

Lab-6: Intelligent Agents (Implementation of Model Based Reflex Agent)

6.1 Objectives:

1. To learn and implement Model Based reflex agent using Python

6.2 Model Reflex Agent

Model-based reflex agents are made to deal with partial accessibility; they do this by keeping track of the part of the world it can see now. It does this by keeping an internal state that depends on what it has seen before so it holds information on the unobserved aspects of the current state.

This time out mars Lander after picking up its first sample, it stores this in the internal state of the world around it so when it come across the second same sample it passes it by and saves space for other samples.

While reading this you are keeping track of where you have got to somewhere internally in your brain just in case you lose your place.

But in order to update this internal store we need 2 things:

1. Information on how the world evolves on its own.

e.g. If our mars Lander picked up the rock next to the one it was going to the world around it would carry on as normal

2. How the world is affected by the agents actions.

E.g. If our mars Lander took a sample under a precarious ledge it could displace a rock and it could be crushed.

We can predict how the world will react with facts like if you remove a supporting rock under a ledge the ledge will fall, such facts are called models, hence the name model-based agent.

- If the world is not fully observable, the agent must remember observations about the parts of the environment it cannot currently observe.
- This usually requires an internal representation of the world (or internal state).

A schematic diagram of a model based reflex agent is shown below:

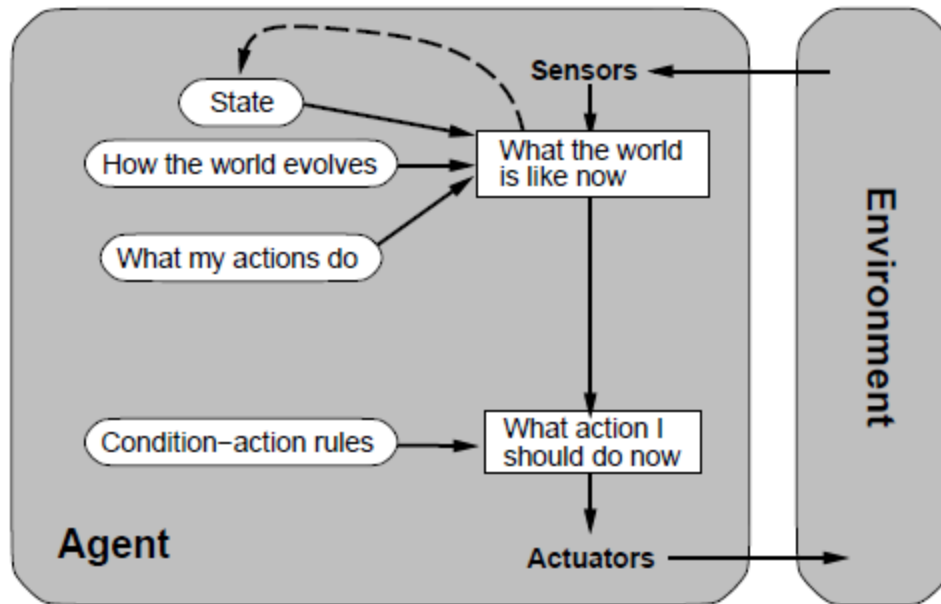


Figure 1: schematic of Model based reflex agent

They use a model of the world to choose their actions. They maintain an internal state.

Model – knowledge about “how the things happen in the world”.

Internal State – It is a representation of unobserved aspects of current state depending on percept history.

Updating the state requires the information about –

- How the world evolves.
- How the agent’s actions affect the world..

6.2.1 Pseudocode:

function REFLEX-AGENT-WITH-STATE(*percept*) **returns** an action

static: *rules*, a set of condition-action rules

state, a description of the current world state

action, the most recent action.

state ← UPDATE-STATE(*state*, *action*, *percept*)

rule ← RULE-MATCH(*state*, *rule*)

action ← RULE-ACTION[*rule*]

return *action*

Example: Vacuum cleaner

Consider the vacuum world shown in the figure below:

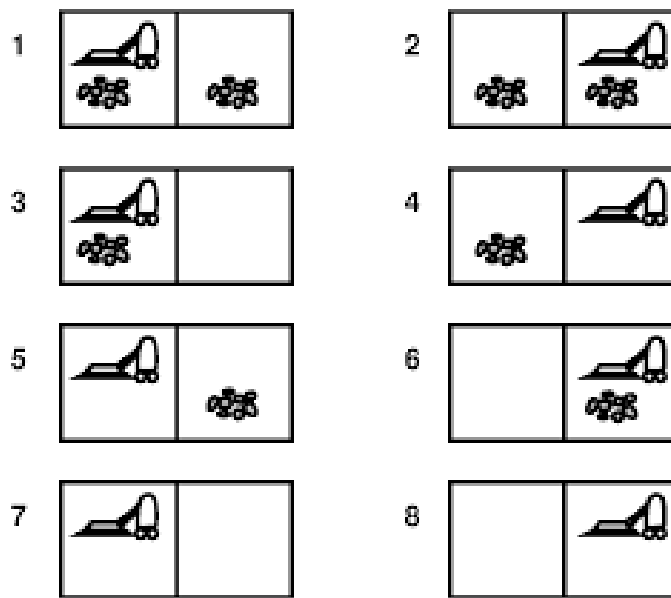


Figure 2: vacuum cleaner problem

This particular world has just two locations: squares A and B. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can choose to move left, move right, suck up the dirt, or do nothing. **Agent function** is the following: if the current square is dirty, then suck, otherwise move to the other square. Write a model based reflex agent for the vacuum cleaner. (Hint: Agent has initial states knowledge)

If the current square is dirty, then suck; otherwise, move to the other square.

Initial state is 1, where square A and Square B, both are dirty.

Pseudocode to the problem is as follows;

function Reflex-Vacuum-Agent([location,status]) **returns** an action

static: last A, last B, numbers, initially ∞

if status = Dirty **then and so on**

code:

```
class ModelBasedVacuumAgent():
    def __init__(self,init_a,init_b):
        self.model = {"Loc_a" : init_a, "Loc_b" : init_b}
    def DoAction(self,location, status):
        self.model[location] = status
        print(self.model)
        if self.model["Loc_a"] == self.model["Loc_b"] == 'clean':
            return 'NoOp'
        elif status == 'dirty':
```

```
        return 'suck'
    elif location == "Loc_a":
        return 'right'
    else:
        return 'left'
a=ModelBasedVacuumAgent('dirty','dirty')
print(a.DoAction("Loc_a","dirty"))
```

TASK:

1. Can you name few model-based reflex agents?
2. Write a program for model-based reflex agent of your own choice.