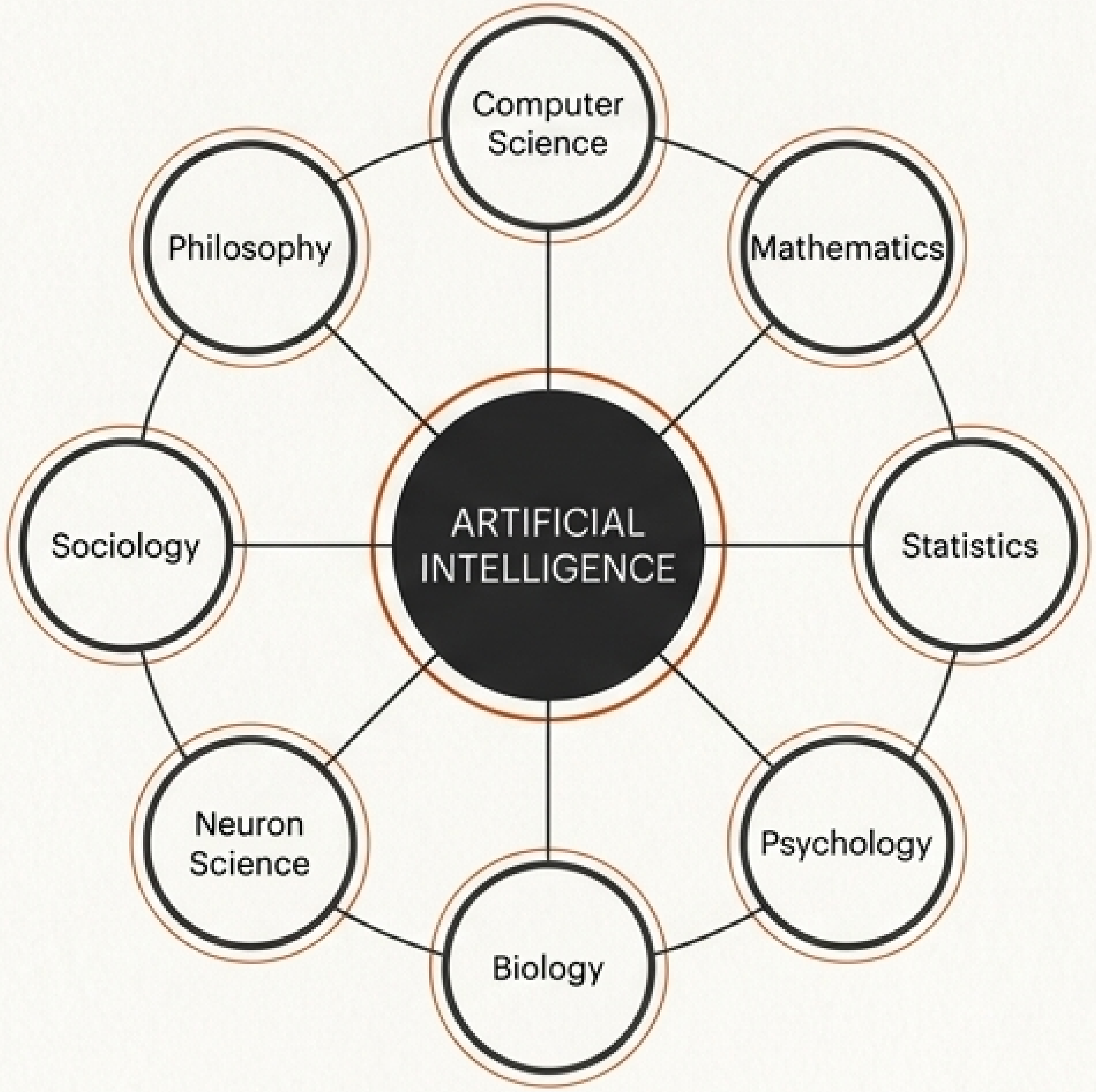
- **Healthcare:** Faster, better diagnosis.
- **Transport:** Self-driving cars for safer journeys.
- **Assistance:** Personal virtual assistants like Siri and Google Assistant.
- **Exploration:** Robots for risky environments where humans cannot go.

