# Genetic Algorithm

# The idea was proposed by John Holland and is based on biological evolution.

# Genetic Algorithms is to search a space of candidate hypotheses to identify the best hypotheses.

# Genetic algorithms are based on the ideas of natural selection and genetics.

# The algorithm operates by iteratively updating a pool of hypotheses called the population.

# Prototypical Genetic Algorithm

# Initialize Population: P <- generate p hypotheses at random.

# Evaluate: for each h in P, compute fitness(h)

# While Maxh Fitness(h) < Threshold do

# Create a new generation PNew

# Select

# Crossover

# Mutate

# Update: P <- PNew

# Evaluate

# Return the hypothesis from P that has the highest fitness.

# Once the initial generation is created, the algorithm evolves the generation using following operators –

# **Selection Operator:** The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to successive generations.

# probabilistically select a fraction of the best p’s in P to add to PNew. Call this new generation.

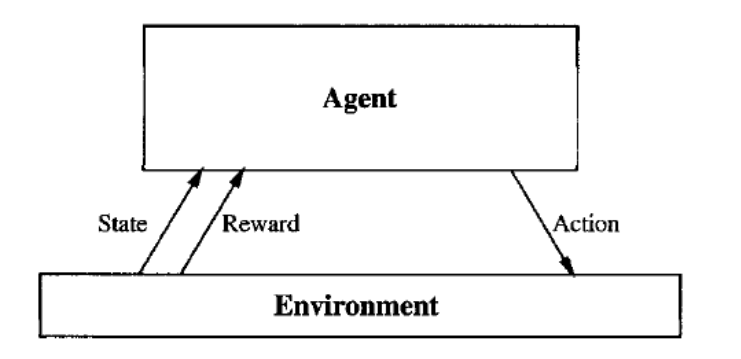
# Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these PNew crossover sites are exchanged thus creating a completely new individuals(offspring).

# 

# 

**Reinforcement Learning**

* It is based on reward and policy-based method.
* Reinforcement Learning (RL) is the science of decision making. It is about learning the optimal behaviour in an environment to obtain maximum reward.
* Agent performs actions in the environment, based on the actions done by the agent, the agent receives reward/penalty.
* Based on the policy that was received to the agent, it performs well in the next action.
* Ex: Q-Learning Algorithm

****

**Intro:**

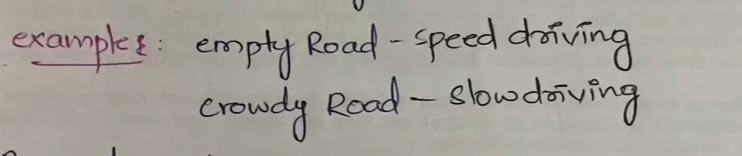
* Consider building a learning robot. The robot, or agent, has a set of sensors to observe the state of its environment, and a set of actions it can perform to alter this state.
* Its task is to learn a control strategy, or policy, for choosing actions that achieve its goals.
* The goals of the agent can be defined by a reward function that assigns a numerical value to each distinct action the agent may take from each distinct state.
* This reward function may be built into the robot, or known only to an external teacher who provides the reward value for each action performed by the robot.
* The task of the robot is to perform sequences of actions, observe their consequences, and learn a control policy.
* The control policy is one that, from any initial state, chooses actions that maximize the reward accumulated over time by the agent.
* In reinforcement learning it doesn’t have any training data.
* Instead, the trainer provides only a sequence of immediate reward values as the agent executes its sequence of actions.
* The agent during its course of learning experience various different situations in the environment it is in. These are called states.

**Q - Learning Algorithm**

* This is a reinforcement learning algorithm.
* Here Q stands for quality.
* In maze problem, the goal is to move the rat from start to destination
* The maze also contains the blockers or walls in order to stop the rat not to move forward, so that the rat should choose the path which makes it not to stop when it moves from start to destination.
* And, also we should find the minimal optimal path from start to destination
* Terms:

1. Policy: Qπ(st, at)
   * S stands for state, a stands for action, t means time
   * State and action with respect to time

* Certain rules & limits are considered
* Policy is undertaken by the agent
  + Agent observes the state and select the best possible action



1. Reward:

* Based on the action performed by the agent the agent receives reward
* When the action is correct only the agent receives the reward
* Ex: In maze game instead of reaching destination, hitting a wall is not a correct path, so here agent receives penalty
* The agent receives a scalar quantity

1. Penalty:

* It is opposite to reward
* It’s a negative reward
* It’s a wrong action done by the agent
* Ex: Parking vehicle in no parking zone, so penalty should be paid here

