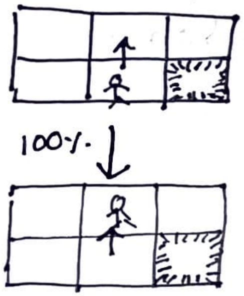
16. Markov Decision Processes : (MDP)

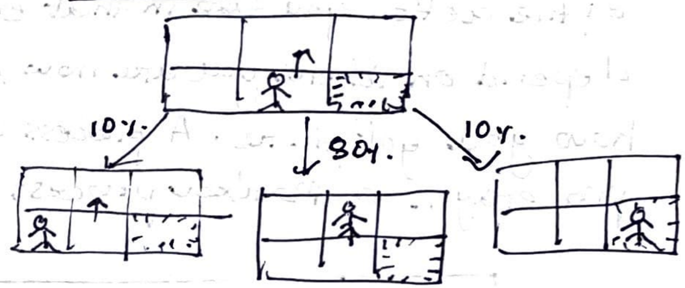
Deterministic Search :



* If the agent wants to go up, it will surely go up.

- For a particular input, the computer will always give the same output
- Can solve the problem in polynomial time.
- Can determine the next step of execution.

Non-Deterministic Search : (Stochastic)



* If the agent wants to go up. 80% chances are he will go up. 10% ~~chang~~ chance that he will go right & 10% that he will go left.

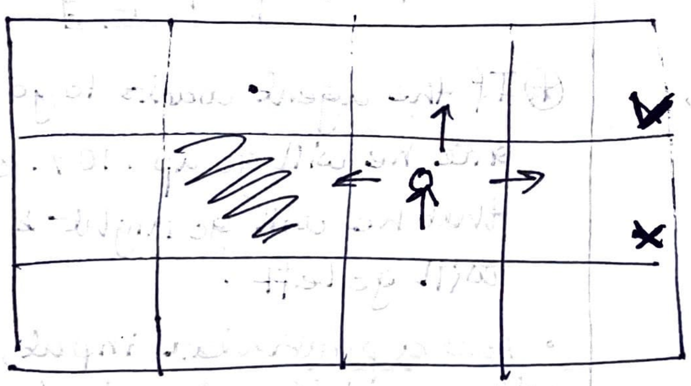
- For a particular input, the computer will give different outputs on different executions.
- Cannot solve in polynomial time.
- Cannot determine the next step of execution due to more than one path the algorithm can take.

Markov property:

A stochastic process has the Markov property if the conditional probability distribution of future states of the process (conditional on both past and present states) depends only upon the present state, not on the sequence of events that preceeded it.

A process with this property is a MARKOV DECISION PROCESS

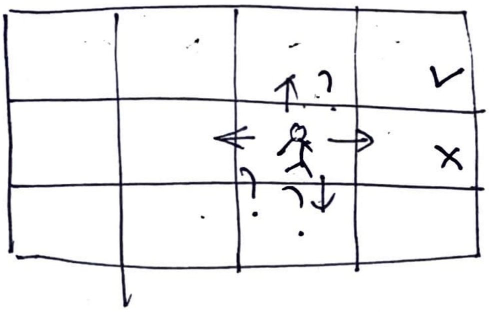
In markov property your future state, not just your choice, your choice and the environment, the results of the action you take in that environment will only depend on where you are now, It will not depend on how you got there. A process which has this property is a Markov process.



- Might go up, left or right.
- Does not matter where it started.
- The probability of up, left right will always be the same if he is in this state now.

Markov Decision Processes (MDP):

Provide a mathematical framework for modelling decision making in situations where outcomes are partly random and partly under the control of a decision maker.



Agent doesn't know what to do.
Thus it applies a framework.
Uses MDP.

From Bellman Equation:

$V(s) = \max_a (R(s,a) + \gamma V(s'))$ but due to stochasticity we don't know what s' will be

up = 80% or 0.8
left = 10% or 0.1
right = 10% or 0.1

Thus we replace that with the expected value of the next step:

up $V(s_1')$, $V(s_2')$ left, $(V s_3')$ right

$V(s) = \max_a (R(s,a) + \gamma V(s'))$

$\Rightarrow$ " " " $+ \gamma(0.8 \times V(s_1') + 0.1 \times V(s_2') + 0.1 \times V(s_3'))$

$\Rightarrow$ " " " $+$ average of this.

$$\boxed{V(s) = \max_a \Big(R(s,a) + \gamma \sum_{s'} \underbrace{P(s,a,s')}_{\text{Probability}} V(s')\Big)}$$

→ New Bellman Eqn

→ This is the framework used by MDP.