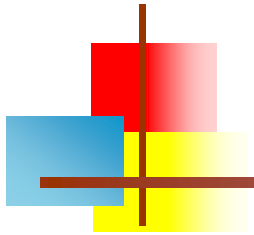


Artificial Intelligence

Intelligent systems

Intelligent systems

- the system that incorporates intelligence into applications being handled by machines
- perform search and optimization along with learning capabilities.
- different types of machine learning such as supervised, unsupervised and reinforcement learning can be modeled in designing intelligent systems.
- Expert systems, intelligent agents and knowledge-based systems are examples of intelligent systems
- from automated vacuums such as the Roomba to facial recognition programs to Amazon's personalized shopping suggestions

Characteristics

- Self-explaining
  - the system can explain how it came to a certain decision
- Robust
  - property of a system means that the system behaves well and adequate not only under ordinary conditions, but also under unusual conditions
- fault tolerant
  - continue to adequately perform even if one or more of its internal system components fail or break.
- Adaptive
  - react to changes, in particular to changes in the environment or the context of the system
- self-optimizing
  - organize their internal components and capabilities in new structures without a central or an external authority in place

Characteristics

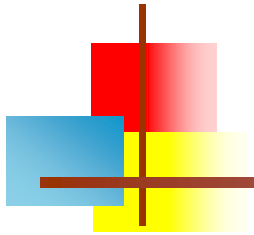
- Deductive
  - based on a set of axioms and rules, they can deduct new insights by applying the rules to the axioms as well as to the resulting new facts.
  - using an underlying inference engine; deductive systems can discover new facts that they can use for their decision process
  - deductive systems can discover new facts that they can use for their decision process
- Learning
  - observe the achieved results and compare them with the desired outcome.
- Cooperative
  - expose social capabilities; interact with other systems – and potentially humans as well
- Autonomous

Characteristics

- Autonomous
  - performs the desired tasks and behaves well and adequate even in unstructured environments without continuous human guidance.
- Agile
  - able to manage and apply knowledge effectively so that they behave well and adequate in continuously changing and unpredicted environments.

Steps to build AI systems

- Identify the problem
  - *What are you trying to solve?*
  - *Which result is desired?*
- Preparation of the data (preprocessing)
  - *structured and unstructured data*
  - *80% of their time cleaning , moving, reviewing, and organizing*
- Choice of algorithms (model)
  - *Supervised learning*
  - *Unsupervised learning & reinforcing learning*
- Training the algorithms
- Choosing the most suitable programming language
- Platform selection
- Test the Model
- Deployment

