Artificial Intelligence

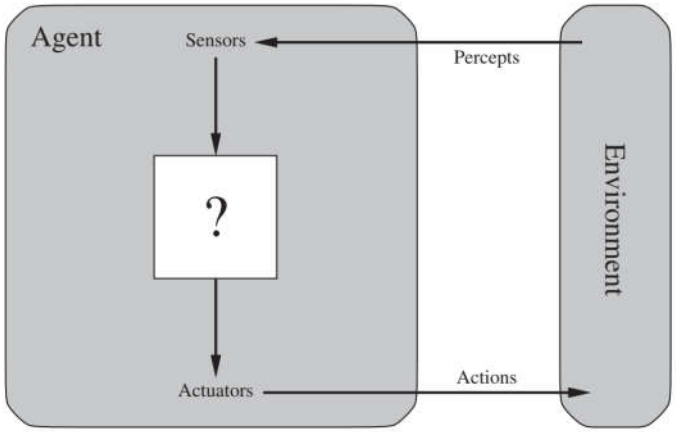
AGENT

Intelligent Agent (IA)

- A computational IA or program or software that gathers information about an environment and takes actions based on that Information.
  - A robot
  - Product/Service Recommendation System of an E-Commerce Portal/Site.
  - Item Sorted in Factory
  - A Smart Traffic Signal
  - Search Engine Auto-Complete and Auto-Suggestion System

Structure of IA

- An agent is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **actuators**.



HUMAN vs ROBOT vs SOFTWARE (w.r.t IA)

	Human Agent	Robot Agent	Software Agent
Sensors	Eye, Ear, Nose and Sensory organs	Camera, LIDAR, SONAR, RADAR, etc.	Text/Image/AV File/Stream, Network Packets, UI Inputs
Actuators	Hand, Legs and Vocal tracts, etc.	Motor, Servos, Display, Buzzer,	UI Outputs, Output File/Streams, Network Packets

GOOD BEHAVIOR: THE CONCEPT OF RATIONALITY

- A rational agent is one that does the **right thing**
- Now the age-old question, what is **right**?
  - by considering the *consequences* of the agent's behavior. When an agent is plunked down in an environment, it generates a sequence of actions according to the percepts it receives. This sequence of actions causes the environment to go through a sequence of states. If the sequence is desirable, then the agent done the **right thing** or performed well.
- Performance of IA

IA Performance Measurement

- Environment detects performance matric rather then Agents.
- Reward/Penalty based task evaluation
- A Rational Agent always tries to maximize its performance.

Task Environment

- PEAS (**P**erformance, **E**nvironment, **A**ctuators, **S**ensors)

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	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, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Nature of Task Environment

- **Fully observable** vs. **partially observable**
- **Single agent** vs. **multiagent**
- **Deterministic** vs. **stochastic**
- **Episodic** vs. **sequential**
- **Static** vs. **dynamic**
- **Discrete** vs. **continuous**
- **Known** vs. **unknown**

Example

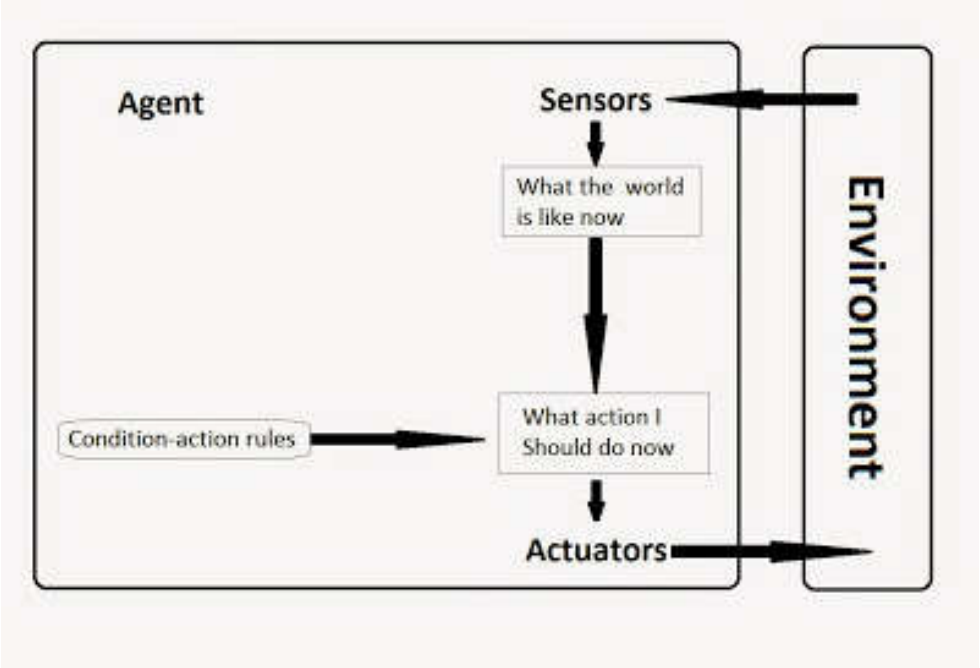
Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	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

