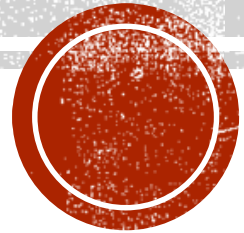
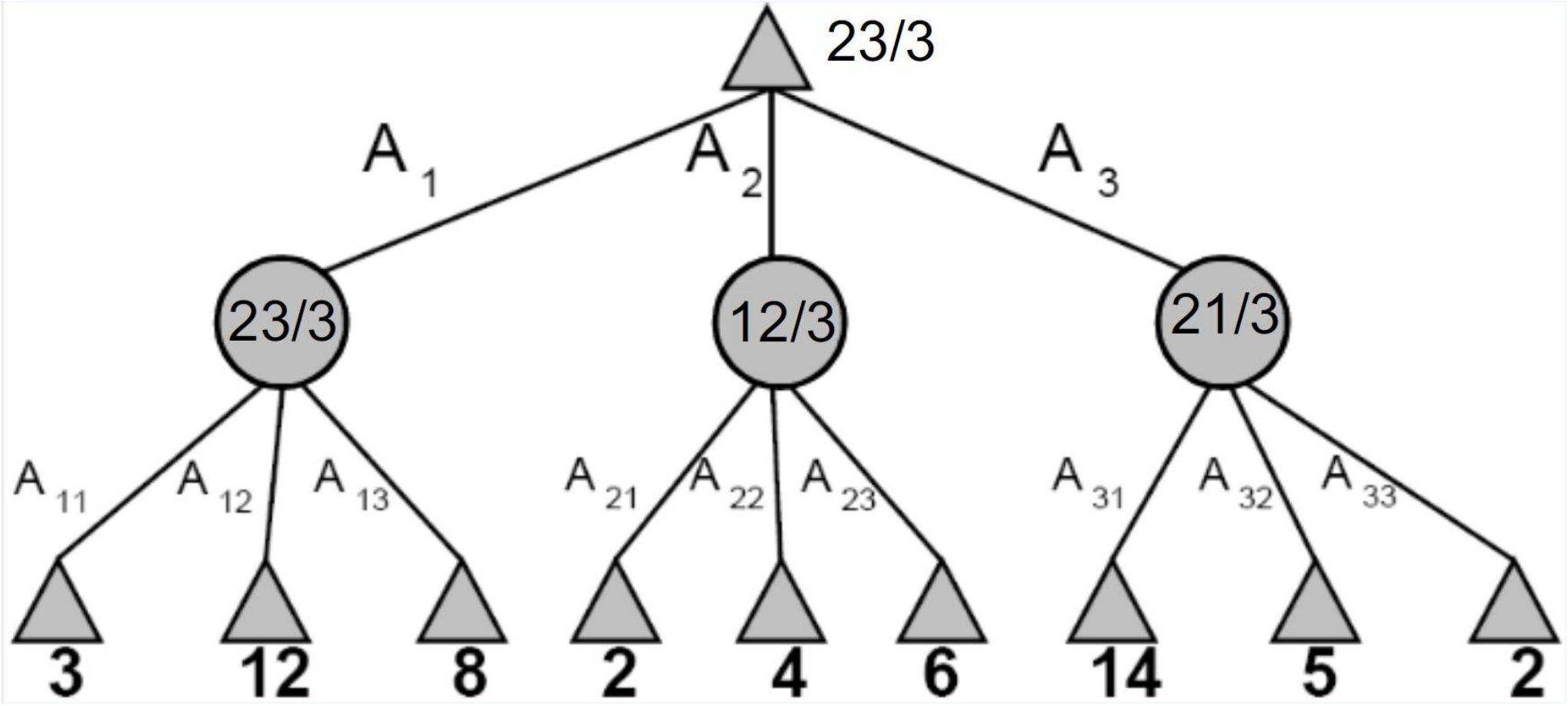
# EXPECTIMAX

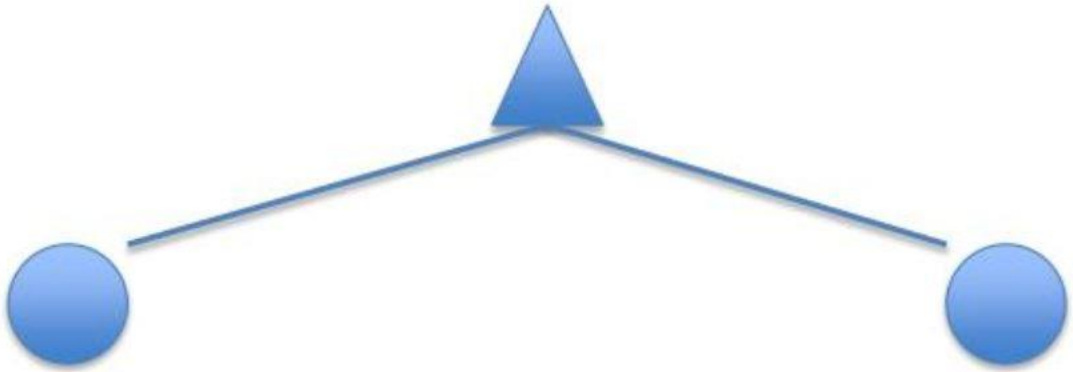
Dr. Poulami Dalapati
Asst. Prof., Dept. of CSE,
LNMIIT Jaipur