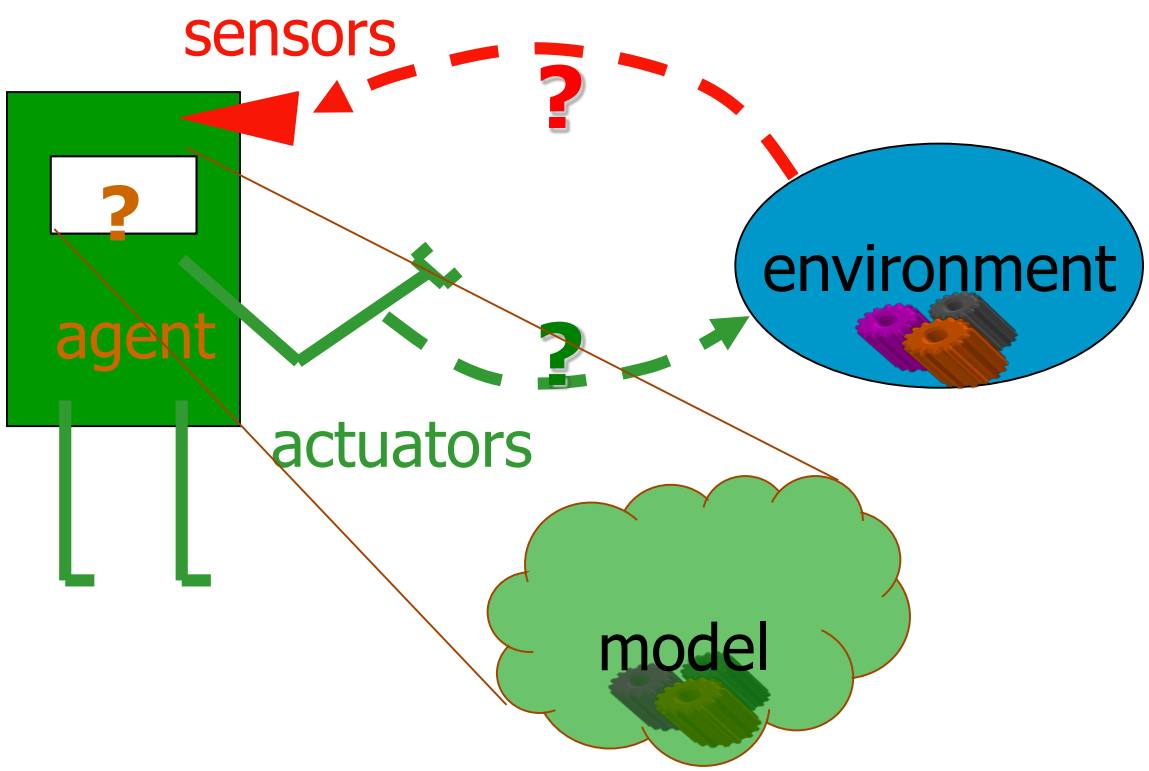


Artificial Intelligence

Intelligent agents

Course – Introduction (recall)

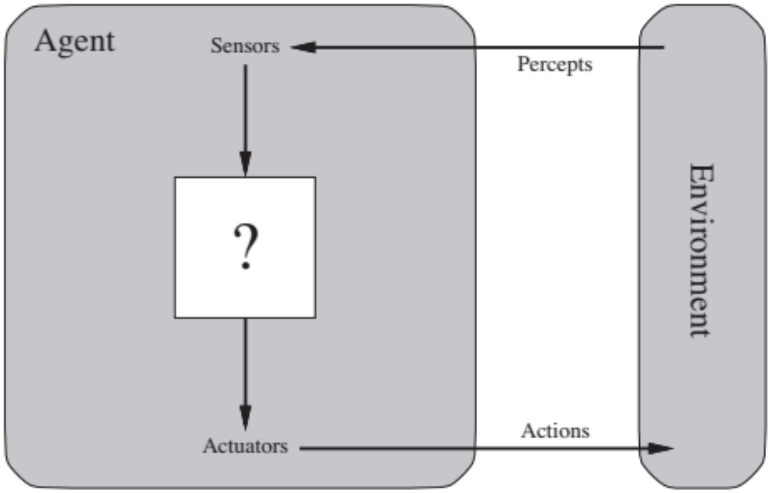
- The main unifying theme is the idea of an **intelligent agent**.
- define AI as the study of agents that receive percepts from the environment and perform actions.



Agents And Environments

- An agent is anything that can be viewed as perceiving its environment through sensors
- acting upon that environment through actuators.
  - A robotic agent might have cameras and infrared range finders for sensors and various motors for actuators.
  - A software agent receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

Intelligent agents are supposed to maximize their performance measure.

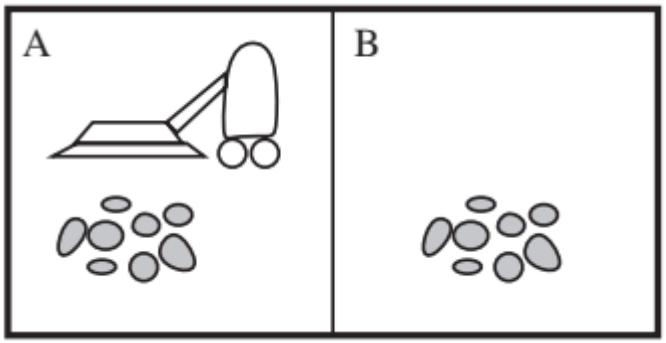


Agent Function and Program

- percept to refer to the agent's perceptual inputs at any given instant
- percept sequence is the complete history of everything the agent has ever perceived.
- Thus, *an agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived.*
- an agent's behavior is described by the **agent function** that maps any given percept sequence to an action
- the agent function for an artificial agent will be implemented by an **agent program**

The agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.

Agent Function



A vacuum-cleaner world with just two locations.

