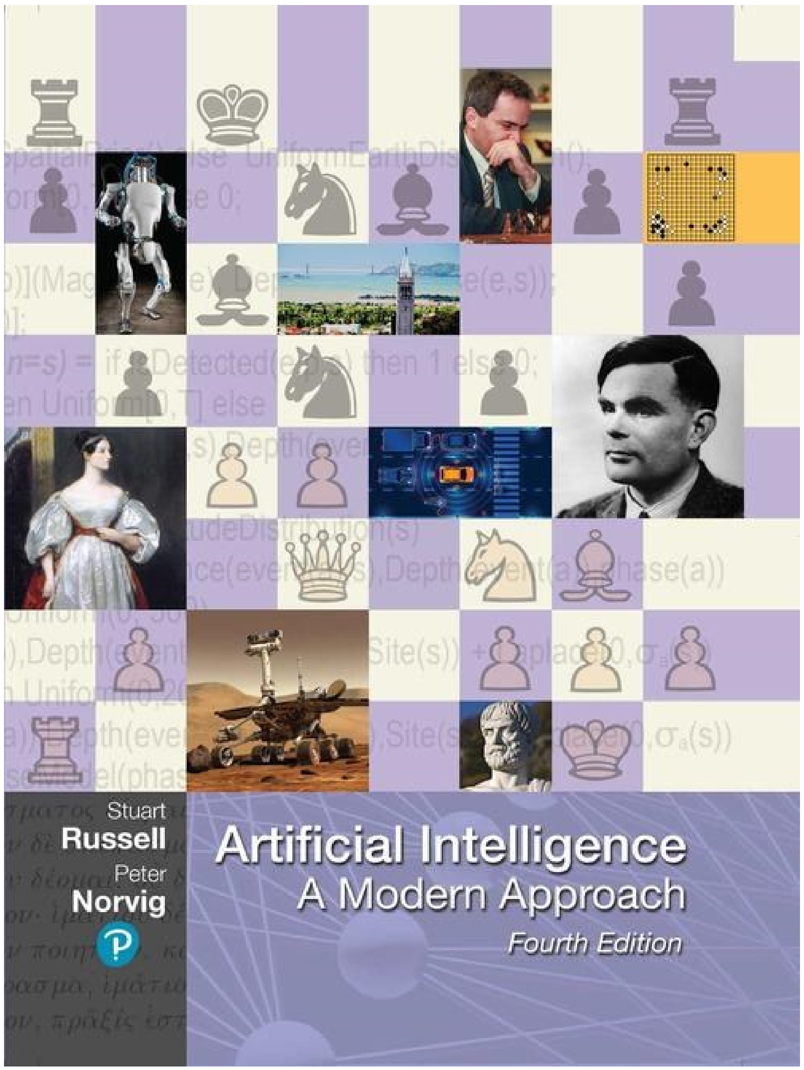
Artificial Intelligence: A Modern Approach

Fourth Edition



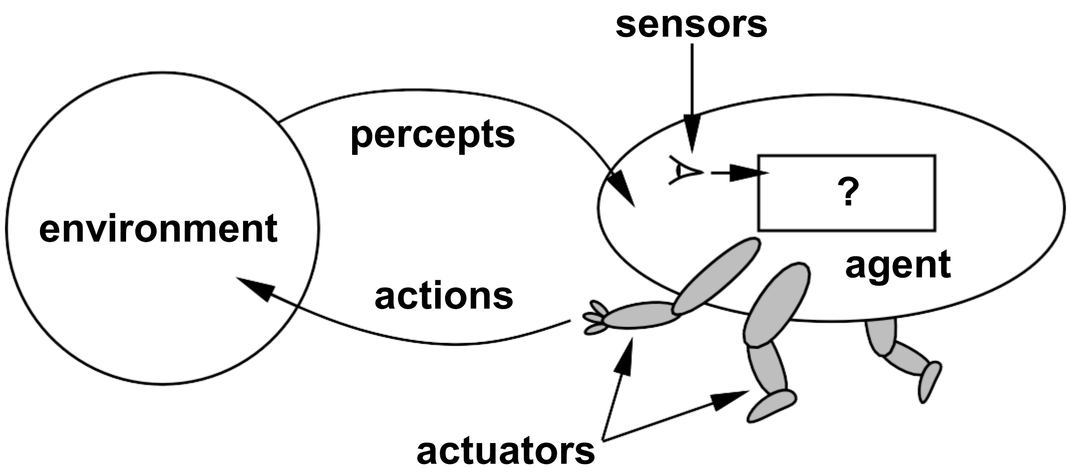
Lecture 1

Intelligent Agents

Outline

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Agents and environments



Agents include humans, robots, softbots, thermostats, etc.