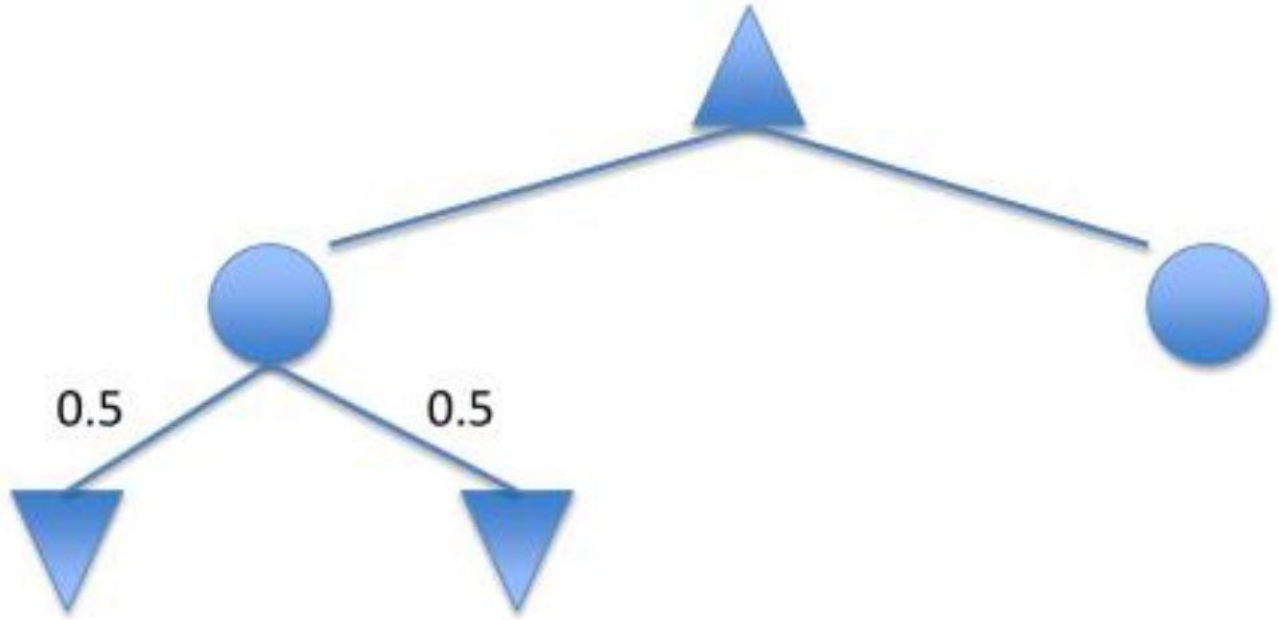
# EXPECTIMAX


23/3
A 1
A 2
A 3
23/3
12/3
21/3
A 11
A 12
A 13
A 21
A 22
A 23
A 31
A 32
A 33
3
12
8
2
4
6
14
5
2

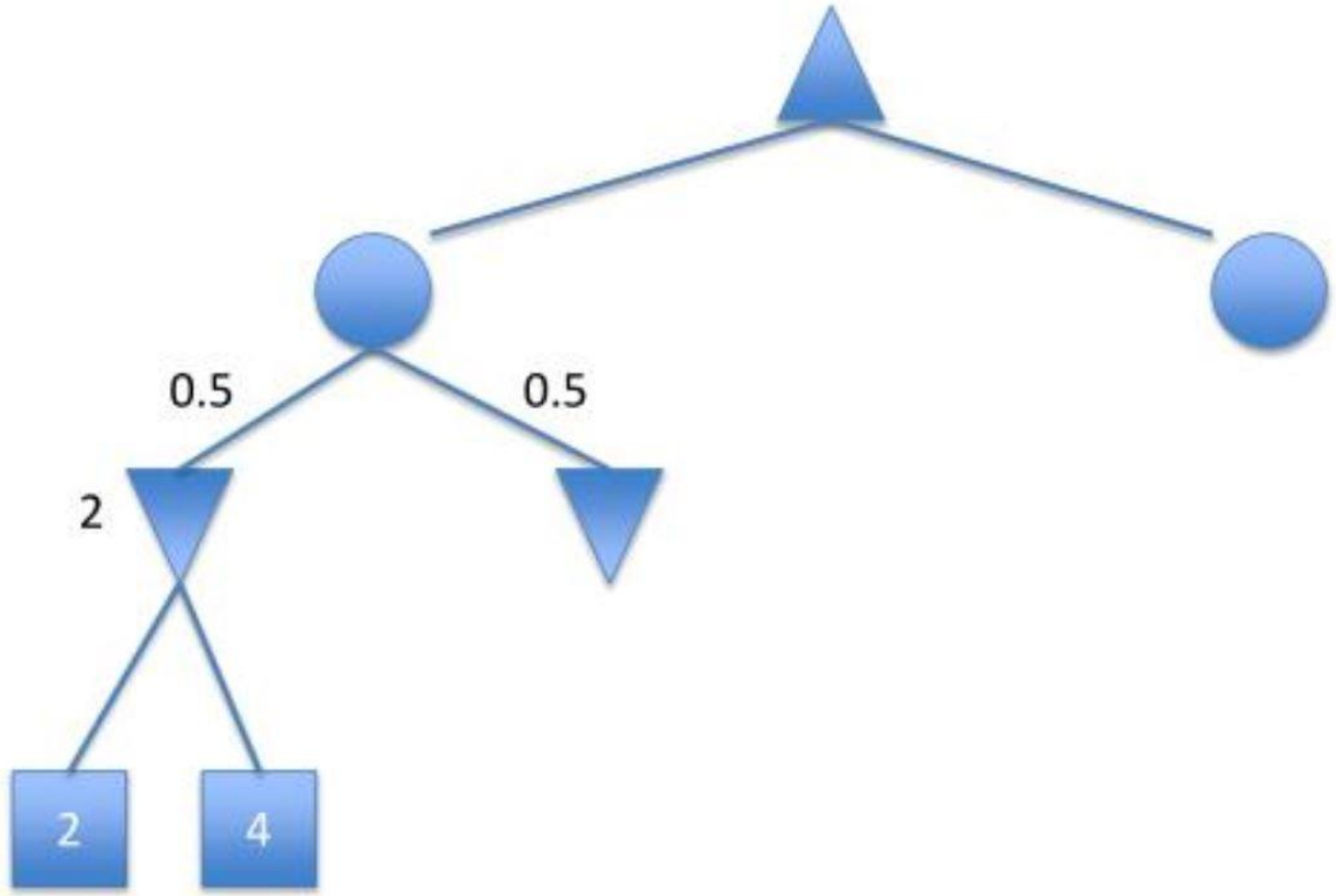
# EXPECTIMAX



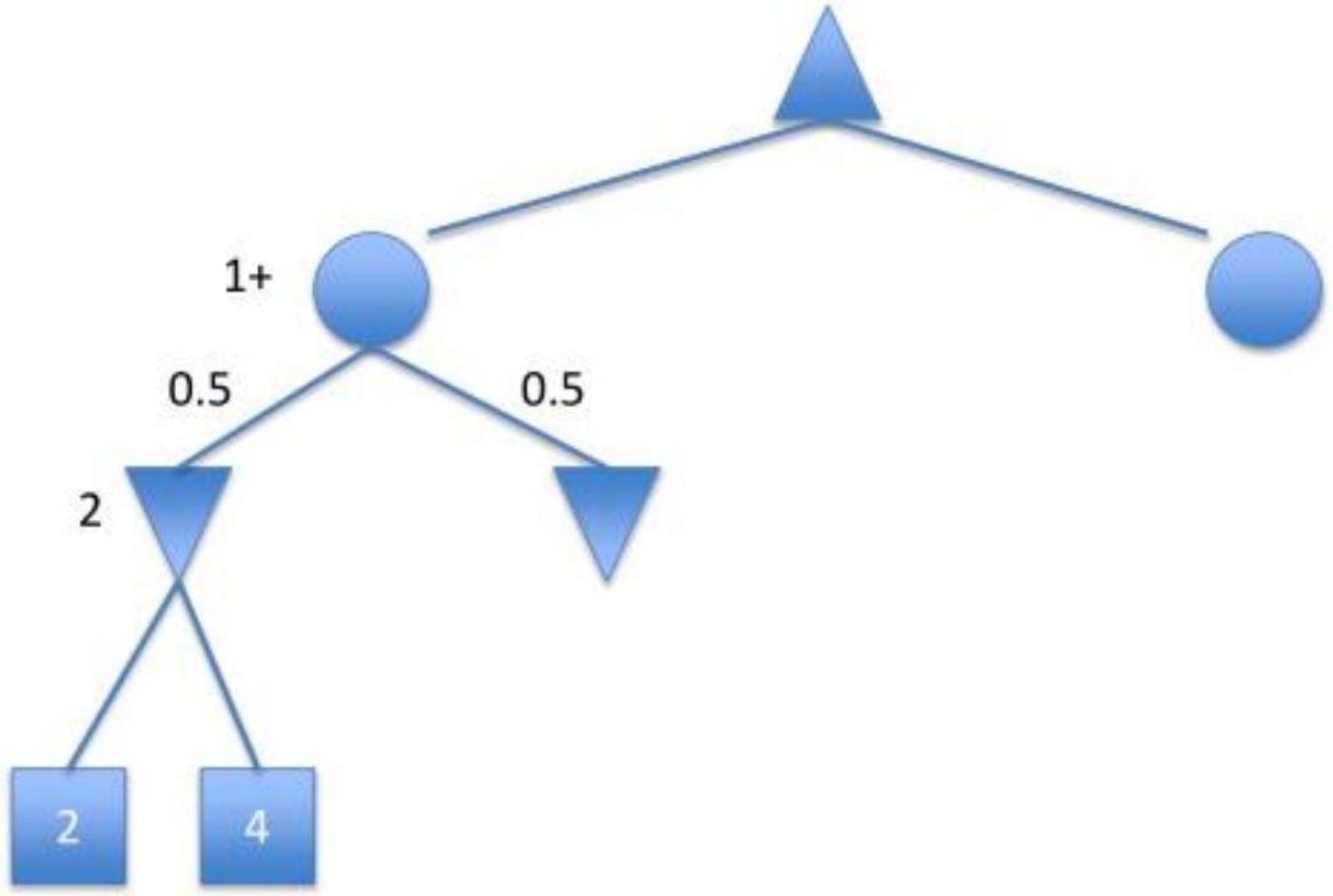
