Artificial and Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

BITS Pilani
Pilani Campus

M1 : Introduction to AI
M2 : Problem Solving Agent using Search

Presented by
V Indumathi –Guest Faculty –BITS-W
indumathi.p@wilp.bits-pilani.ac.in

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP**:** Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1 Introduction to AI

M2 Problem Solving Agent using Search

M3 Game Playing

M4 Knowledge Representation using Logics

M5 Probabilistic Representation and Reasoning

M6 Reasoning over time

M7 Ethics in AI

Learning Objective

At the end of this class , students Should be able to:

1. Identify dimensions of TASK environment
2. Design problem solving agents
3. Create search tree for given problem
4. Apply uninformed search algorithms to the given problem

Dimensions of Task Environment

Sensor Based:

- Observability : Full Vs Partial

Action Based:

- Dependency : Episodic Vs Sequential

State Based:

- No.ofState : Discrete Vs Continuous

Agent Based:

> Cardinality : Single Vs MultiAgent

Action & State Based:

- State Determinism : Deterministic Vs Stochastic | Strategic
- Change in Time : Static Vs Dynamic

Task Environment

A rational agent is built to solve a specific task. Each such task would then have a
different environment which we refer to as Task Environment

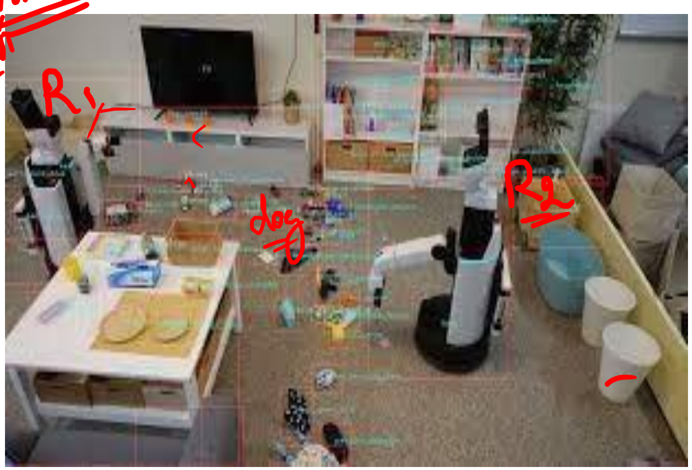
Based on the applicability of each technique for agent implementation its task environment design is determined by multiple dimension

1 **Sensor Based:**

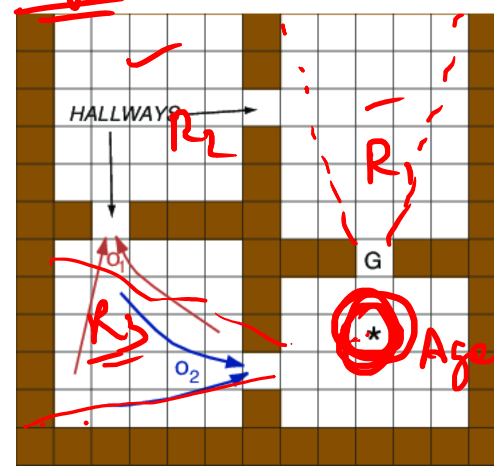
- Observability : Full Vs Partial

Controlled Env

eg2



eg 1



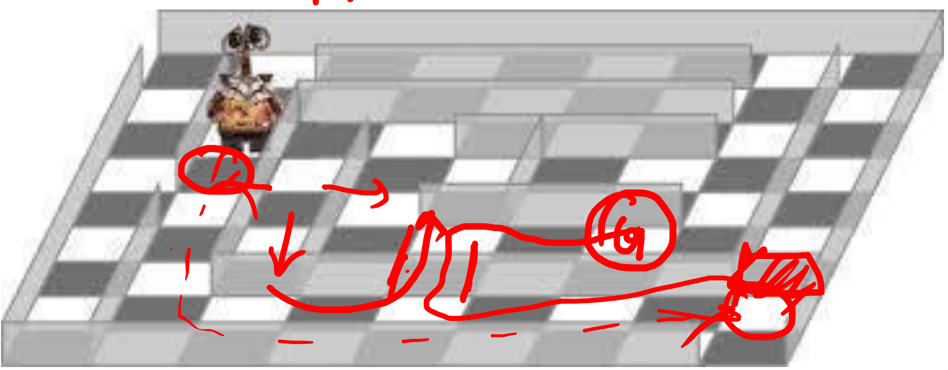
Task Environment

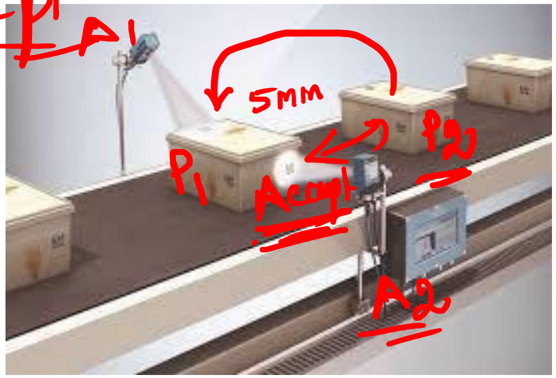
Action Based:

