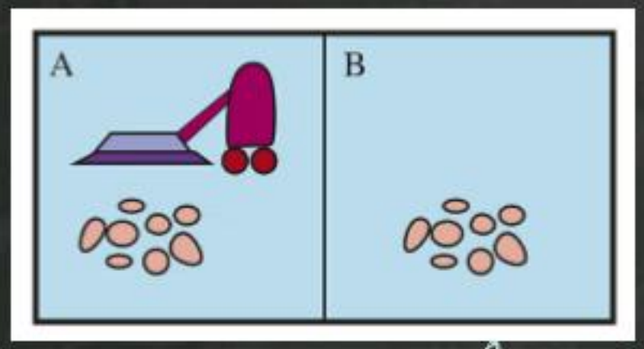
# هوش مصنوعی - جلسه هشتم

## معماری کنشگر



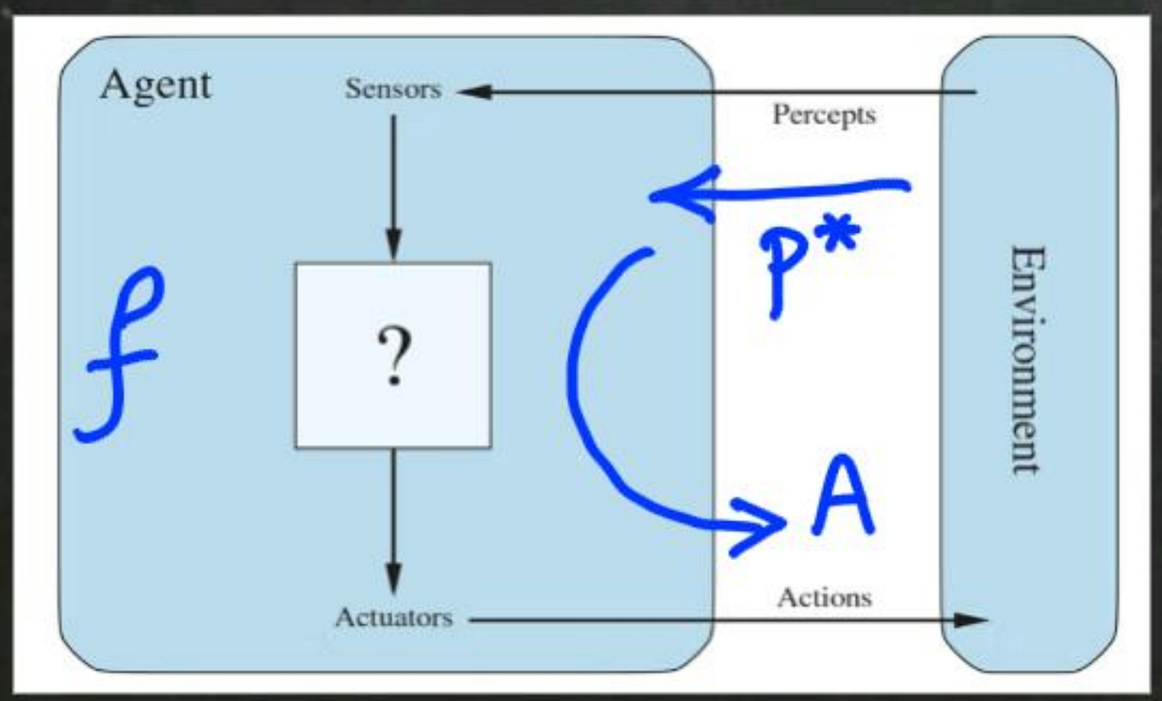

Agent Architecture

$f: P^* \rightarrow A$

A ← Agent(Percept P)

Agent function

Agent Program

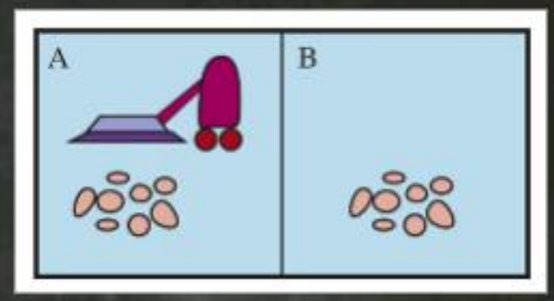



معماری کنشگر، ساختار روشمند روابط بین اجزاء برنامه کنشگر است.