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*
⋮	⋮
[A, Clean], [A, Clean], [A, Clean]	*Right*
[A, Clean], [A, Clean], [A, Dirty]	*Suck*
⋮	⋮

a simple agent function for the vacuum-cleaner world

Rational agent

- *Def:*

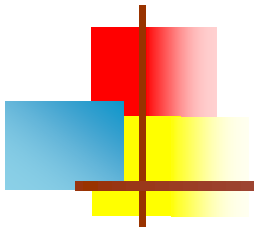
  For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- task environment - PEAS (Performance, Environment, Actuators, Sensors)

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Rational agent

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry



Properties of Task Environment

- Observable or partially observable?
- Discrete or Continuous?
- Deterministic or Stochastic?
- Static or Dynamic?
- Episodic or Sequential?
- Multiple or Single Agent?

Agent Program

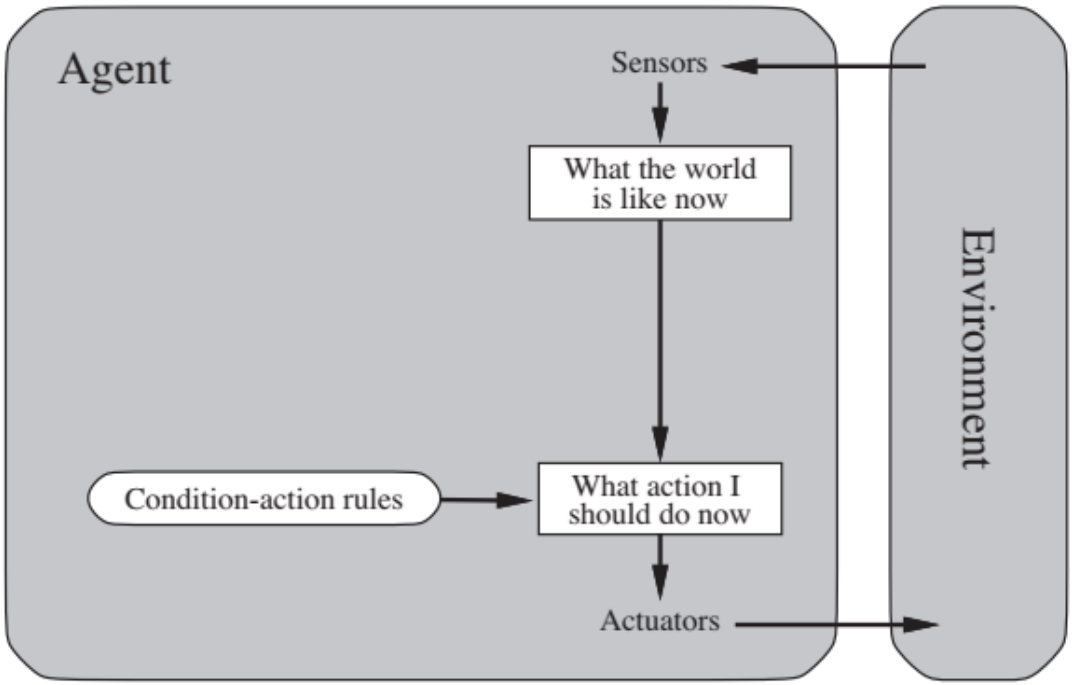
- The job of AI is to design an agent program that implements the agent function — the mapping from percepts to actions.
- program will run on some sort of computing device with physical sensors and actuators — architecture

$$agent = architecture + program$$

- Four basic kinds of agent programs
  - Simple reflex agents;
  - Model-based reflex agents;
  - Goal-based agents; and
  - Utility-based agents
  - Learning based agents
- Each kind of agent program combines particular components in particular ways to generate actions