- Dependency : Episodic Vs Sequential

Epi eg A1 Obj

Maze





A/Rejected

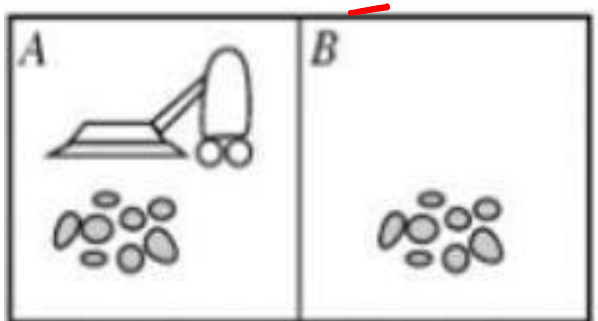
1st Seq of Moves

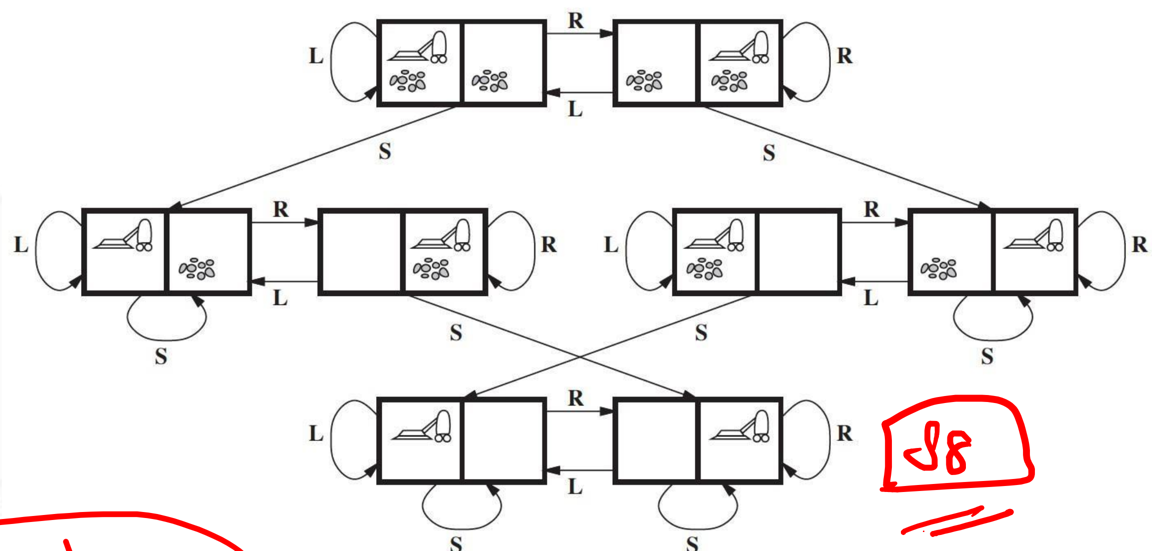
Task Environment

State Based:

- No.of.State : **Discrete** Vs Continuous

S0





S8

L R, SA|SB

Task Environment

State Based:

- No.of.State : Discrete Vs Continuous

VS.

Task Environment

i/p o/p

t_1 $T(S_1, a_1) \rightarrow S_2$

Action & State Based:

t_{100} " $\rightarrow S_2$

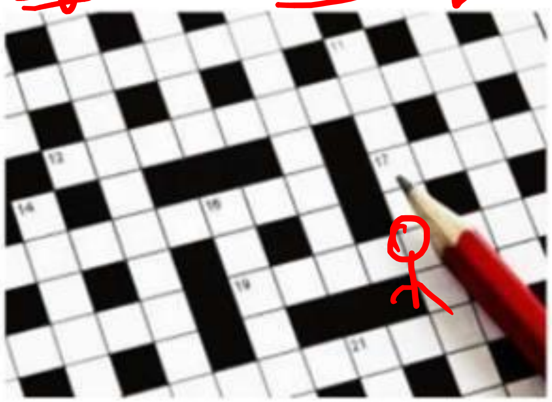
$S_1\ a_1 \rightarrow S_2\ 0.8$

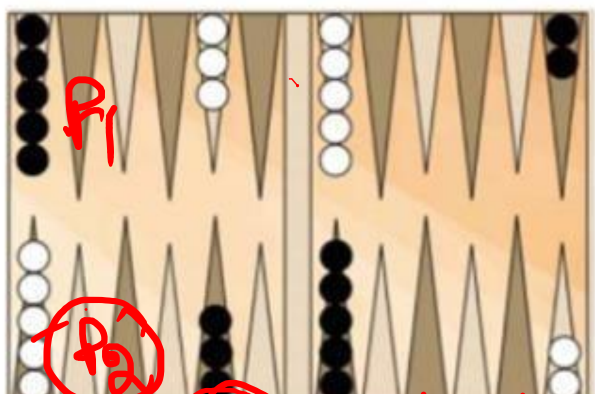
$S_1, a_1 \rightarrow S_3\ 0.1$

$S_1, a_1 \rightarrow S_4\ 0.1$

- State Determinism : Deterministic Vs Stochastic | Strategic
(If the environment is deterministic except for the actions of other agents, then the environment is strategic)

eg i/p & o/p





S_1 : with Dice → stochastic

S_2 : W/O Dice

Task Environment

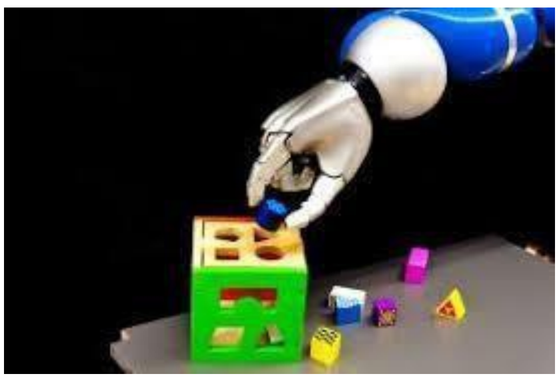
Agent Based:

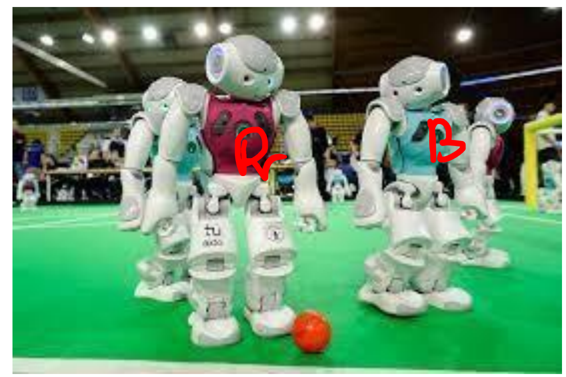
> Cardinality : Single Vs MultiAgent

Cooperative

Competitive Agent env-

Resume obj Same

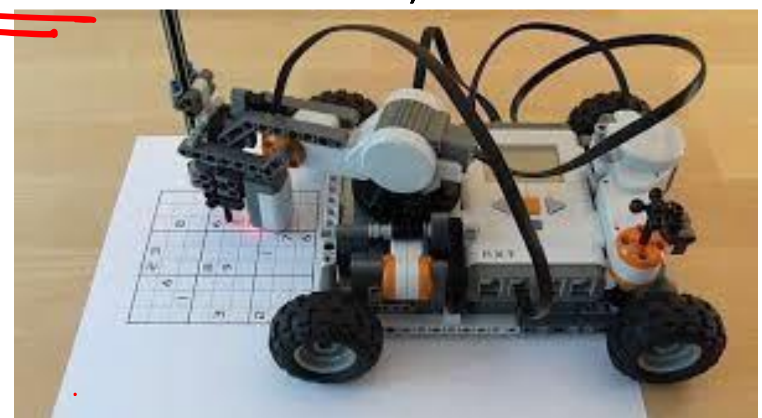




Task Environment

Action & State Based:

- Change in Time : Static Vs Dynamic
- (The environment is semi dynamic if the environment itself does not change with the passage of time but the agent's performance score does)



NO

eg:

8hr attentiveness ⇒ 6 to 7 90%

⇒ 7 to 8 95% 80 60%

Task Environment

Task Environment	Fully vs Partially Observable	Single vs Multi-Agent	Deterministic vs Stochastic	Episodic vs Sequential	Static vs Dynamic	Discrete vs Continuous
Medical diagnosis system	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Satellite Image Analysis System	Fully	Single	Deterministic	Episodic	Static	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

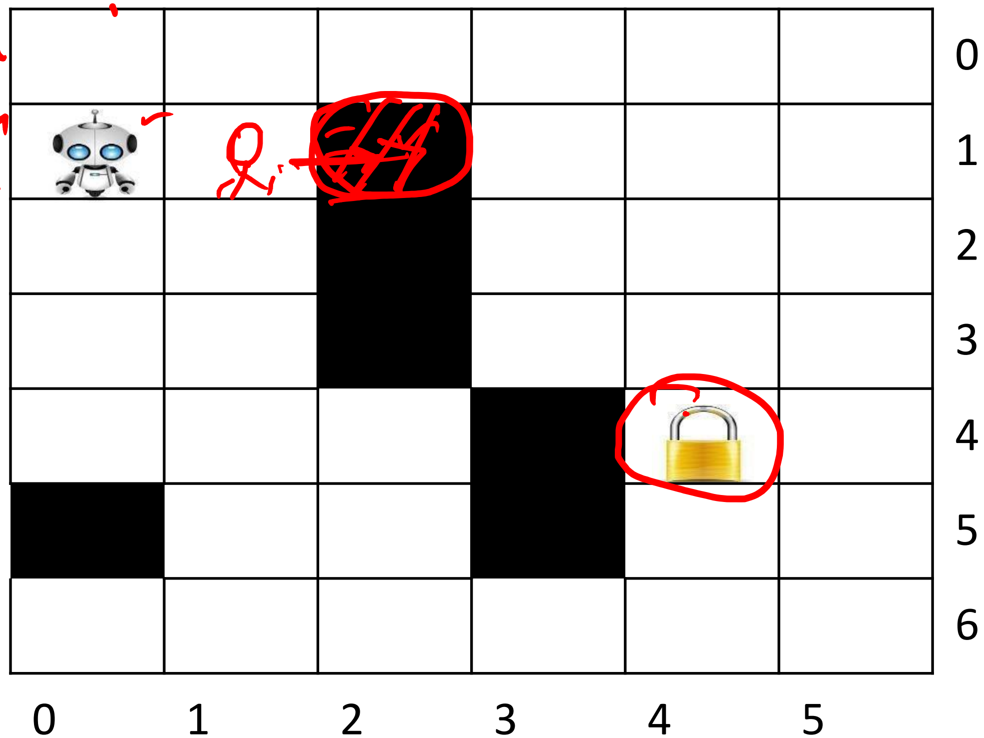
Path finding Robot - Lab Example

P, E, A, S

safe/unsafe

shortest path

L(R,U,D)



(r, c)
(1, 0)

(4, 4)

Agent
Observability
No.of.Agents
No.of.States
Determinism
Dynamicity
Output Dependency

Path finding Robot - Lab Example

1 0

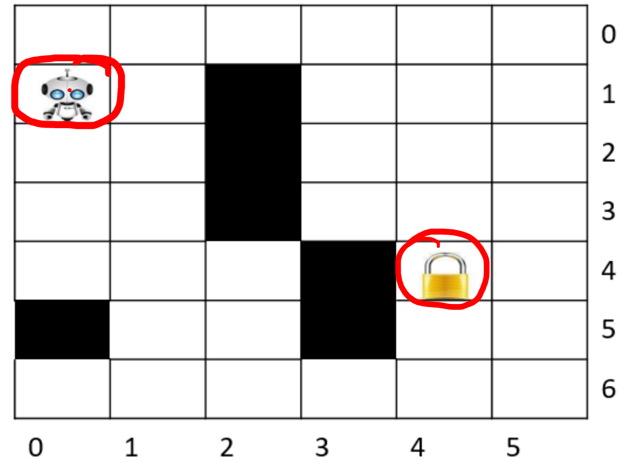
Not → 0

x,y | S | VS | G

[(1,0), 1, 0], MU → [(0,0), 1, 0]

