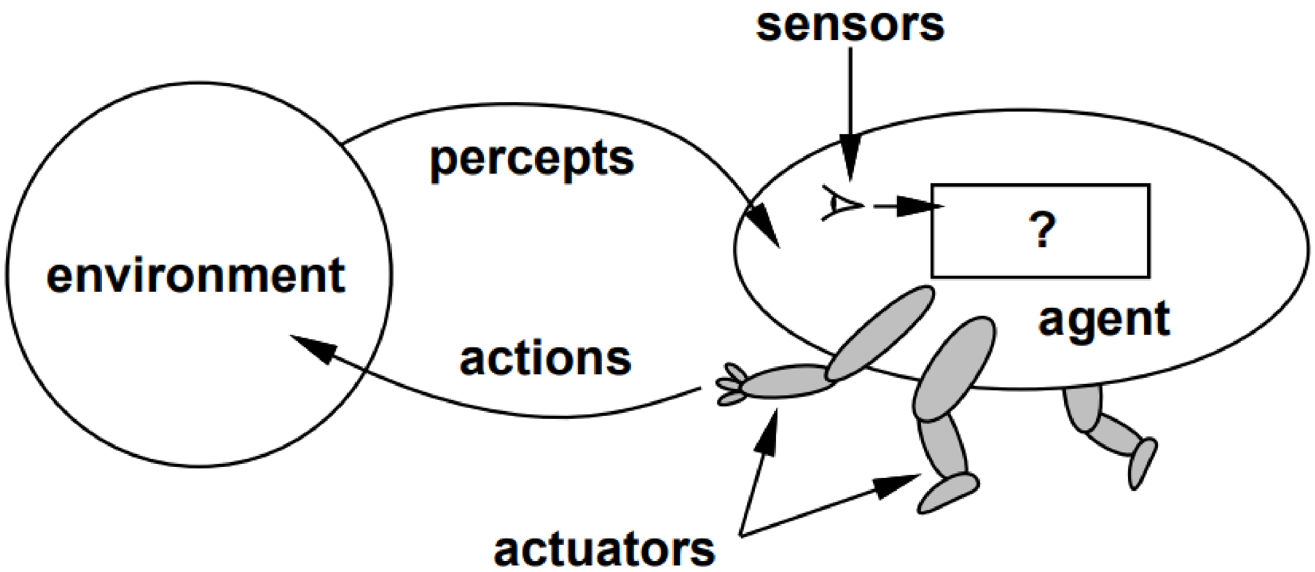
Computer Science and Artificial Intelligence (P3)

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Content

- Agent
- Perception
- Reasoning
- Additional Areas of Research



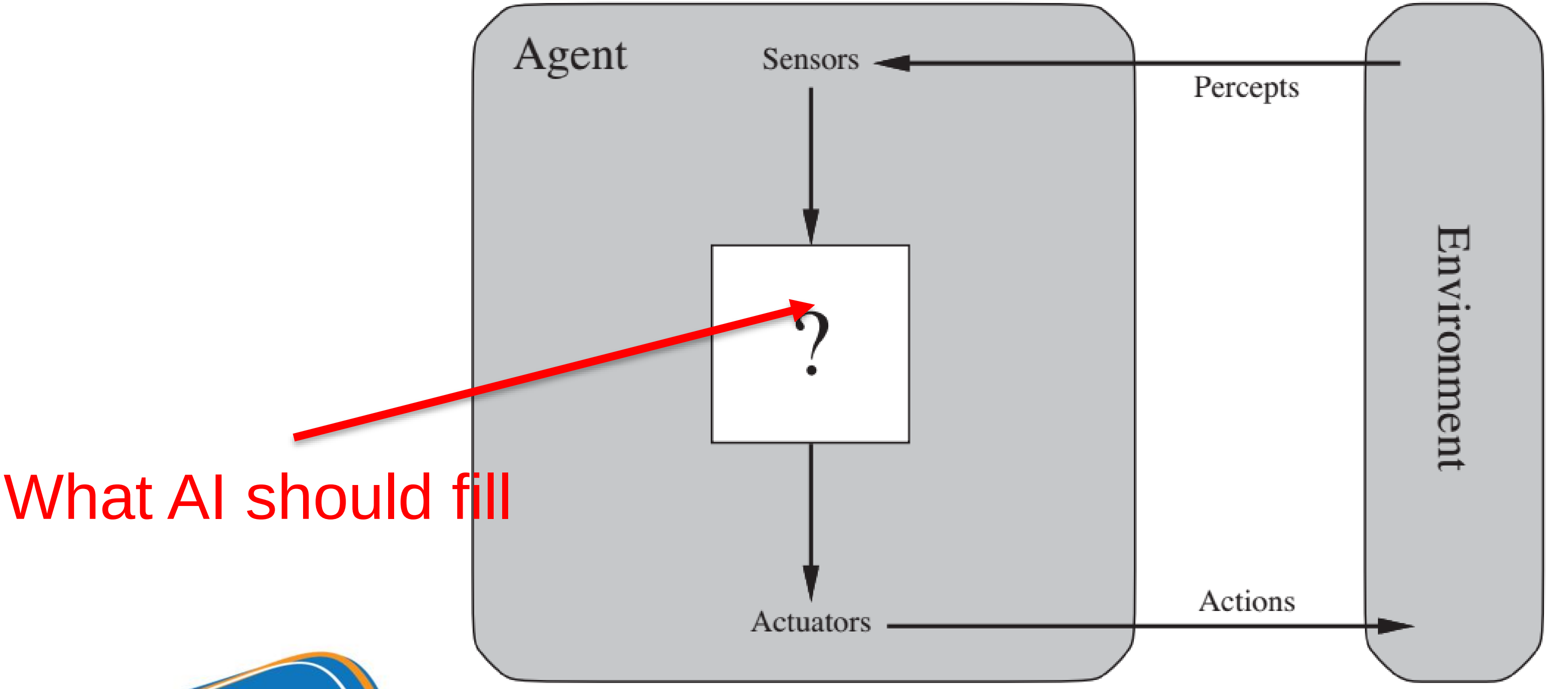
AGENTS AND ENVIRONMENTS

What is Agent?

- AI studies how to make computers do things that people are better at if they could
  - Extend what they do to huge data sets
  - Do it fast, in near real-time
  - Not make mistakes
- Such systems are called **Agents**.

What is Agent?

- An agent perceives its environment through sensors and acts upon that environment through actuators.



Agents interact with environments through sensors and actuators.

Examples of agents

Human agent

Sensors: eyes, ears, and other organs.

Actuators: hands, legs, vocal tract, etc.

Robotic agent

Sensors: cameras, infrared range finders, etc.

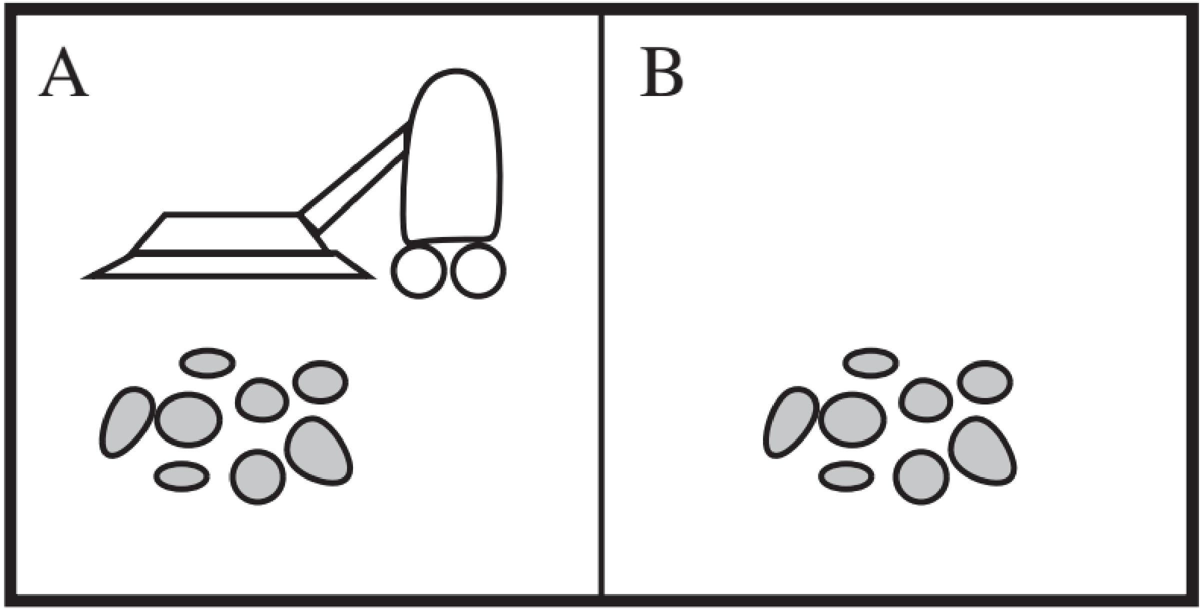
Actuators: levels, motors, etc.

Software agent

Sensors: keystrokes, file contents, network packets, etc.

Actuators: monitor, physical disk, routers, etc.

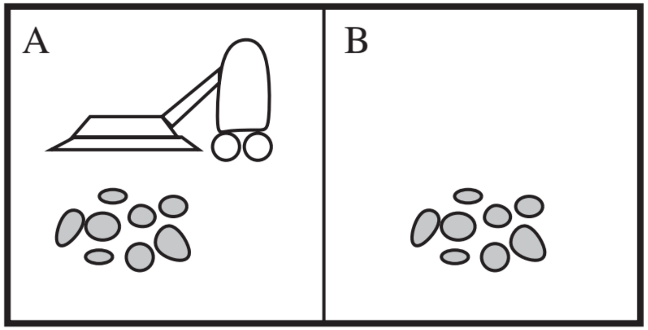
The Vacuum-cleaner world



A vacuum-cleaner world with just two locations

- Percepts: location and contents, e.g., [A,Dirty]
- Actions: Left, Right, Suck, Do Nothing

The Vacuum-cleaner world



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*

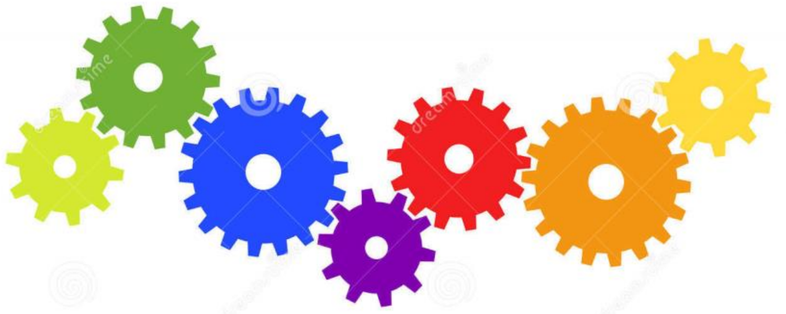
The agent program for a simple reflex agent in the two-state vacuum environment.

Why do we need agents?

- Research in artificial intelligence can be characterized in the context of building agents that behave intelligently
- The actions of the agent's actuators must be rational responses to the data received though its sensors.
- In turn, we can classify this research by considering different levels of these responses

Why do we need agents?

- A tool for analyze systems
- All areas of engineering can be seen as designing artifacts that interact with the world.
- AI designs artifacts that have significant computational resources and the task environment requires nontrivial decision making



THE CONCEPT OF RATIONALITY

Rational agents

- A rational agent is one that does the right thing.
- What is "right" thing?
  - The actions that cause the agent to be most successful
- We need ways to measure success.

Performance measure

- An agent, based on its percepts → generates actions sequence → environment goes to sequence of states
  - If this sequence of states is desirable, then the agent performed well.
- Performance measure evaluates any given sequence of environment states (remember, not agent states!!!).
  - An objective function that decides how the agent does successfully. E.g., 90%? 30%?

Design performance measures

- General rule: Design performance measures according to
  - What one **actually wants** in the environment
  - Not how one **thinks** the agent should behave
- For example, in vacuum-cleaner world
  - The amount of dirt cleaned up in a single eight-hour shift, or
  - The floor clean, no matter how the agent behaves
  - *Which one is better?*

Rationality

- What is rational at any given time depends on

Performance measure Define the criterion of success	**Prior knowledge** What the agent knows about the environment
Percept sequence The agent's percept to date	**Actions** What the agent can perform

Definition of a rational agent

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- For example, in an exam,
  - Maximize marks based on the questions on the paper and your knowledge

The Vacuum-cleaner agent

- Performance measure
  - Award one point for each clean square at each time step, over 10000 timesteps
- Prior knowledge about the environment
  - The geography of the environment (2 squares)
  - The effect of the actions
- Actions that can perform
  - Left, Right, Suck and Do Nothing
- Percept sequences
  - Where is the agent?
  - Whether the location contains dirt?
- Under this circumstance, the agent is **rational**.

Types of agent programs

- Simple reflex agent
- Model-based reflex agent
- Goal-based agents
- Utility-based agents

Simple reflex agents

- The simplest kind of agent, limited intelligence
- Select actions based on the current percept, ignoring the rest of the percept history
- The connection from percept to action is represented by condition-action rules.