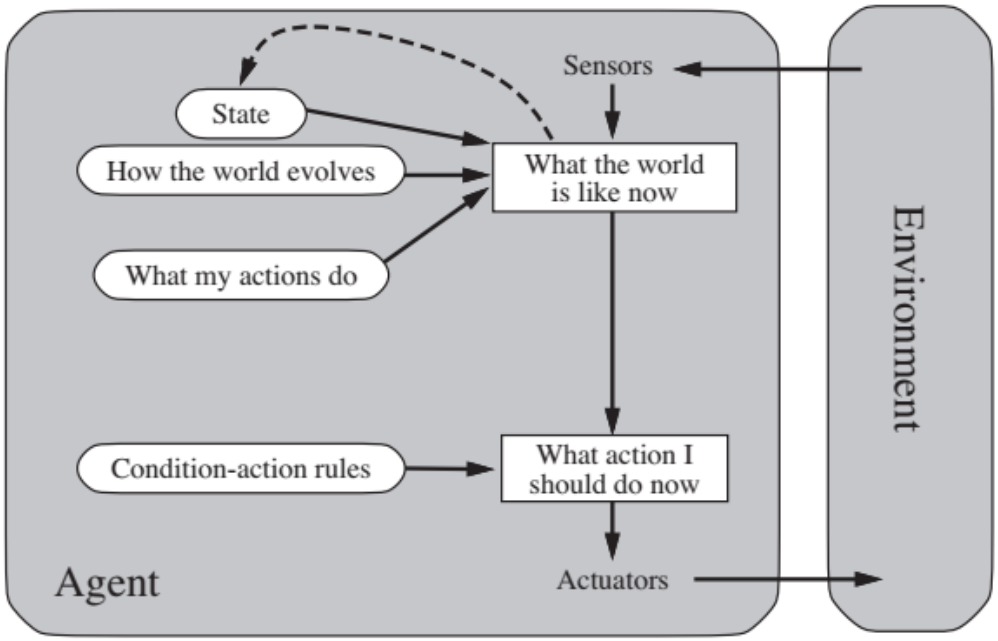
Simple reflex agents

- agents select actions on the basis of the **current percept**, ignoring the rest of the percept history
- **condition–action rule**



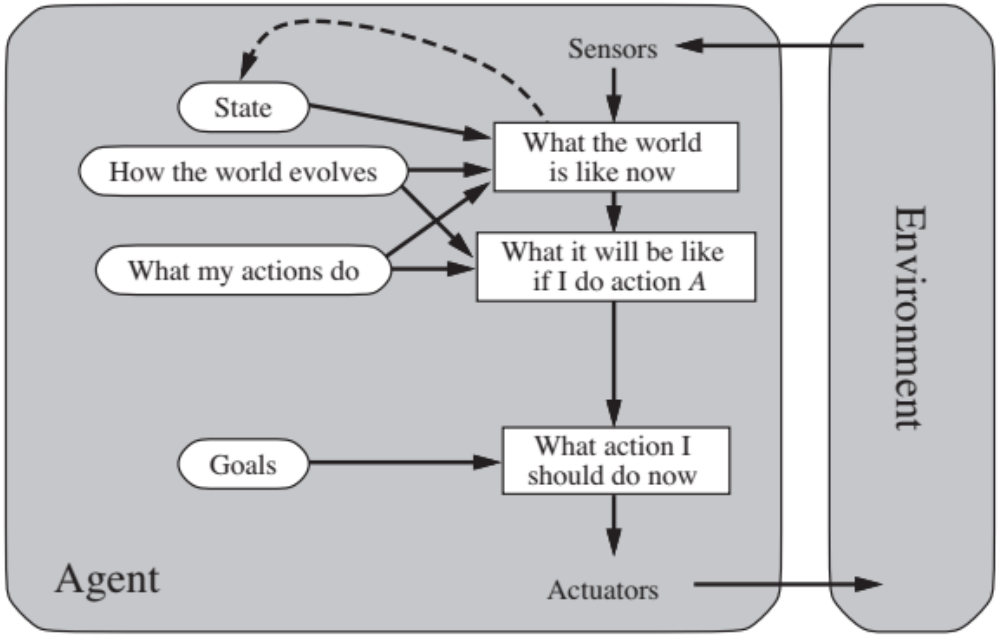
Model-based reflex agents

- 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.
- Internal state - percept history
- Requires two knowledge,
  - First, some information about how the world evolves independently of the agent
  - Second, some information about how the agent's own actions affect the world