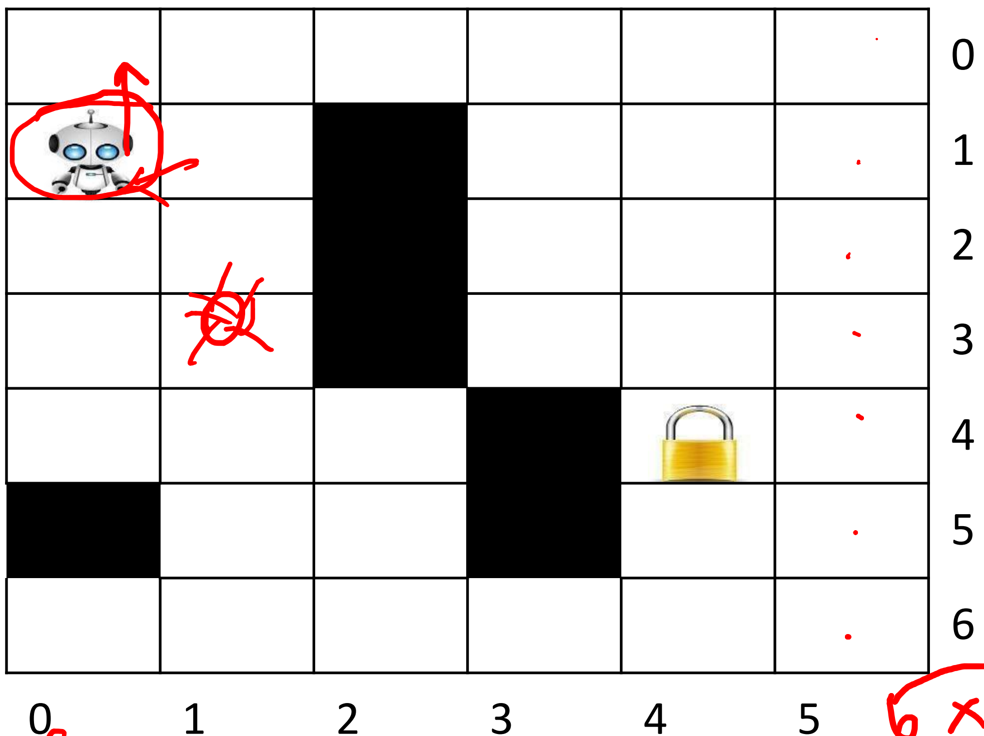
i/p

MR



Path finding Robot - Lab Example

Controlled Envi



6 x 7

7x6

b, w

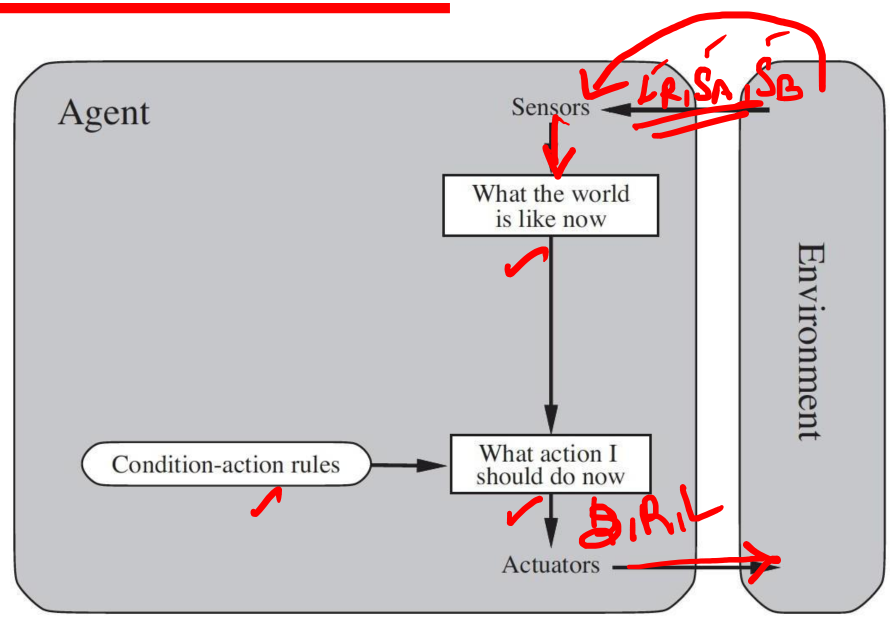
risk

Agent
Observability
Fully
No.of.Agents
Single
No.of.States
Discrete
Determinism
Deterministic
Dynamicity
static /
Output Dependency
seq

Agents Architectures

Simple Reflex Agent

```
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
```