IF *current percept* **THEN** *action*

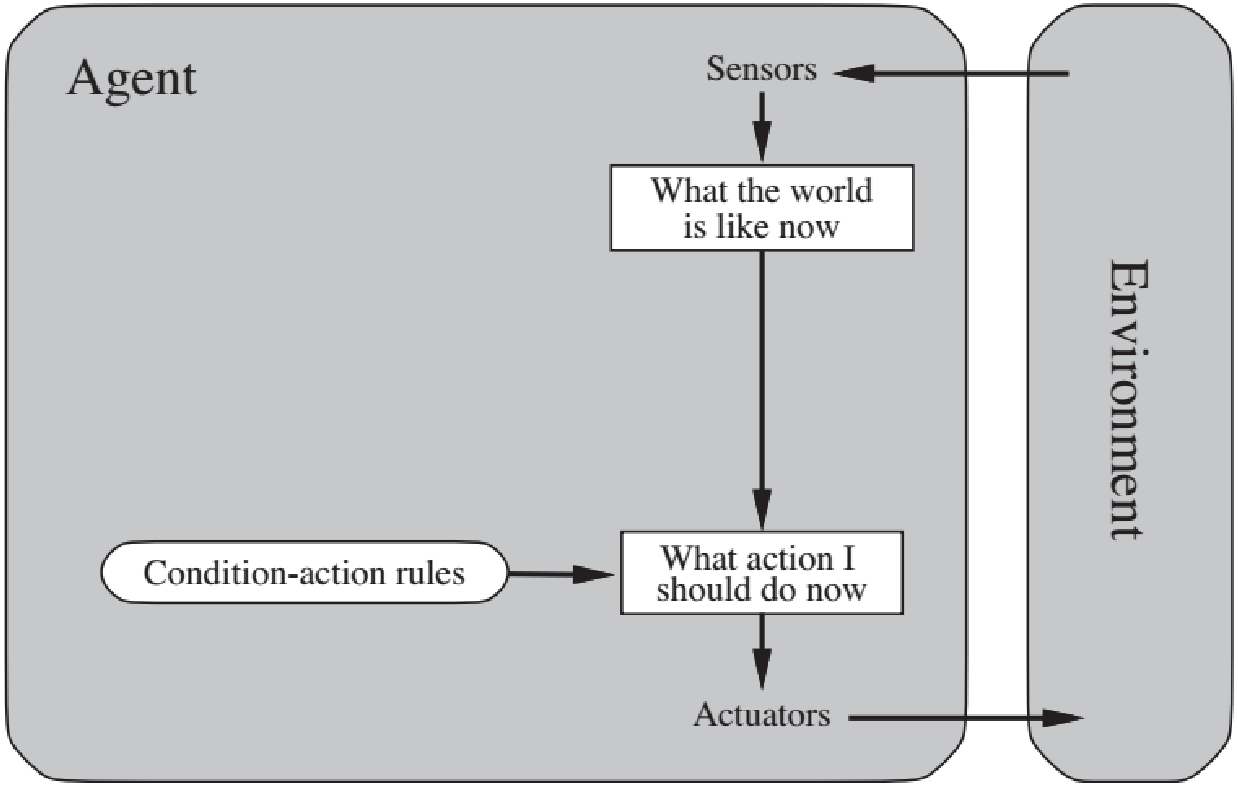
  - E.g., IF *car-in-front-is-braking* THEN *initiate-braking*.
- Limitations
  - Knowledge sometimes cannot be stated explicitly → low applicability
  - Work only if the environment is fully observable

Simple reflex agents

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

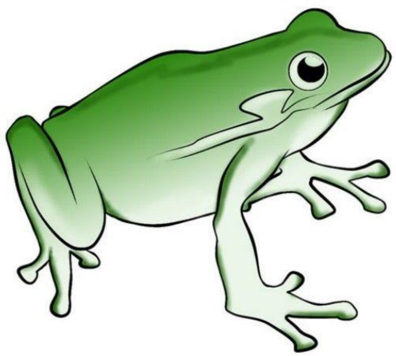
if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*

- The vacuum agent program is very small compared to the corresponding table
- The most obvious reduction: ignoring the percept history, which cuts down the number of possibilities from 4^T to just 4
- A further, small reduction comes from the fact that when the current square is dirty, the action does not depend on the location.


Agent
Sensors
What the world
is like now
Condition-action rules
What action I
should do now
Actuators
Environment

A Simple reflex agent in nature

percepts
(size, motion)



Action: SNAP or AVOID or NOOP

RULES:

(1) If small moving object, then activate SNAP

(2) If large moving object, then activate AVOID and inhibit SNAP

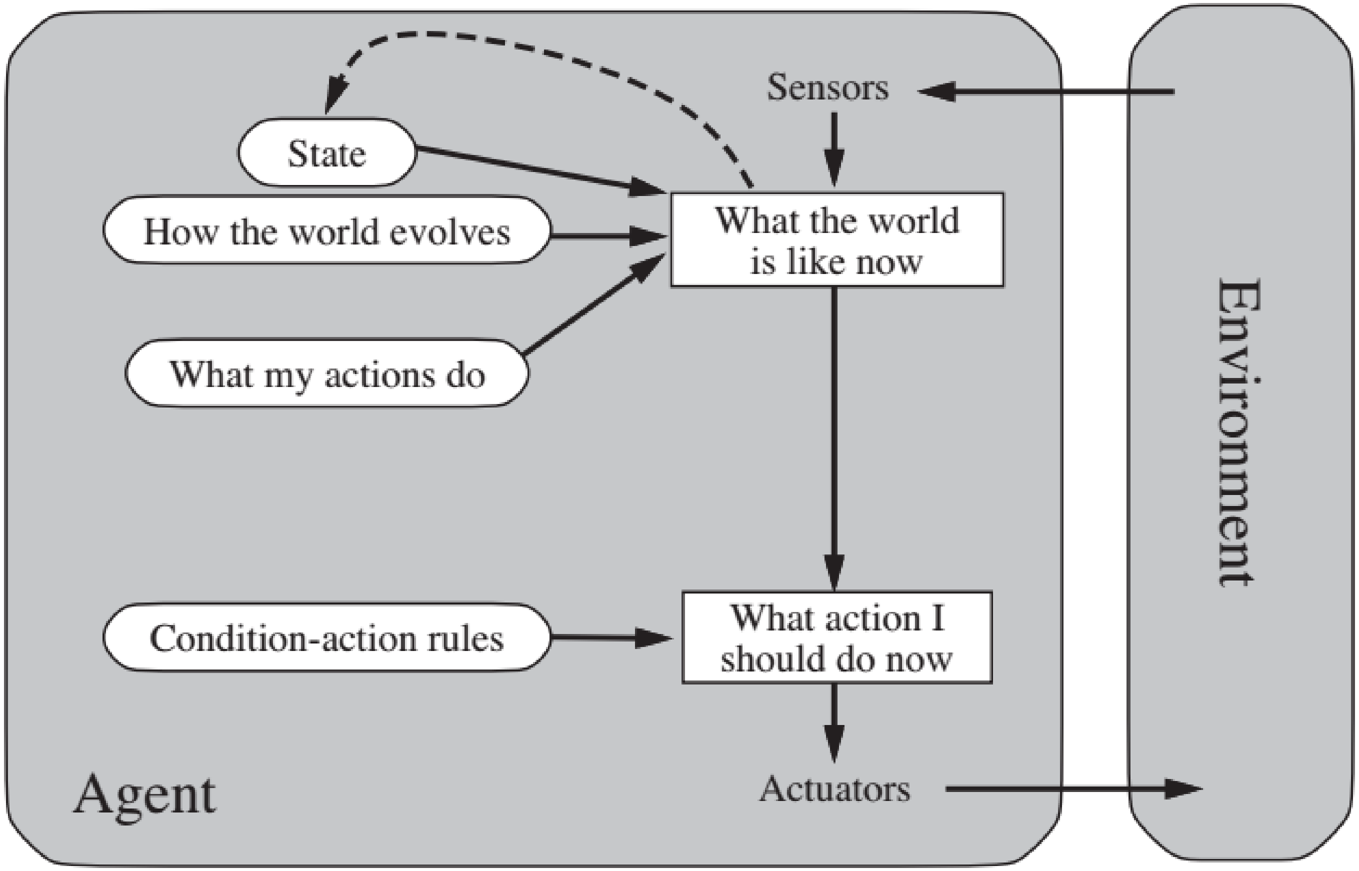
ELSE (not moving) then NOOP

Needed for completeness

Model-based reflex agents

- The agent must keep track of an internal state in partially observable environments.
  - It depends on the percept history and reflects some of the unobserved aspects, e.g., driving a car and changing lane.
- The agent program updates the internal state information as time goes by by encoding two kinds of knowledge
  - How the world evolves independently of the agent
  - How the agent's actions affect the world

model of the world

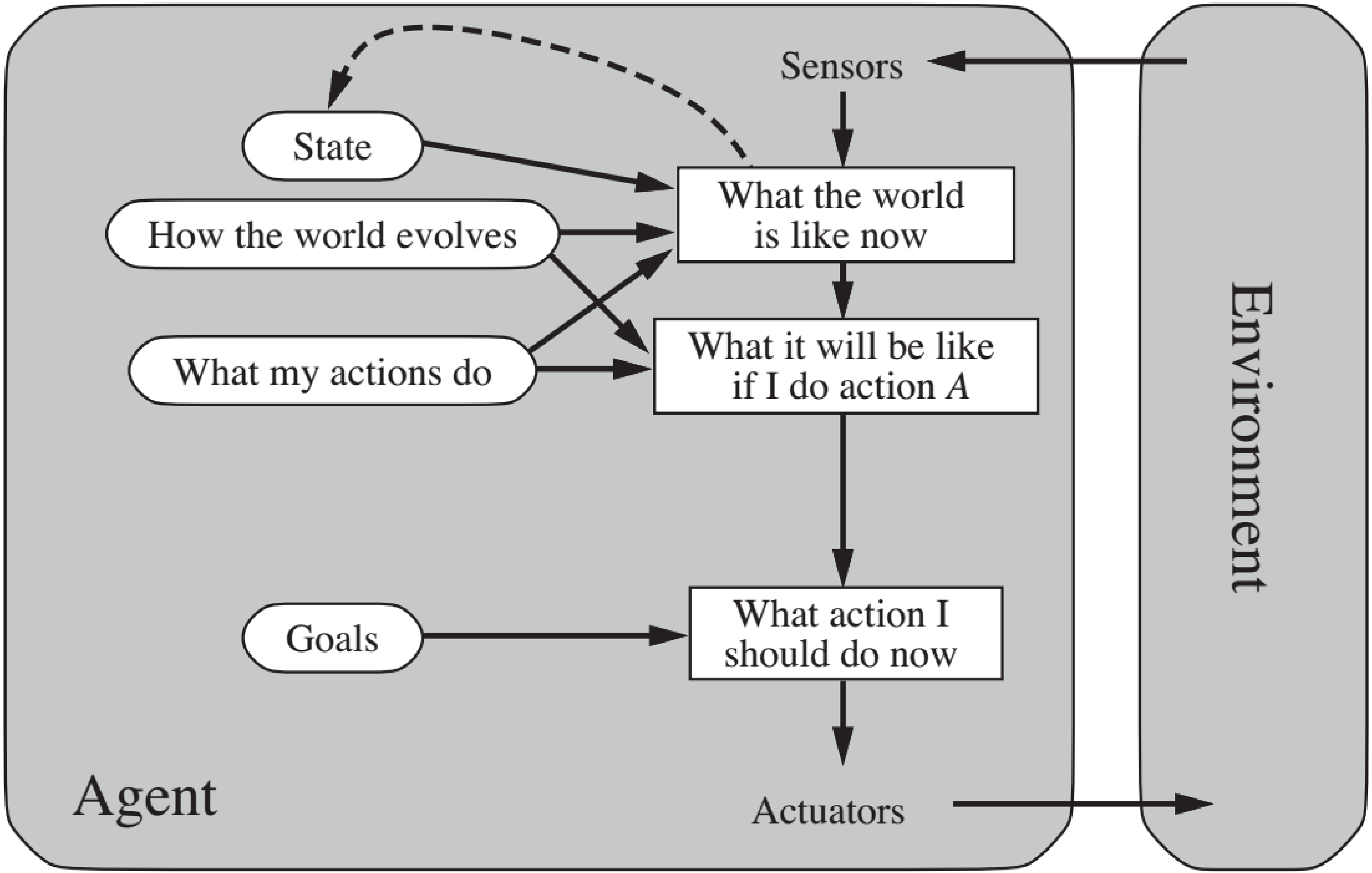


Example table agent with internal state

Goal-based agents

- Current state of the environment is always not enough
- The agent further needs some sort of goal information that describes desired situations.
  - E.g., at a road junction, the taxi can turn left, turn right, or go straight on, depending on where the taxi is trying to get to.
- Less efficient but more flexible
  - Knowledge supporting the decisions is represented explicitly and can be modified.