The Quest to Replicate Intelligence

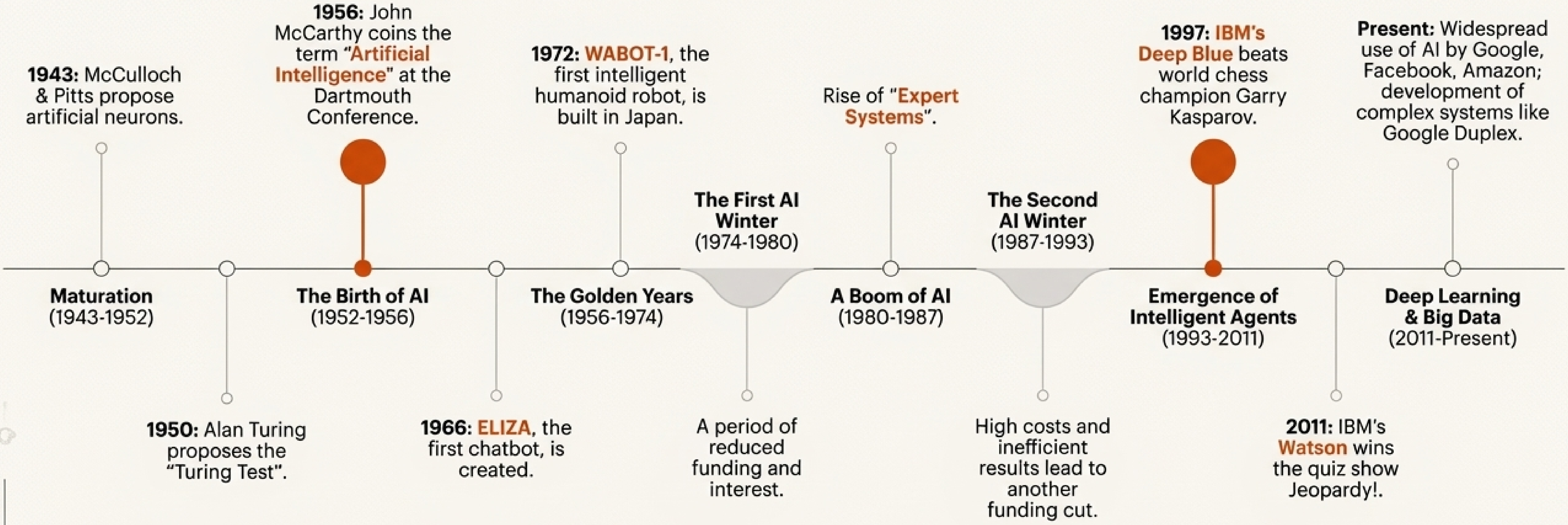
The Primary Goals of AI

1. Replicate human intelligence.
2. Solve knowledge-intensive tasks (e.g., proving theorems, planning surgery).
3. Build machines that can perform tasks requiring human intelligence (e.g., playing chess, driving in traffic).
4. Create systems that can learn, demonstrate, and explain on their own.

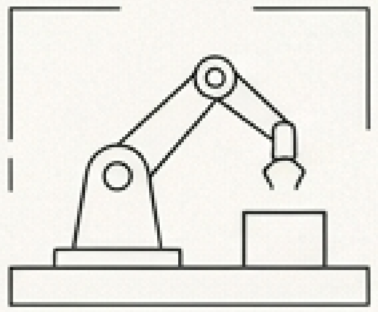
The Foundations of AI



A Timeline of Key AI Milestones



The Spectrum of Artificial Intelligence



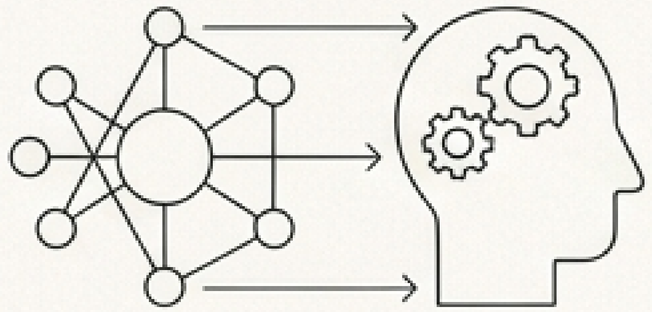
Narrow AI (Weak AI)

Able to perform a dedicated task with intelligence. Cannot perform beyond its specific limitations.

Status: The most common and currently available AI.

Examples: Apple's Siri, self-driving cars, image recognition, purchasing suggestions on e-commerce sites.

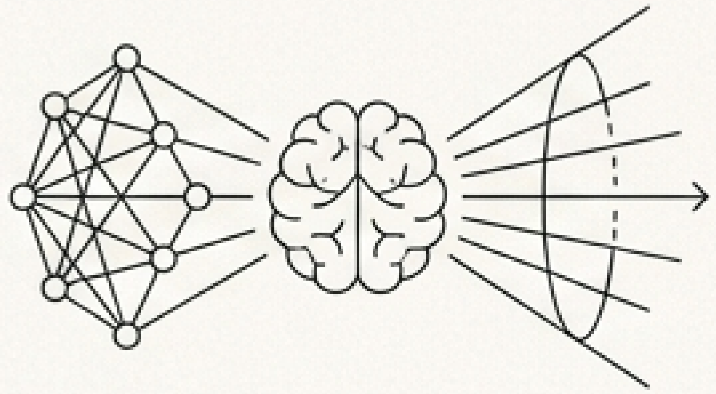
WE ARE HERE



General AI

Could perform any intellectual task with the same efficiency as a human. A system that could be smarter and think like a human on its own.

Status: Still under research; no such system currently exists.



