Introduction to Computational intelligence

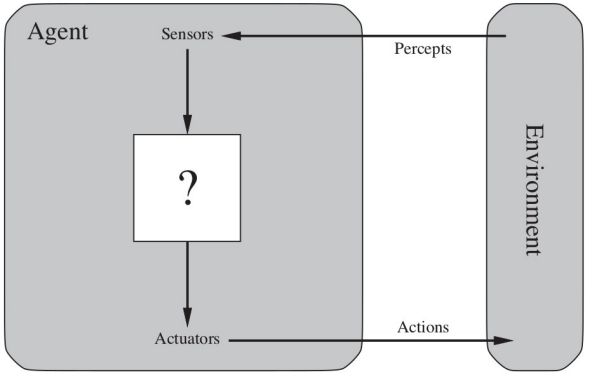
Intelligent agents

Igor Farkaš
Centre for Cognitive Science
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava

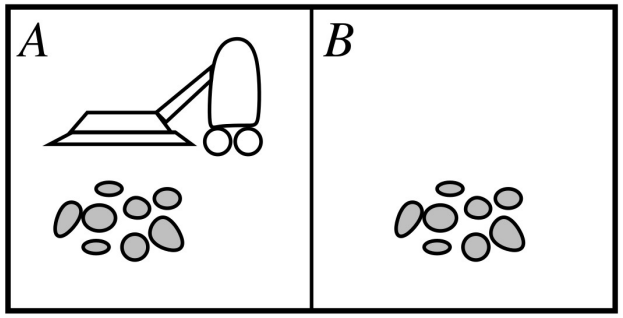
Based on Russel & Norvig: Artificial Intelligence: a Modern Approach, 3rd ed., Prentice Hall, 2010.

Definition of an agent

- Agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Percept refers to the agent's inputs at any given instant.



Example: Vacuum-cleaner world



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

Actions: $Left$, $Right$, $Suck$, $NoOp$

Vacuum-cleaner agent operation

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*

Rational behavior

- A rational agent is the one that does the “right thing”.
- Assessing the actions is captured by a performance measure (PM) that evaluates any given sequence of environment states (not agent states, to avoid self “delusion”).
- PM is introduced by a designer, more possibilities available
- *It is better to design PMs according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.*
- Definition of a rational agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its PM, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rational behavior (ctd)

- Rational ≠ omniscient
  - omniscient agent has perfect perception and knows the actual outcome of its actions
- Rational ≠ clairvoyant
  - action outcomes may not be as expected
- Rational ≠ successful
- Rational ⇒ exploration, learning, autonomy
- A rational agent should be autonomous – it should learn what it can do to compensate for partial or incorrect prior knowledge.

Agent examples and PEAS descriptions

Agent type	Performance measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	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, ...
Interactive English tutor	Student’s score on test	Set of students, Testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Properties of environments

- Observability: full vs. partial
  - Full, if complete state of environment accessible to the agent
- Single agent vs. multiagent
  - Multiagent: cooperative or competitive
- Deterministic vs. stochastic
  - Det. if next state is determined by current state & agent’s action
- Episodic vs. sequential
  - Episodic if agent’s decisions can be divided into independent epis.
- Static vs. dynamic
  - static if env. cannot change while agent is thinking
- Discrete vs. continuous (refers to env. states, agent’s actions)
- Known vs. unknown (refers to world model)

Examples of task environments

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Structure of agents

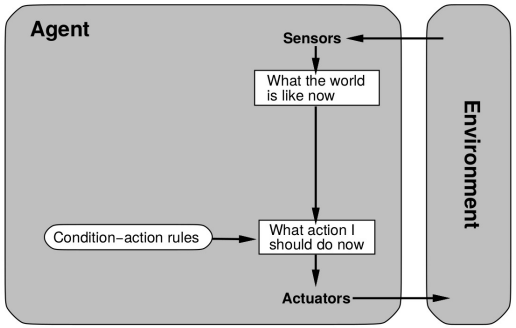
- Agent = architecture + program
- Table-driven agent:

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified

  append *percept* to the end of *percepts*
  action ← LOOKUP(*percepts*, *table*)
  return *action*