Q-Learning mainly depends on 2 factors:

1. Q-function
2. Q-Table

Q-function

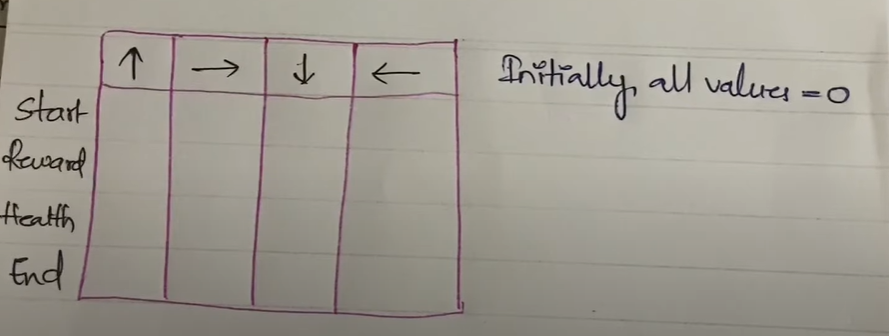
* Qπ(st, at) = E (Rt+1+rRt+2+…+ γnRt+n| St,at)
* This is bellman equation
* γ is discount factor, R is reward

Q-Table

* Combination of states and action

In maze game

* Actions: up, left, down, right
* State: start, end, reward, health

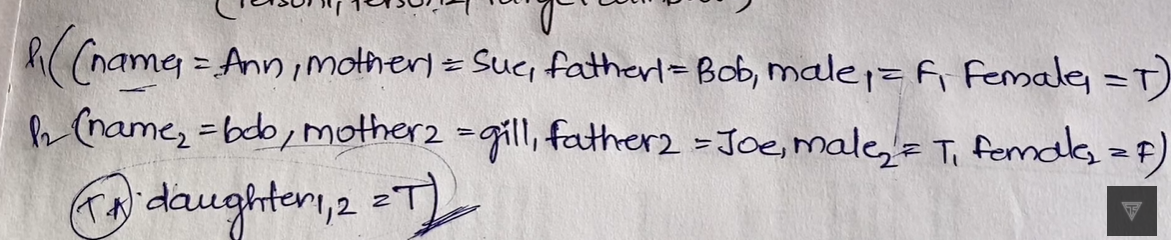
****

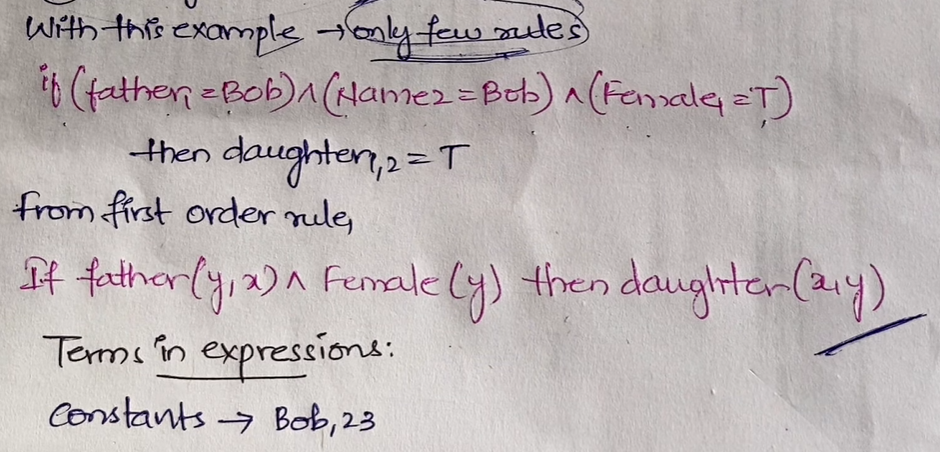
Steps followed:

* Exploration: finding all possible paths
* Exploitation: remove the unwanted paths, and find the best possible path
* Initializing Q table
* Choose action
* Perform action
* Measure reward
* Update Q-table

**Learning sets of First-Order rules**

* Implemented with the help of PROLOG.
* First order logic is much expressive than prepositional logic.
  + It allows a finer grain of specific and reasoning
* Ex: daughter (x, y)
  + Where (x, y) are {Name, Mother, Father, Male, Female}
* Training examples are represented as (Person1, Person2, target attribute)





FOIL ALGORITHM:

* First Order Inductive Learning
* Extension of Sequential Covering algorithm
* 1 similarity, 1 difference
* Foil searches its hypothesis using 2 loops.

1. Outer Loop: Disjuctive(v)
2. Inner Loop: Conjunctive(^)

* From most general to more specific