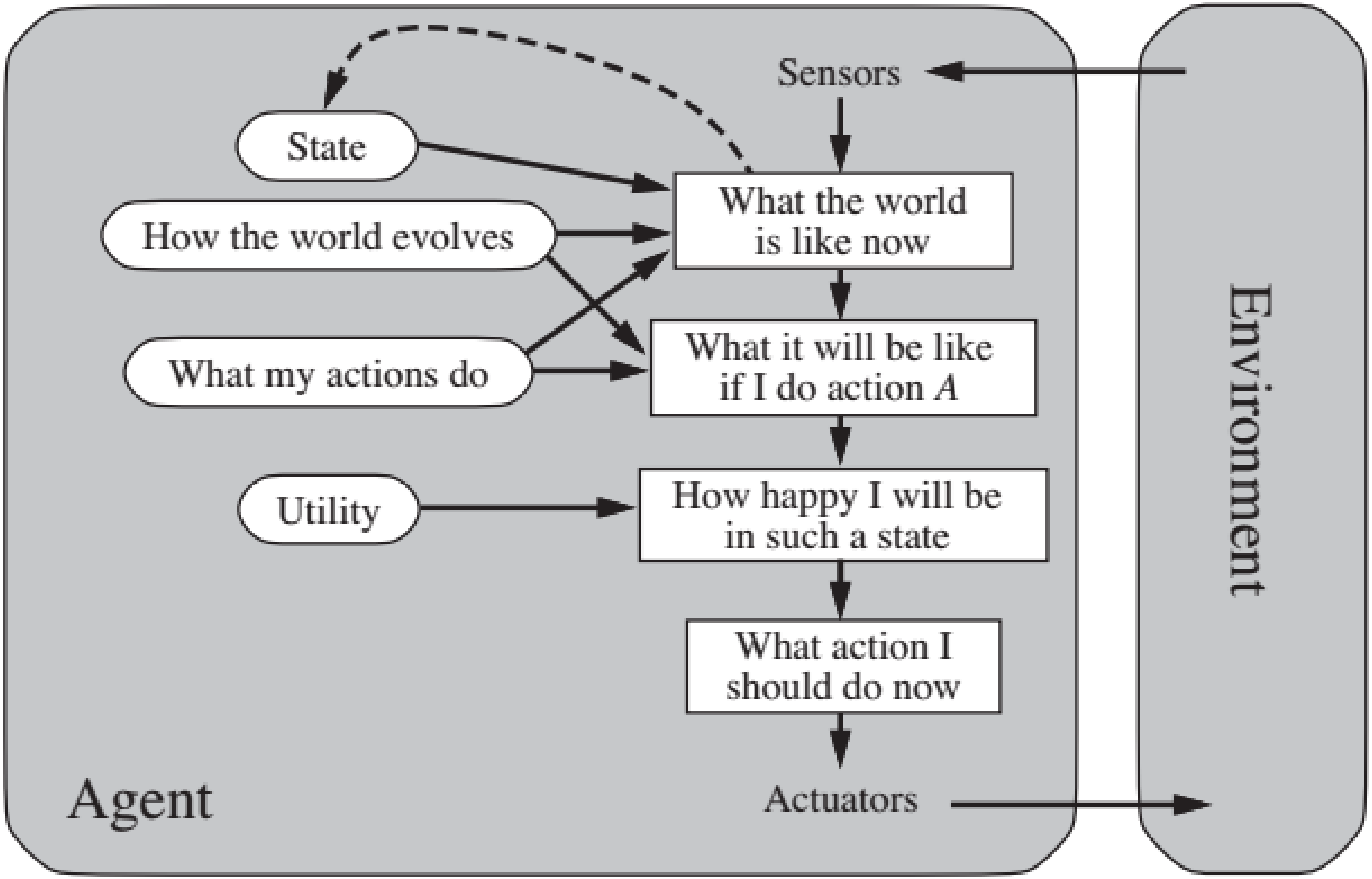
Goal-based agents



Utility-based agent

- Goals are inadequate to generate high-quality behavior in most environments.
  - Many action sequences can get the goals, some are better, and some are worse, e.g., go home by taxi or Grab car?
- An agent's utility function is essentially an internalization of the performance measure.
  - Goal $\rightarrow$ success, utility $\rightarrow$ degree of success (how successful it is)
  - If state A is more preferred than others, then A has higher utility.

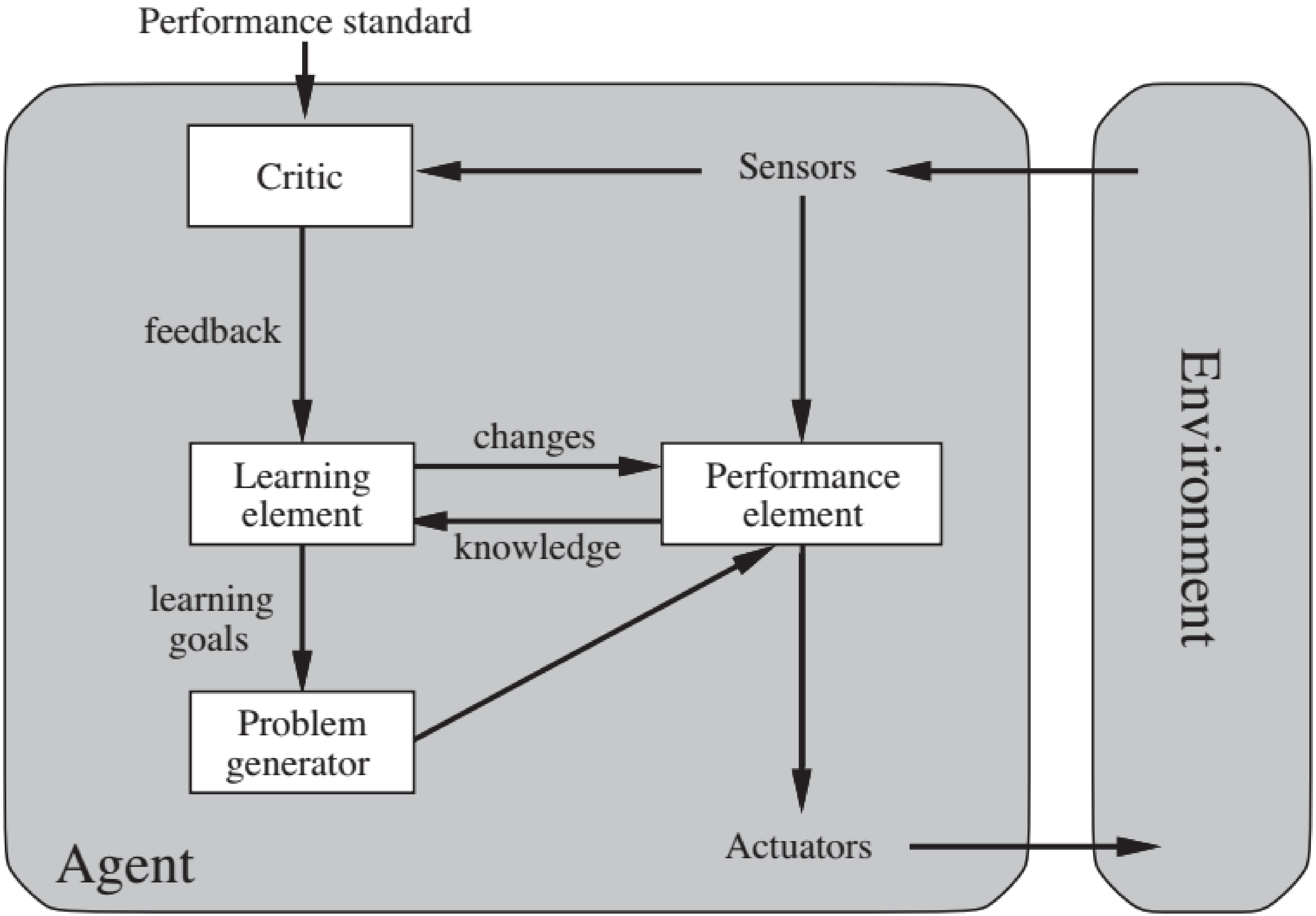
Utility-based agent



Learning agents

- After an agent is programmed, can it work immediately?
  - No, it still need teaching
- Once an agent is done, what can we do next?
  - Teach it by giving it a set of examples
  - Test it by using another set of examples
- We then say the agent **learns** → learning agents

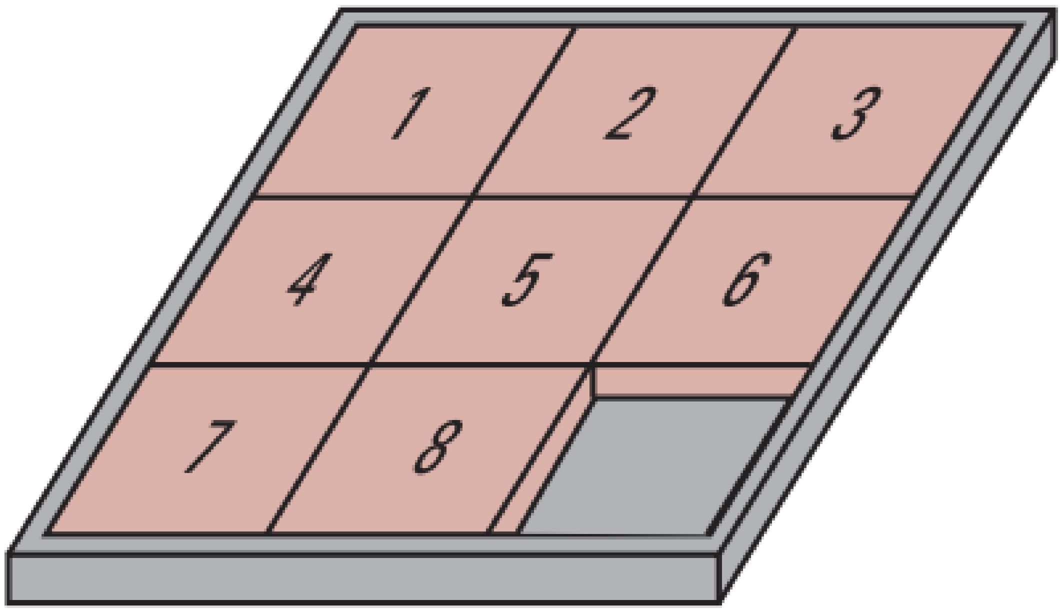
Learning agents



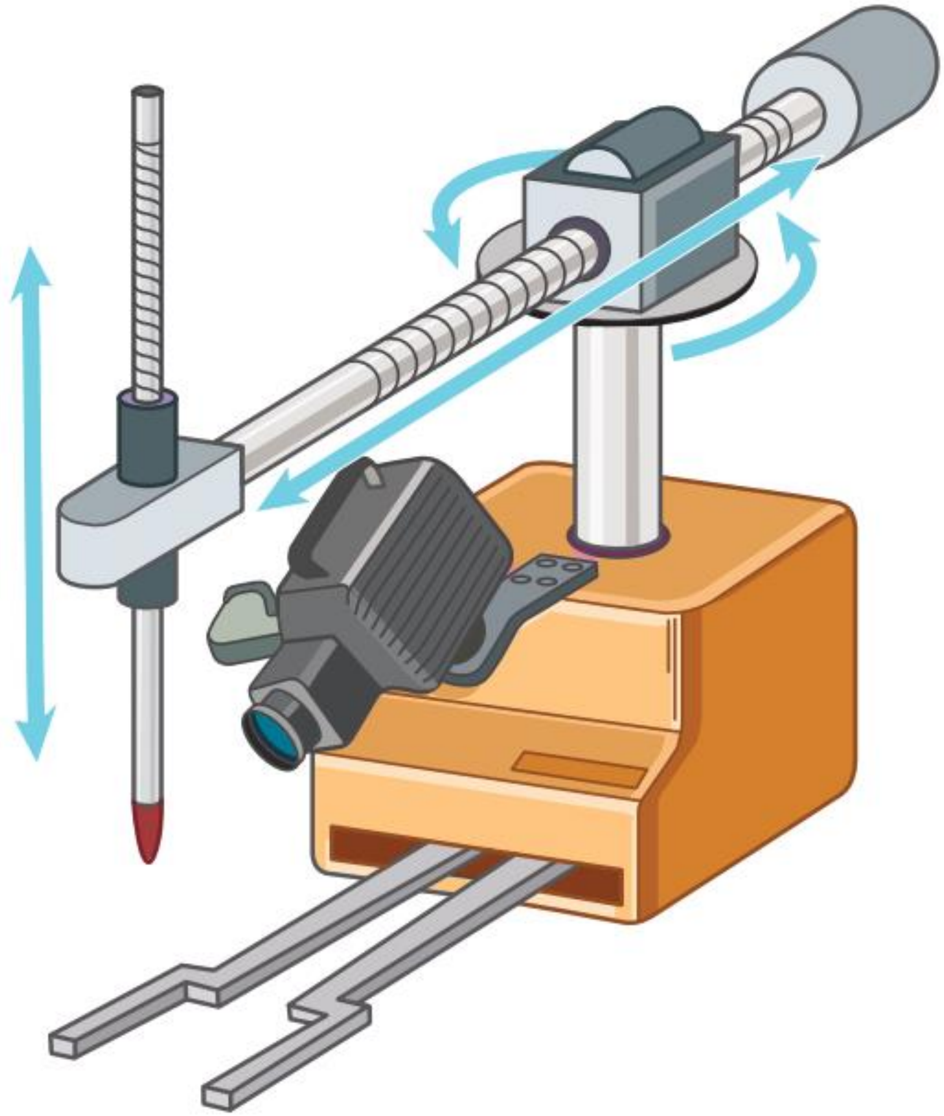
Learning agents

- A learning agent is divided into four conceptual components
  1. Learning element → Make improvement
  2. Performance element → Select external actions
  3. Critic → Tell the Learning element how well the agent is doing with respect to fixed performance standard. (Feedback from user or examples, good or not?)
  4. Problem generator → Suggest actions leading to new and informative experiences

The eight-puzzle in its solved configuration



Our puzzle-solving machine

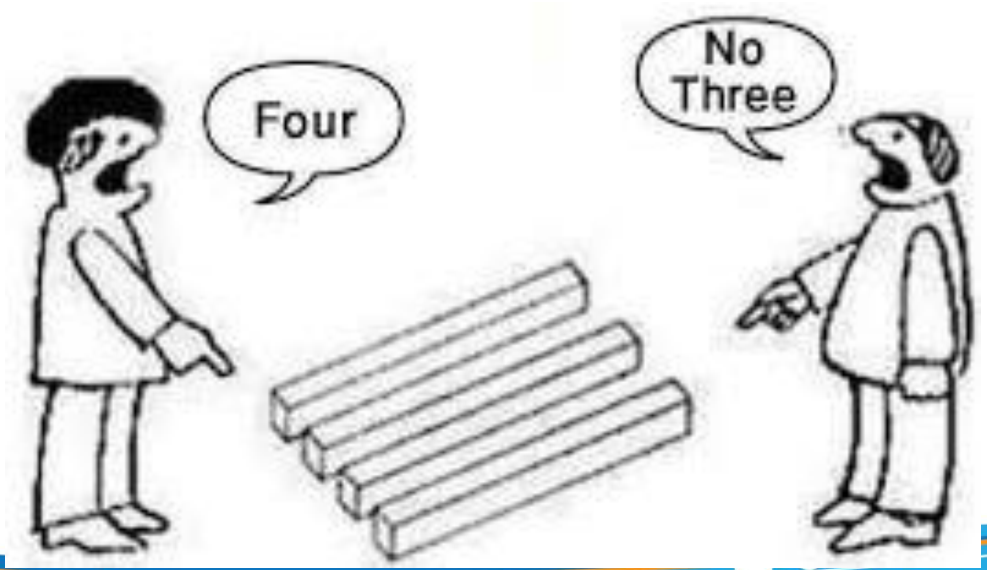


Content

- Agent
- **Perception**
- Reasoning
- Additional Areas of Research

Perception

- To respond intelligently to the input from its sensors, an agent must be able to understand that input (perceive).
- Two areas of research in perception that have proven to be especially challenging:
  - Understanding images (Computer Vision) and language (NLP).

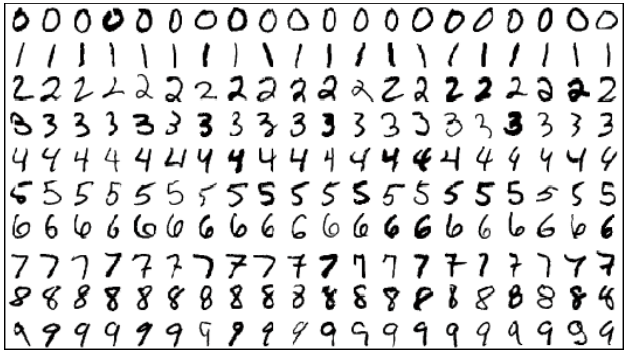


Computer vision

- **Computer vision** is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information.
- If AI enables computers to think, computer vision enables them to see, observe and understand.

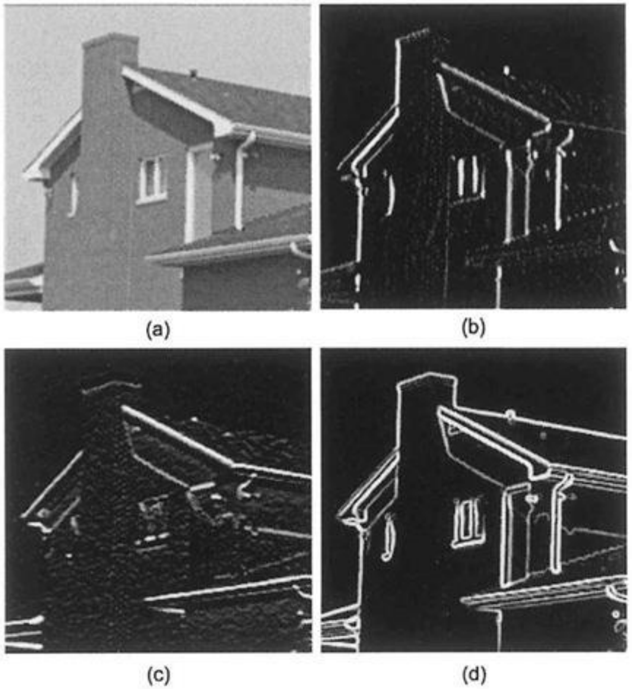
Techniques for Understanding Images

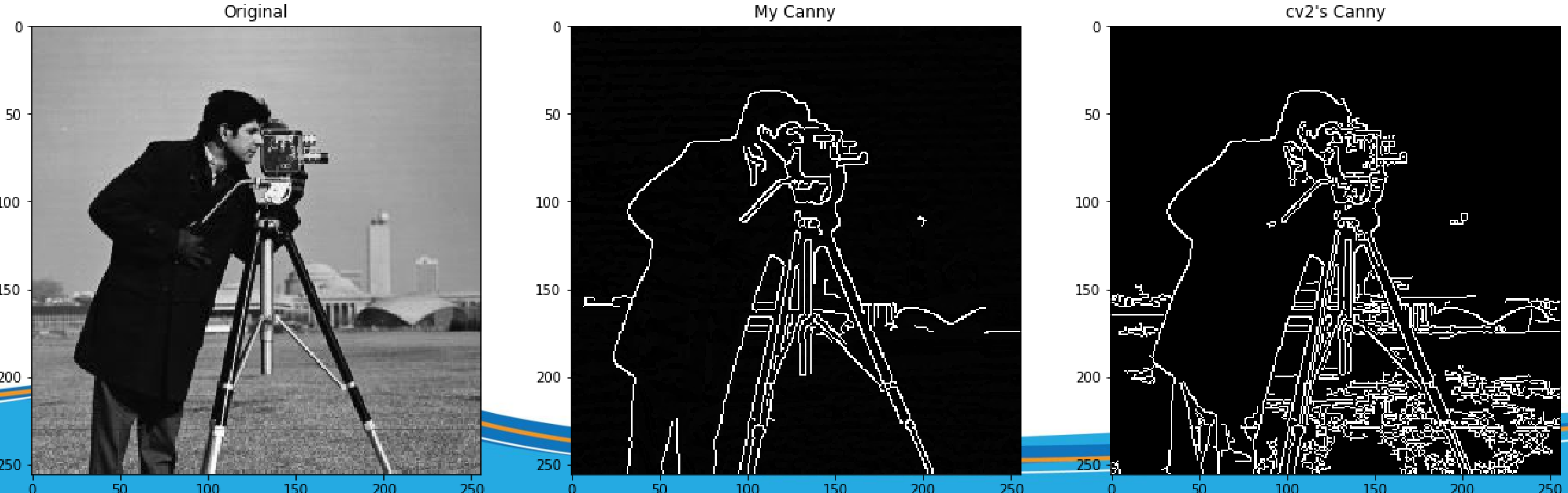
- Template matching
- Image processing (identifying characteristics of the image)
  - edge enhancement
  - region finding
  - smoothing



- Image analysis (the process of understanding what these characteristics mean)
- Some demo: https://setosa.io/ev/image-kernels/

Edge enhancement





Region finding

Smoothing

Image analysis

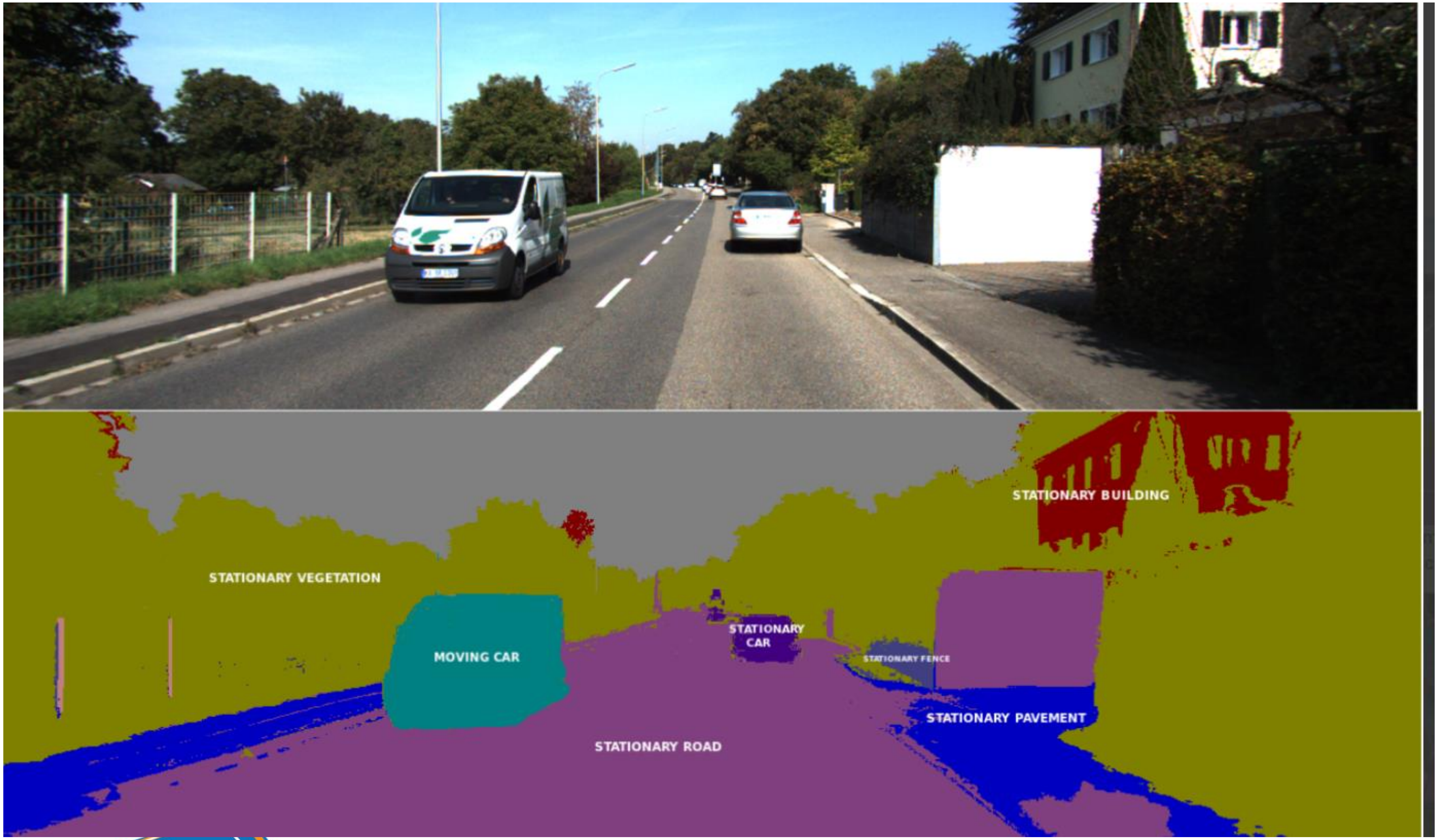
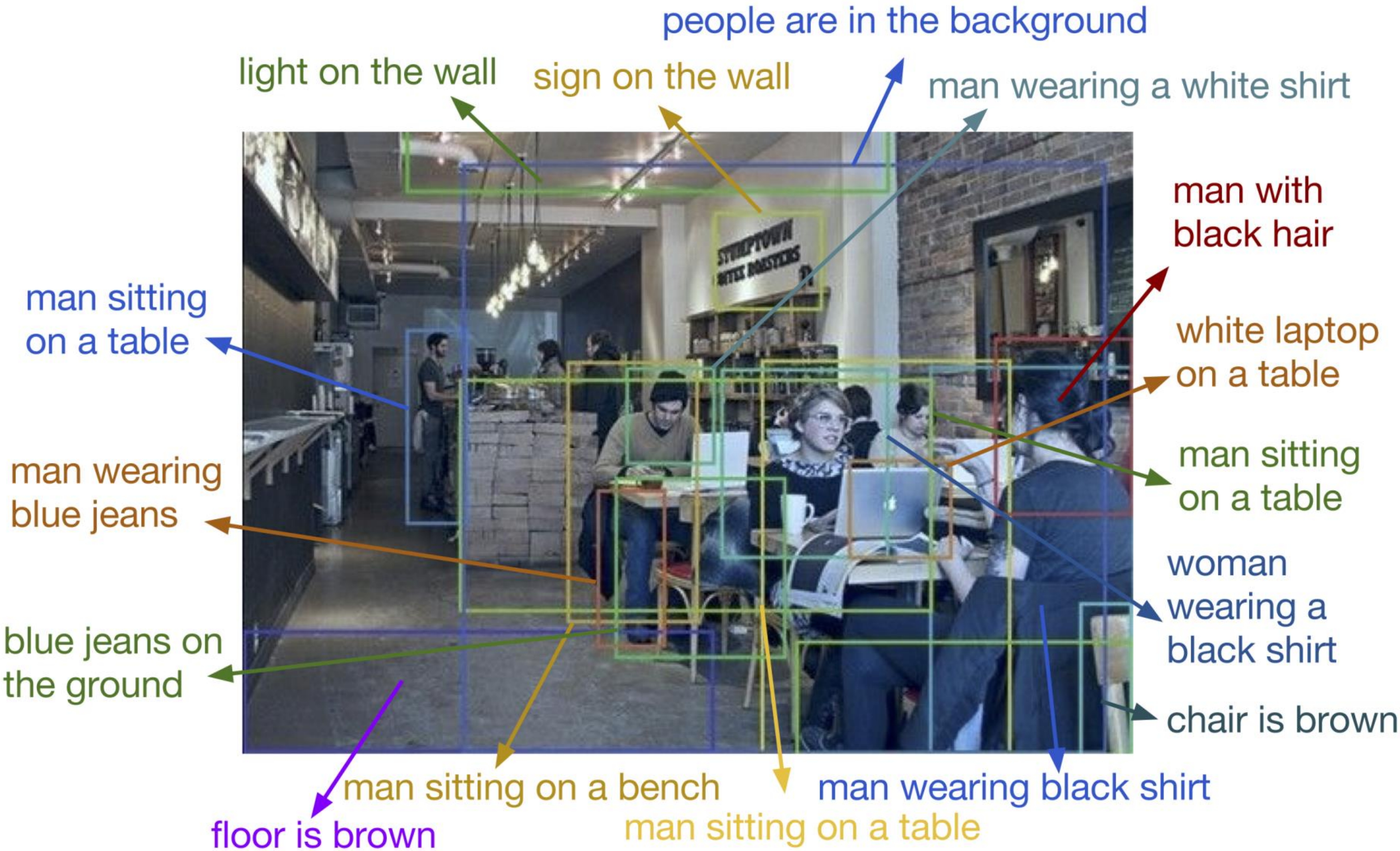
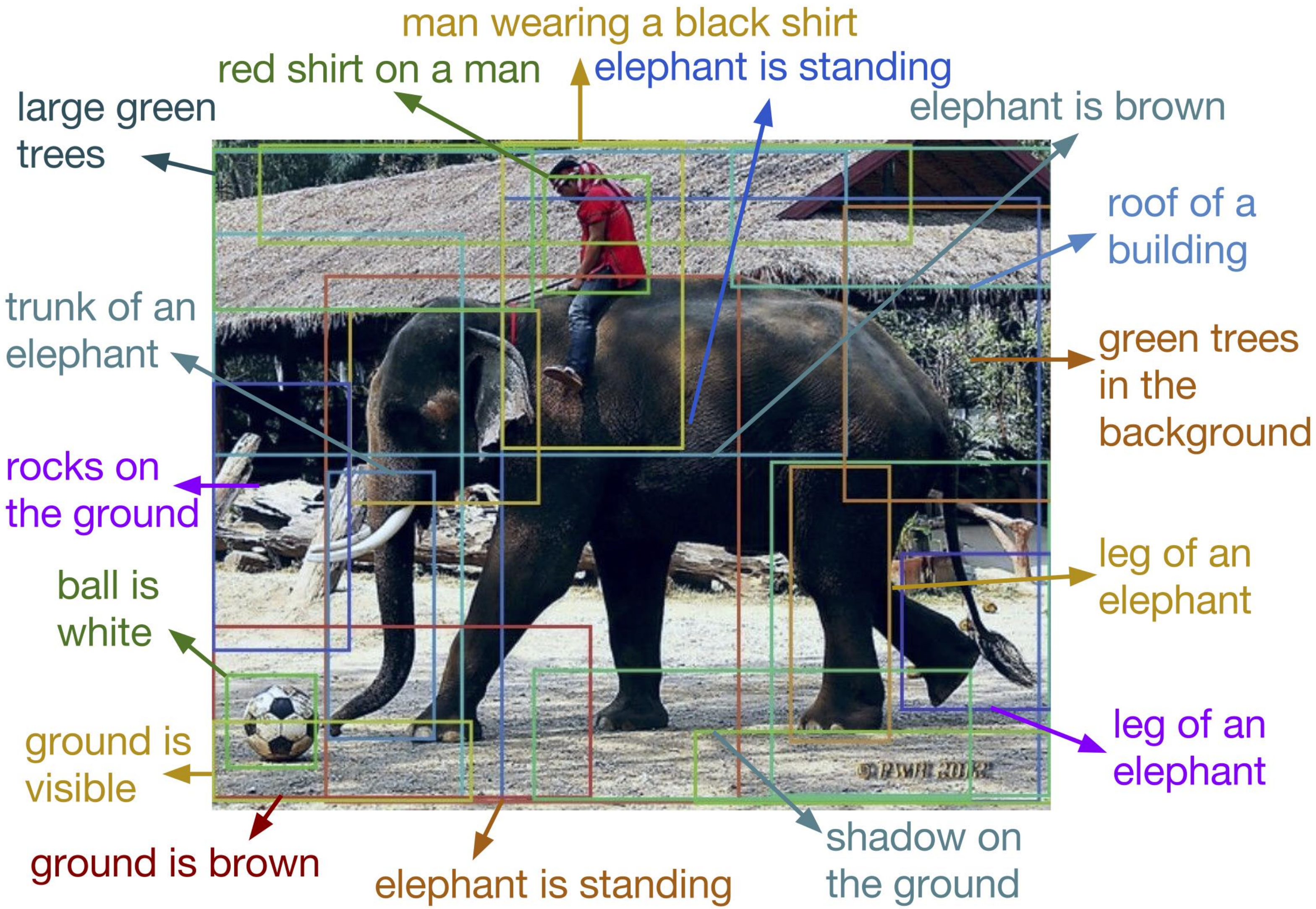


Image analysis



https://github.com/jcjohnson/densecap/blob/master/imgs/resultsfig.png

Image analysis



https://github.com/jcjohnson/densecap/blob/master/imgs/resultsfig.png

Natural language processing

- Natural language processing (NLP) is the intersection of computer science, linguistics and machine learning.
- NLP is all about making computers understand and generate human language.

Natural language processing

- Speech recognition
  - The translation of spoken language into text.
- Natural language understanding
  - a computer’s ability to understand language.
- Natural language generation
  - the generation of natural language by a computer.

Language Processing

- Programming languages are constructed from well-designed primitives so that each statement has only one grammatical structure and only one meaning.
- A statement in a natural language can have multiple meanings depending on its context or even the manner in which it is communicated.
- To understand natural language, humans rely heavily on additional knowledge.

Language Processing

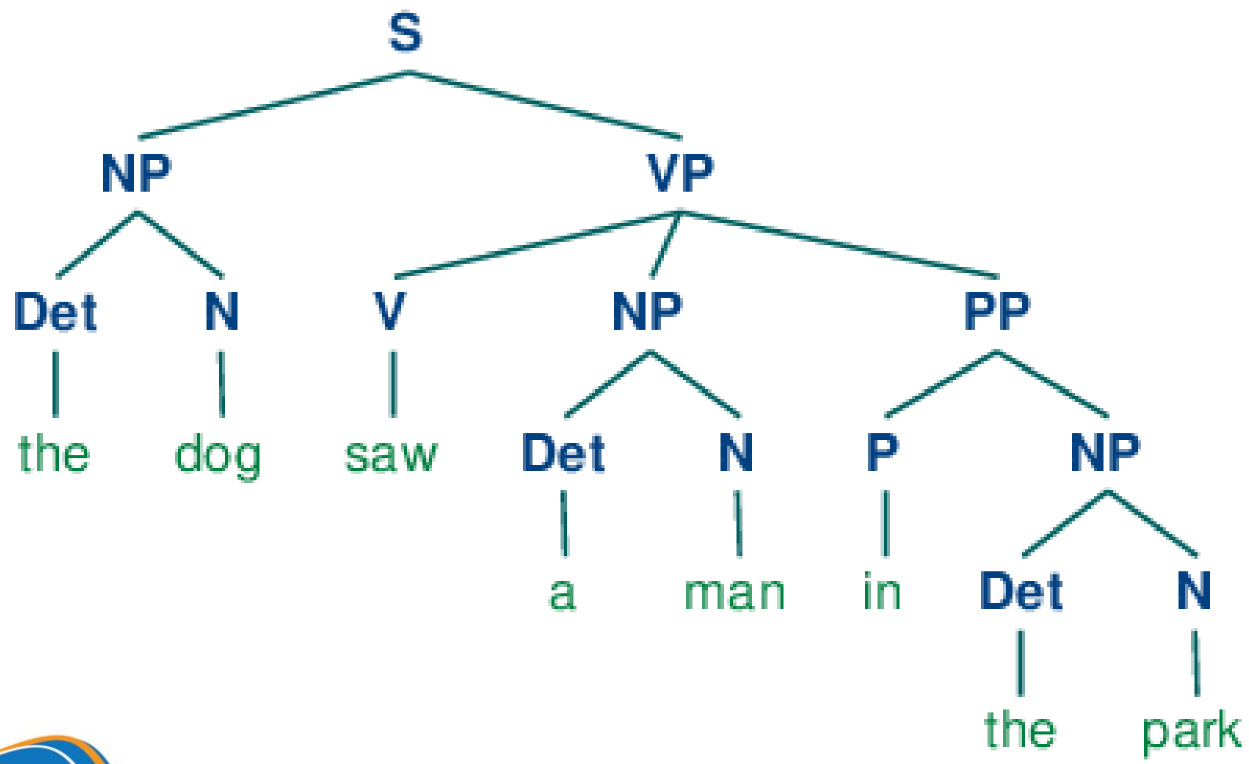
- Consider these sentences:
  - Norman Rockwell painted people.
  - Cinderella had a ball.
  - Do you know what time it is?

Language Processing

- To understand the meaning of a statement in a natural language requires several levels of analysis
  - Syntactic Analysis
  - Semantic Analysis
  - Contextual Analysis

Syntactic analysis

- Identifies the grammatical role of each word



Syntactic analysis

- Identifying the semantic role of each word.
- A technique that can analyze the meaning of a text
- Syntactic analysis focuses on “form” and syntax, meaning the relationships between words in a sentence.
- Semantic analysis focuses on “meaning,” or the meaning of words together and not just a single word.

Syntactic analysis

- Example:
  - The customer, says Mister Johnson, is satisfied.
  - The customer says: Mister Johnson is satisfied.
- Understanding the logical meaning that connects the parts of a sentence

Contextual analysis

- The context of the sentence is brought into the understanding process
- Example:
  - The bat fell to the ground.
  - Do you know what time it is?

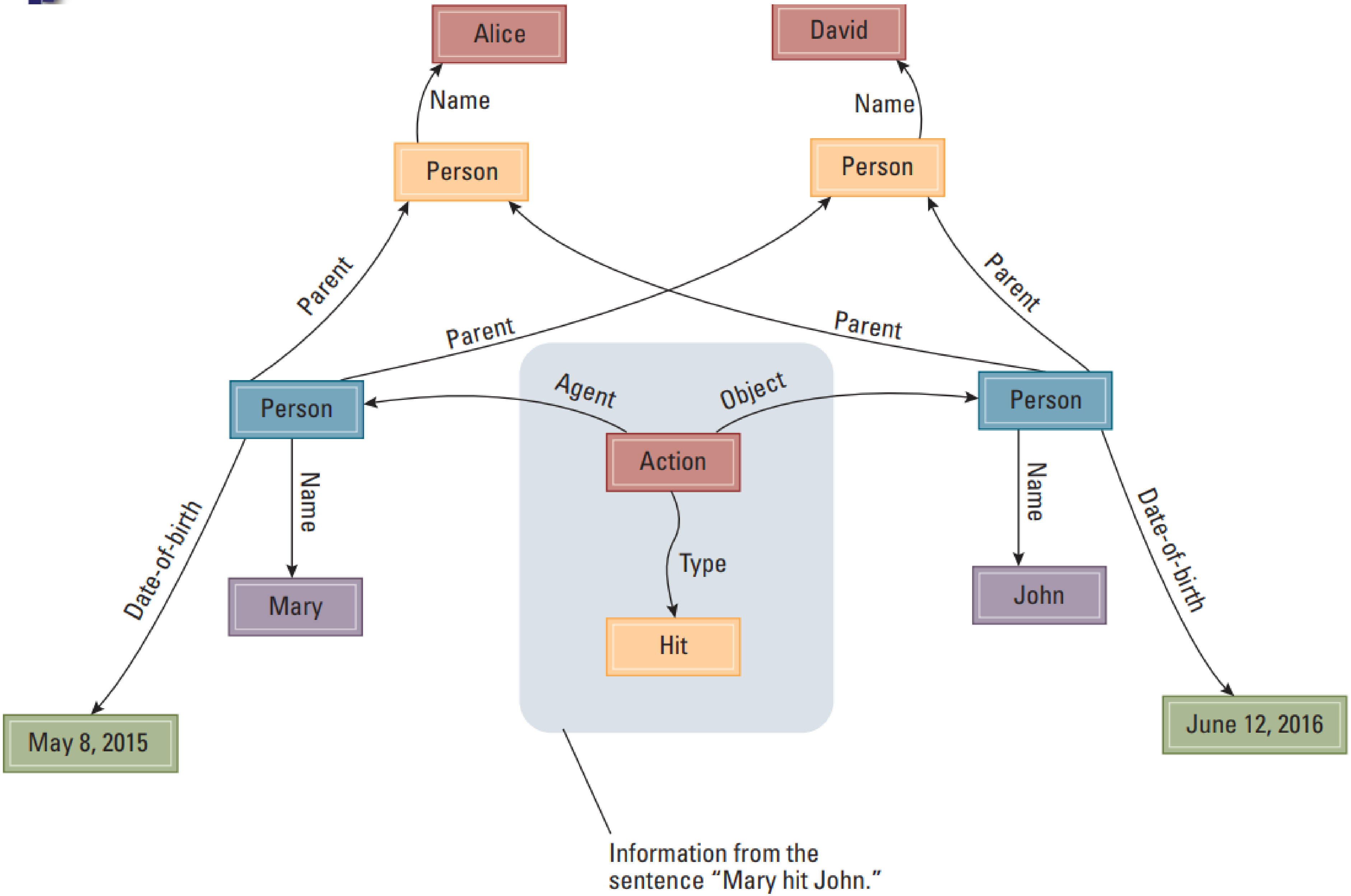
Other research in NLP

- Information retrieval
  - Task of identifying documents that relate to the topic at hand.
  - Search sites for key words. “Automobiles” vs “cars.”
  - Needed a search mechanism that understands the contents of the sites being considered.

Other research in NLP

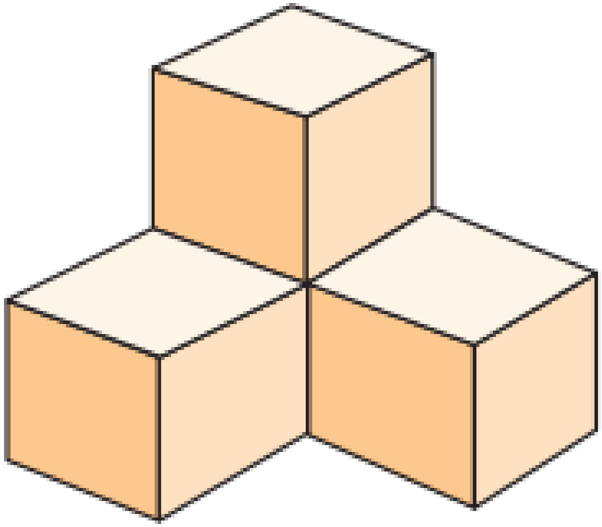
- Information extraction
  - Task of extracting information from documents so that it takes a form that is useful in other applications
  - Frame and semantic net

A semantic net



Quiz 1

- How many blocks are in the stack represented next?
- How could a machine be programmed to answer such questions accurately?