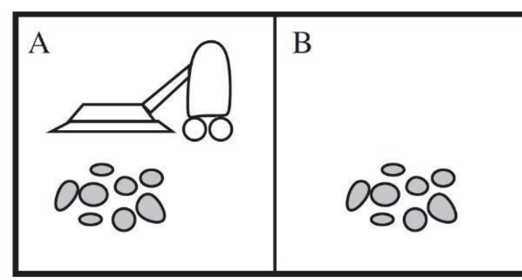



```
function REFLEX-VACUUM-AGENT( [location,status]) returns an action
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

Simple Reflex Agents

Fully observable

D → C| suck

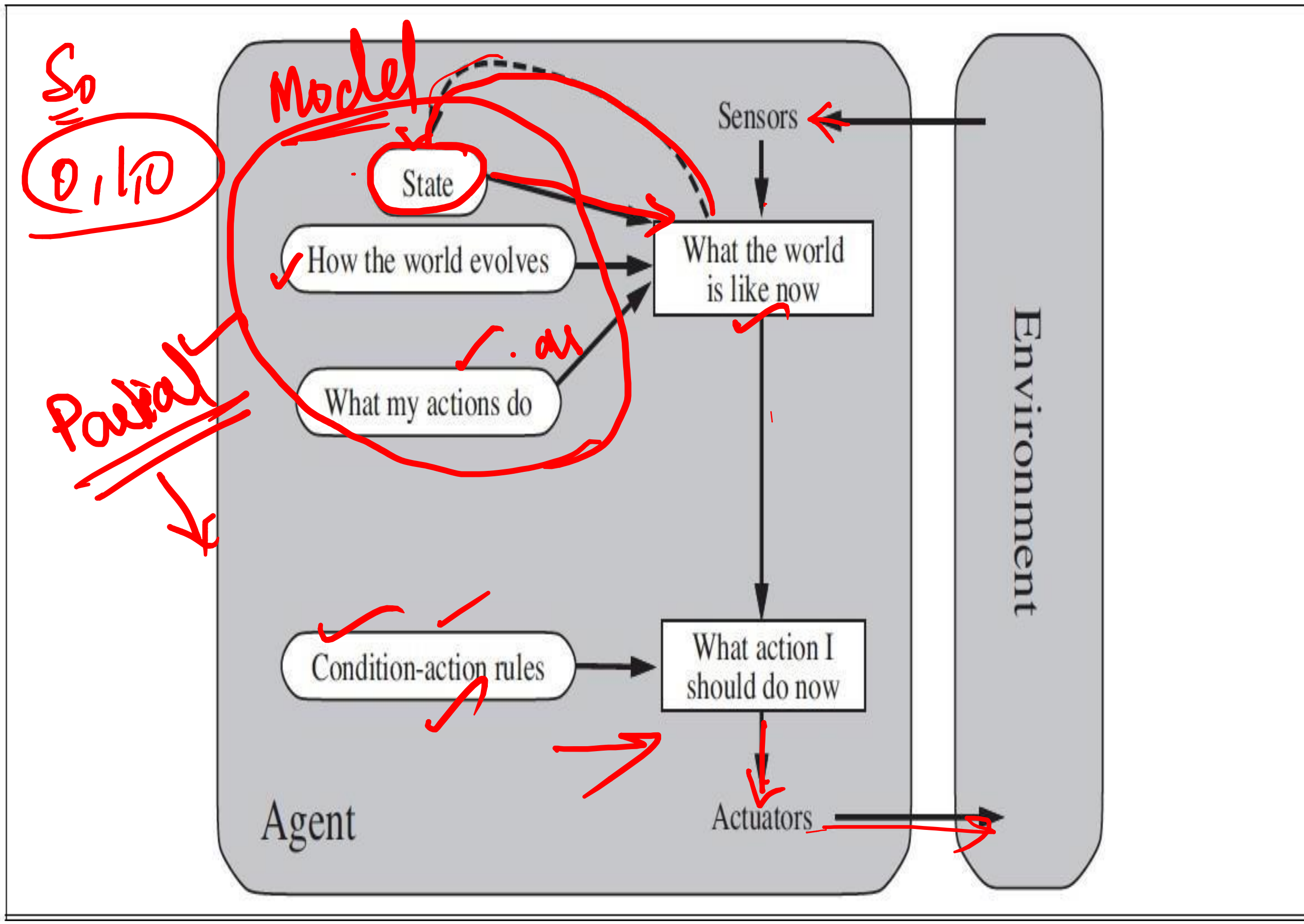


Agent Architectures

Model based Agent

Simple Reflex Agents

Model Based Agents



Agent Architectures

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

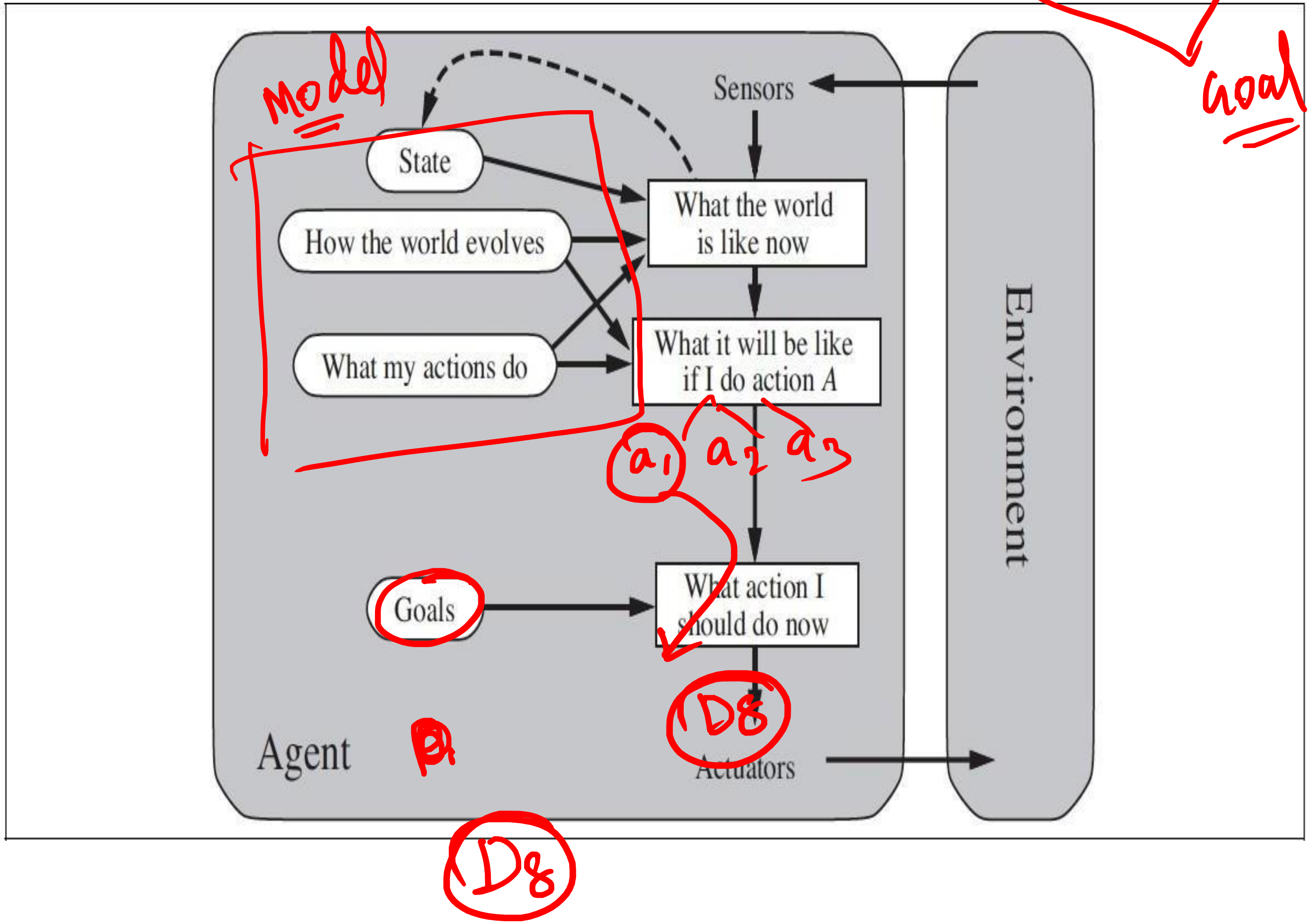
state←UPDATE-STATE(*state, action, percept, transition model, sensor model*)
rule←RULE-MATCH(*state, rules*)
action ←*rule*.ACTION
return *action*

Agent Architectures

Goal based Agent

P1 P3 P5
goal

- Simple Reflex Agents
- Model Based Agents
- Goal Based Agents



Agent Architectures

Utility based Agent

Simple Reflex Agents

↓

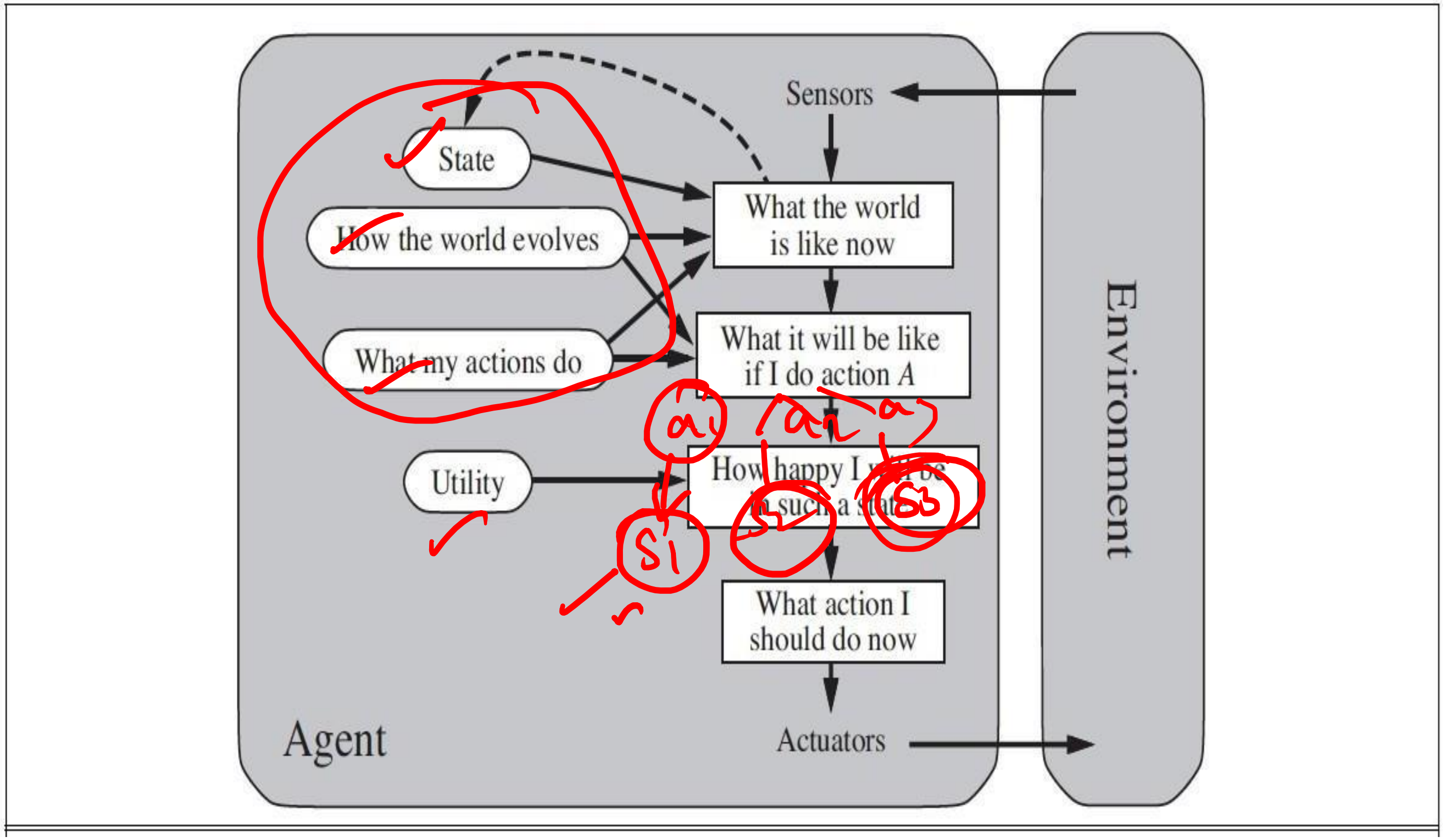
Model Based Agents

↓

Goal Based Agents

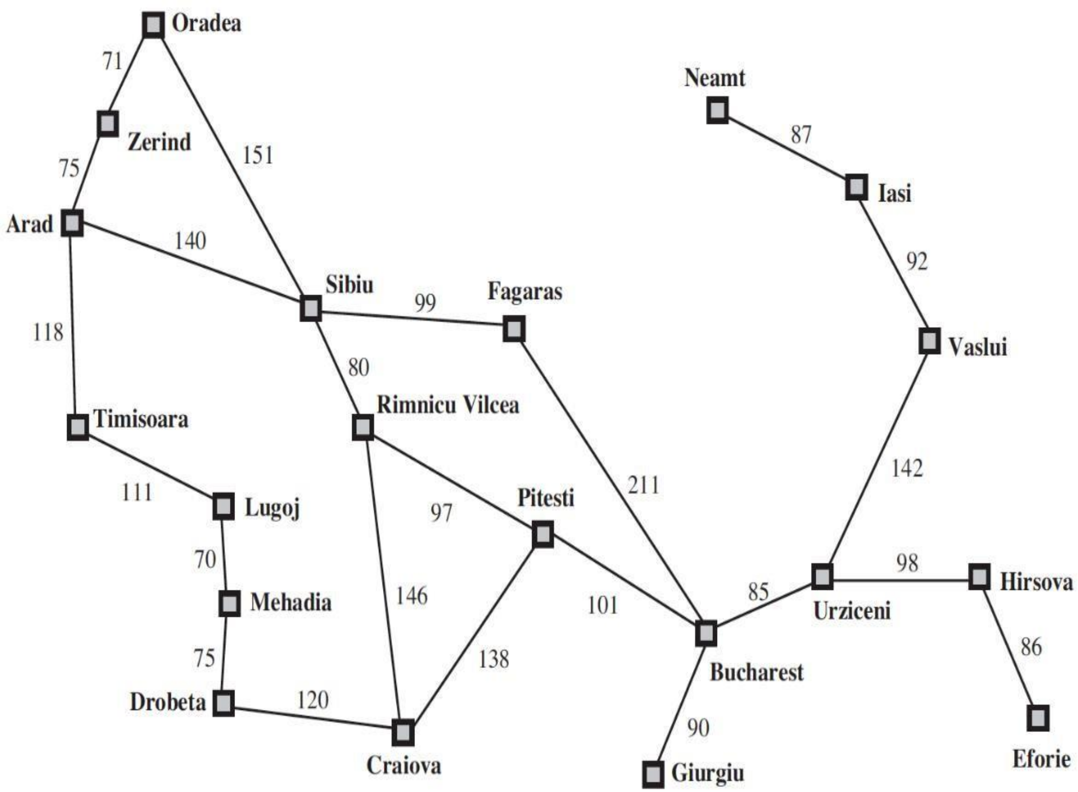
↓

Utility Based Agents



Agent Architectures