# Agent function (Explicit rep. is not possible)

| 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* |
| ⋮ | ⋮ |

مثال: ربات جاروبرقی




# Agent program

**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*

# انواع معماری کنشگر

| | واکنشی | معطوف به هدف/سود |
|---|---|---|
| بدون حافظه | Simple Reflex | ؟ |
| با حافظه | Model-based Reflex | Goal-based Agent / Utility-based Agent |

یادگیرنده ← Learning Agent

مثال: ربات جاروبرقی یک کنش‌گر واکنشی ساده است.

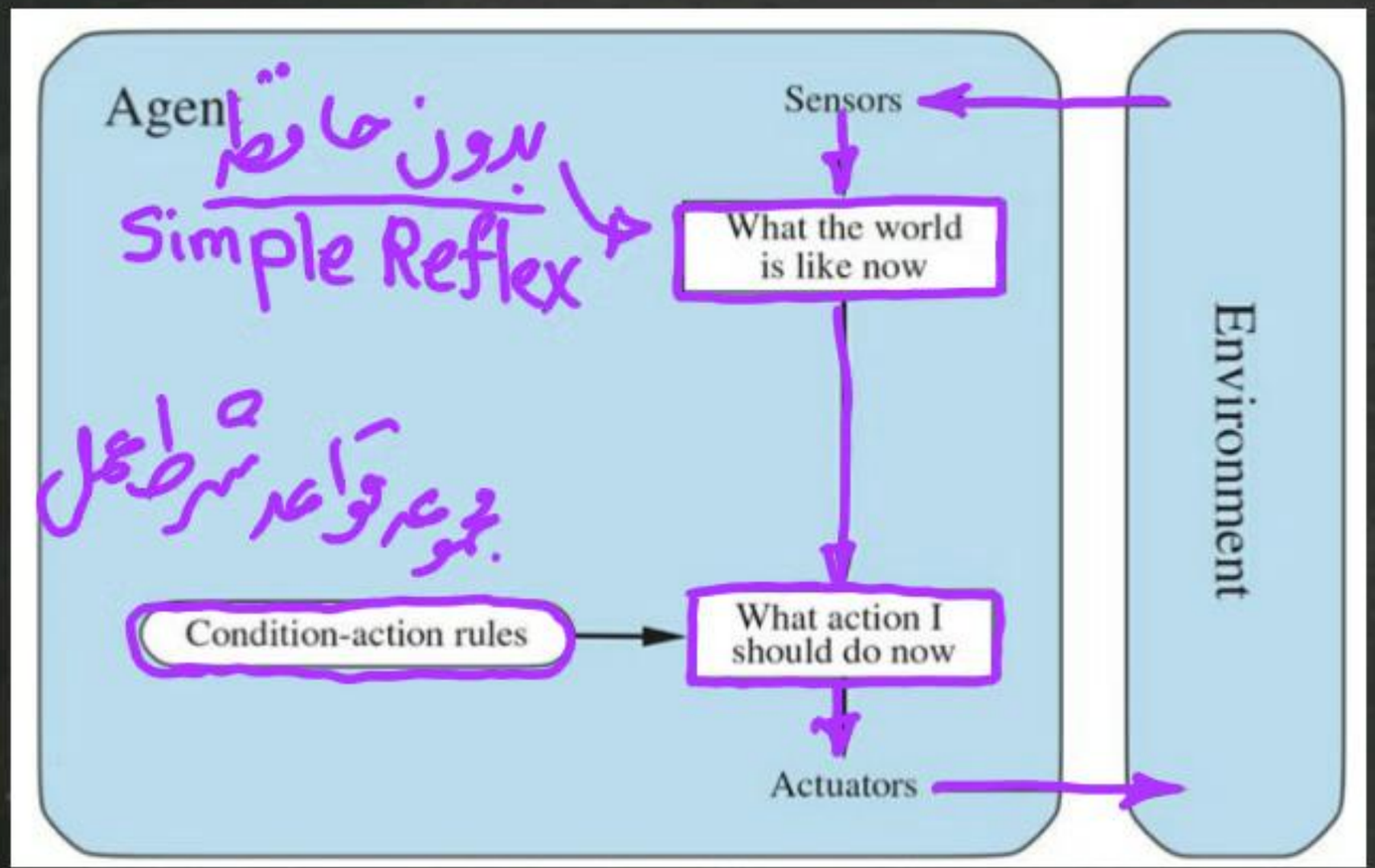
```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

---

الگوی کلی پیاده‌سازی تابع کنش‌گر در قالب برنامه کنش‌گر