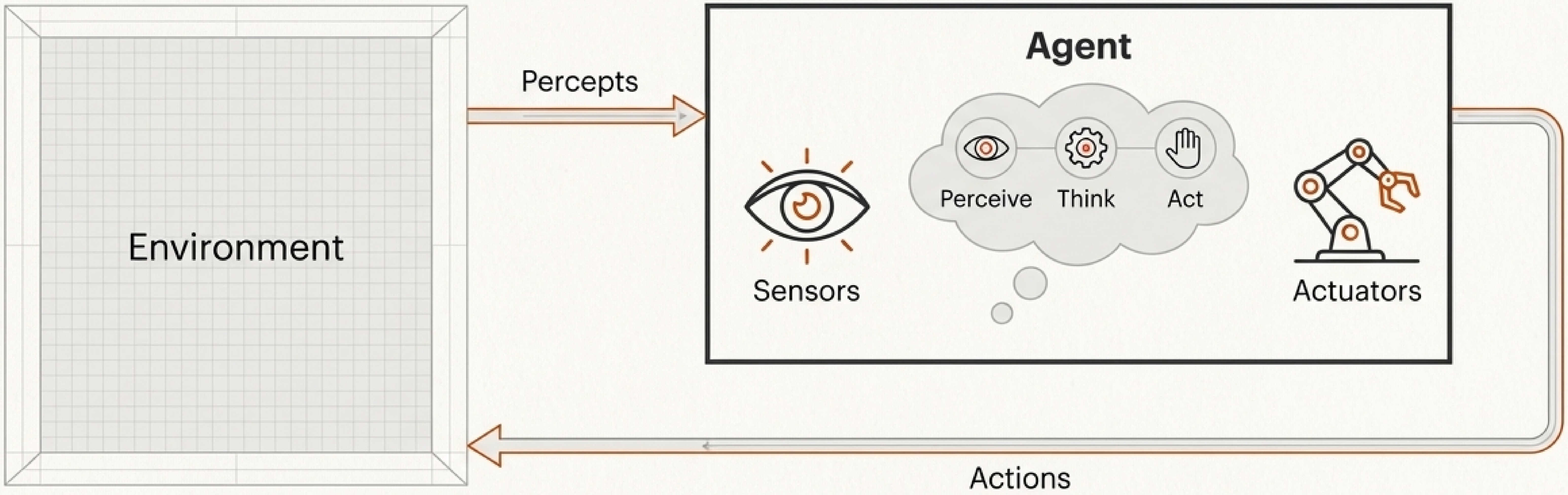
Super AI

A level of intelligence at which machines could surpass human intelligence and perform any task better than a human, with cognitive properties.

Status: A hypothetical concept; an outcome of General AI.

The Core of AI: The Intelligent Agent

An agent is anything that can perceive its environment through **sensors** and act upon that environment through **actuators**.



Human Agent: Sensors (eyes, ears), Actuators (hands, legs).
Robotic Agent: Sensors (cameras, infrared), Actuators (motors, arms).
Software Agent: Sensors (keystrokes, file contents), Actuators (display output on screen).

Analysing an Agent with the PEAS Framework

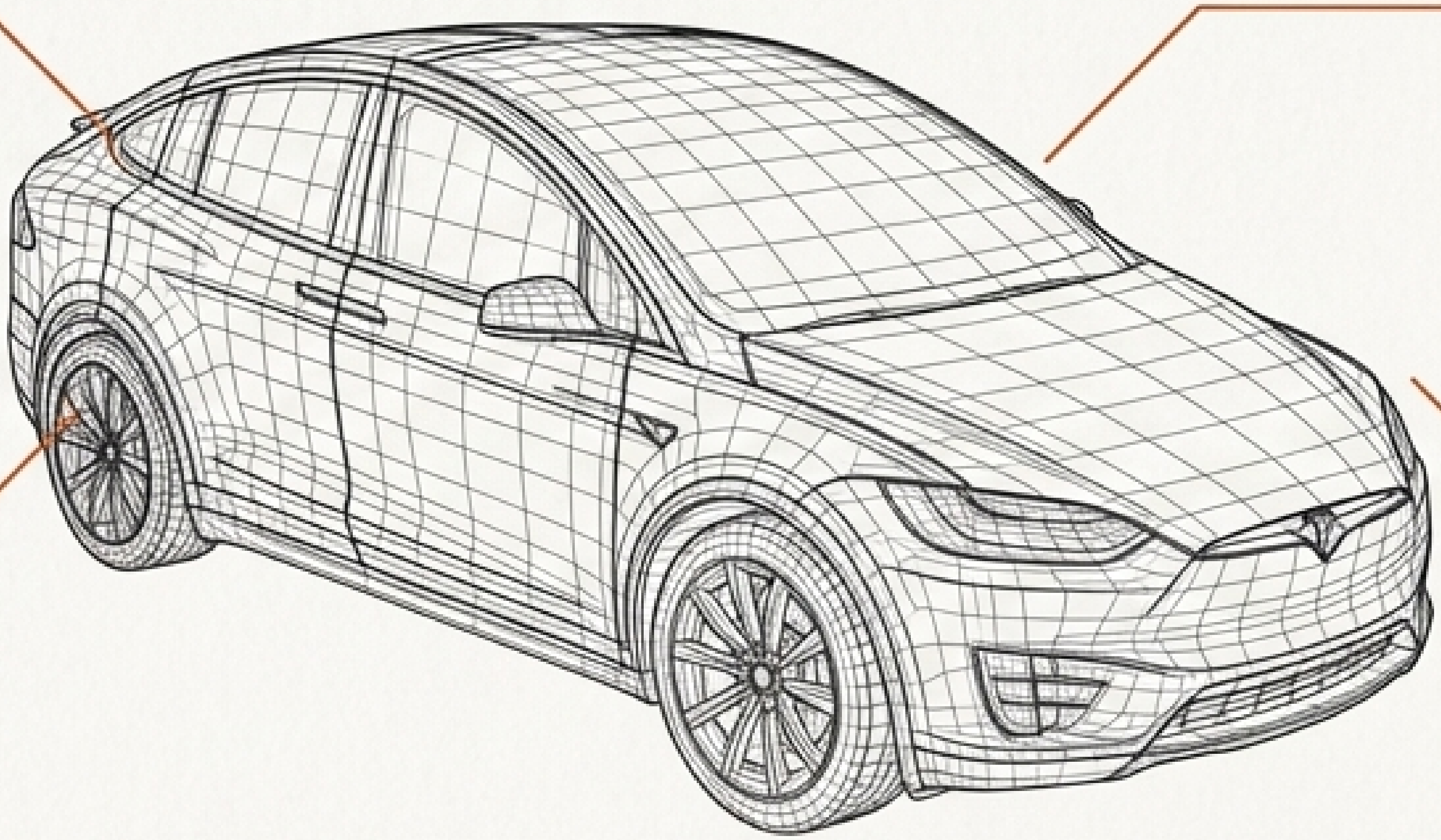
PEAS is a model used to define a rational agent. It groups an agent's properties into four categories.