# EXPECTIMAX



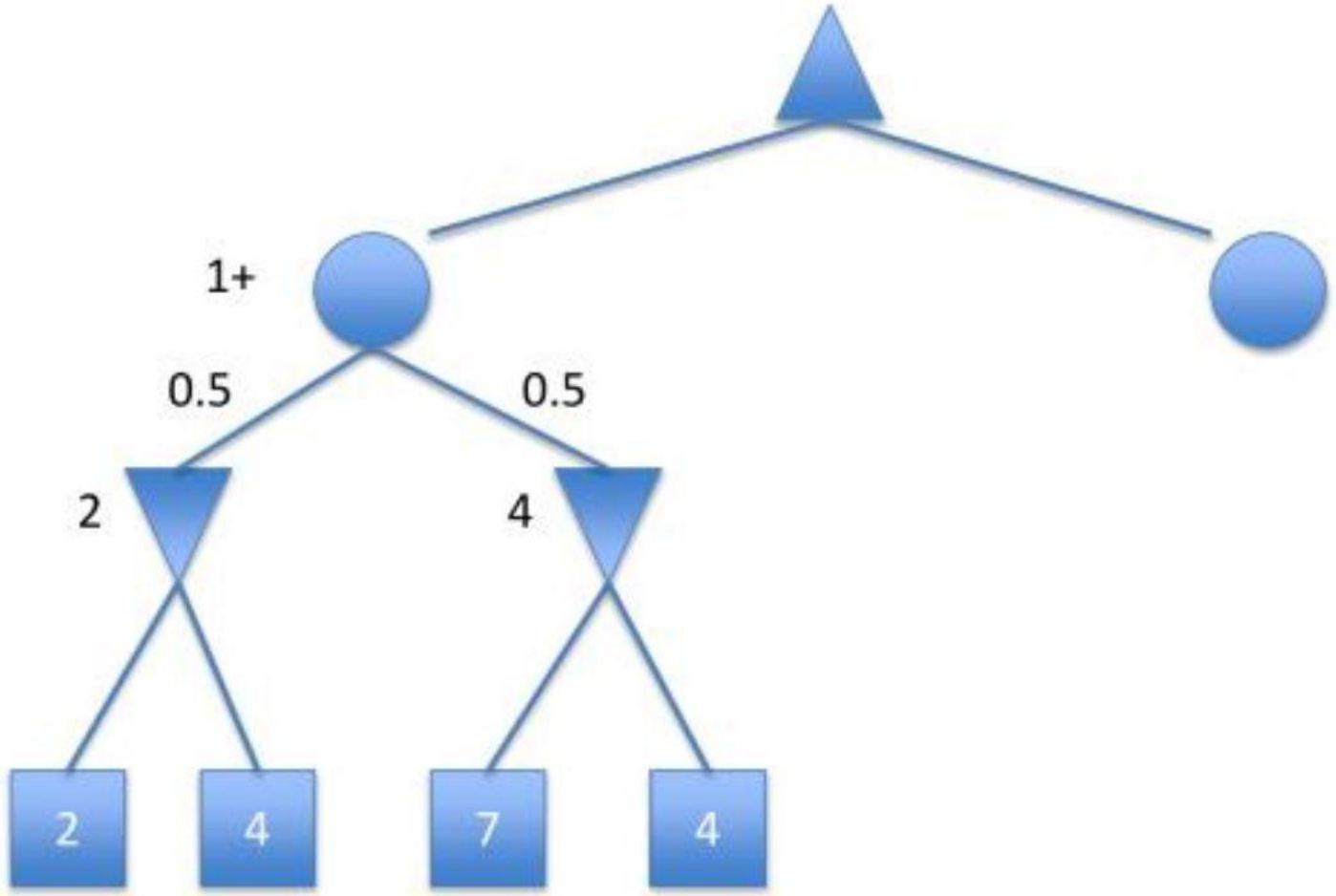
# EXPECTIMAX



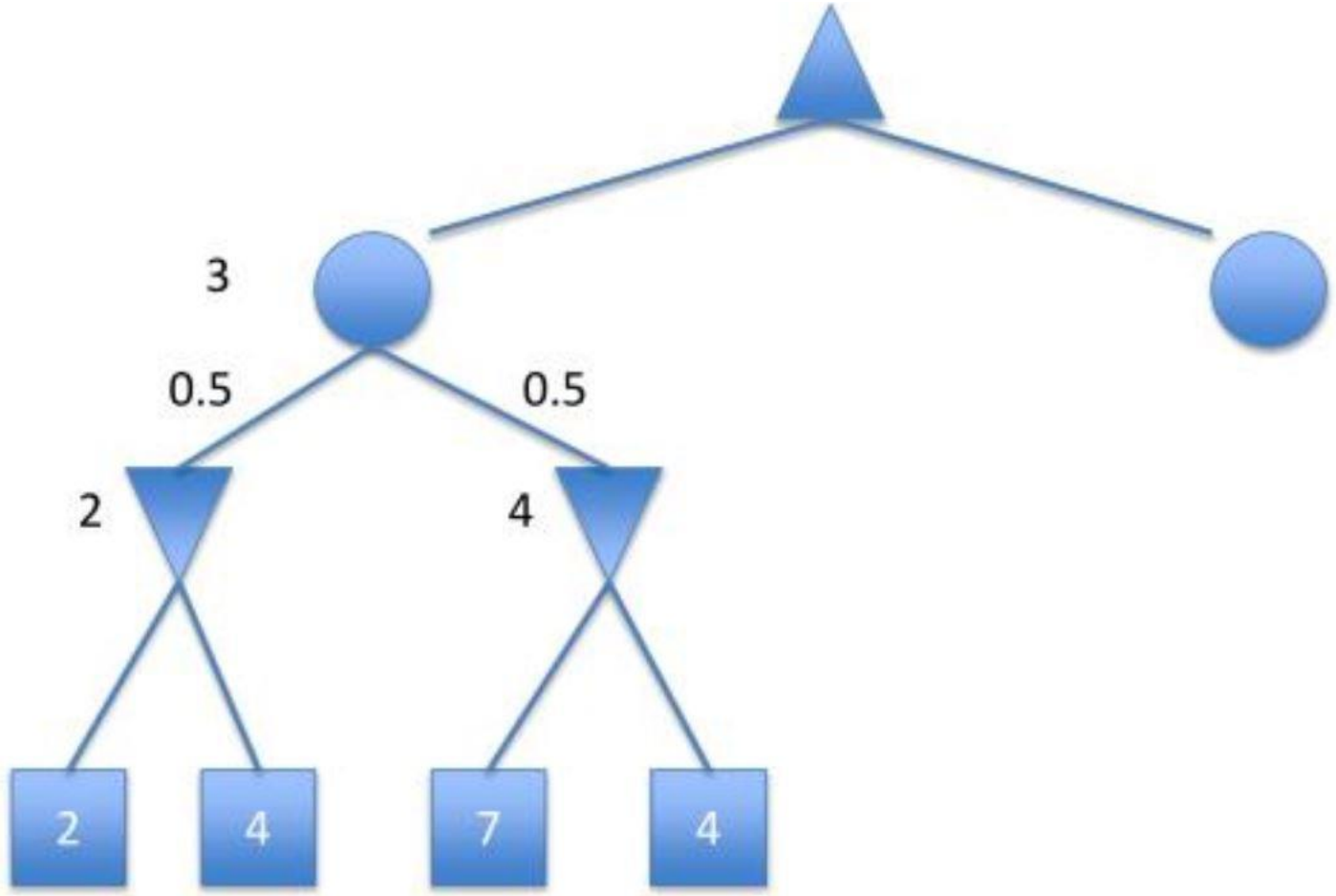
# EXPECTIMAX



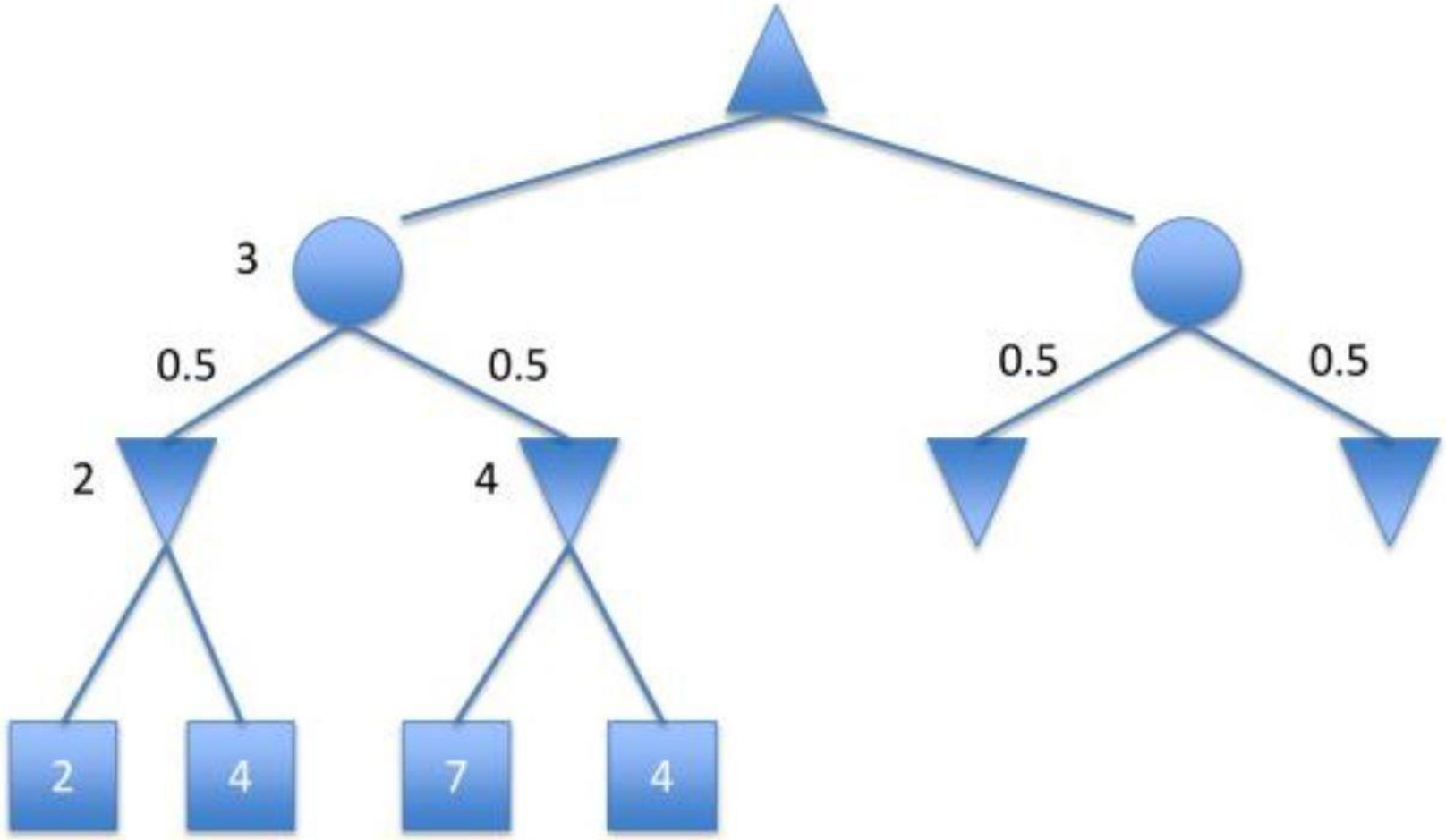
# EXPECTIMAX



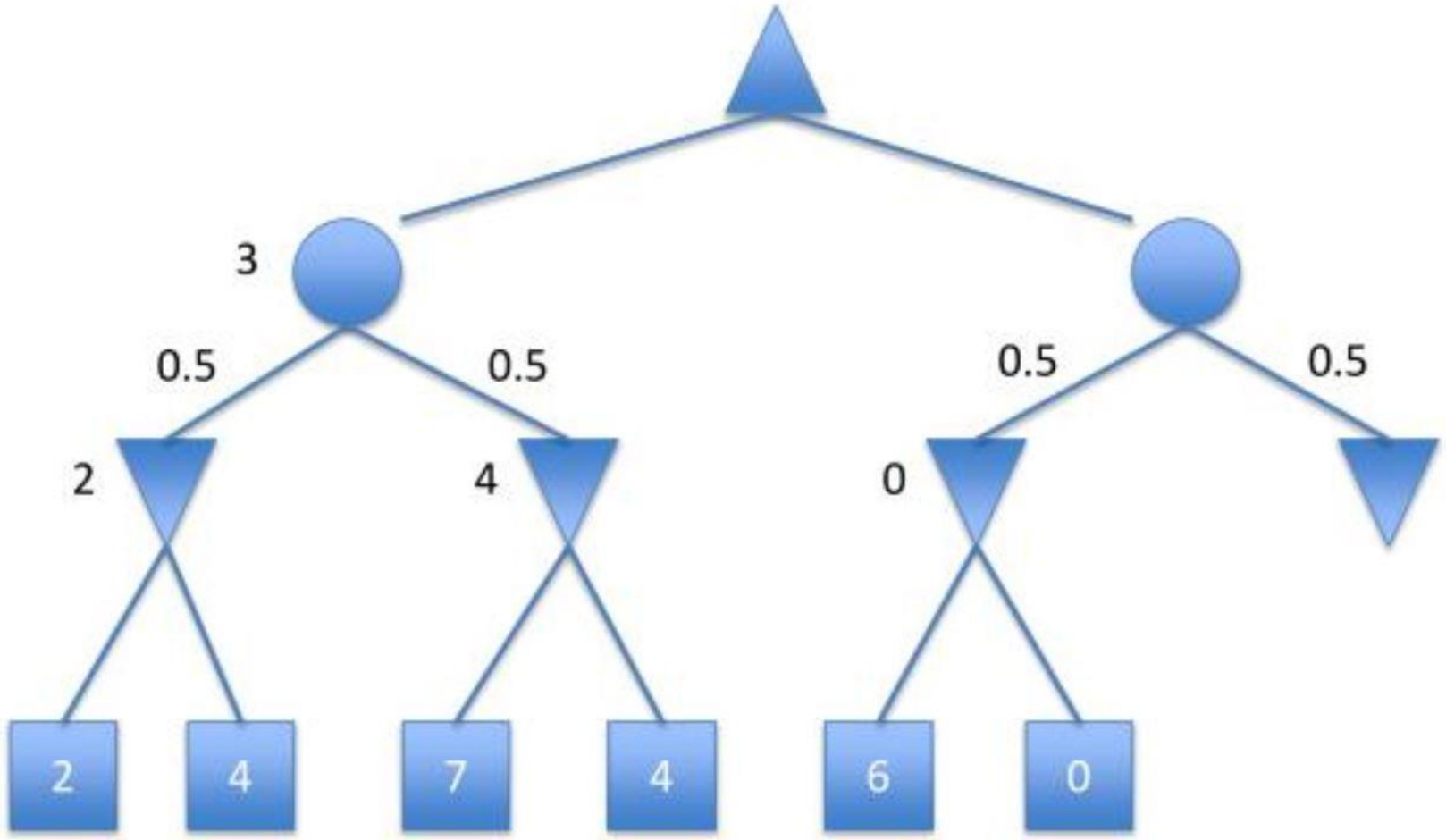
# EXPECTIMAX



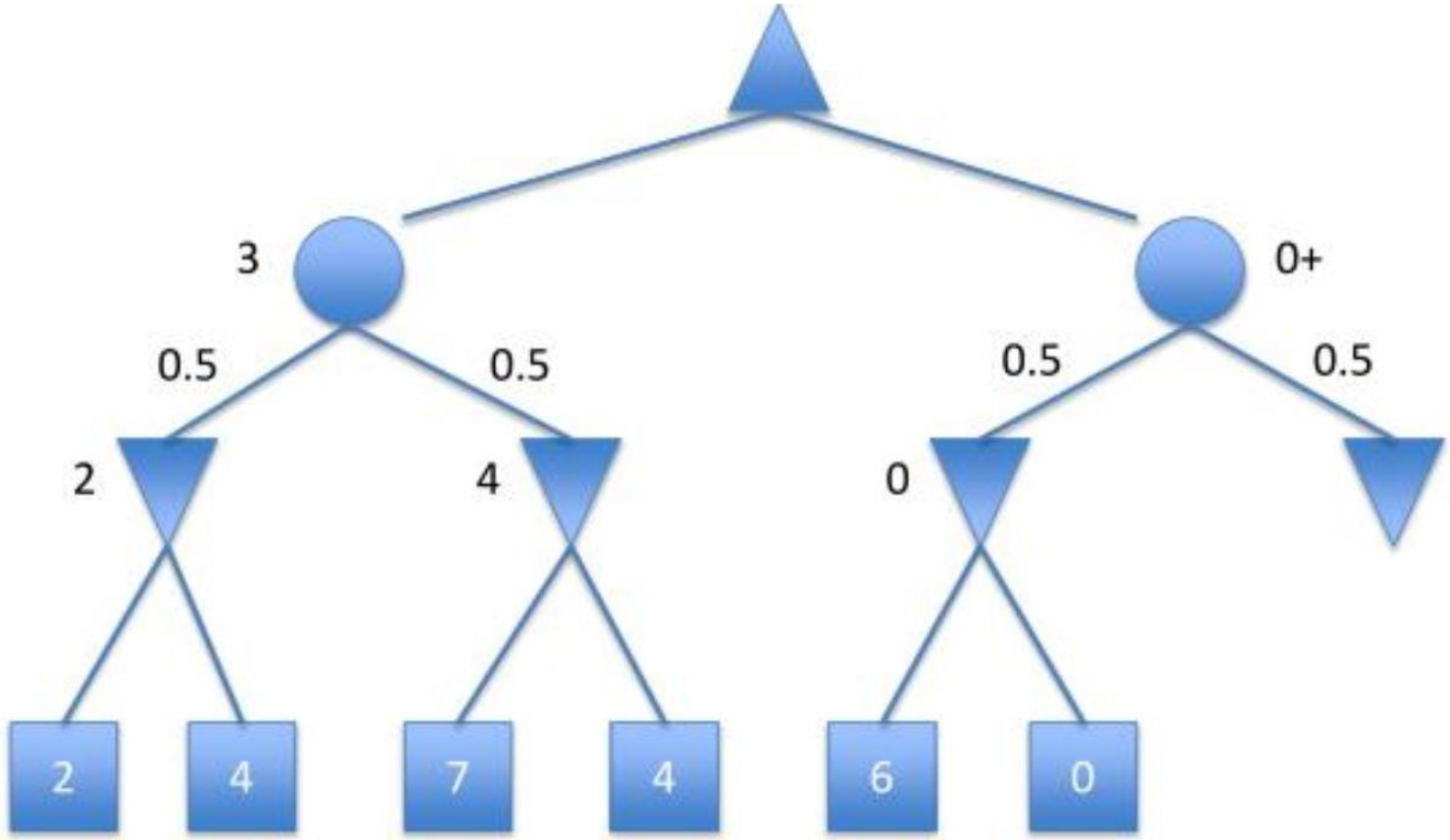