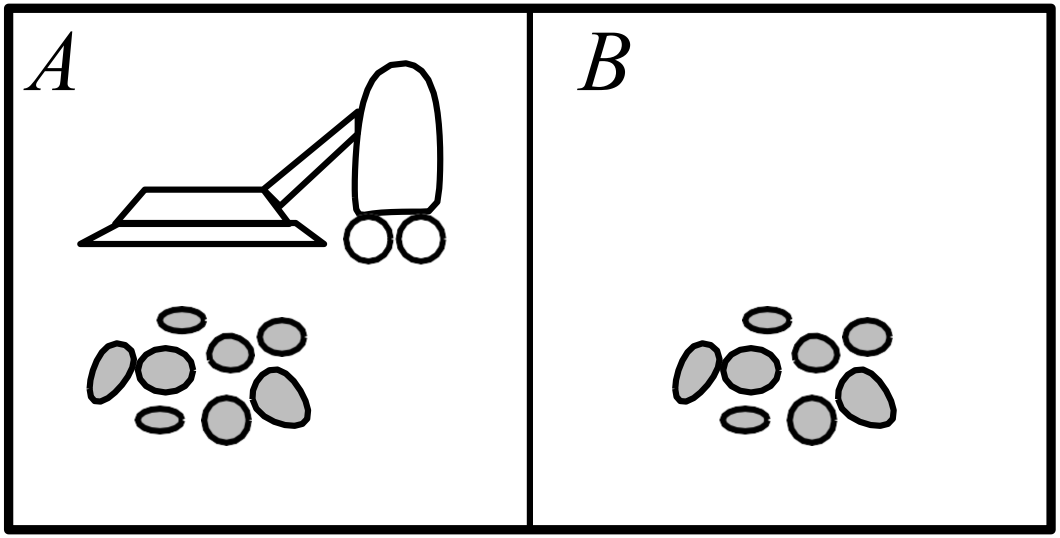
An agent can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators

The agent function maps from percept histories to actions:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

The agent program runs on the physical architecture to produce f

Vacuum-cleaner world



Percepts: location and contents, e.g., [*A*, *Dirty*]

Actions: *Left*, *Right*, *Suck*, *N oOp*

A vacuum-cleaner agent

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*
.	.

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*

What is the right way to fill out this table?
What makes the agent good or bad?
What is the right function?
Can it be implemented in a small agent program?

Rationality

What is rational at any given time depends on four things:

- The performance measure that defines the criterion of success.
- The agent’s prior knowledge of the environment.
- The actions that the agent can perform.
- The agent’s percept sequence to date.

This leads to a **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.

Rationality

What is rational at any given time depends on four things:

- The performance measure that defines the criterion of success.

  The performance measure awards one point for each clean square at each time step, over a "lifetime" of 1000 time steps.

- The agent's prior knowledge of the environment.

  The "geography" of the environment is known a priori but the dirt distribution and the initial location of the agent are not. Clean squares stay clean and sucking cleans the current square. The and actions move the agent one square except when this would take the agent outside the environment, in which case the agent remains where it is.

- The actions that the agent can perform.

  The only available actions are *Right, Left* and *Suck*.

- The agent's percept sequence to date.

  The agent correctly perceives its location and whether that location contains dirt.

This leads to a **definition of a rational vacuum-cleaner agent**: *Under these circumstances the agent is indeed rational; its expected performance is at least as good as any other agent's.*

Rationality

Fixed performance measure evaluates the environment states sequence

- one point per square cleaned up in time T?
- one point per clean square per time step, minus one per move?
- penalize for $> k$ dirty squares?

A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rational ≠ omniscient (note example pg.59)

- percepts may not supply all relevant information
  Rational ≠ clairvoyant
- action outcomes may not be as expected
  Hence, rational ≠ successful

Rational ⇒ exploration, learning, autonomy

PEAS

The characteristics of the performance measure, environment, action space and percepts dictate approaches for selecting rational actions. They are summarized as the task environment.

Consider, e.g., the task of designing a self-driving car:

Performance measure??

Environment??

Actuators??

Sensors??

PEAS

To design a rational agent, we must specify the task environment

Consider, e.g., the task of designing a self-driving car:

Performance measure?? safety, destination, profits, legality, comfort, . . .

Environment?? US streets/freeways, traffic, pedestrians, weather, . . .

Actuators?? steering, accelerator, brake, horn, speaker/display, . . .

Sensors?? video, accelerometers, gauges, engine sensors, keyboard, GPS,speedometer . . .

Internet shopping agent

Performance measure??

Environment??

Actuators??

Sensors??

Internet shopping agent

Performance measure?? price, quality, appropriateness, efficiency

Environment?? current and future WWW sites, vendors, shippers

Actuators?? display to user, follow URL, fill in form

Sensors?? HTML pages (text, graphics, scripts)

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

Examples of agent types and their PEAS descriptions.

Environment types

Fully observable vs. partially observable

Whether the agent sensors give access to the complete state of the environment, at each point in time.

Deterministic vs. stochastic

Whether the next state of the environment is completely determined by the current state and the action executed by the agent.

Episodic vs. sequential

Whether the agent's experience is divided into atomic independent episodes.

Static vs. dynamic

Whether the environment can change, or the performance measure can change with time.

Cont…

Discrete vs. continuous

Whether the state of the environment, the time, the percepts or the actions are continuous.

Single agent vs. multi-agent

Whether the environment include several agents that may interact which each other.

Known vs. unknown

Reflects the agent's state of knowledge of the "law of physics" of the environment.

Environment types Examples

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	
Single-agent??	No Yes	No	Yes (except auctions)	No

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agent programs

Our goal is to design an agent program that implements the agent function.