Simple Reflex Agent

SimRA

Simple Reflex Agent

- Action taken based on current percept
- Implement through condition – action rules

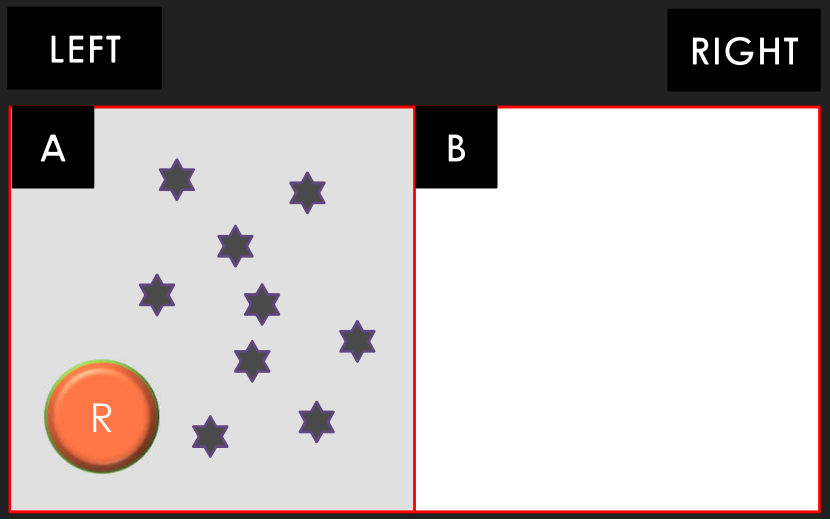


SimRA Function

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*

Example – Floor Cleaning Robot (Roomba v0.1)



PEAS – Roomba v0.1

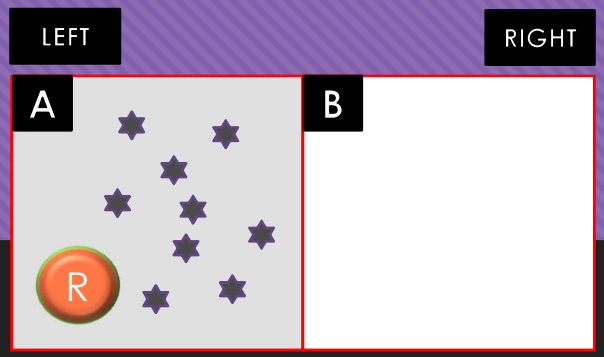
- Performance
  - Cleanliness
  - Battery Power Conservation
- Environment
  - Consists of 2 tiles named A and B respectively
- Actuator
  - Vacuum Cleaning Unit (Action = VAC)
  - Left-Right Locomotion Unit (Action = LEFT/RIGHT)
- Sensor
  - Dirt Sensor (Percept = Dirty/Clean)

