



Intelligent Agents

AGENTS AND ENVIRONMENT

Structure of Agents

- Agent = Architecture + Program

Percept sequence	Action
<i>[A, Clean]</i>	<i>Right</i>
<i>[A, Dirty]</i>	<i>Suck</i>
<i>[B, Clean]</i>	<i>Left</i>
<i>[B, Dirty]</i>	<i>Suck</i>
<i>[A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>
<i>[A, Clean], [A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>

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*

Agent Program

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

Simple reflex agents

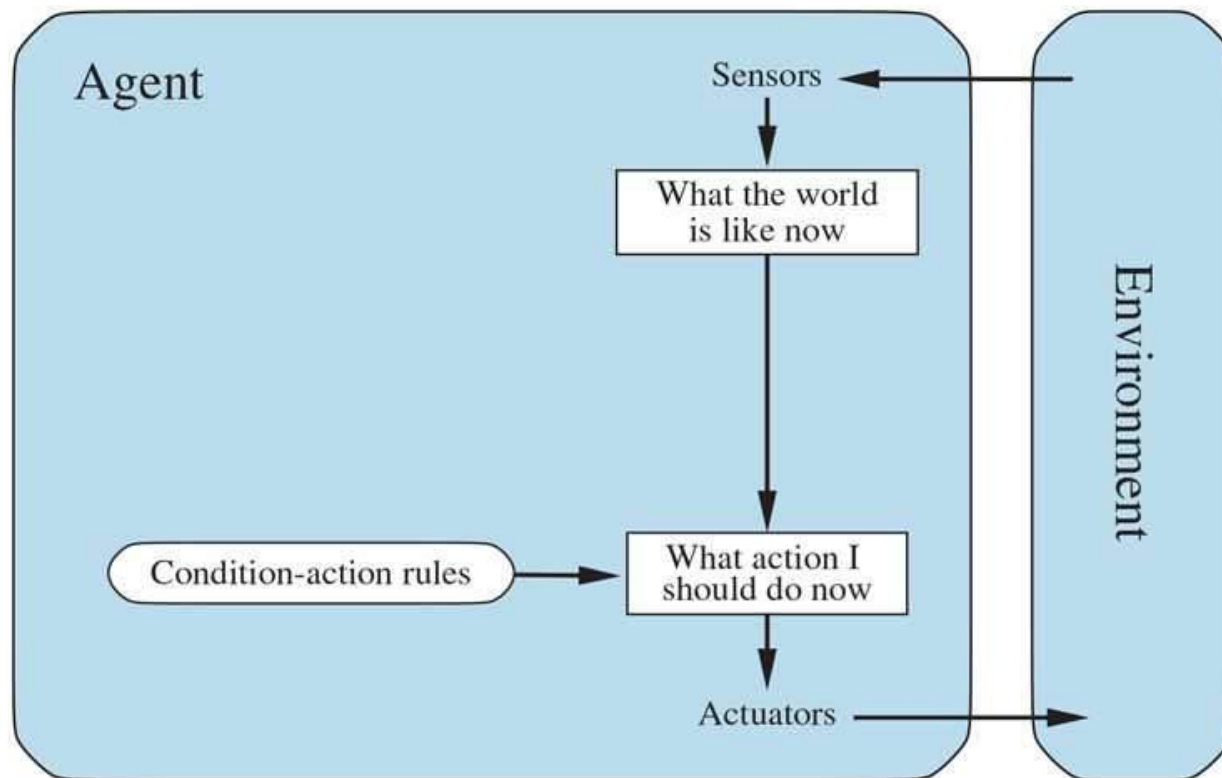
- Action depends only on current percept.

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*

- Condition-Action rule
 - If car-in-front-is-braking then initiate braking.