Agent programs can be designed and implemented in many ways:

- with tables
- with rules
- with search algorithms
- with learning algorithms

Agent types

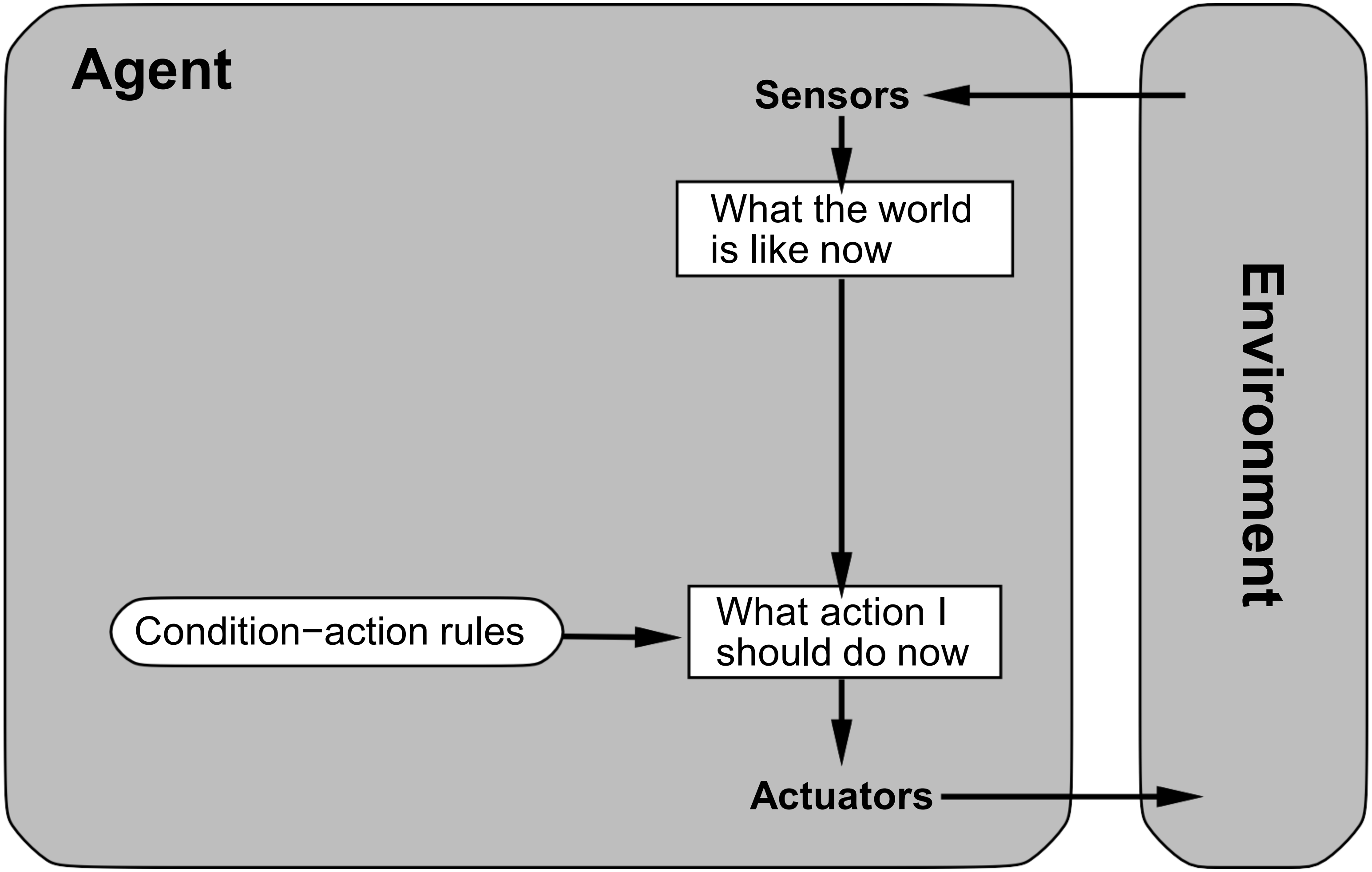
Four basic types in order of increasing generality:

– simple reflex agents
– model-based reflex agents
– goal-based agents
– utility-based agents

All these can be turned into learning agents

- Simple reflex agents select actions on the basis of the current percept, may have memory or model of the world's current state, ignoring the rest of the percept history.
- They implement condition-action rules that match the current percept to an action.
  - Rules provide a way to compress the function table.
  - Example (autonomous car): If a car in front of you slow down, you should break. The color and model of the car, the music on the radio or the weather are all irrelevant.
- They can only work in a Markovian environment, that is if the correct decision can be made on the basis of only the current percept. In other words, if the environment is fully observable.

