Robot Learning

Assignment - 2

Member 1

Name: Hojun Lim
Mat. No: 3279159

Member 2

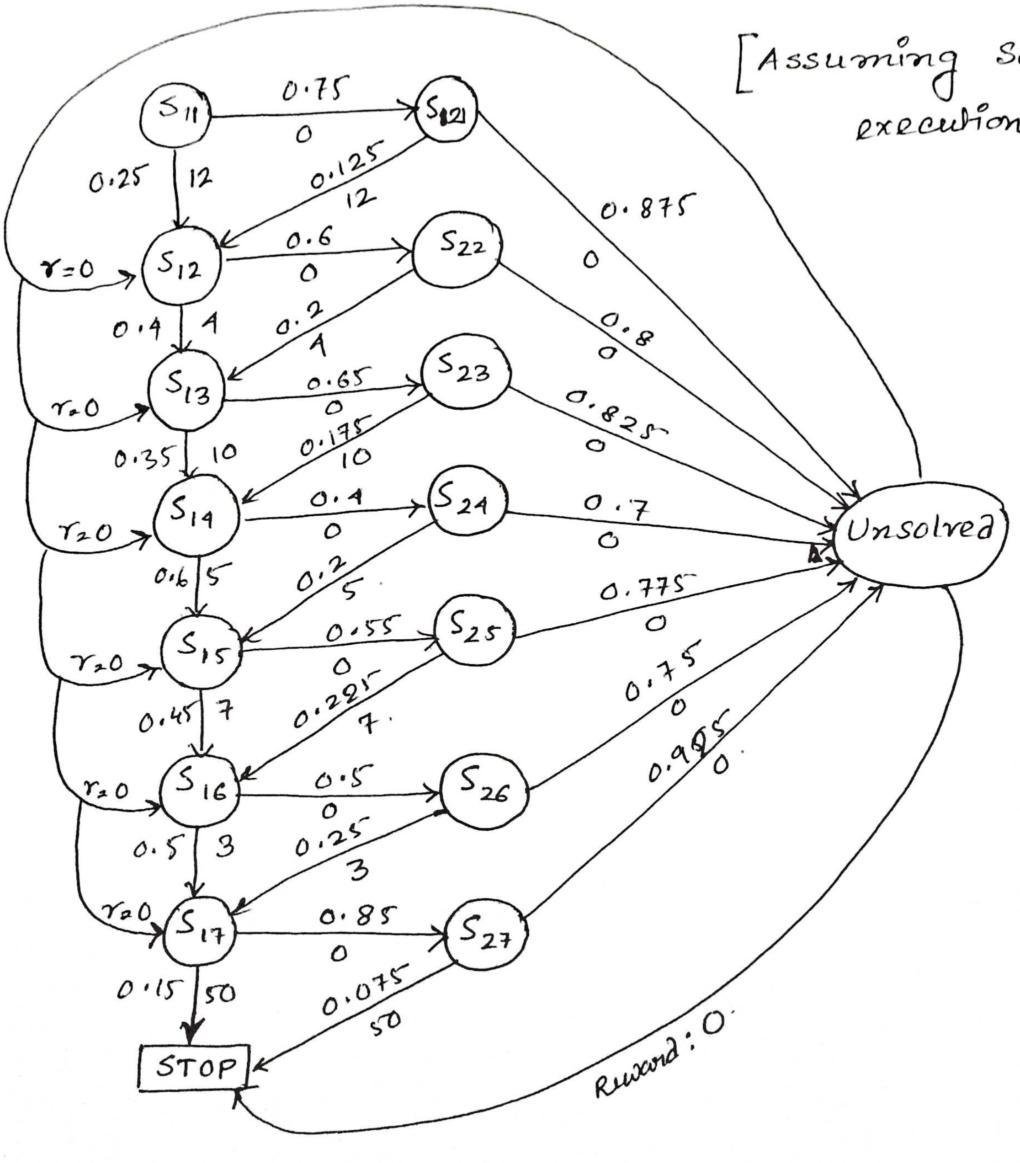
Name: Kajaree Das
Mat. No: 3210311

2.1 states : $\{S_{1i}, S_{2i}\}$, unsolved, stop }

S_{1i} → 1^{st} Attempt on i^{th} problem / task

S_{2i} → 2^{nd} Attempt on i^{th} problem / task

task (i)	rewards (r_i)	p_{1i}	$1-p_{1i}$	p_{2i}	$1-p_{2i}$
1	12	0.25	0.75	0.125	0.875.
2	4	0.4	0.6	0.2	0.8
3	10	0.35	0.65	0.175	0.825
4	5	0.6	0.4	0.3	0.7
5	7	0.45	0.55	0.225	0.775
6	3	0.5	0.5	0.25	0.75
7	50.	0.15	0.85	0.075	0.925



[Assuming sequential execution]

2.5

Example -1:

Suppose, we have a bag of r red balls, g green balls and b black balls. We start drawing one ball a day without replacing it back in the bag.
Then, the probability that we get a red ball (say) tomorrow, will not only depend on the ball we draw today, but also on the color of the ball that was drawn yesterday.
So, the Markov assumption: “the conditional probability distribution of future states of the process depends only upon the present state. And given the present, the future does not depend on the past.”; does not hold for our process.
But, if in the above experiment, if we replace the ball after each draw, the process will have Markov Property.

Example - 2:

Consider a frog jumping from a lotus leaf to a lotus leaf on in a small forest pond. Suppose that there are N leaves so that the state space can be described as S = {1, 2, . . . , N}. The frog starts on leaf 1 at time n = 0, and jumps around in the following fashion: at time 0 it chooses any leaf except for the one it is currently sitting on (with equal probability) and then jumps to it. At time n > 0, it chooses any leaf other than the one it is sitting on and the one it visited immediately before (with equal probability) and jumps to it. The position {Xn}n∈N of the frog is not a Markov chain. Indeed, we have P[X3 = 1|X2 = 2, X1 = 3] = 1/N – 2 ,

while P[X3 = 1|X2 = 2, X1 = 1] = 0. A more dramatic version of this example would be the one where the frog remembers all the leaves it had visited before, and only chooses among the remaining ones for the next jump.

Making a non-Markov chain into a Markov chain:

The problem is that the frog has to remember the number of the leaf it came from in order to decide where to jump next. The way out is to make this information a part of the state. In other words, we need to change the state space. Instead of just S = {1, 2, . . . , N}, we set S = {(i1, i2) : i1, j2 ∈ {1, 2, . . . N}}. In words, the state of the process will now contain not only the number of the current leaf (i.e., i1) but also the number of the leaf we came from (i.e., i2). There is a bit of freedom with the initial state, but we simply assume that we start from (1, 1). Starting from the state (i1, i2), the frog can jump to any state of the form (i3, i2), i3 6= i1, i2 (with equal probabilities). Note that some states will never be visited (like (i, i) for i 6= 1), so we could have reduced the state space a little bit right from the start.

Exercise 2.2

Calculated the probability of passing the exam: 8.9619796875%

Exercise 2.3

$Expected\ Reward\ of\ the\ Policy\!:{}'Sequential\ Order\ \pi_A{}'is\ 31.021249999999995.$
$Expected\ Reward\ of\ the\ Policy\!:{}'Increasing\ Order\ \pi_B{}'is\ 31.02125.$

Exercise 2.4

'Increasing Reward Order', 'Decreasing Difficulty Order', 'Decreasing Reward Order' policies showed slightly better expected rewards of 31.02125000000001, 31.021250000000002 and 31.021250000000002 respectively.