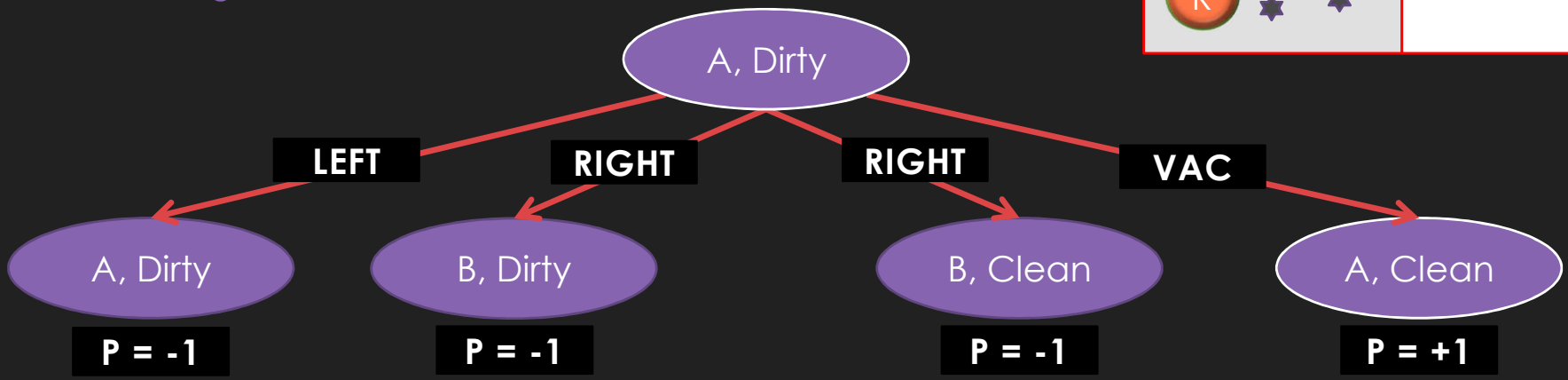
Performance Measurement

Clean	VAC	-1
	LEFT/RIGHT	0
Dirty	VAC	+1
	LEFT/RIGHT	-1

Any Invalid move to LEFT/RIGHT is -1

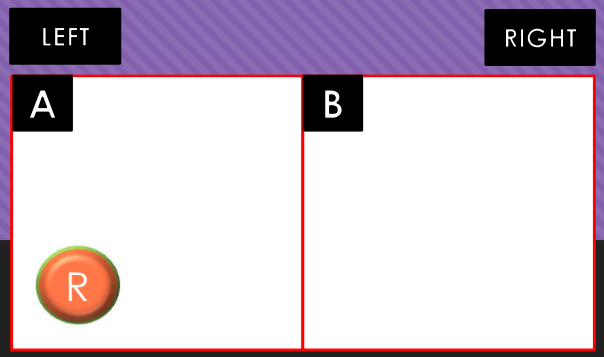
Decision Tree

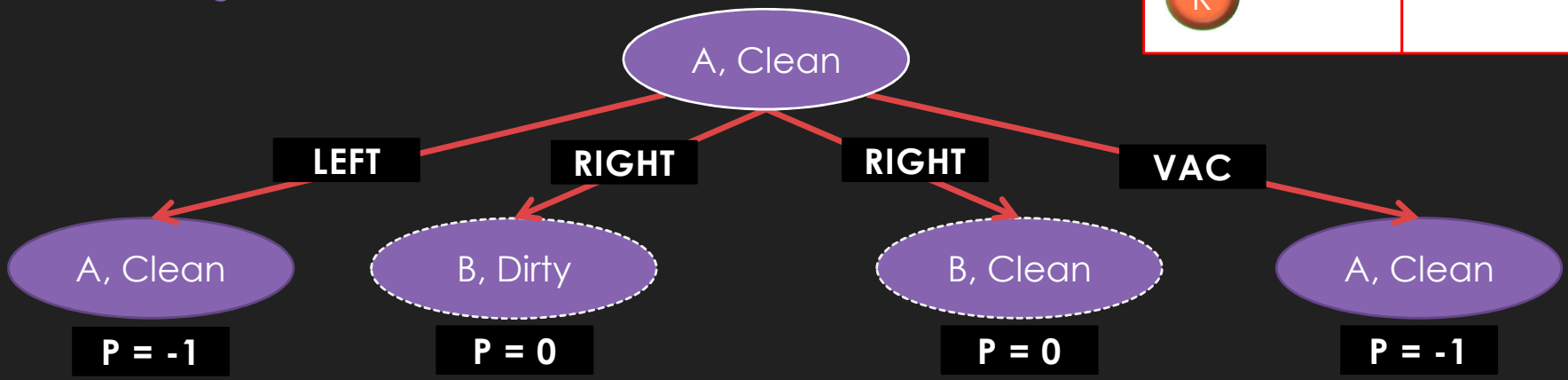




If (Loc = A AND Percept = Dirty) *then* Action = VAC

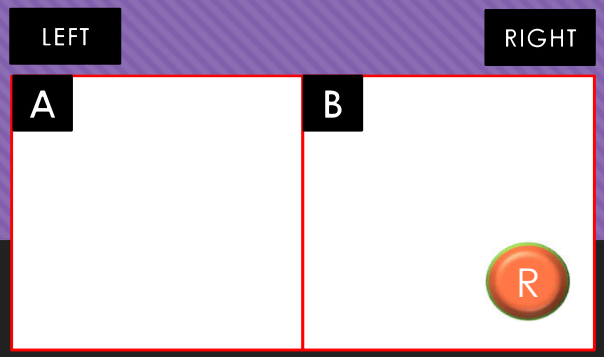
Decision Tree

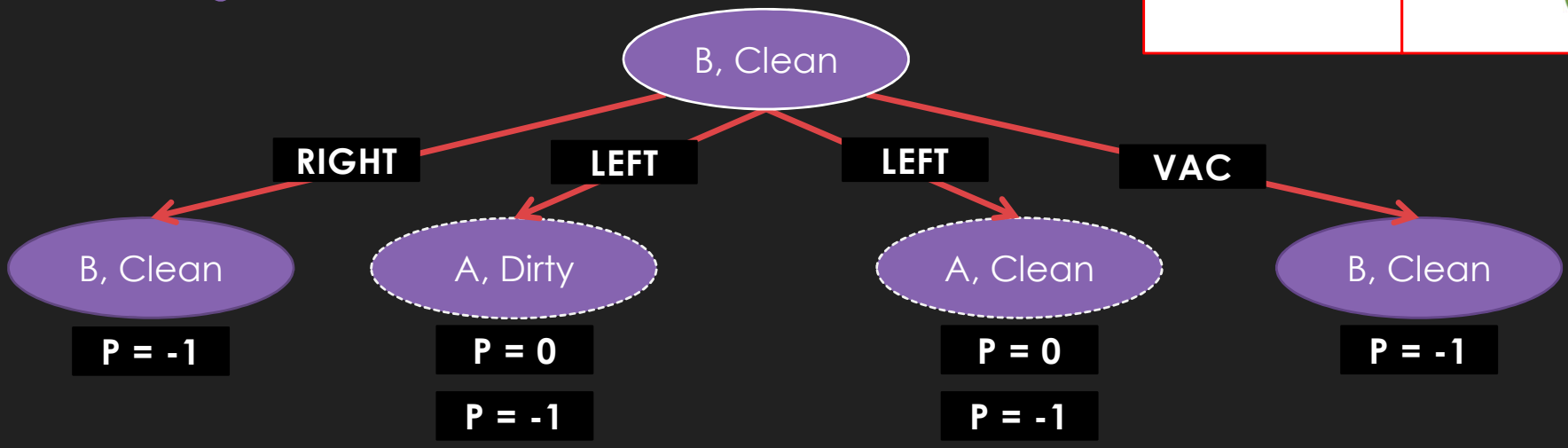




If (Loc = A AND Percept = Clean) *then* Action = RIGHT

Decision Tree





P = 0 (LOOP)	*If* (Loc = B AND Percept = Clean) *then* Action = LEFT
P = -1 (NOLOOP)	*If* (Loc = B AND Percept = Clean) *then* Action = NoOP

SimRA Function

```
function getRoombaAction(Percept, Loc){
    if (Percept = "Dirty"){
        Action = VAC;
    }elseif (Loc = A){
        Action = RIGHT;
    }elseif (Loc = B){
        Action = LEFT;
    }
    return Action;
}
```