Simple reflex agents



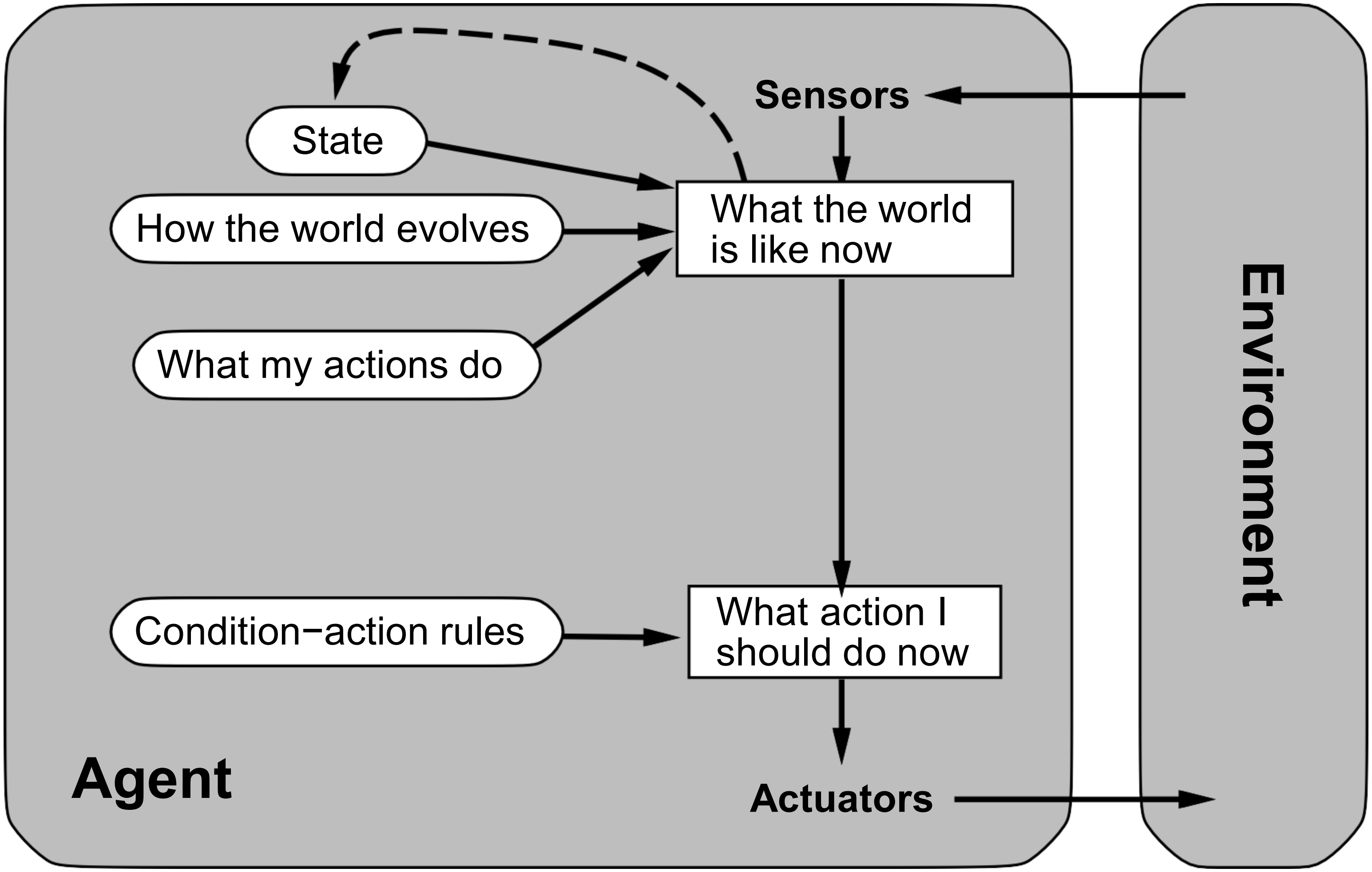
Example

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*

```
(setq joe (make-agent :name 'joe :body (make-agent-body)
                      :program (make-reflex-vacuum-agent-program))

(defun make-reflex-vacuum-agent-program ()
  #'(lambda (percept)
      (let ((location (first percept)) (status (second percept)))
        (cond ((eq status 'dirty) 'Suck)
              ((eq location 'A) 'Right)
              ((eq location 'B) 'Left)))))
```