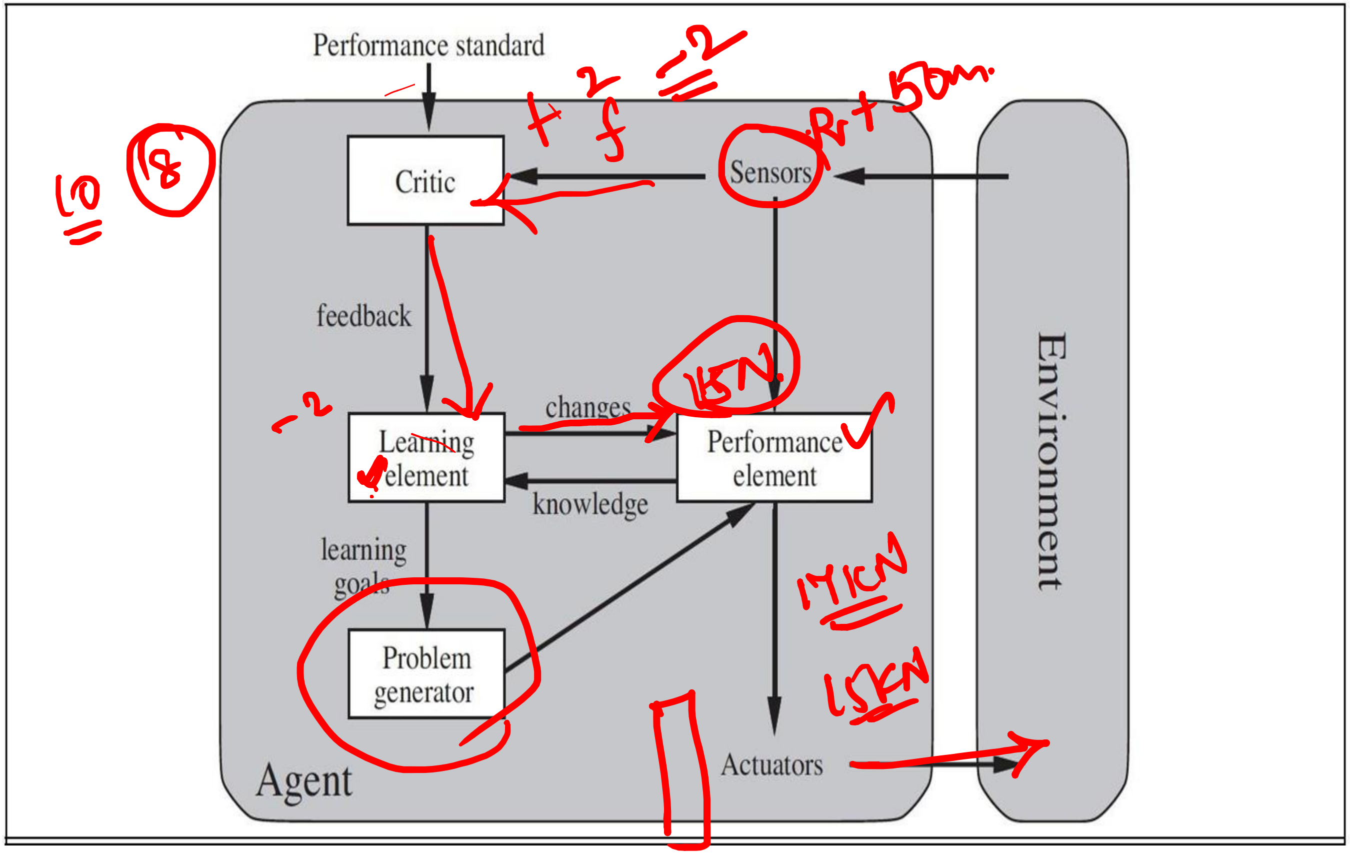
Learning Agent



Agent Architectures

Learning Agent

- Simple Reflex Agents
- Model Based Agents
- Goal Based Agents
- Utility Based Agents
- Learning Agents



Agent Architectures

Learning Agent

- Simple Reflex Agents
- Model Based Agents
- Goal Based Agents
- Utility Based Agents
- Learning Agents

Role of Learning

- **Performance Element** – taking a decision of action based on percepts
- **Learning Element** – Make the performance element select better actions such that the utility function is optimized
- **Critic** – Provides feedback on the actions taken
- **Problem Generator** – Make the Performance Element select sub-optimal actions such that you would learn from unseen actions

Role of Learning

Agents that improve their performance by learning from their own experiences

Input Percept

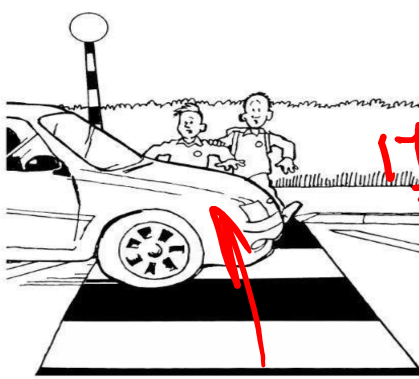
Rr

Performance Element

Output Action

APPLY BRAKE

Resulting State