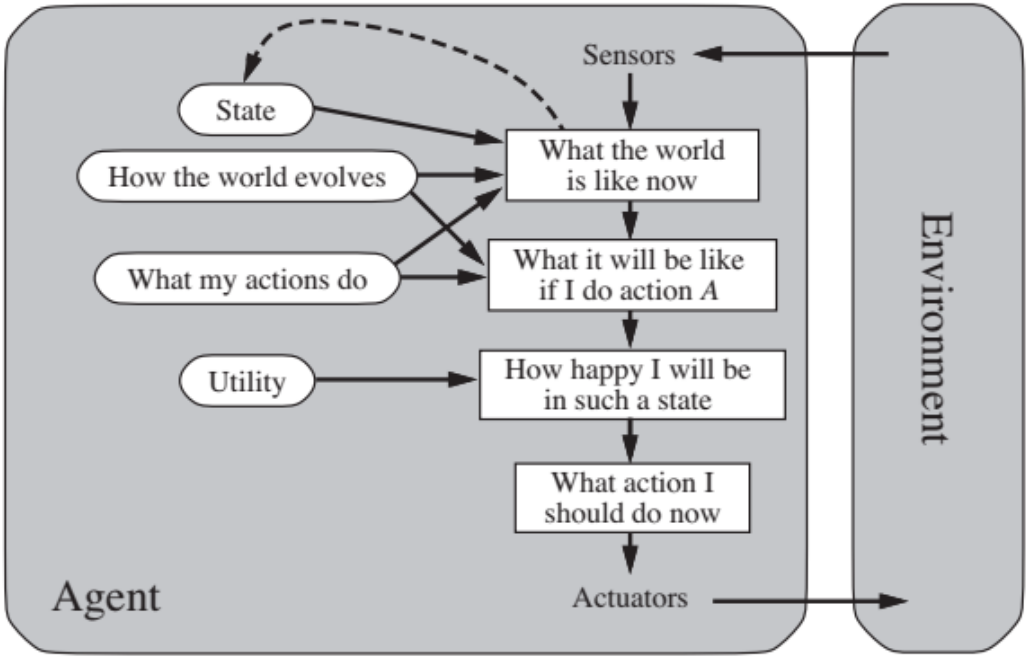
Goal-based agents

- Knowing something about the current state of the environment is not always enough to decide what to do.
  - at a road junction, the taxi can turn left, turn right, or go straight on.
- the agent needs some sort of goal information that describes situations that are desirable
  - being at the passenger's destination
- goal-based action selection is
  - straightforward
  - tricky
- Search - subfields of AI devoted to finding action sequences that achieve the agent's goals.