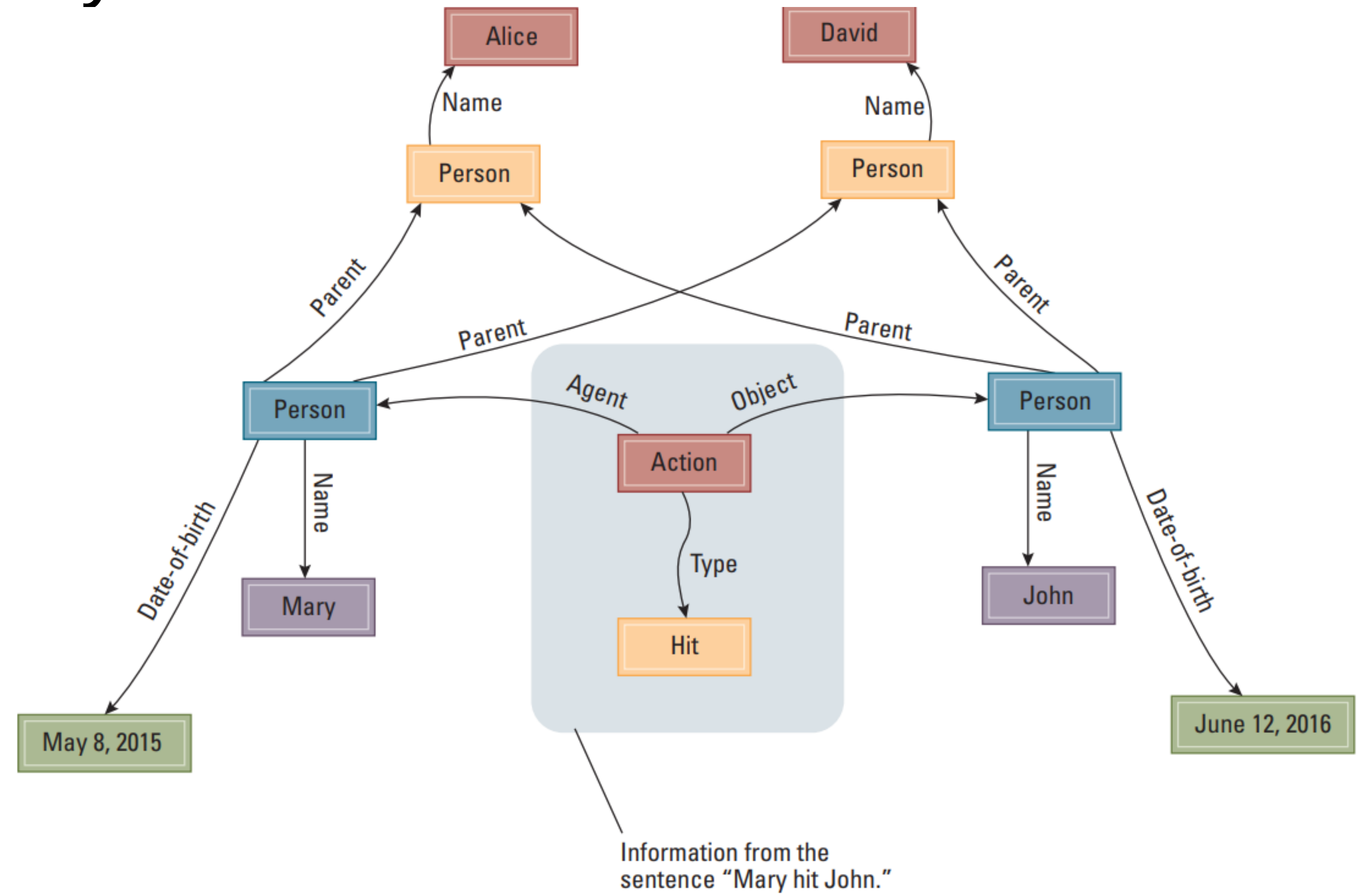


Quiz 2

- Compare the results of parsing the following two sentences. Then, explain how the sentences differ semantically
  - The farmer built the fence in the field.
  - The farmer built the fence in the winter.

Quiz 3

- What is the family relationship between Mary and John?



Content

- Intelligence and Machines
- Perception
- **Reasoning**
- Additional Areas of Research
- Artificial Neural Networks
- Robotics
- Considering the Consequences
- About Computer Science Department

Reasoning

- Reasoning is the act of deriving a conclusion from certain premises using a given methodology.
- Reasoning is a process of thinking.
- Reasoning is logically arguing.
- Reasoning is drawing inference.
- It must figure out what it needs to know from what it already knows

Reasoning Example

If we know :

Robins are birds, and

All birds have wings

Then if we ask:

Do robins have wings?

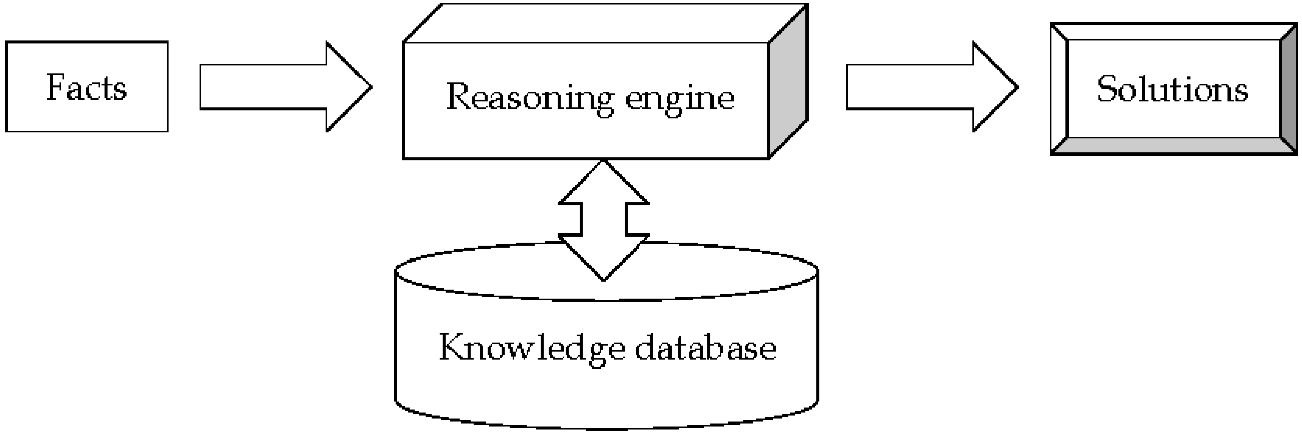
To answer this question - some reasoning must go.

Uncertainty in Reasoning

- The world is an uncertain place; often the Knowledge is imperfect (Incomplete, Inconsistent, Changing) which causes uncertainty.
  - Therefore reasoning must be able to operate under uncertainty.
- AI systems must have ability to reason under conditions of uncertainty.

Reasoning under uncertainty

- Many reasoning systems provide capabilities for reasoning under uncertainty.
- This is important when building situated reasoning agents which must deal with uncertain representations of the world.



Reasoning Systems

- Production systems
- Theorem provers
- Deductive retrieval systems
- Semantic networks
- …

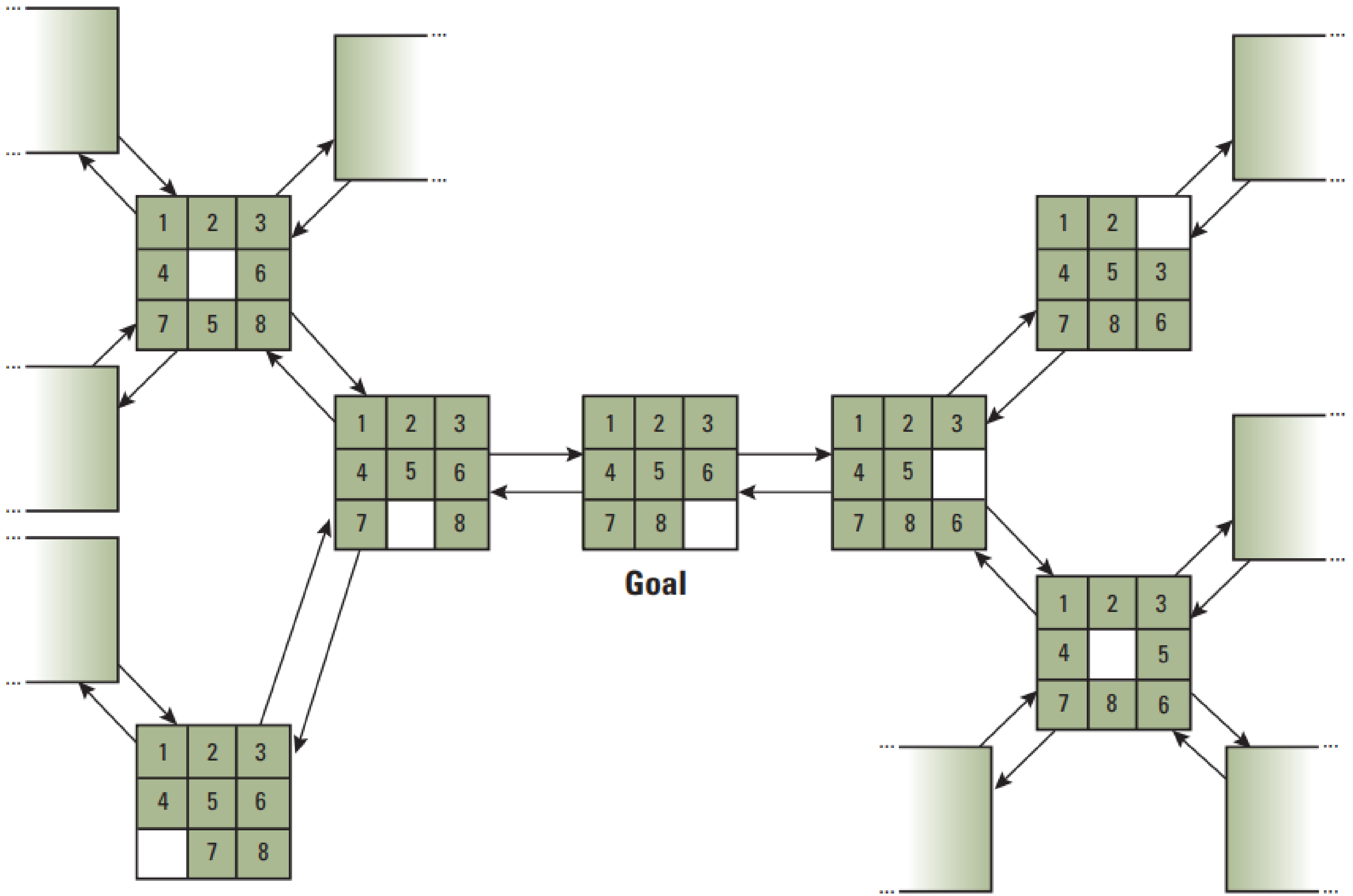
Production Systems

1. Collection of states
   - Start (or initial) state
   - Goal state (or states)
2. Collection of productions: rules or moves
   - A production is an operation that can be performed in the application environment to move from one state to another
   - Each production may have preconditions
3. Control system: decides which production to apply next

Reasoning by Searching

- **State Graph:** All states and productions
- **Search Tree:** A record of state transitions explored while searching for a goal state
  - Breadth-first search
  - Depth-first search

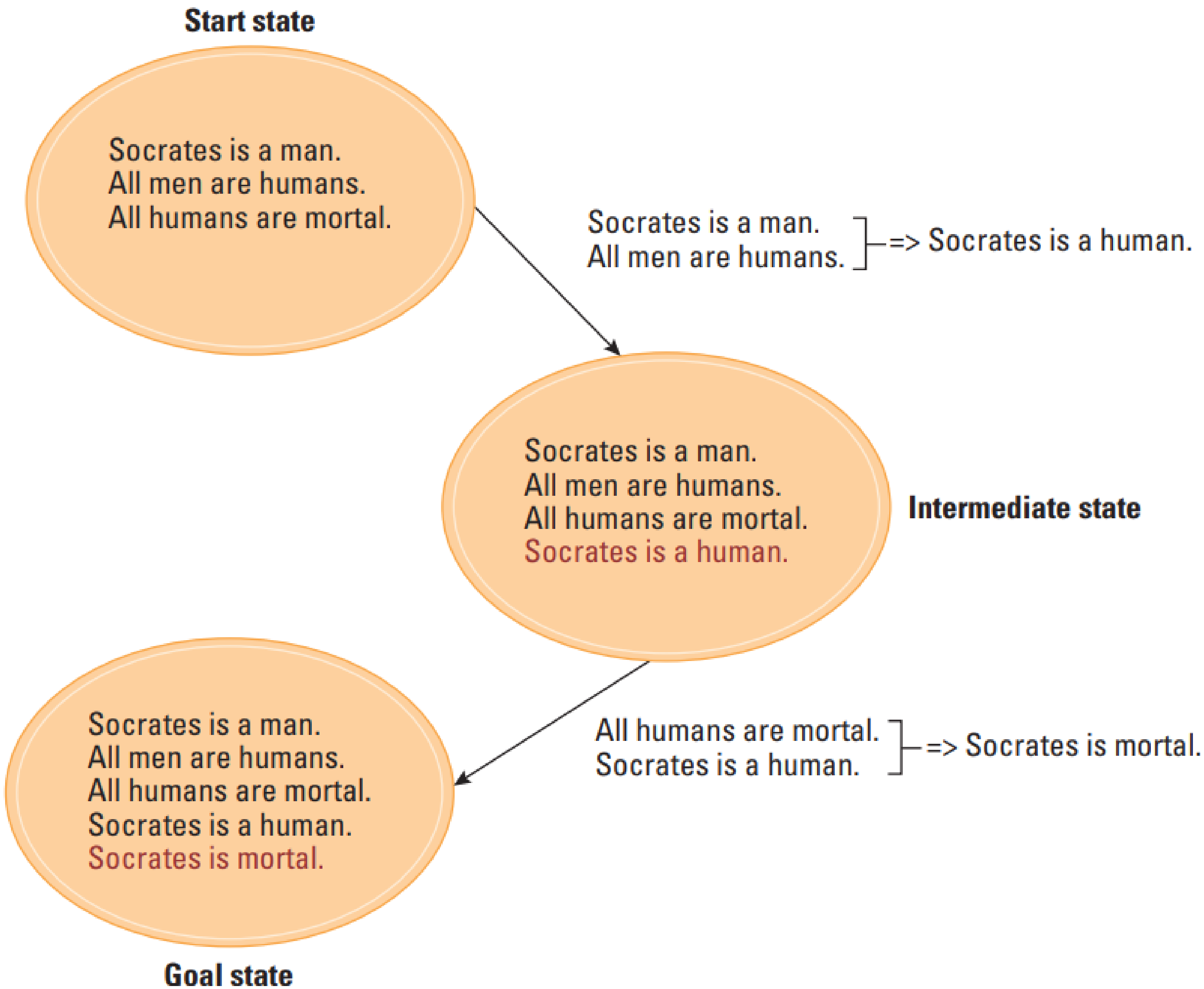
A small portion of the eight-puzzle's state graph