P

Performance Measure: Safety, journey time, legal driving, passenger comfort.

E

Environment: Roads, other vehicles, pedestrians, road signs.



A

Actuators: Steering wheel, accelerator, brake, signal, horn.

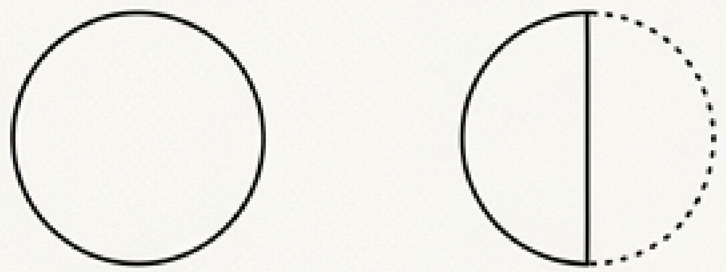
S

Sensors: Cameras, GPS, speedometer, sonar, accelerometer.

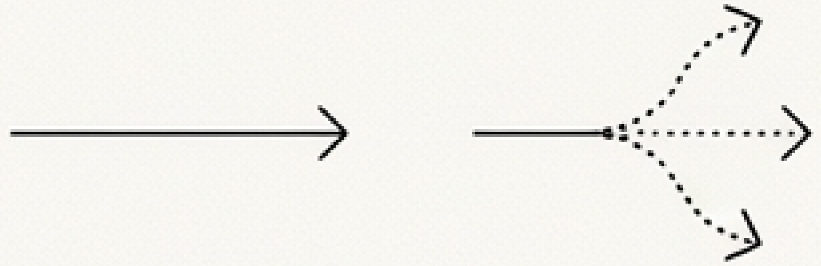
Agent	Performance	Environment	Actuators	Sensors
Vacuum Cleaner	Cleanness, efficiency	Room, obstacles, carpet	Brushes, vacuum, wheels	Camera, dirt sensor, bump sensor
Medical Diagnosis	Healthy patient, cost	Patient, hospital, staff	Screen display (treatments)	Keyboard (symptom entry)

The Nature of an Agent’s Environment

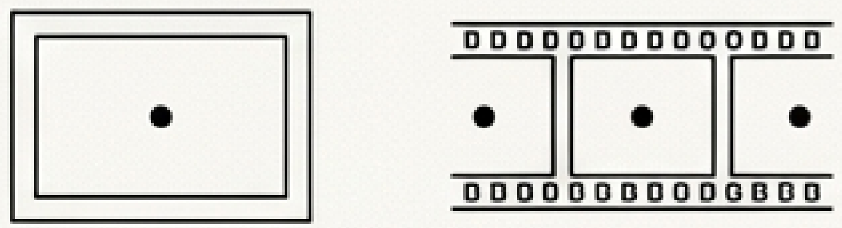
The characteristics of an environment determine the challenges an agent faces.
An environment can be classified along several key axes.



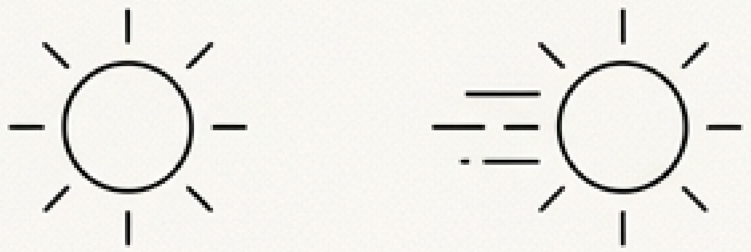
Fully Observable vs. Partially Observable
Can the agent see the complete state of the world?



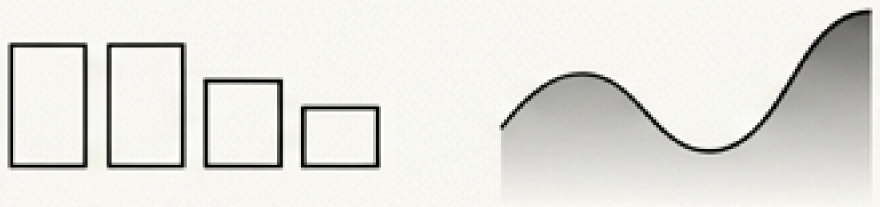
Deterministic vs. Stochastic
Is the next state completely determined by the current state and the action?



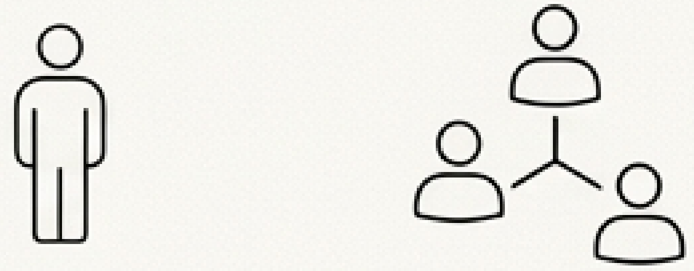
Episodic vs. Sequential
Is the agent’s experience a series of one-shot actions, or does it require memory of past actions?



Static vs. Dynamic
Can the environment change while the agent is thinking?



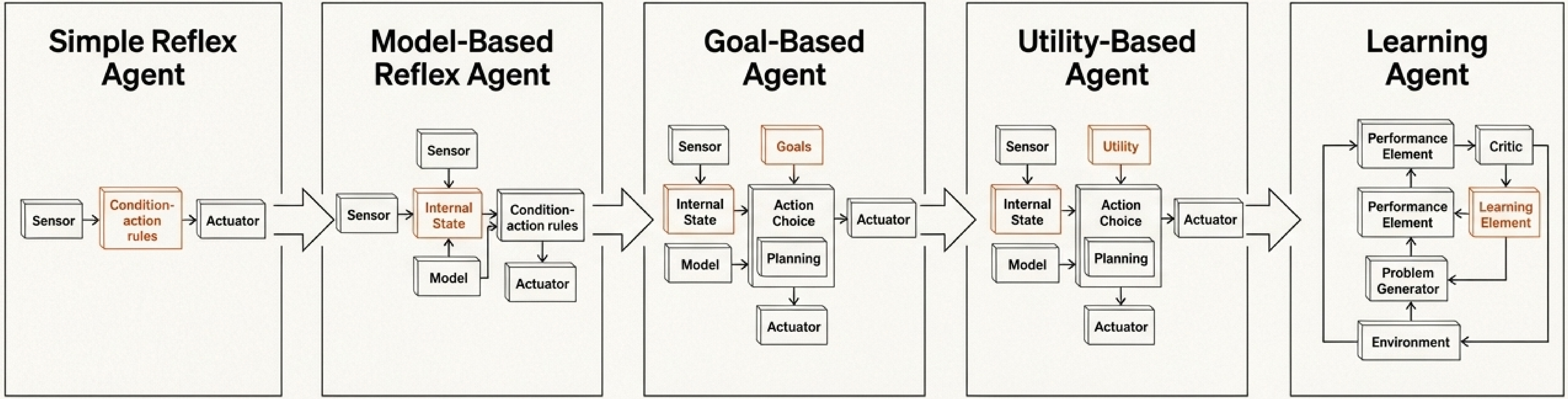
Discrete vs. Continuous
Are there a finite number of distinct states, percepts, and actions?



Single-agent vs. Multi-agent
Is the agent operating by itself or with/against others?

A self-driving car operates in a **Partially Observable**, **Stochastic**, **Sequential**, **Dynamic**, **Continuous**, and **Multi-agent** environment.

Blueprints for Intelligence: Five Five Types of Agent



Acts only on the basis of the current percept. Uses condition-action rules.

Only succeeds in fully observable environments.

Maintains an internal state (a 'model' of the world) to track the parts of the world it can't see.

Can handle partially observable environments.

Knows its goal and chooses actions to achieve it.

Considers future actions ('searching' and 'planning').

Chooses actions that maximise its 'utility' or happiness, not just achieving a goal.

Can make optimal choices when there are multiple goals or paths.

Can learn from its experiences to improve its performance over time.

Can adapt and operate in unknown environments.

The Guiding Principle: What is a Rational Agent?

A rational agent is one that acts to maximise its expected performance measure, given the evidence provided by its percept sequence and whatever built-in knowledge the agent has.

In simpler terms: A rational agent performs the 'right' action to be as successful as possible.

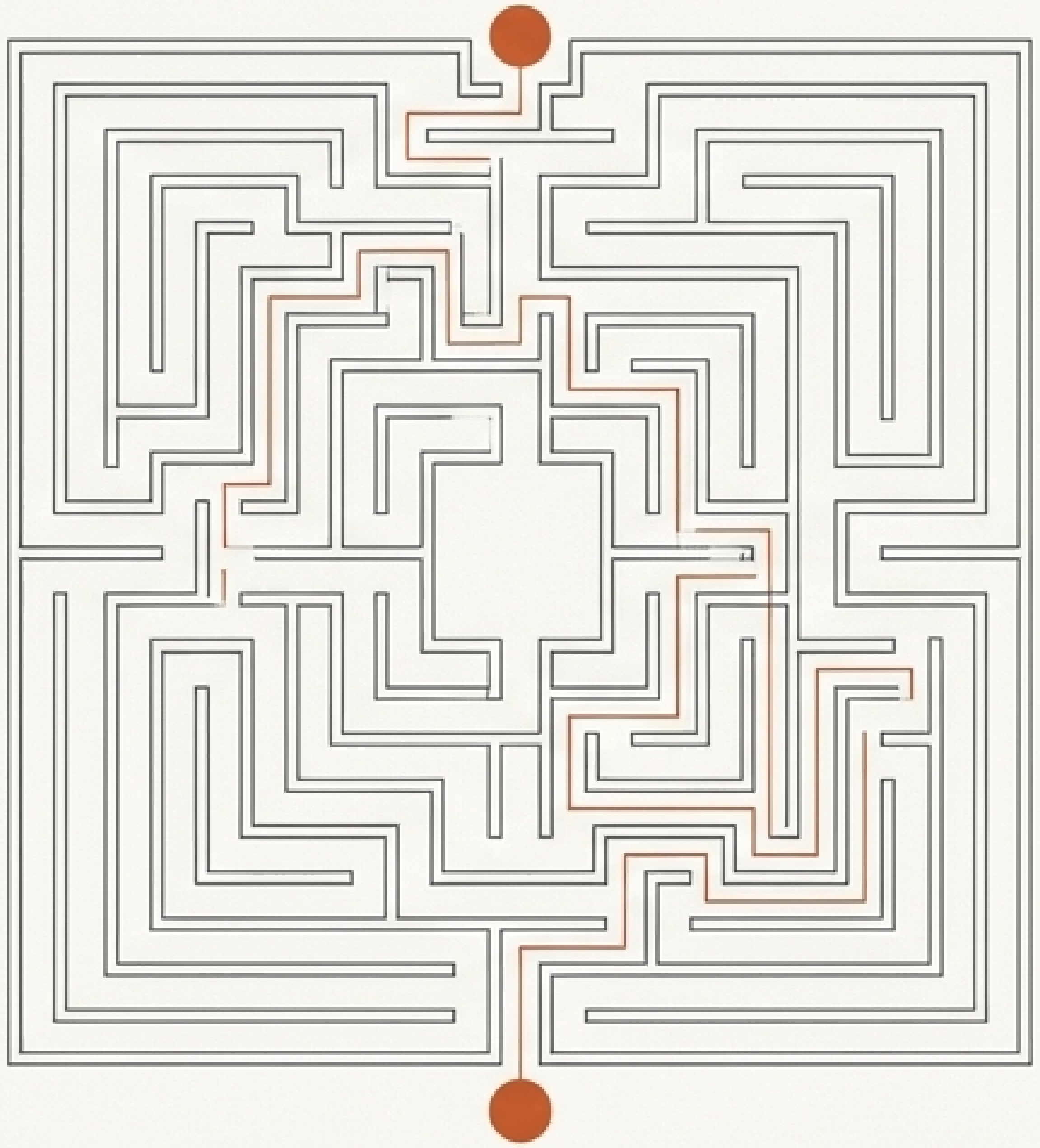
Rationality is Judged Based On Four Factors:

1. **The Performance Measure**: Defines the criterion for success.
2. **The Agent's Prior Knowledge**: What the agent knows about its environment.
3. **The Actions Available**: The range of possible actions the agent can perform.
4. **The Percept Sequence**: The agent's perceptual history up to the current time.

Rationality depends on the performance measure, what the agent knows, and what it can perceive and do. It is not about being omniscient or perfect.

The Agent's Quest: Finding Solutions Through Search

Goal-based agents solve problems by searching for a sequence of actions that leads from an initial state to a goal state.



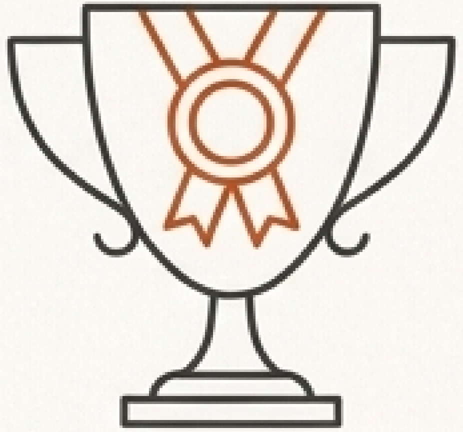
- **Search Space**: The set of all possible solutions to a problem.
- **Start State**: Where the agent begins its search.
- **Goal Test**: A function that determines if the current state is the goal state.
- **Solution**: A sequence of actions (a path) from the start state to the goal state.
- **Optimal Solution**: The solution with the lowest path cost.

Measuring an Algorithm's Performance

Not all search algorithms are created equal. We evaluate their performance against four key properties.

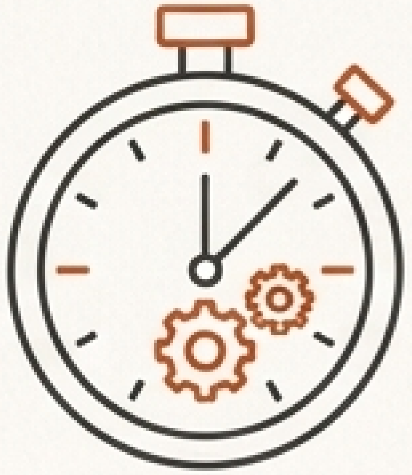
Completeness

Is the algorithm guaranteed **to find a solution if one exists?**



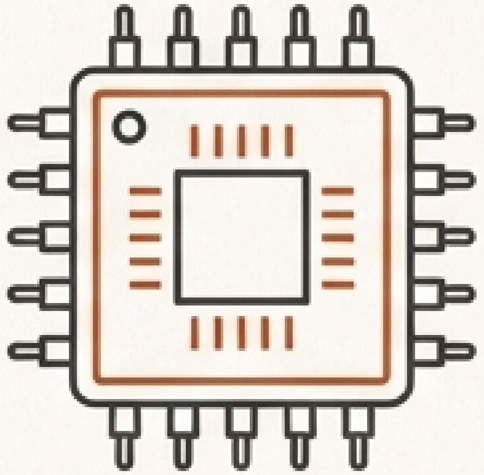
Optimality

Does the algorithm find the **best solution** (the one with **the lowest path cost**)?



Time Complexity

How long does it take to find **a solution? (Measured in terms of nodes generated).**