```
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
```




بدون حافظه
Simple Reflex

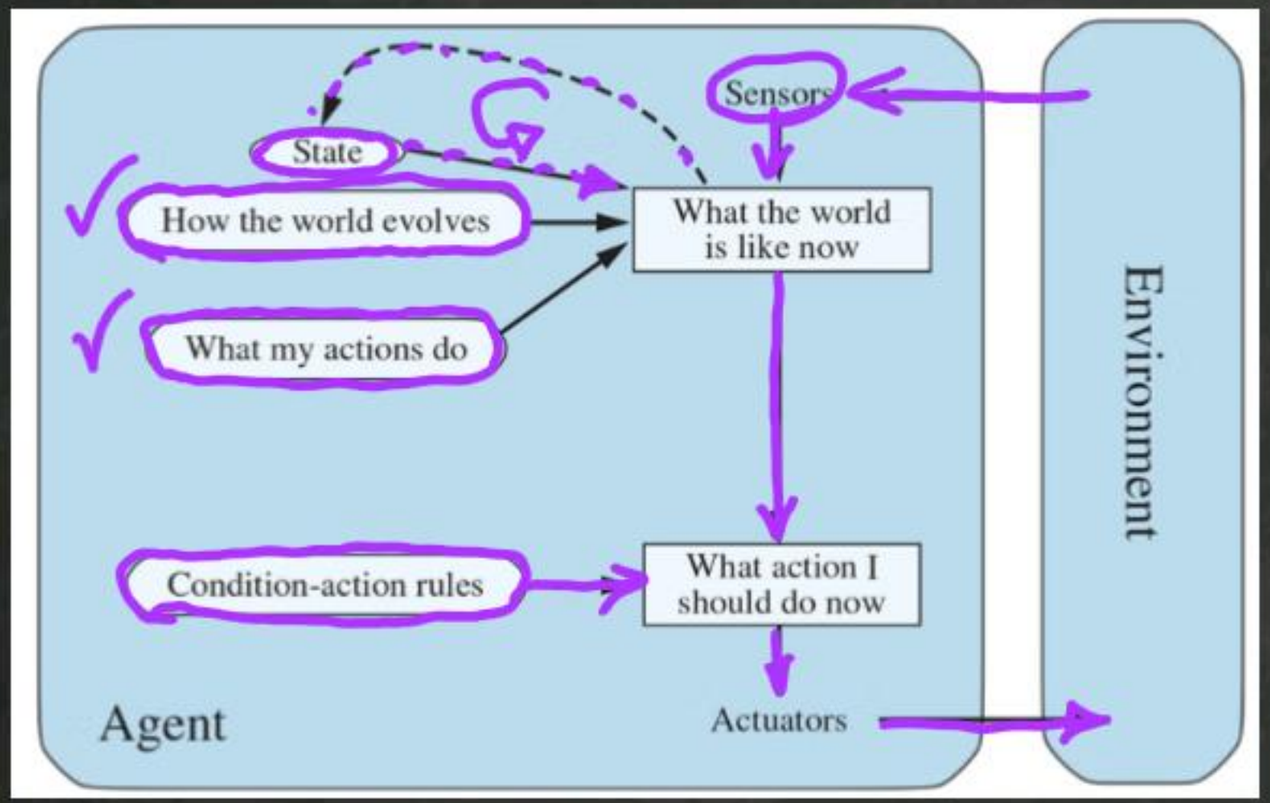
مجموعه قواعد شرط عمل

**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*

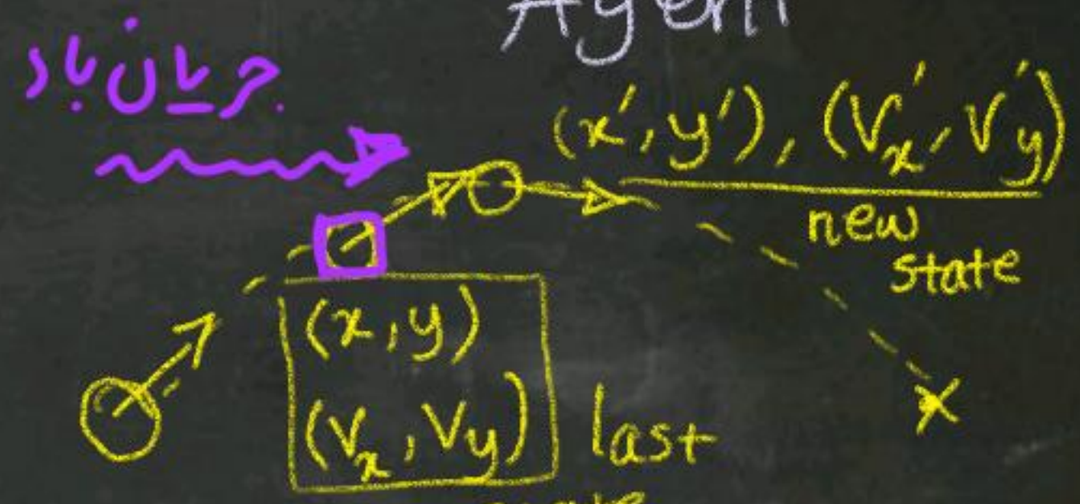
## کنشگر واکنشی ساده
## Simple Reflex