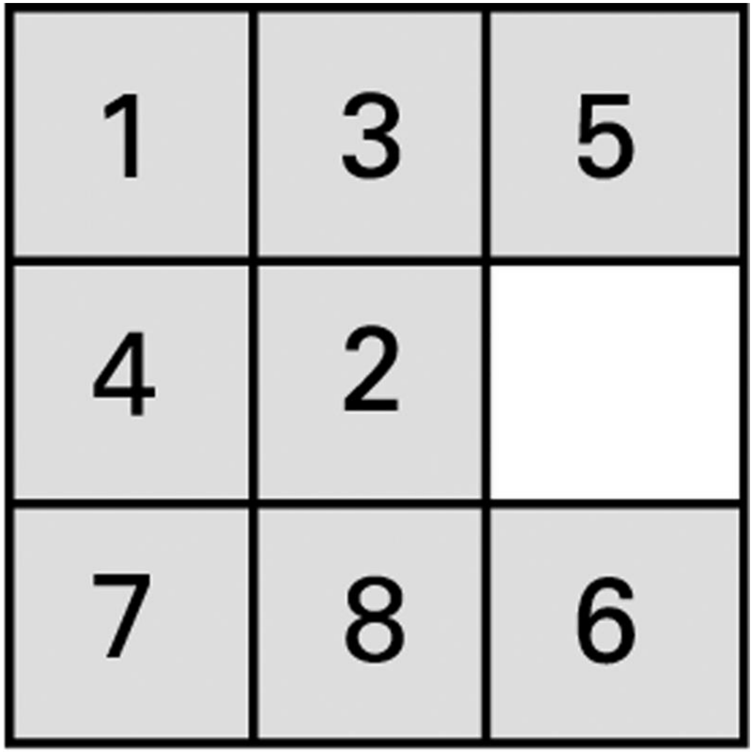
Reasoning by Searching

- The problem faced by the control system becomes that of finding a sequence of arrows that leads from the start state to the goal state.
- The task of the control system can be viewed as that of finding a path through a state graph.
- If a problem can be characterized in terms of a production system, then its solution can be formulated in terms of searching for a path.
- Ex: chess

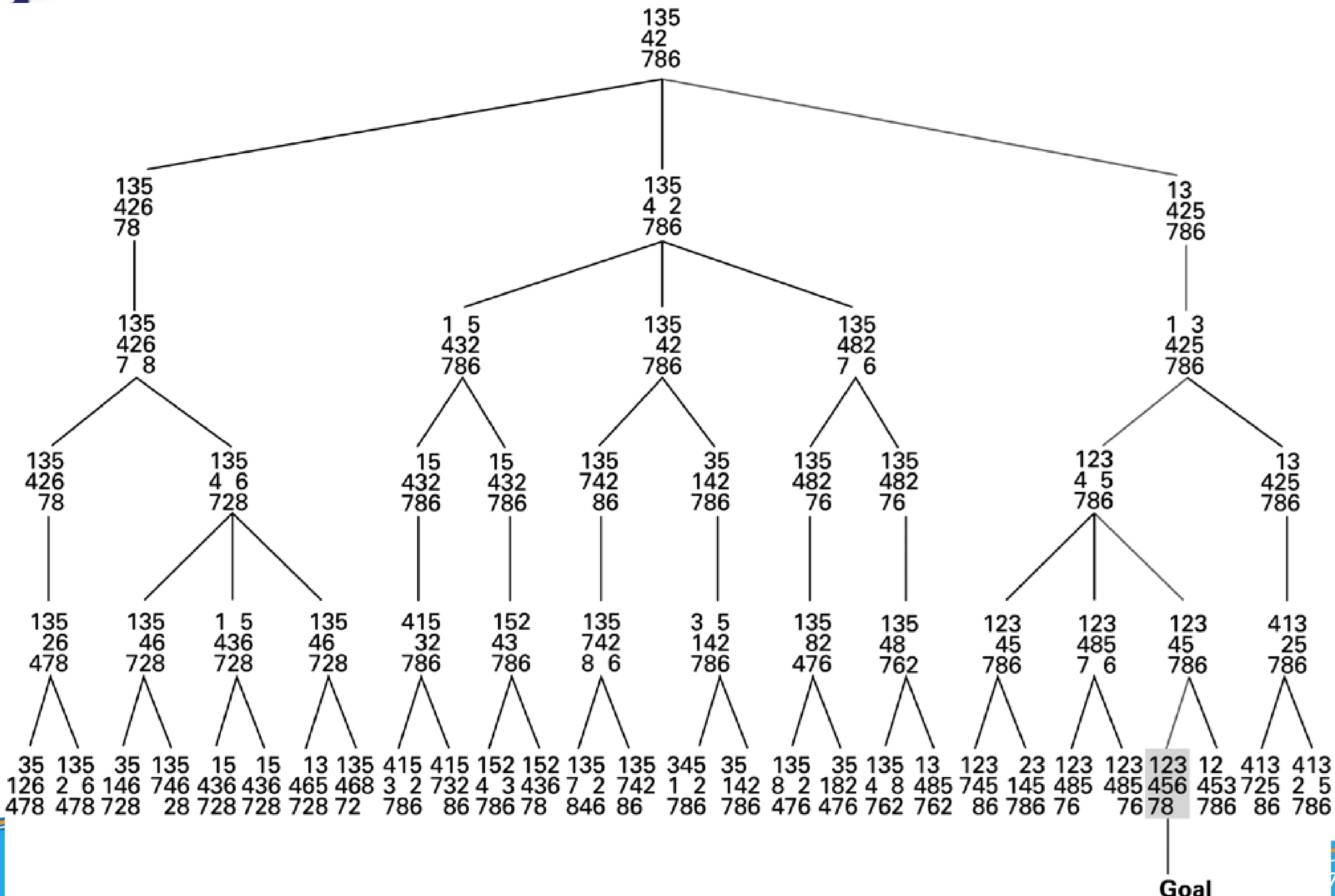
Deductive reasoning in the context of a production system



An unsolved eight-puzzle



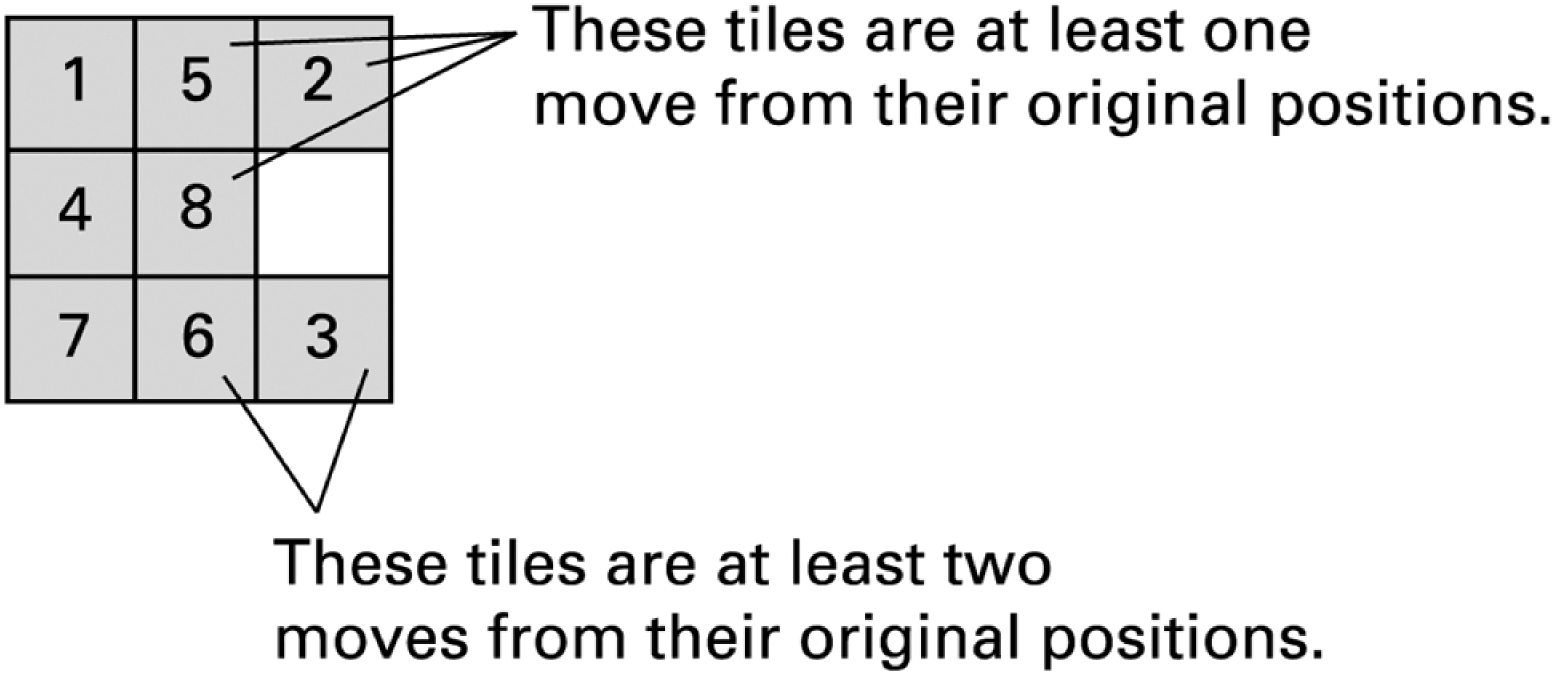
A sample search tree



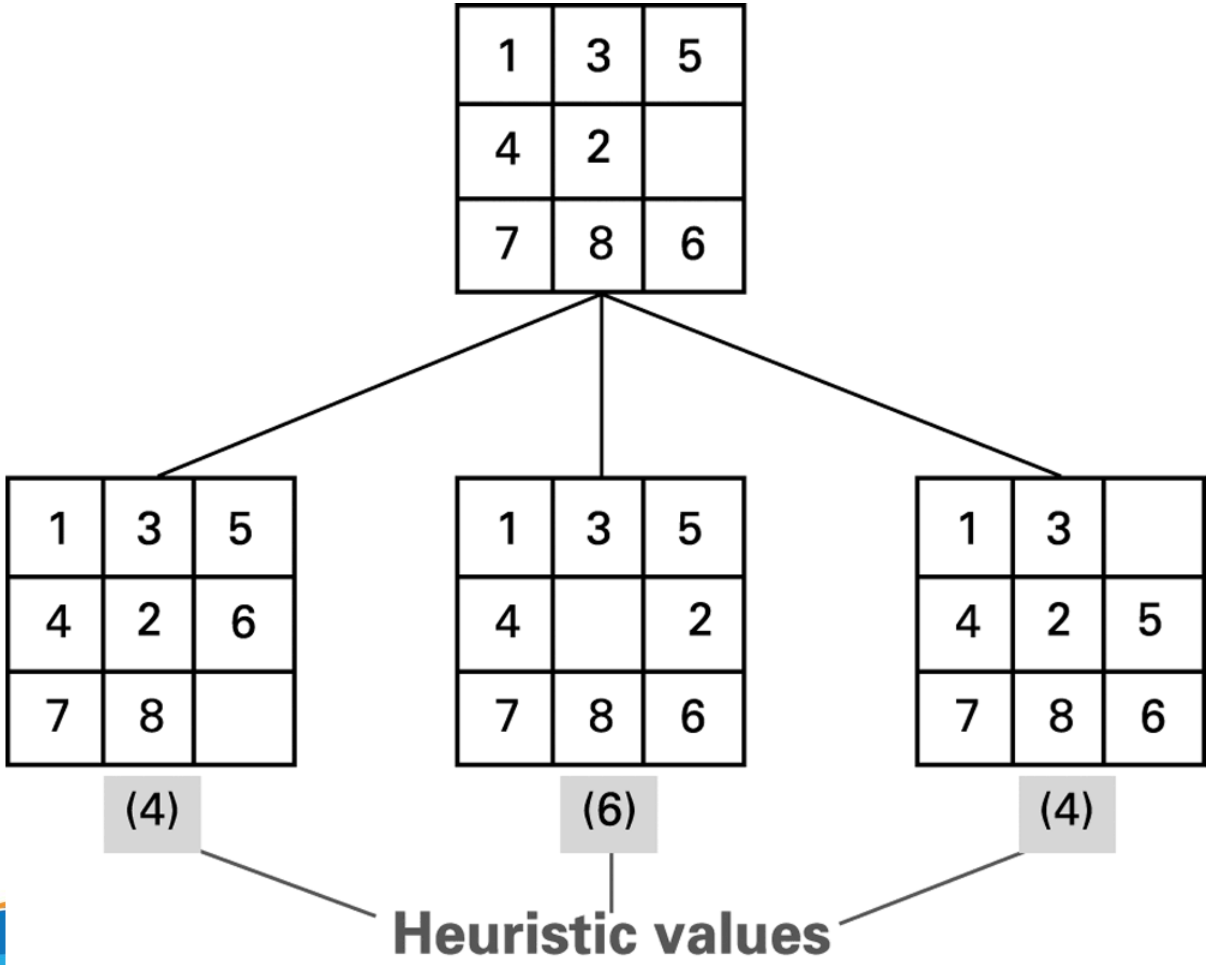
Heuristic Strategies

- **Heuristic:** A “rule of thumb” for making decisions
- Requirements for good heuristics
  - Must be easier to compute than a complete solution
  - Must provide a reasonable estimate of proximity to a goal