Utility-based agents

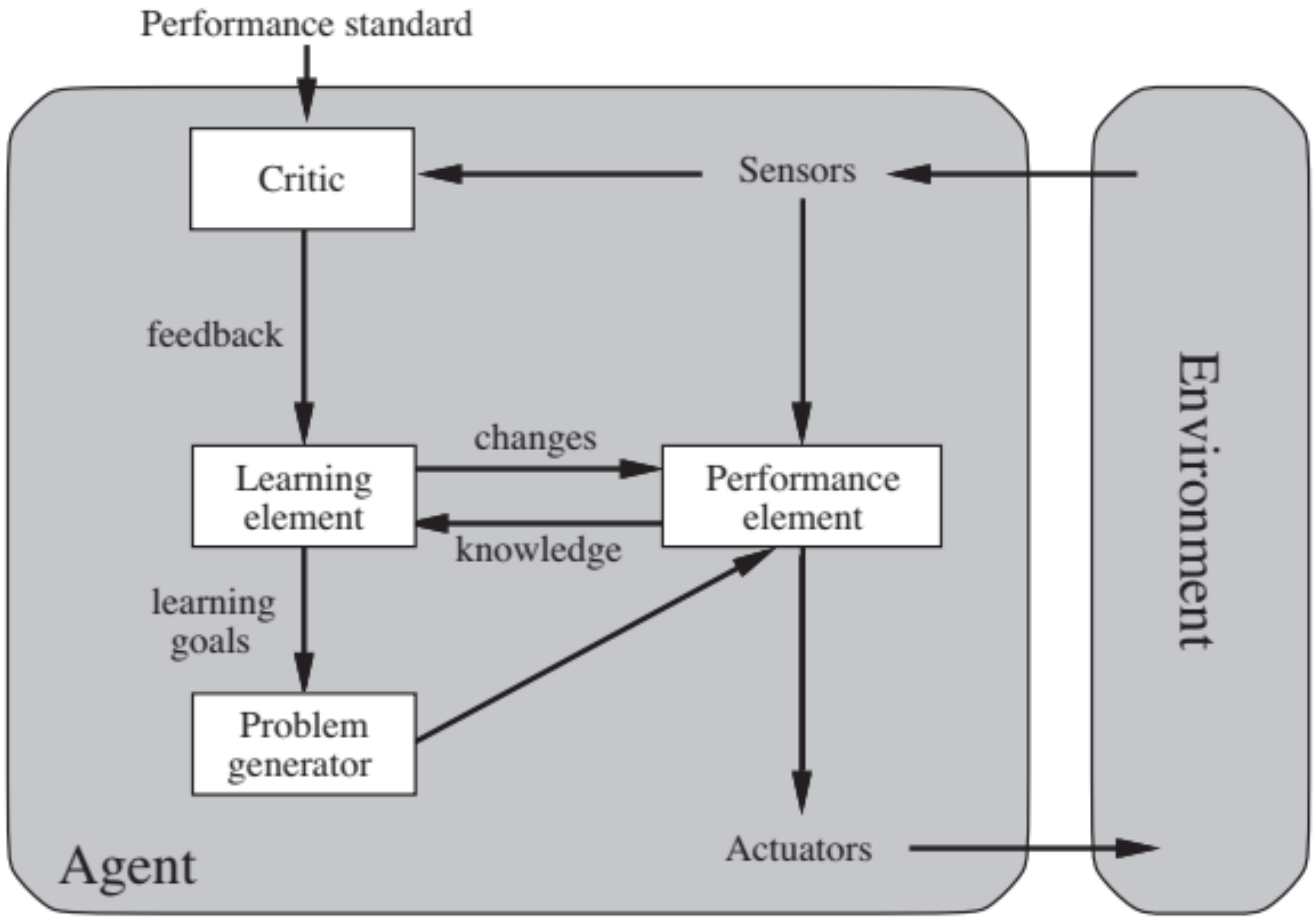
- Utility - general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent.
- utility function that measures its preferences among states of the world.
- it chooses the action that leads to the best expected utility
- An agent's utility function is essentially an internalization of the performance measure



Learning agents

- learning element is responsible for making improvements
- performance element is responsible for selecting external actions.
- The learning element uses feedback from the critic on how the agent is doing
  - determines how the performance element should be modified to do better in the future
- critic tells the learning element how well the agent is doing with respect to a fixed performance standard
- problem generator suggesting actions that will lead to new and informative experiences
  - job is to suggest these exploratory actions