Reflex agents with state



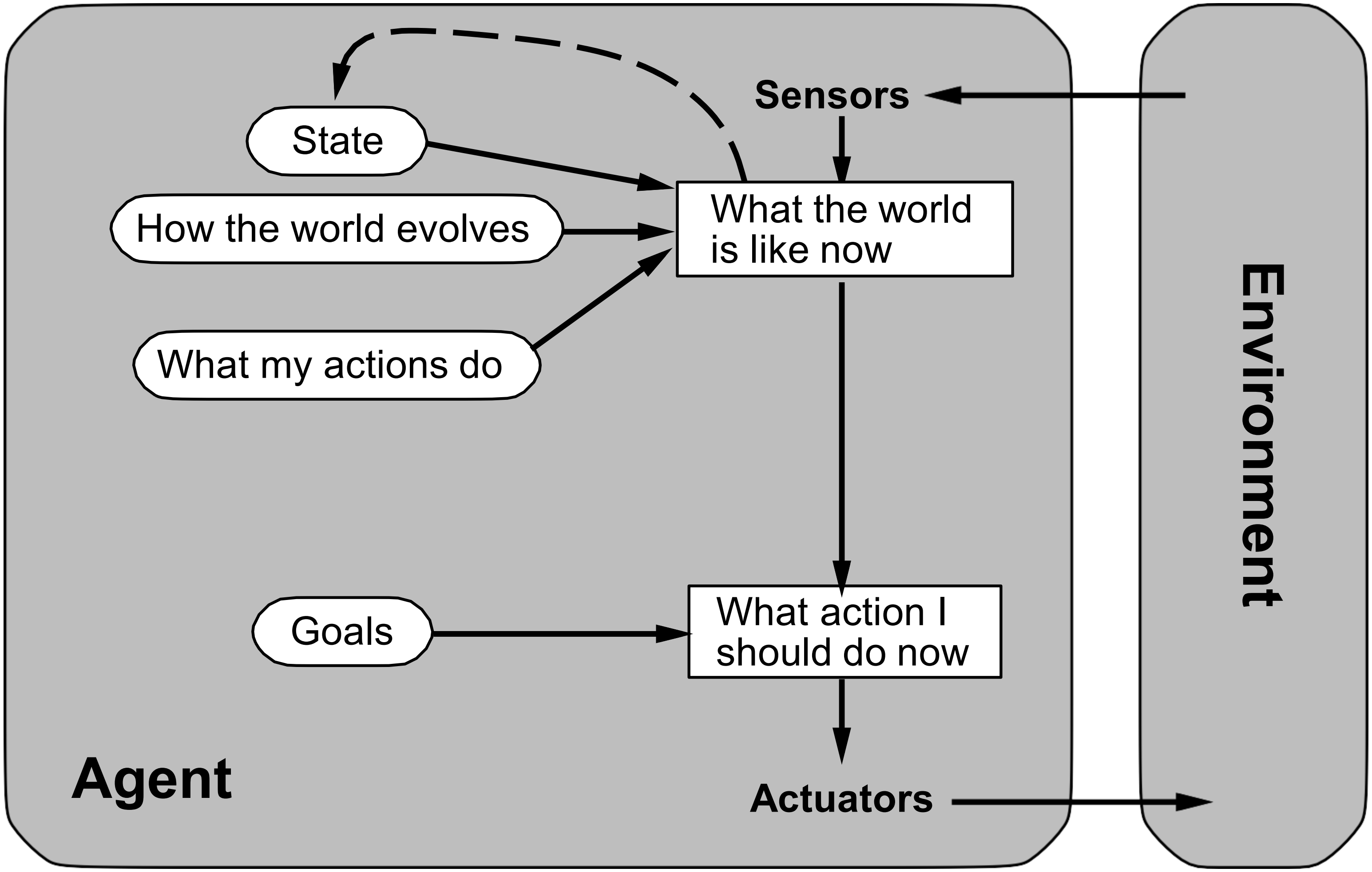
Example

function Reflex-Vacuum-Agent([*location*,*status*]) returns an action
static: *last_A*, *last_B*, numbers, initially ∞

if *status* = *Dirty* then . . .

```
(defun make-reflex-vacuum-agent-with-state-program ()
  (let ((last-A infinity) (last-B infinity))
  #'(lambda (percept)
      (let ((location (first percept)) (status (second percept)))
        (incf last-A) (incf last-B)
        (cond
         ((eq status 'dirty)
          (if (eq location 'A) (setq last-A 0) (setq last-B 0))
          'Suck)
         ((eq location 'A) (if (> last-B 3) 'Right 'NoOp))
         ((eq location 'B) (if (> last-A 3) 'Left 'NoOp)))))))
```

Model-based agents

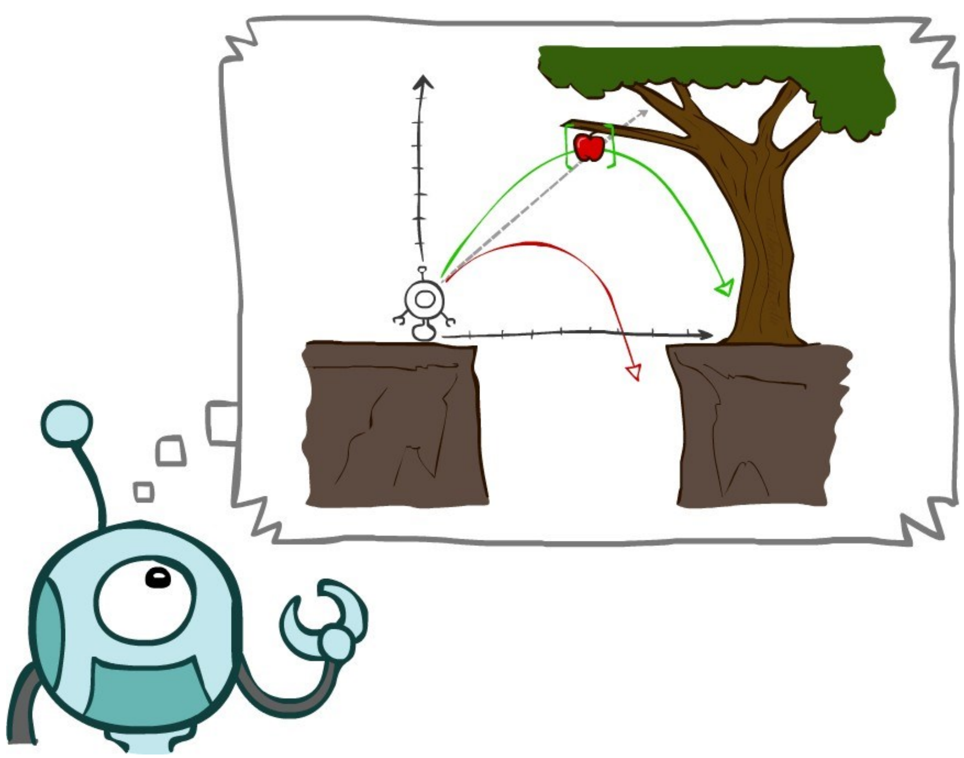


- Model-based agents handle partial observability of the environment by keeping track of the part of the world they cannot see now.
- The internal state of model-based agents is updated on the basis of a model which determines:
  - how the environment evolves independently of the agent;
  - how the agent actions affect the world.