```
class Percept;
class Action;
class State;
class SimpleReflex{
  dict rules;
  ----
  ----
  Action getAction
  (Percept P){
  ---
  return action;
};}
```




# کنشگر مدل گرا

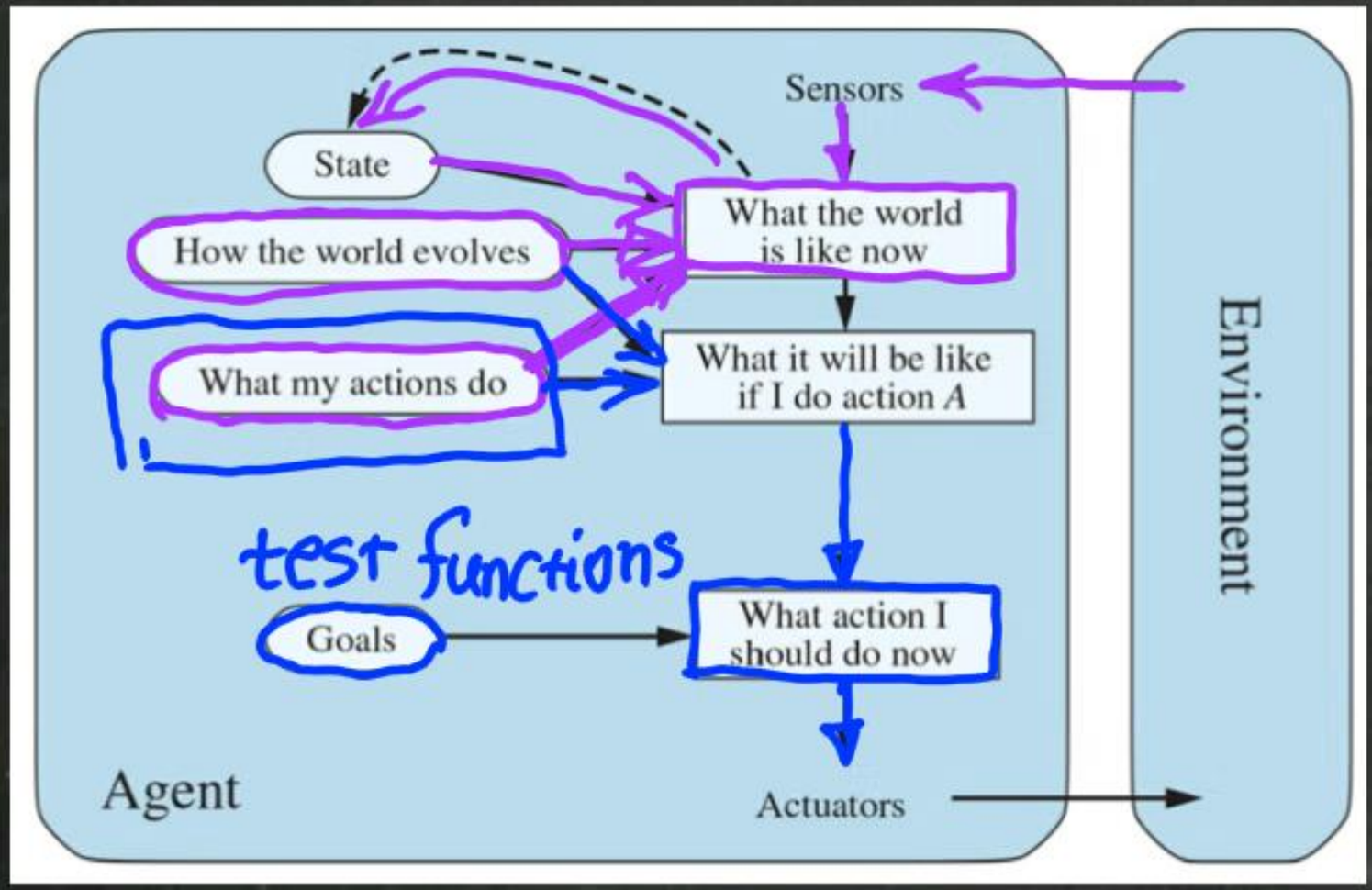