- implements the desired agent function but is very limited

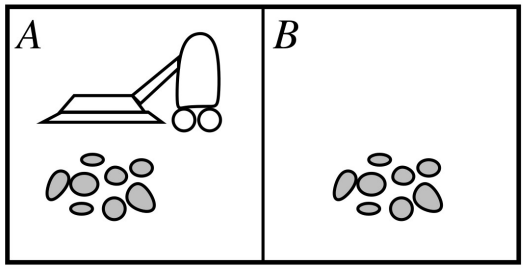
Simple reflex agents



function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

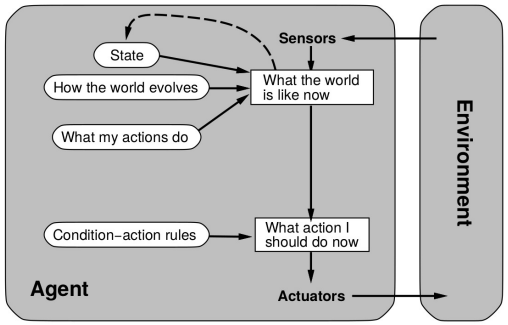
Simple reflex agent: example



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*

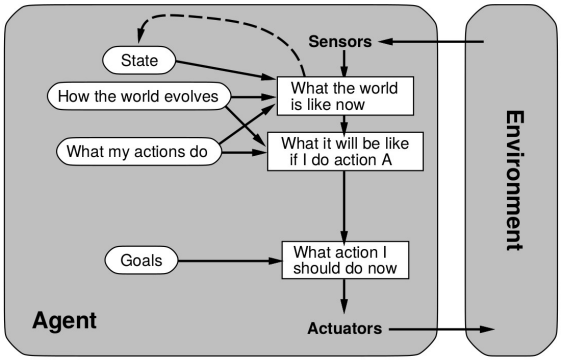
Model-based reflex agents



function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
persistent: *state*, the agent's current conception of the world state
model, a description of how the next state depends on current state and action
rules, a set of condition–action rules
action, the most recent action, initially none

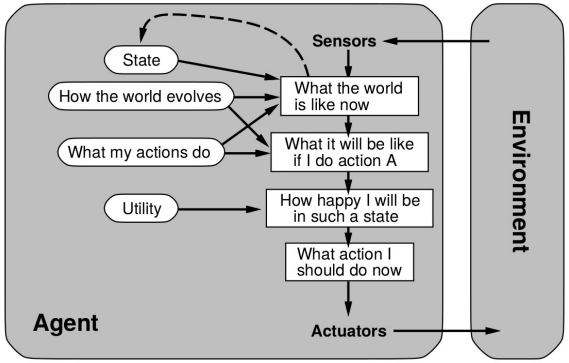
state ← UPDATE-STATE(*state*, *action*, *percept*, *model*)
rule ← RULE-MATCH(*state*, *rules*)
action ← *rule*.ACTION
return *action*

Goal-based agents



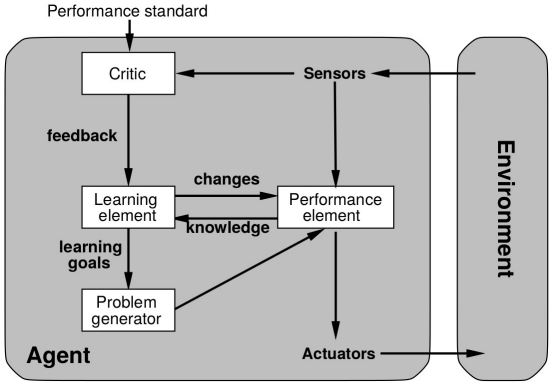
- Agent not only considers the past, but also looks into the future
- May require search or planning ability

Utility-based agents



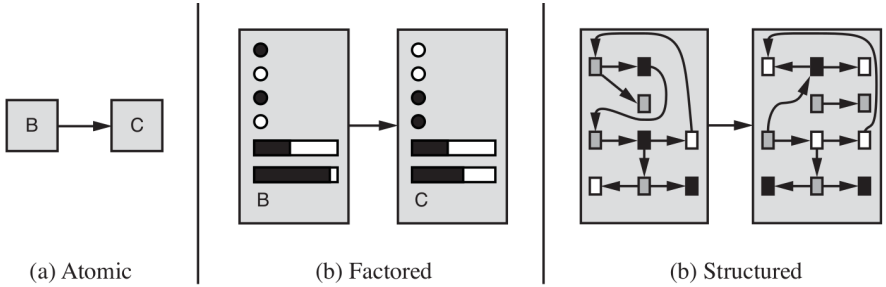
- Agent uses a utility function
- which is essentially an internalization of PM
- Agent tries to maximize expected utility

Learning agents



Learning in intelligent agents is a process of modification of each component of the agent to bring the components into closer agreement with the available feedback information, thereby improving the overall performance of the agent.

How the components of agent programs work



Three ways to represent states and transitions between them:

- Atomic: a state is a black box with no internal structure;
- Factored: a state consists of a vector of attribute values (Boolean, real-valued, or one of a fixed set of symbols);
- Structured: a state includes objects, which may have attributes of its own as well as relationships to other objects.

Summary

- Types of environment (7 features)
- Agent architectures
- Performance measure evaluates the behavior of the agent in an environment.
- Rational agent acts so as to maximize the expected value of PM.
- All agents can improve their performance through learning.