Planning agents

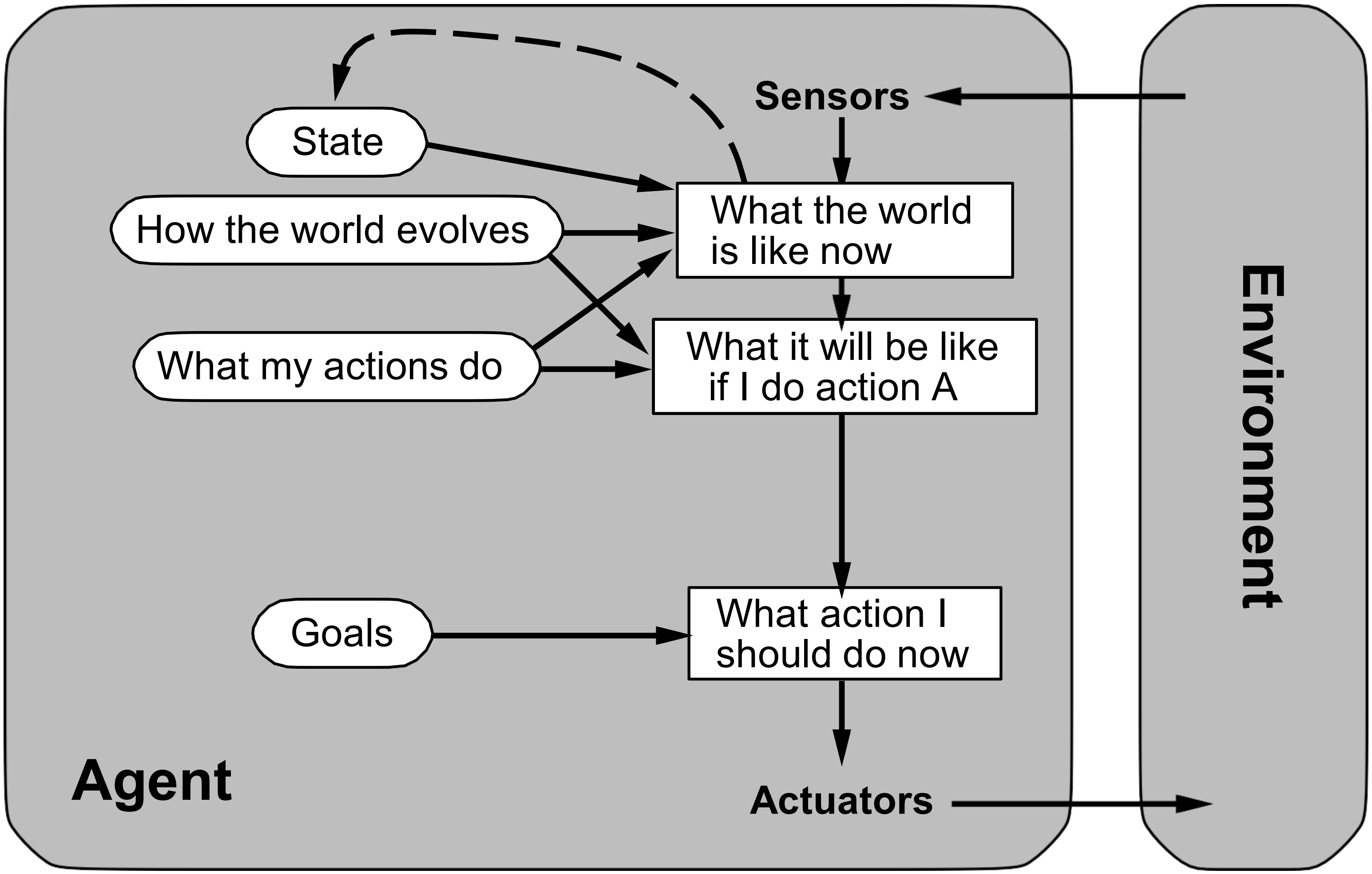
Planning agents:

- ask "what if?";
- make decisions based on (hypothesized) consequences of actions;
- must have a model of how the world evolves in response to actions;
- must formulate a goal.