# EXPECTIMAX



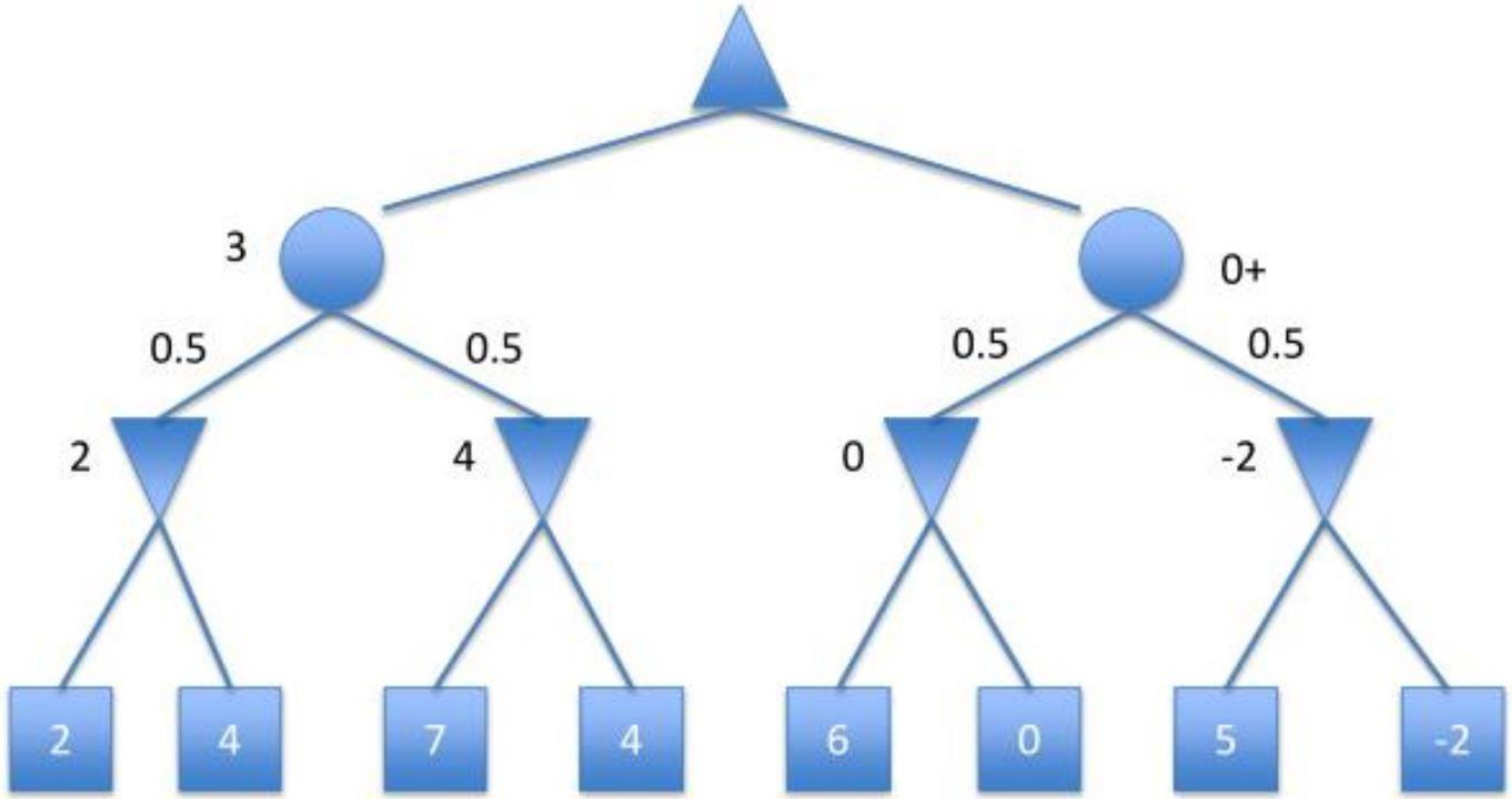
# EXPECTIMAX



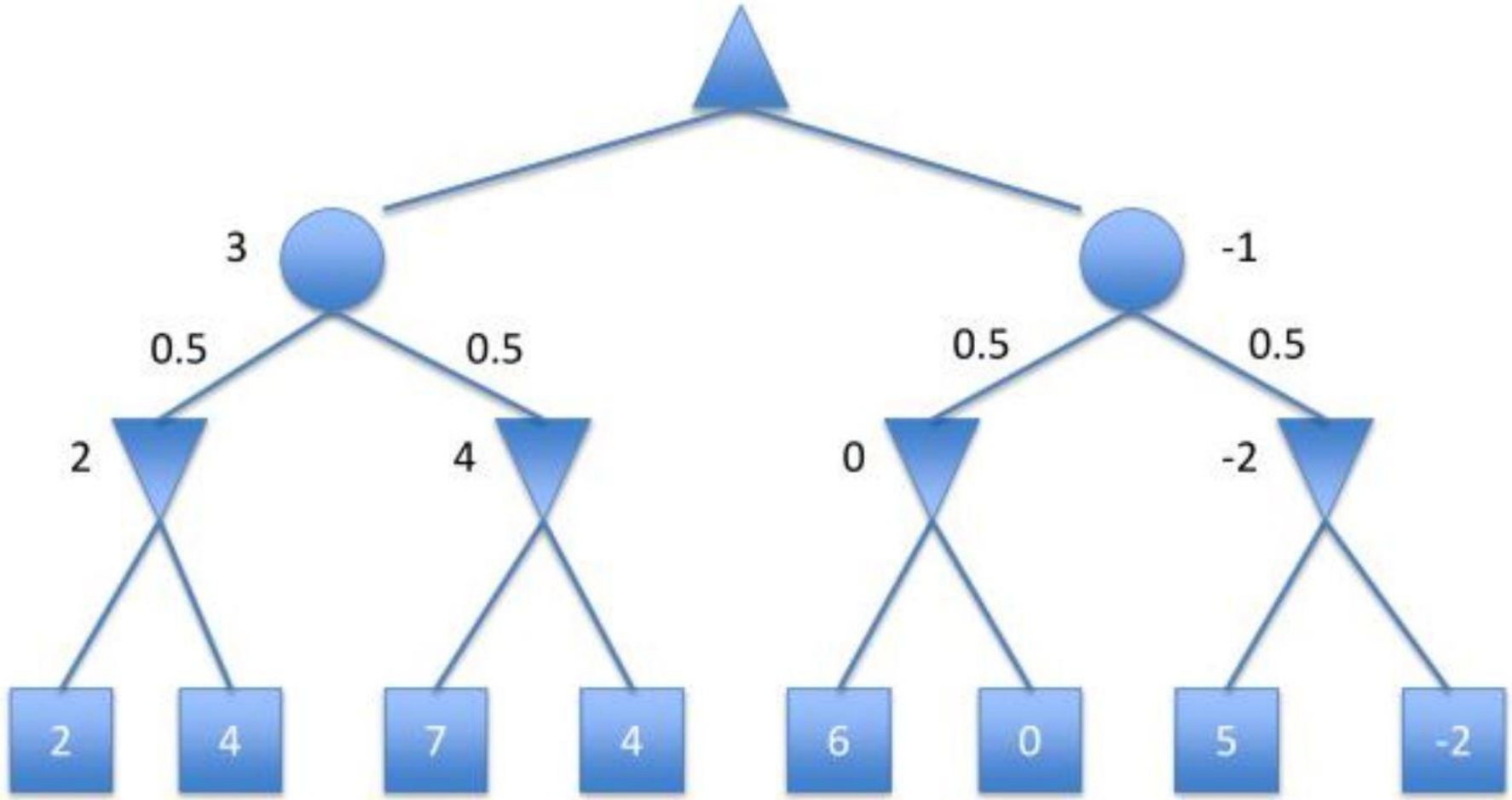
# EXPECTIMAX



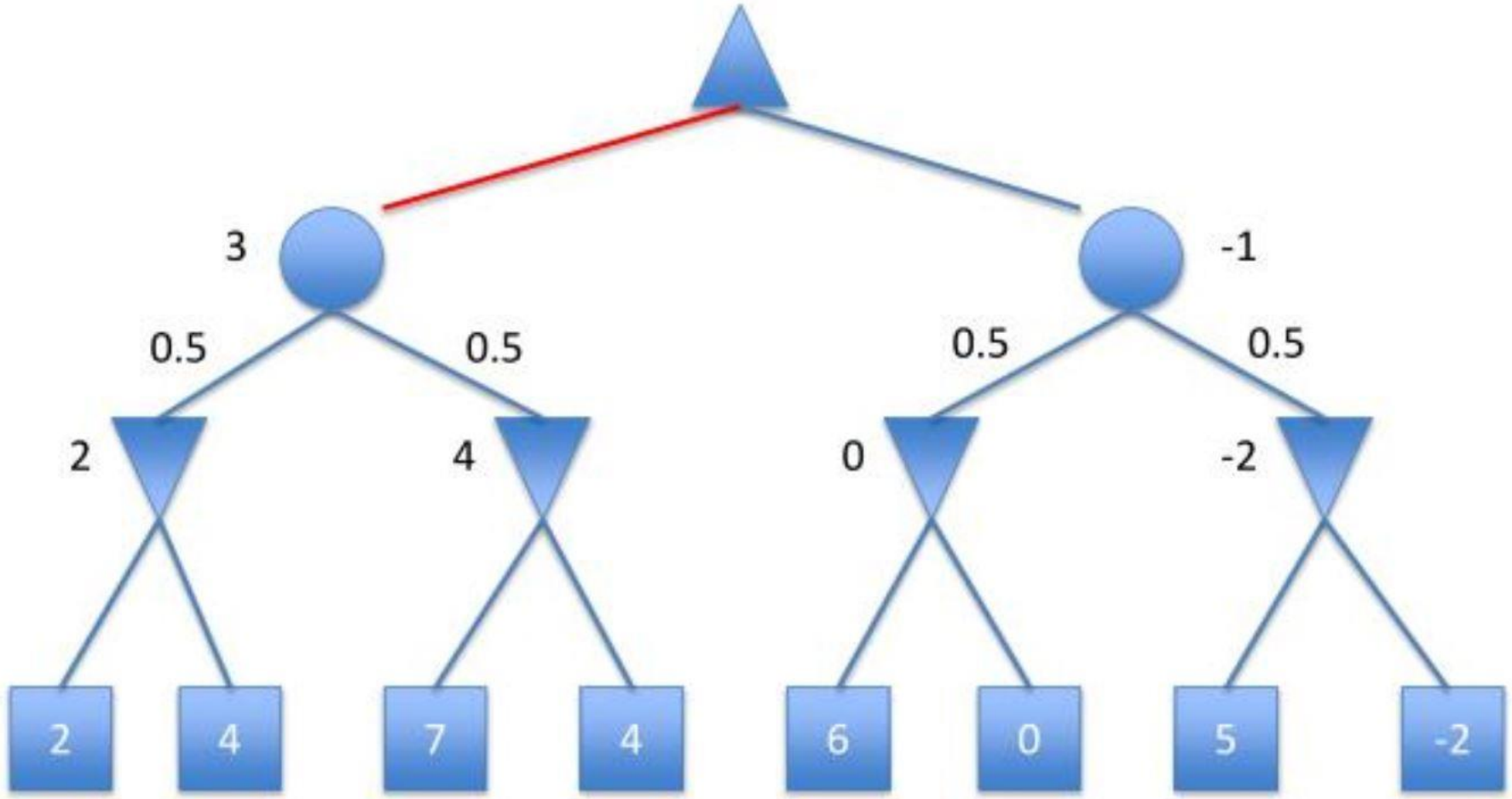
# EXPECTIMAX



# EXPECTIMAX



# EXPECTIMAX



# EXPECTIMAX

**Advantages of Expectimax over Minimax:**

- Expectimax algorithm helps take advantage of non-optimal opponents.
- Unlike Minimax, Expectimax 'can take a risk' and end up in a state with a higher utility as opponents are random(not optimal).

**Disadvantages:**

- Expectimax is not optimal. It may lead to the agent losing (ending up in a state with lesser utility).
- Expectimax requires the full search tree to be explored. There is no type of pruning that can be done, as the value of a single unexplored utility can change the expectimax value drastically. Therefore it can be slow.