# Model-based Agent




**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*




# گنشگر هدف گرا

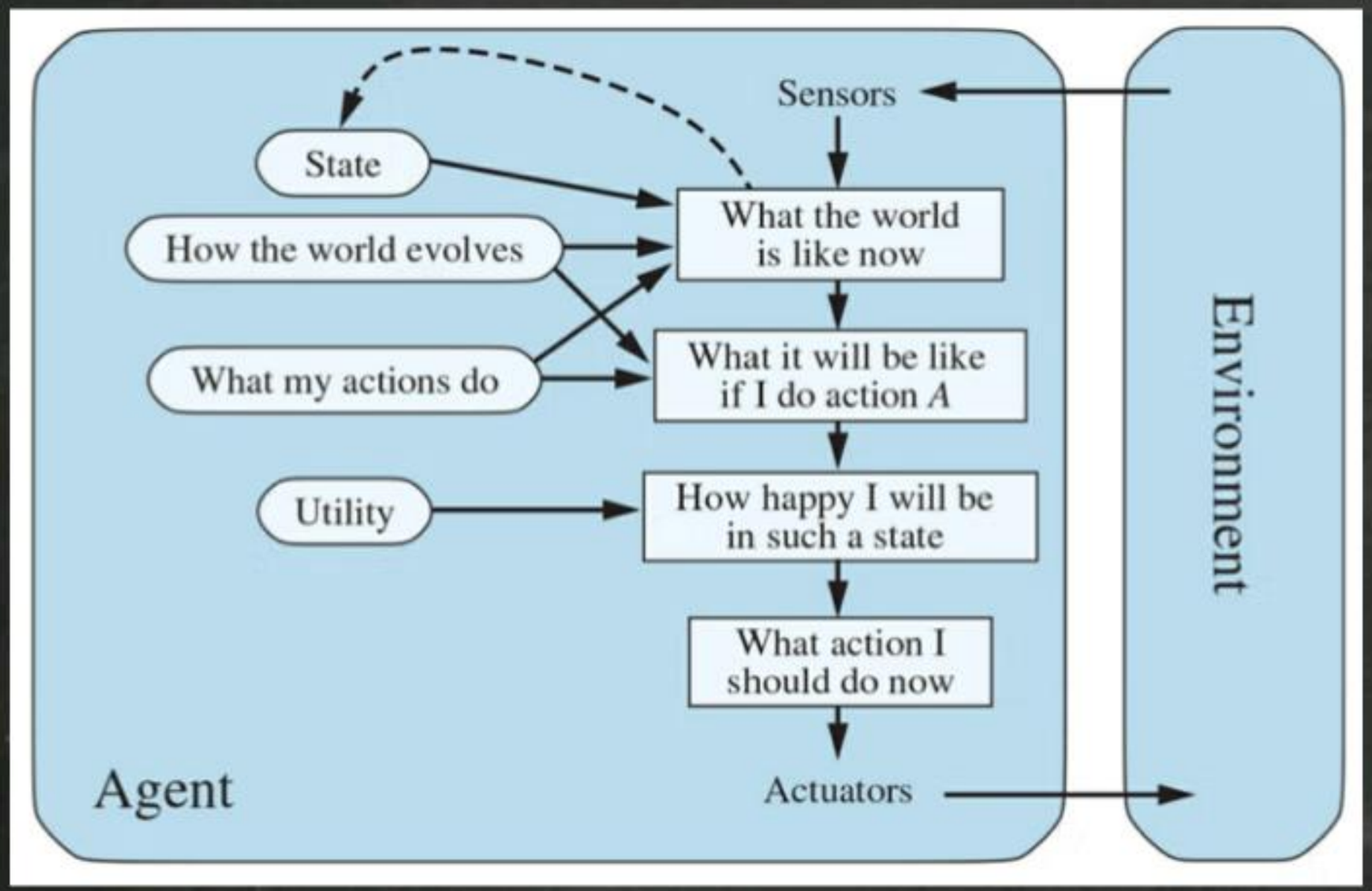
# Goal-based Agent

function isGoal(state)

return True/False

# کنشگر سودگرا

## Utility-based Agent



# کنشگر یادگیرنده