Courtesy of UC Berkley

Goal-based agents



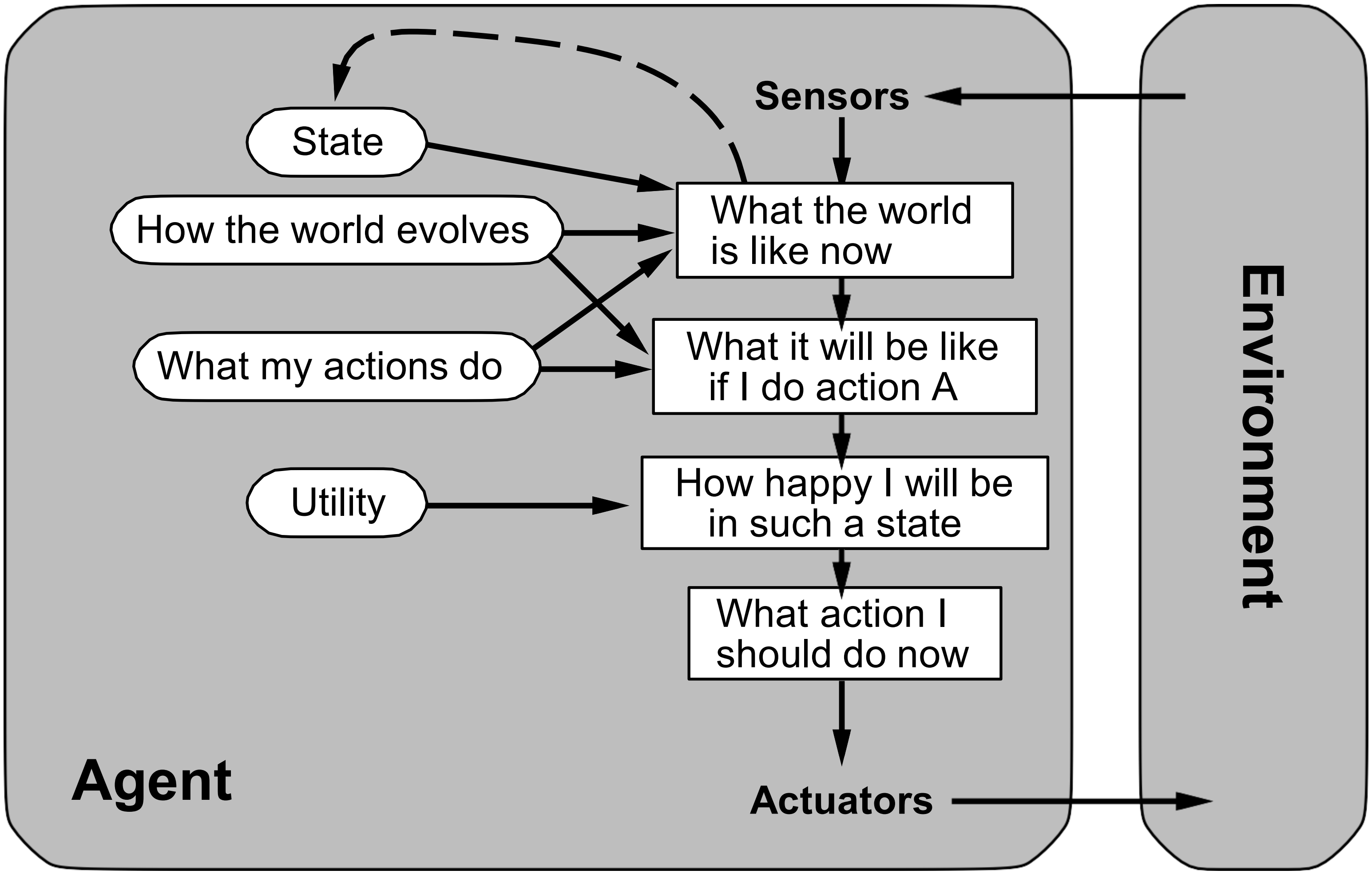
Decision process:

1. generate possible sequences of actions
2. predict the resulting states
3. assess goals in each.

A goal-based agent chooses an action that will achieve the goal.

- More general than rules. Goals are rarely explicit in condition-action rules.
- Finding action sequences that achieve goals is difficult. Search and planning are two strategies.