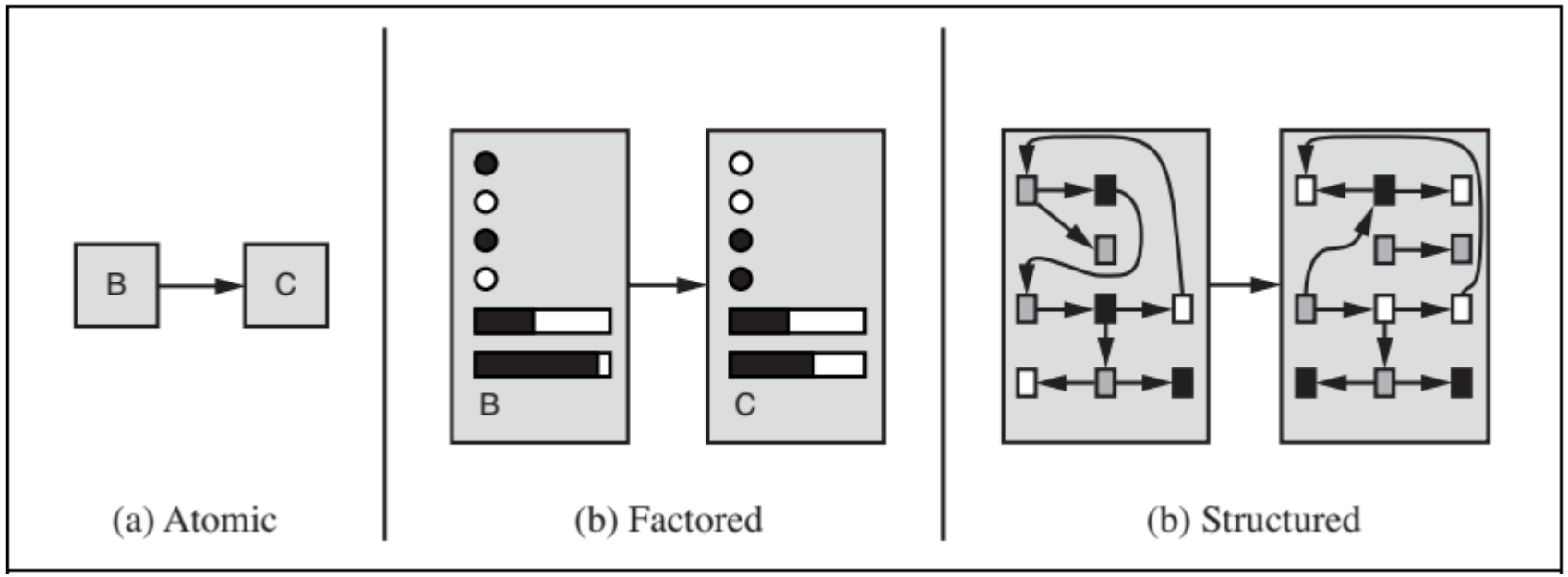
Learning agents



State Representation

- Three ways to represent states and the transitions between them,
- Atomic : a state is a black box with no internal structure
  - each state of the world is indivisible—it has no internal structure
  - search and game-playing - work with atomic representations
- Factored : a state consists of a vector of attribute values
  - splits up each state into a fixed set of variables or attributes, each of which can have a value.
  - values can be Boolean, real valued, or one of a fixed set of symbols
  - propositional logic and machine learning algorithms
- Structured : a state includes objects, each of which may have attributes of its own as well as relationships to other objects.
  - relational databases and first-order logic

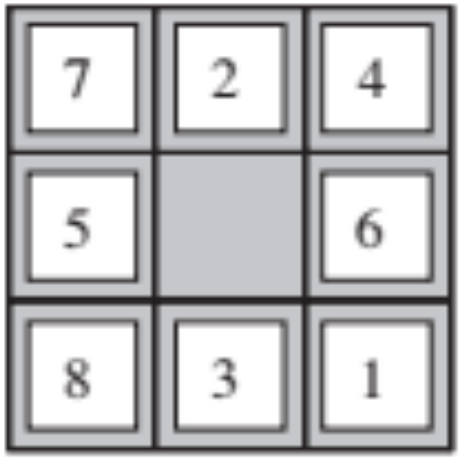
State Representation



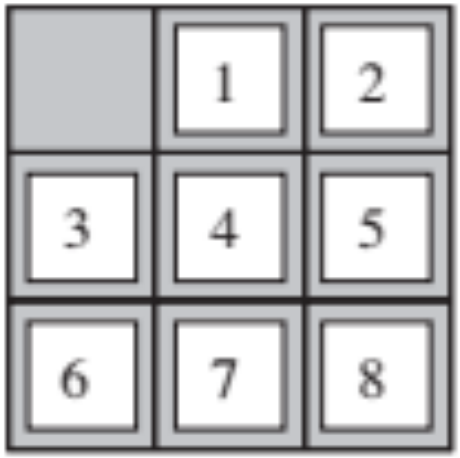
Problem Solving Agent

- The simplest agents were the reflex agents - base their actions on a direct mapping from states to actions.
  - cannot operate well in environments for which this mapping would be too large to store
- Goal-based agents consider future actions and the desirability of their outcomes
- One kind of goal-based agent called a problem-solving agent
- A **problem** can be defined formally by five components:
  - Initial state
  - Actions
  - Transition model
  - Goal test
  - Path cost

8-puzzle



Start State



Goal State

8-puzzle

- **States**: A state description specifies the location of each of the eight tiles and the blank in one of the nine squares.
- **Initial state**: Any state can be designated as the initial state. Note that any given goal can be reached from exactly half of the possible initial states.
- **Actions**: The simplest formulation defines the actions as movements of the blank space Left, Right, Up, or Down. Different subsets of these are possible depending on where the blank is.
- **Transition model**: Given a state and action, this returns the resulting state; for example, if we apply Left to the start state, the resulting state has the 5 and the blank switched.
- **Goal test**: This checks whether the state matches the goal configuration
- **Path cost**: Each step costs 1, so the path cost is the number of steps in the path