Space Complexity

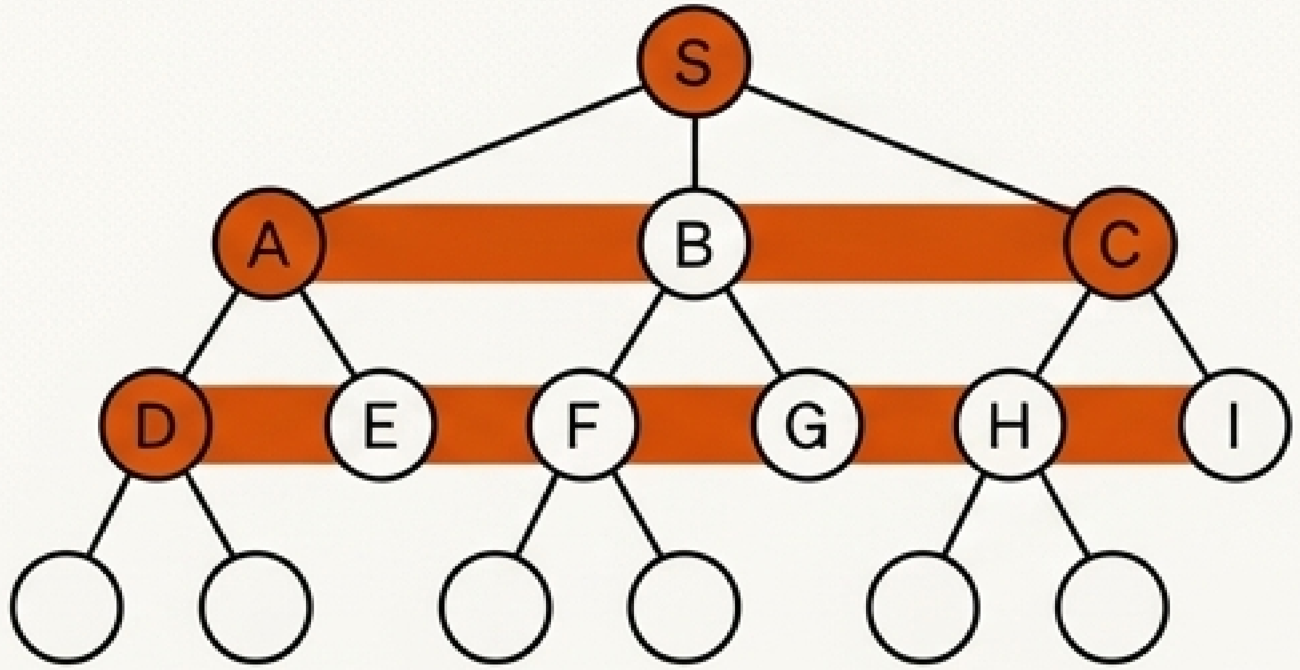
How much memory is needed **to perform the search? (Measured in terms of nodes stored in memory).**

Foundational Strategies: Uninformed (or 'Blind') Search

Uninformed search algorithms have no domain knowledge beyond the problem definition. They operate by systematically generating new states and testing them against the goal.

Breadth-First Search (BFS)

Method: Explores all neighbour nodes at the present depth prior to moving on to nodes at the next depth level. It explores level by level.

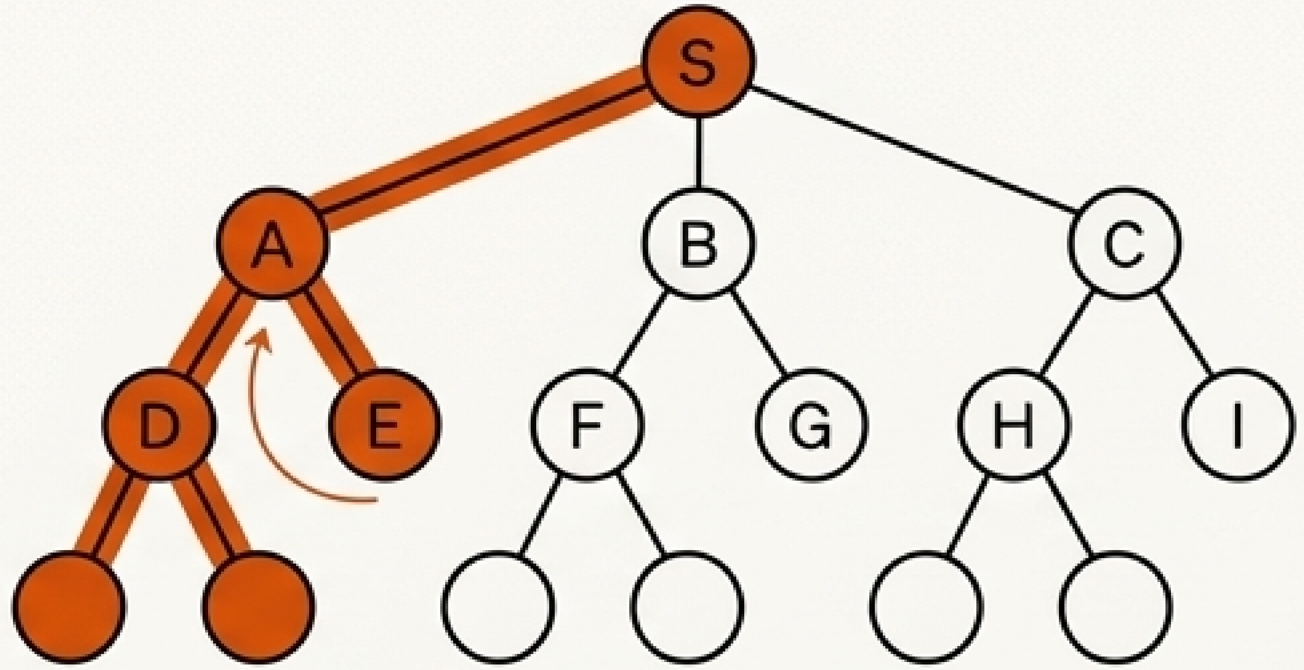


Data Structure: FIFO Queue

Completeness: Yes. | **Optimality:** Yes (if all step costs are equal). | **Complexity:** Requires a lot of memory (O(b^d)).

Depth-First Search (DFS)

Method: Explores as far as possible along each branch before backtracking. It follows one path to its end.

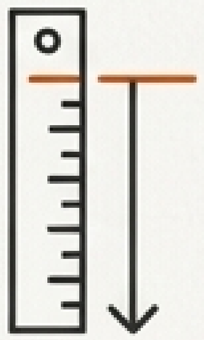


Data Structure: LIFO Stack

Completeness: No (can get stuck in infinite loops). | **Optimality:** No. | **Complexity:** Very memory efficient (O(bm)).