Utility-based agents



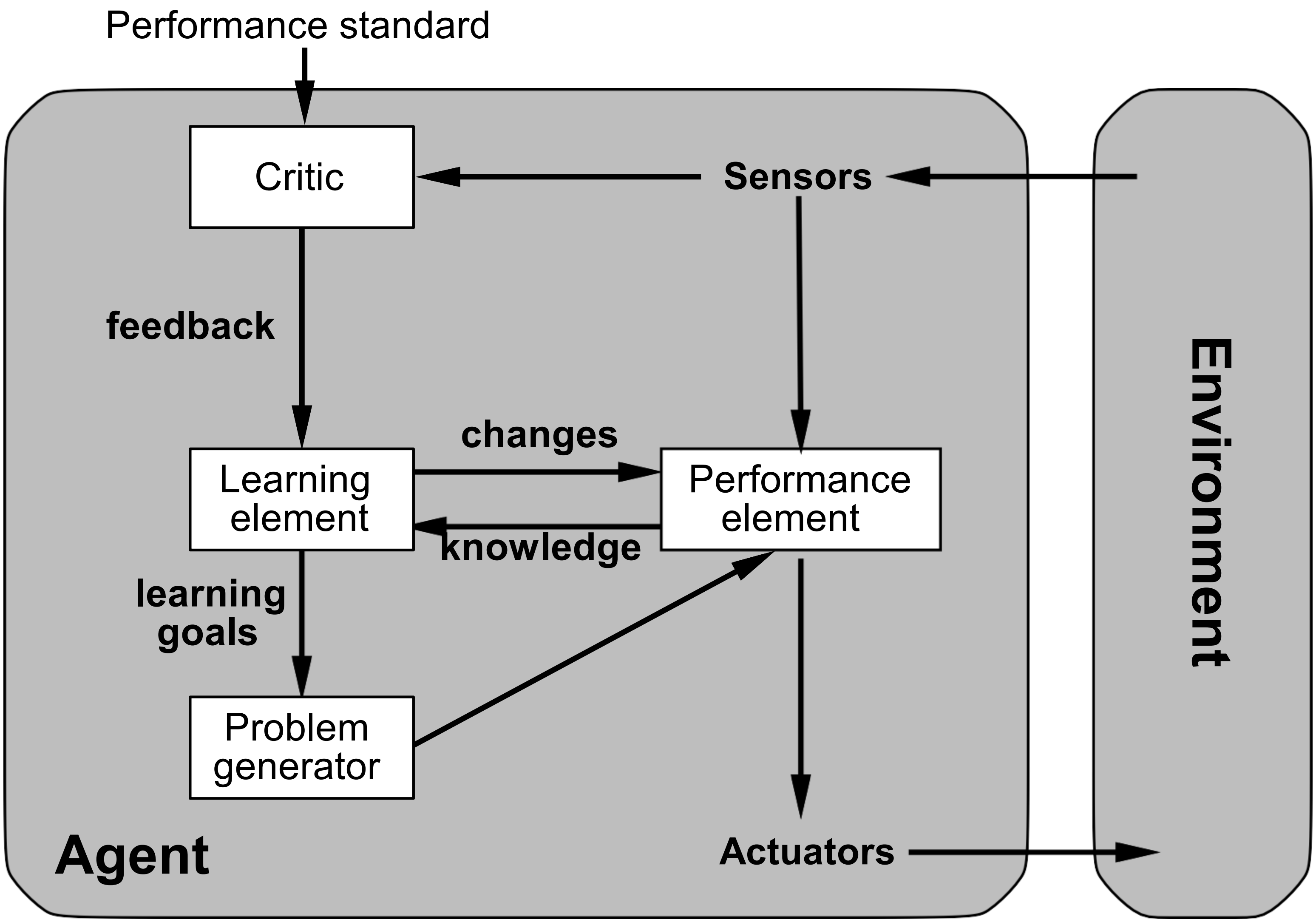
- Goals are often not enough to generate high-quality behavior.
    - Example (autonomous car): There are many ways to arrive to destination, but some are quicker or more reliable.
    - Goals only provide binary assessment of performance.
-

A utility function scores any given sequence of environment states.

- The utility function is an internalization of the performance measure.

- A rational utility-based agent chooses an action that maximizes the expected utility of its outcomes.

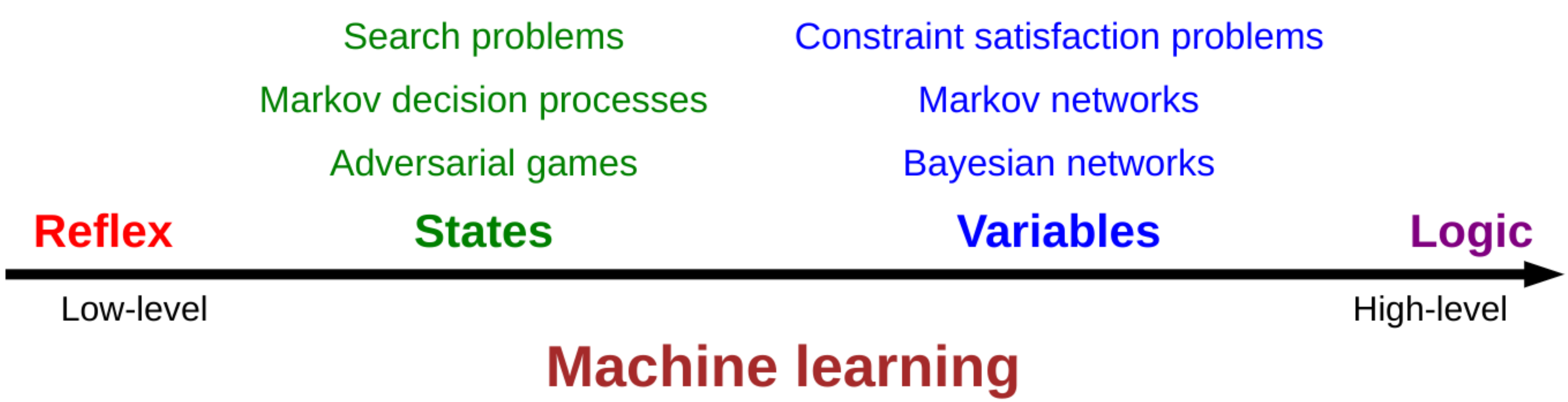
Learning agents



Learning agents are capable of self-improvement. They can become more competent than their initial knowledge alone might allow.

They can make changes to any of the knowledge components by:

- learning how the world evolves;
- learning what are the consequences of actions;
- learning the utility of actions through rewards.



Summary

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances

The performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance

Agent programs implement (some) agent functions

PEAS descriptions define task environments

Environments are categorized along several dimensions:

observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist:

reflex, reflex with state, goal-based, utility-based