# EXPECTIMAX

**Time complexity**: $O(b^m)$

**Space complexity**: O(b*m), where *b* is branching factor and *m* is the maximum depth of the tree.

**Applications:** Expectimax can be used in environments where the actions of one of the agents are random. Following are a few examples,

- In **Pacman**, if we have random ghosts, we can model Pacman as the maximizer and ghosts as chance nodes. The utility values will be the values of the terminal states(win, lose or draw) or the evaluation function value for the set of possible states at a given depth.
- We can create a **minesweeper** AI by modelling the player agent as the maximizer and the mines as chance nodes.

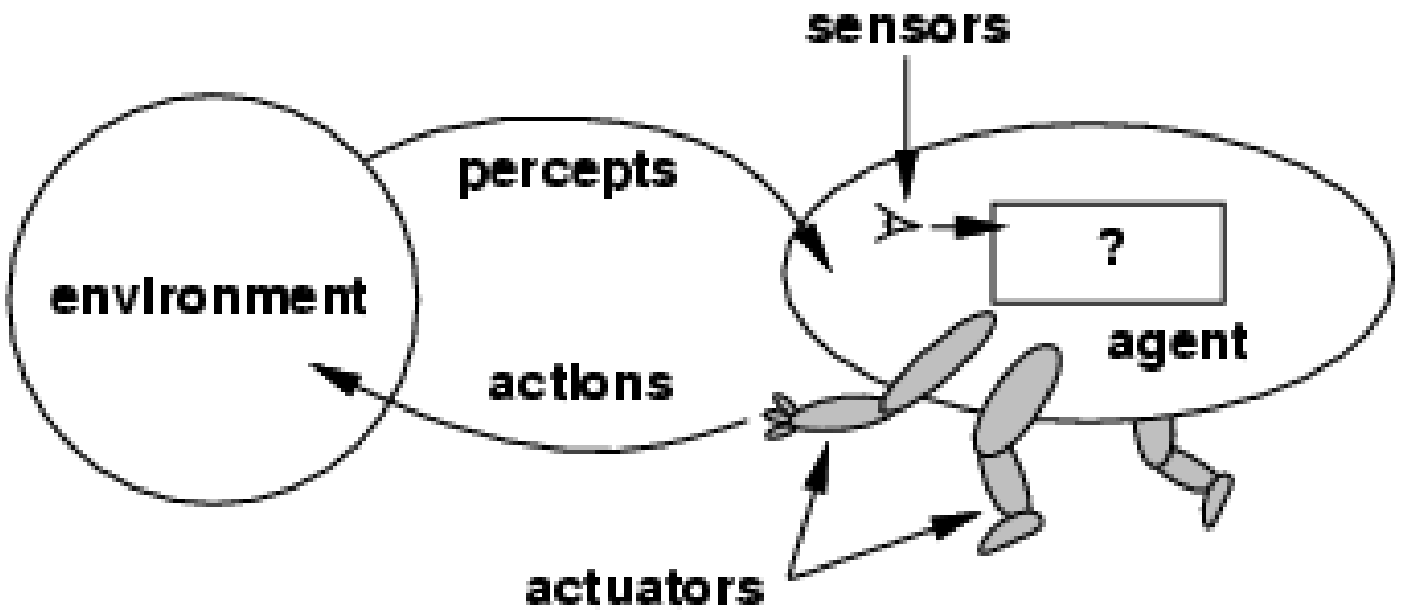
# INTELLIGENT AGENTS

- Agent= <P,E,A,S>
- P: Performance measure
- E: Environment
- A: Actuator
- S: Sensor

# AGENTS

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- *Human agent:* eyes, ears, and other organs for sensors; hands,
- legs, mouth, and other body parts for actuators.
- *Robotic agent:* cameras and infrared range finders for sensors; various motors for actuators

# AGENTS AND ENVIRONMENTS



# ENVIRONMENT TYPES

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- 
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
- 
- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

# CONTD..

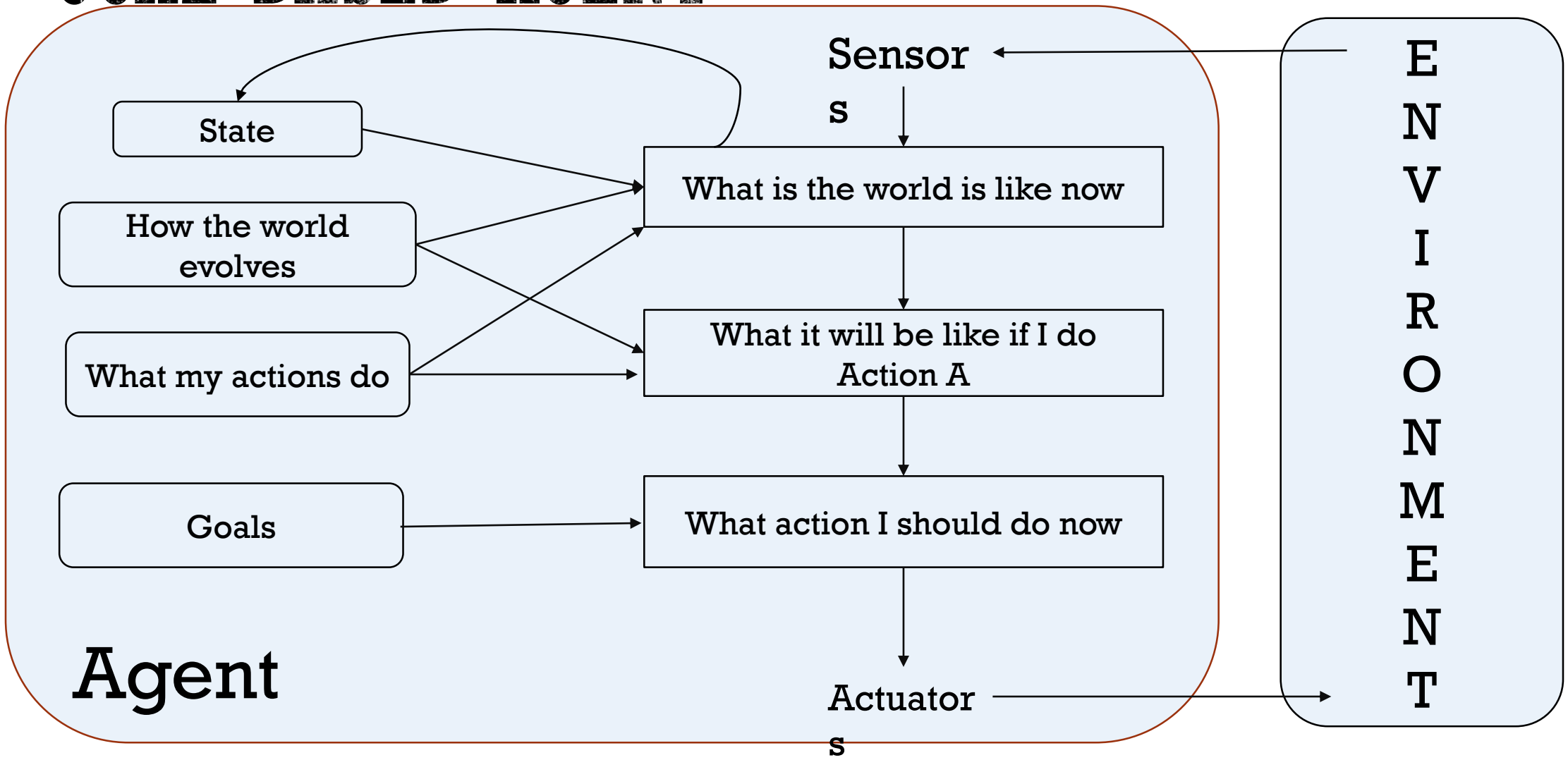
- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)
- 
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- 
- Single agent (vs. multiagent): An agent operating by itself in an environment.