17KM

Critic

Feedback

-2

Negativ

Penalti

Role of Learning

Input Percept

color + dist

Possible Actions

Brake
Change Gear to Lower
Change Gear to Higher
Accelerate
Steer left
Steer right

PG

Random

Selected Action

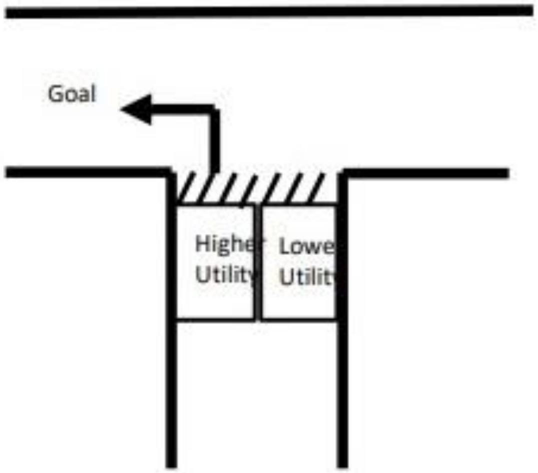
Change Gear to Lower

Role of Learning

Performance Element – Takes decision on action based on percept

$$f(\text{red signal}, \quad \text{distance}) = 15k\ N\ \text{brake}$$
$$\text{distance} = f'(\text{percept sequence})$$
$$f(\text{percepts}, \text{distance}, \text{raining})$$

- $f(state_0, actionA) = 0.83$,
- $f(state_0, actionB) = 0.45$



Role of Learning

Learning : Supervised Vs Unsupervised Vs Reinforcement

Input Percept

at 50 mts away

→ **Performance Element*** → APPLY BRAKE

Output Action

→ Force of **17k** N

Input Percept

at 50 mts away

→ Performance Element → APPLY BRAKE

Output Action

Force of 15k N

+ Critic Feedback -2

Role of Learning