Refining the Search: Advanced Uninformed Strategies

Beyond BFS and DFS, other uninformed strategies offer solutions to specific challenges like path cost and search depth.

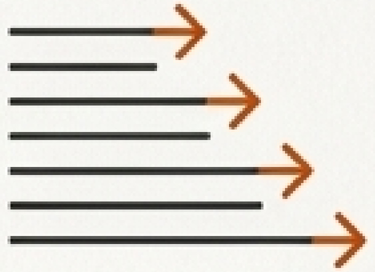


1. Depth-Limited Search (DLS)

Concept: A depth-first search with a pre-determined depth limit.
Solves: The infinite path problem of standard DFS.
Drawback: Not complete if the solution is beyond the limit.

2. Uniform-Cost Search (UCS)

Concept: Expands the node with the lowest path cost from the root, regardless of depth.
Use Case: Essential when path costs vary and the goal is to find the cheapest path, not the shortest. **Property:** Optimal.

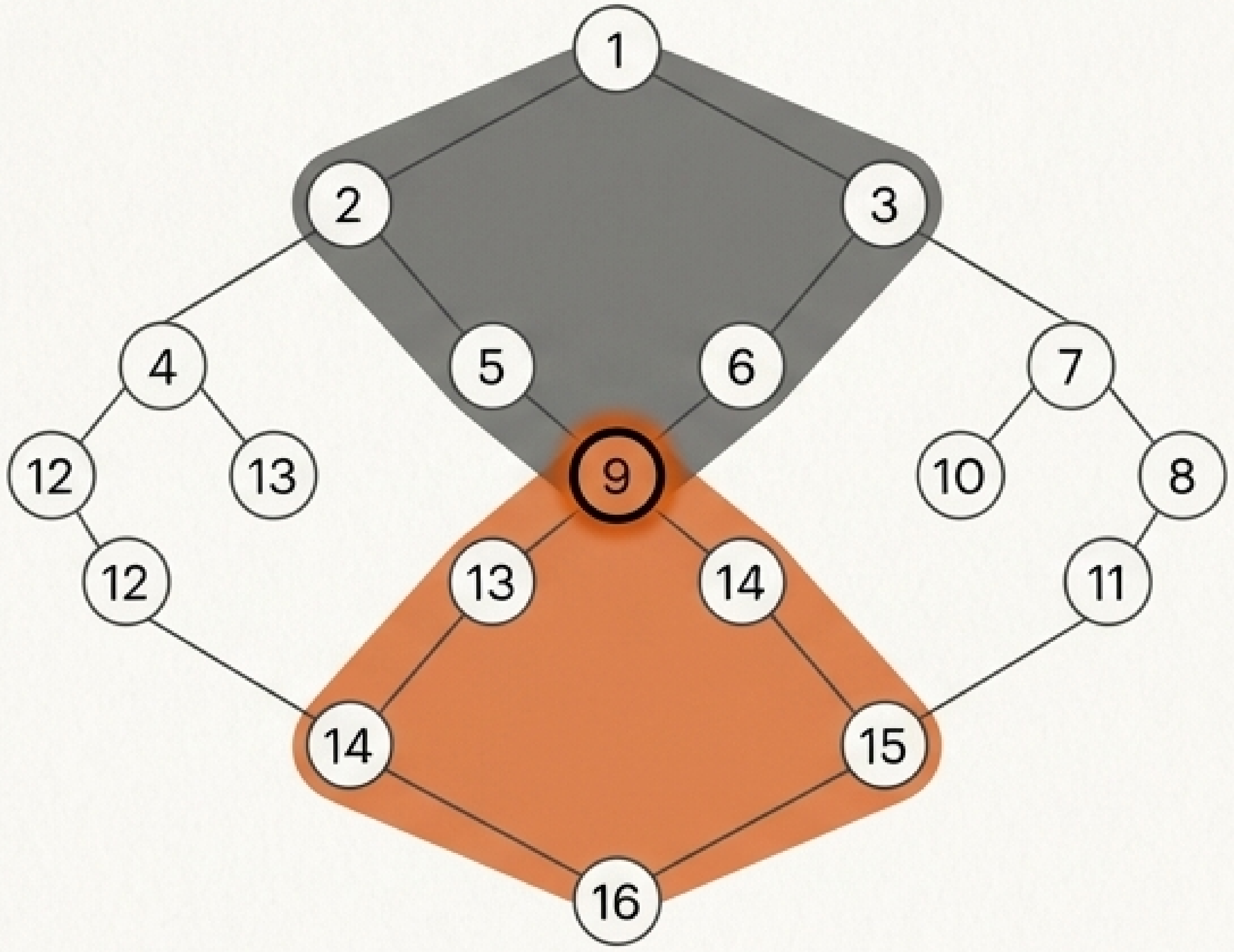


3. Iterative Deepening Depth-First Search (IDDFS)

Concept: Performs a series of depth-limited searches, gradually increasing the depth limit.
Benefit: Combines the memory efficiency of DFS with the completeness of BFS. Often the preferred uninformed search method when the search space is large.

A Smarter Path: Bidirectional Search

Bidirectional search runs two simultaneous searches—one forward from the initial state and one backward from the goal state. The search stops when the two searches meet in the middle.



Key Advantages

Speed: Drastically reduces the search space. Instead of one large search of depth d, it performs two smaller searches of depth d/2.

Memory: Requires less memory than a full BFS.

From defining intelligence to engineering rational agents and designing efficient search strategies, the field of AI provides a powerful toolkit for problem-solving. This foundational understanding is the first step.