# AGENT TYPES

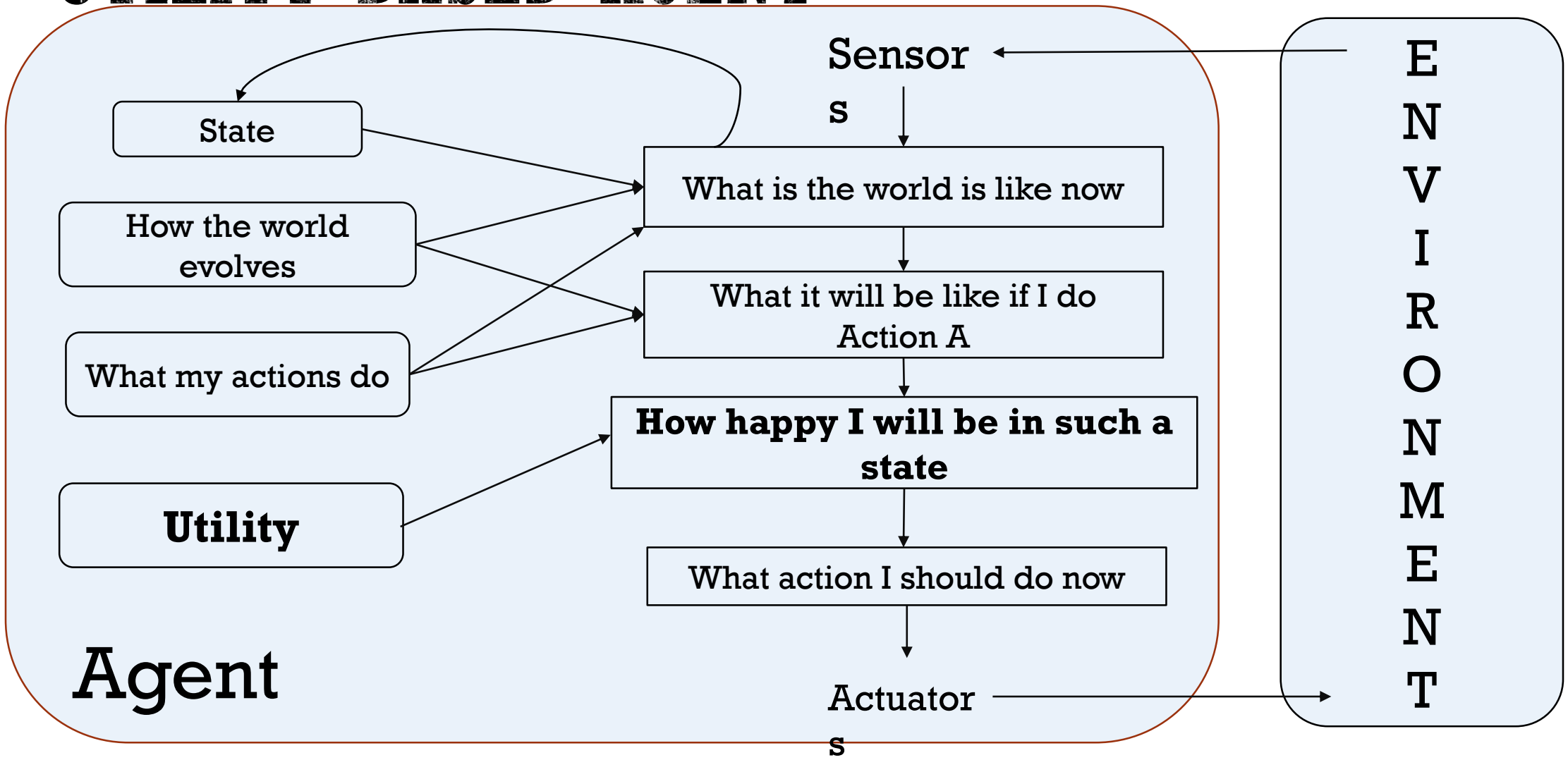
- *Four basic types in order of increasing generality:*

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

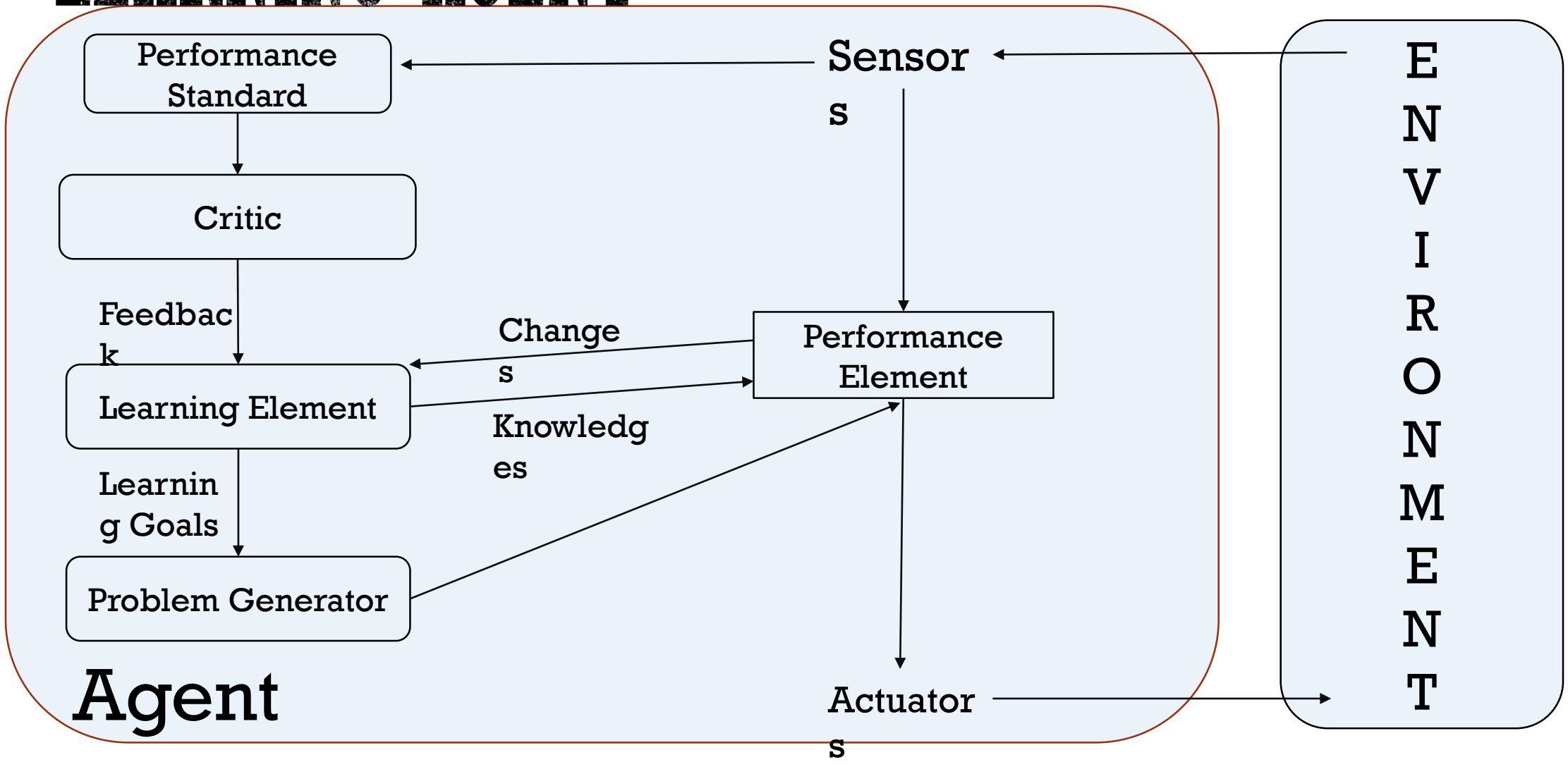
# GOAL BASED AGENT



# UTILITY BASED AGENT



# LEARNING AGENT



# MULTI-AGENT UTILITIES