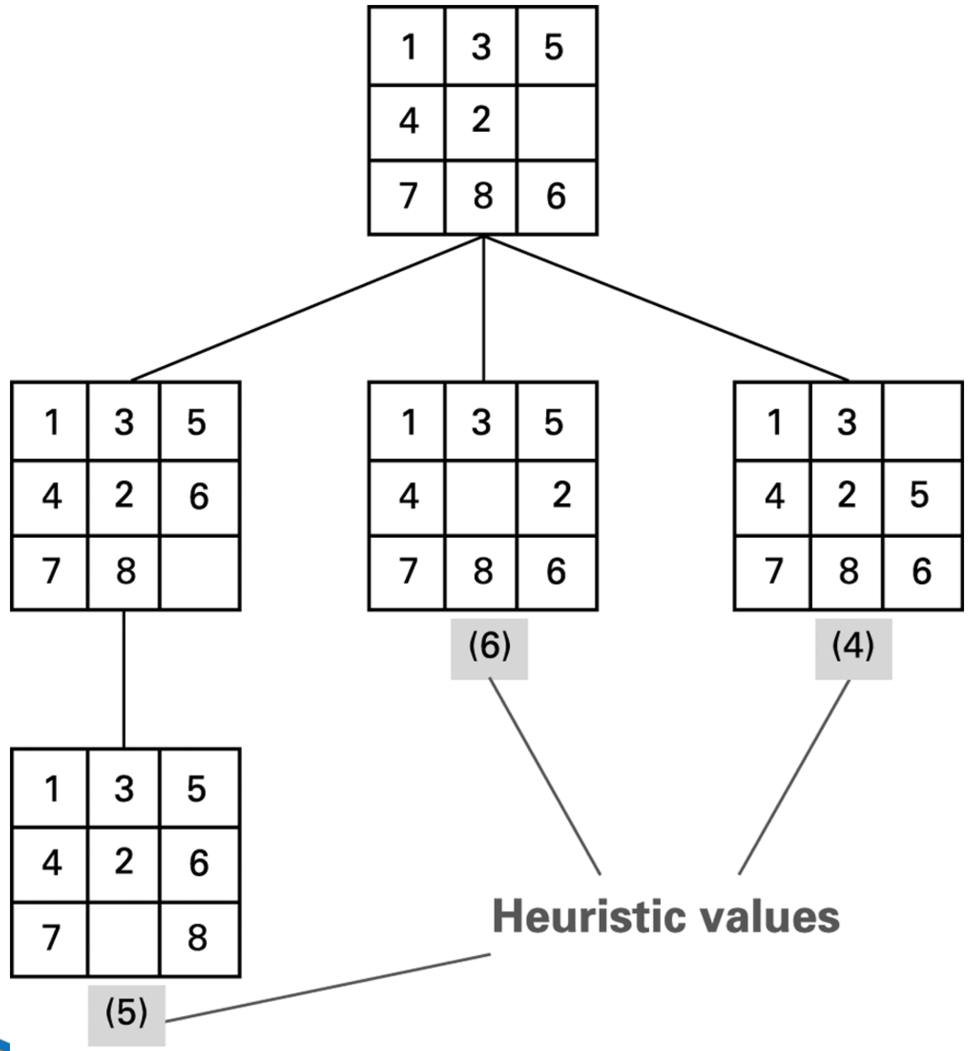
An unsolved eight-puzzle



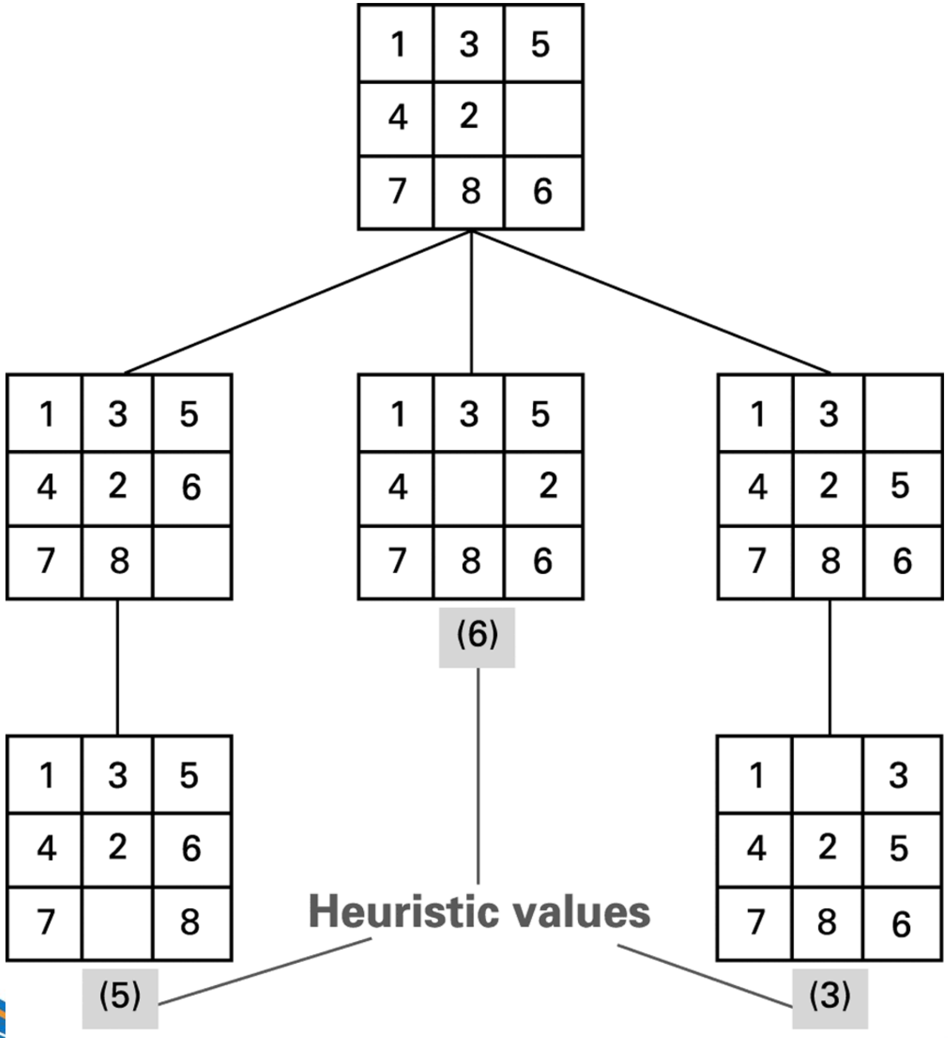
The beginnings of heuristic search

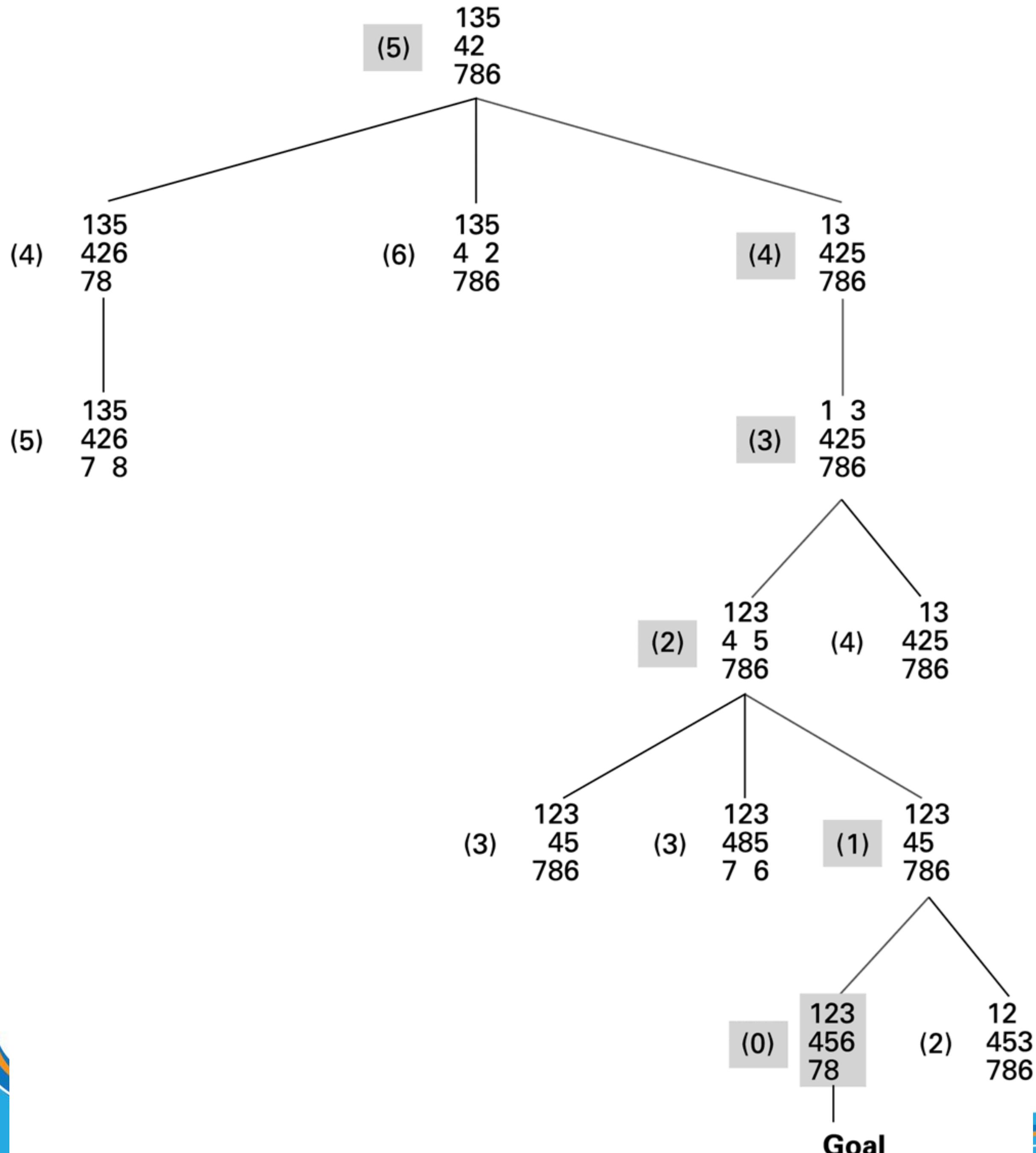


The search tree after two passes



The search tree after three passes





(5) 135 42 786
(4) 135 426 78
(6) 135 4 2 786
(4) 13 425 786
(5) 135 426 7 8
(3) 1 3 425 786
(2) 123 4 5 786
(4) 13 425 786
(3) 123 45 786
(3) 123 485 7 6
(1) 123 45 786
(0) 123 456 78
(2) 12 453 786
Goal

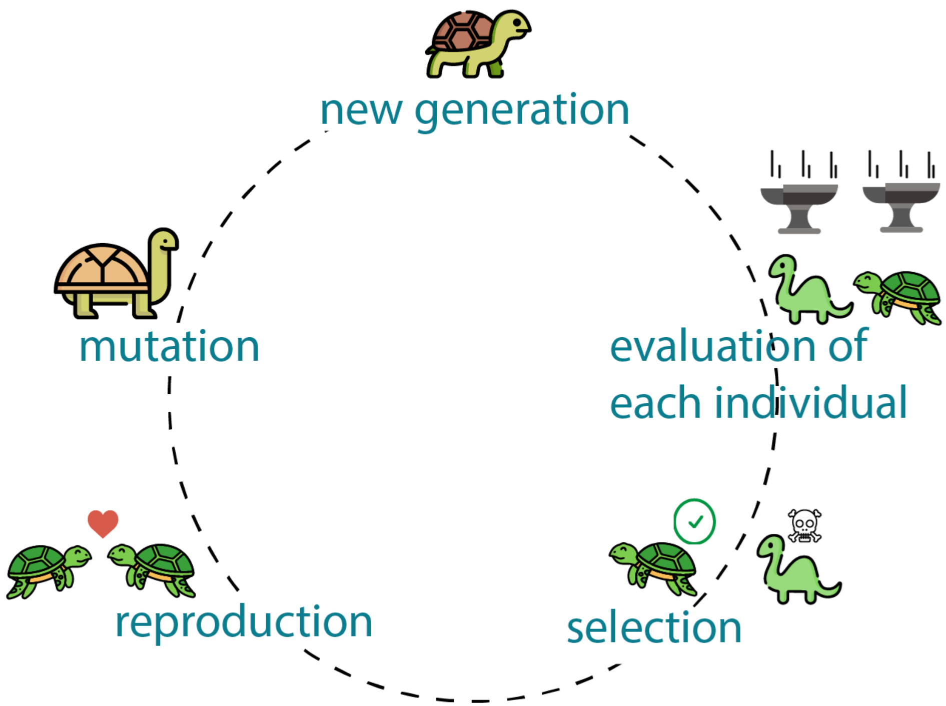
Content

- Agent
- Perception
- Reasoning
- **Additional Areas of Research**

Genetic Algorithms

- There are some problems that are too complex to be solved such as:
  - execution exceeds available memory or
  - cannot be completed within a reasonable amount of time.
- A solution can sometimes be discovered through an evolutionary process involving many generations of trial solutions.
  - This strategy is the foundation for what is called genetic algorithms.

Genetic Algorithms



Genetic Algorithms

- Begins by generating a random pool of trial solutions:
  - Each solution is a **chromosome**
  - Each component of a chromosome is a **gene**
- Repeatedly generate new pools
  - Each new chromosome is an offspring of two parents from the previous pool
  - Probabilistic preference used to select parents
  - Each offspring is a combination of the parent's genes

Genetic Algorithms