Simple reflex agents



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

state \leftarrow INTERPRET-INPUT(*percept*)

rule \leftarrow RULE-MATCH(*state*, *rules*)

action \leftarrow *rule*.ACTION

return *action*

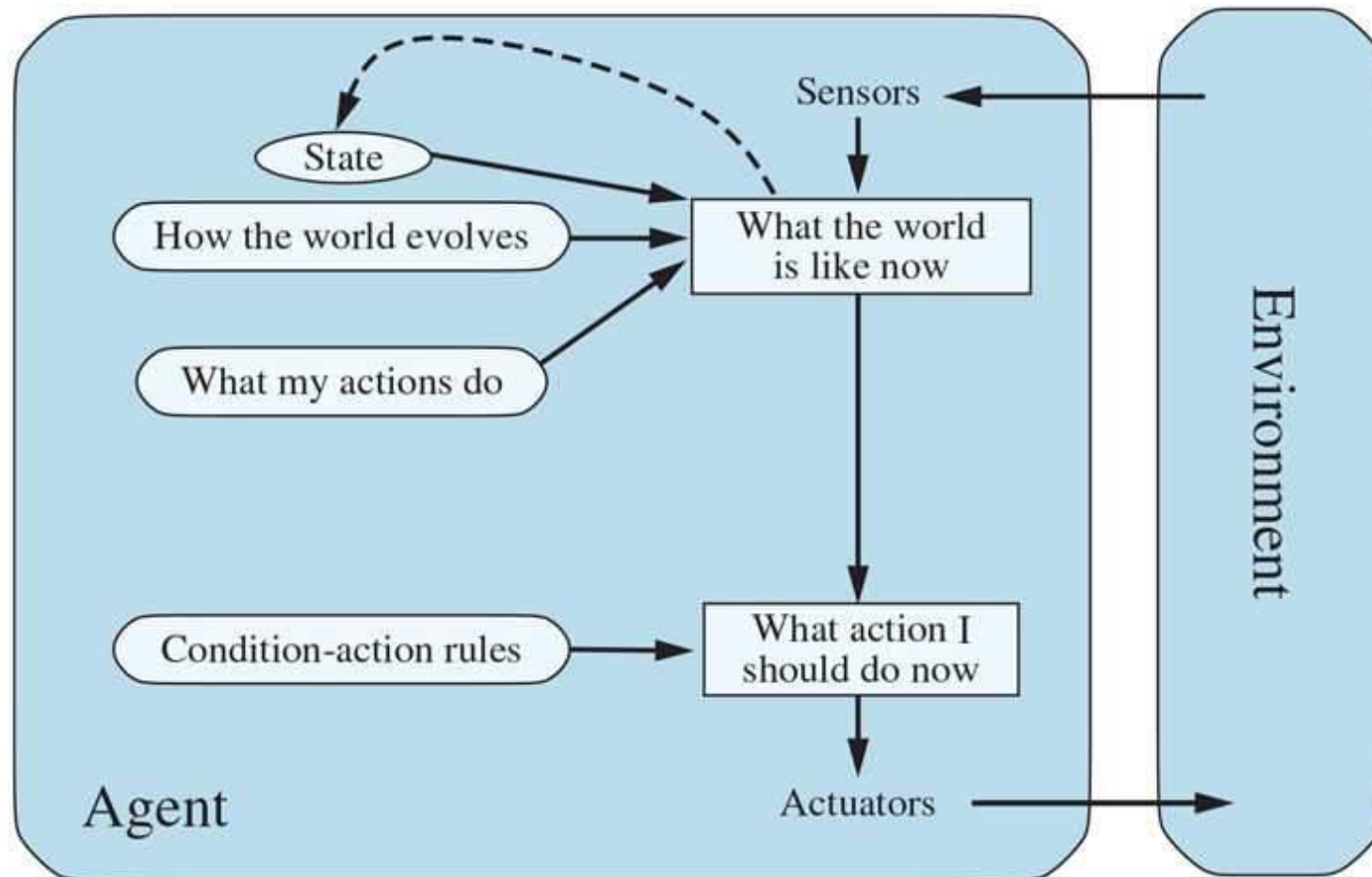
Simple reflex agents

- Simple but limited intelligence
- Correct decision provided environment is fully observable
- Randomized actions

Model-based reflex agents

- Maintain **Internal** state that depends on the percept history
- Transition model: Knowledge about “how the world works”
- Sensor model: Knowledge about “how the state of the world is reflected in the agent’s percepts”

Model-based reflex agents



Model-based reflex agents

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action

persistent: *state*, the agent's current conception of the world state
transition_model, a description of how the next state depends on
the current state and action
sensor_model, a description of how the current world state is reflected
in the agent's percepts
rules, a set of condition-action rules
action, the most recent action, initially none

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