Learning : Supervised Vs Unsupervised Vs Reinforcement

Input Percept

Rt

at 50 mts away

Performance Element*

Output Action

APPLY BRAKE

Force of **17k** N

Input Percept

at 50 mts away

Performance Element

Output Action

APPLY BRAKE

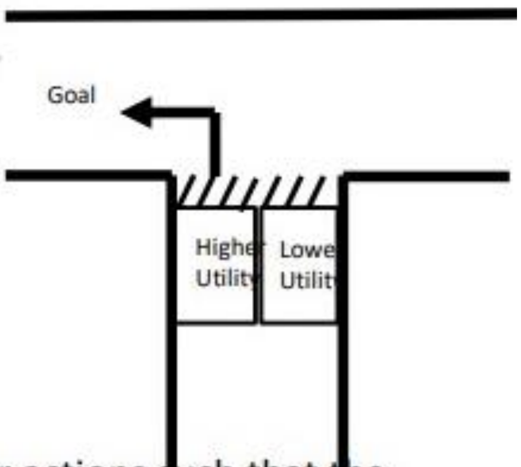
Force of 15k N

+ Critic Feedback -2

Role of Learning

Performance Element – Takes decision on action based on percept

$$f(red\ signal, \quad distance) = 15k\ N\ brake$$
$$distance = f'(percept\ sequence)$$
$$f(percepts, distance, raining)$$



- $f(state_0, actionA) = 0.83,$
- $f(state_0, actionB) = 0.45$

Learning Element – Make the performance element select better actions such that the utility function is optimized

Critic – Provides feedback on the actions taken

Problem Generator – Make the Performance Element select sub-optimal actions such that you would learn from unseen actions

Required Reading: AIMA - Chapter #1, 2, 3.1, 3.2, 3.3

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials