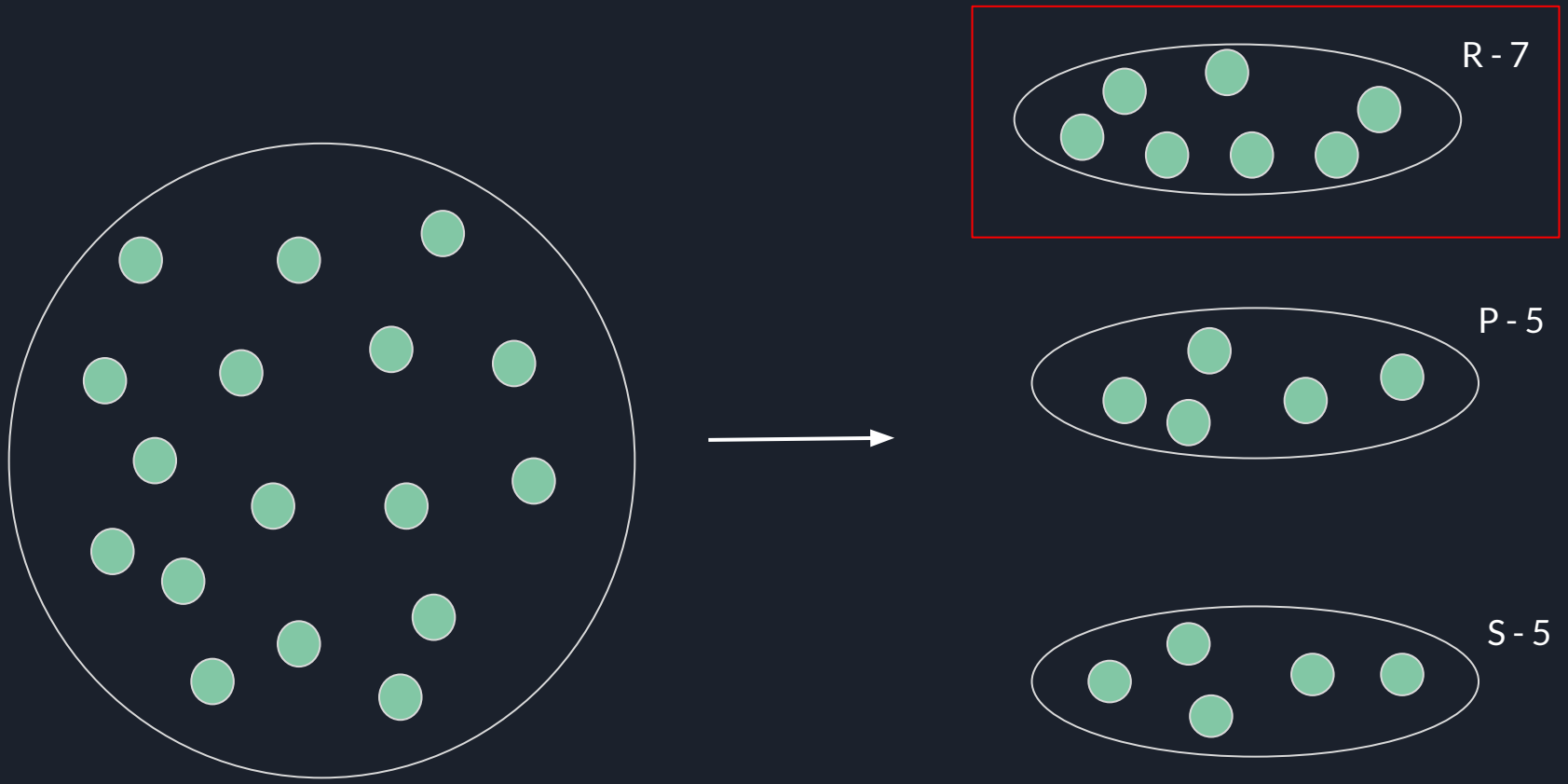
Q2 Eval

Approach

Each one represents a state in the dfa

Now I should be able to win maximum cases in 1st round , regardless of where It starts , so i should defend the RPS which has maximum States:

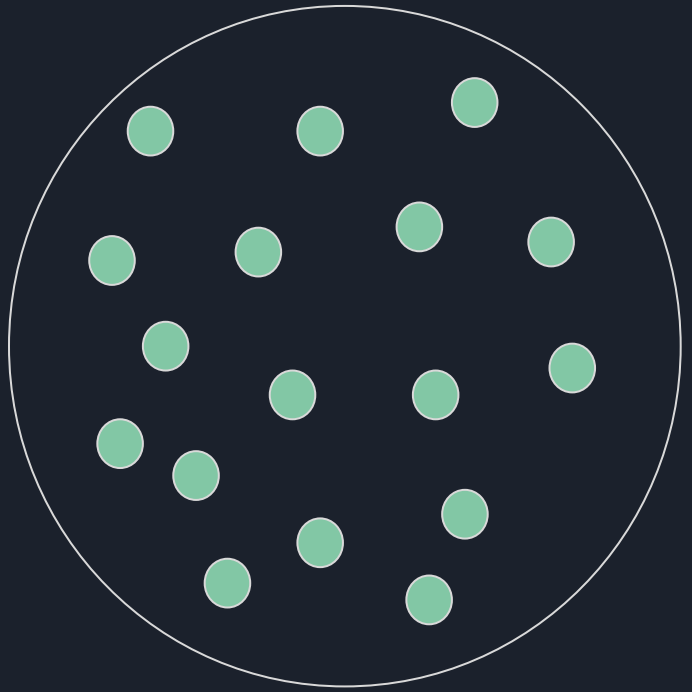


OUR DFA:

Since R is maximum , we have to win against R

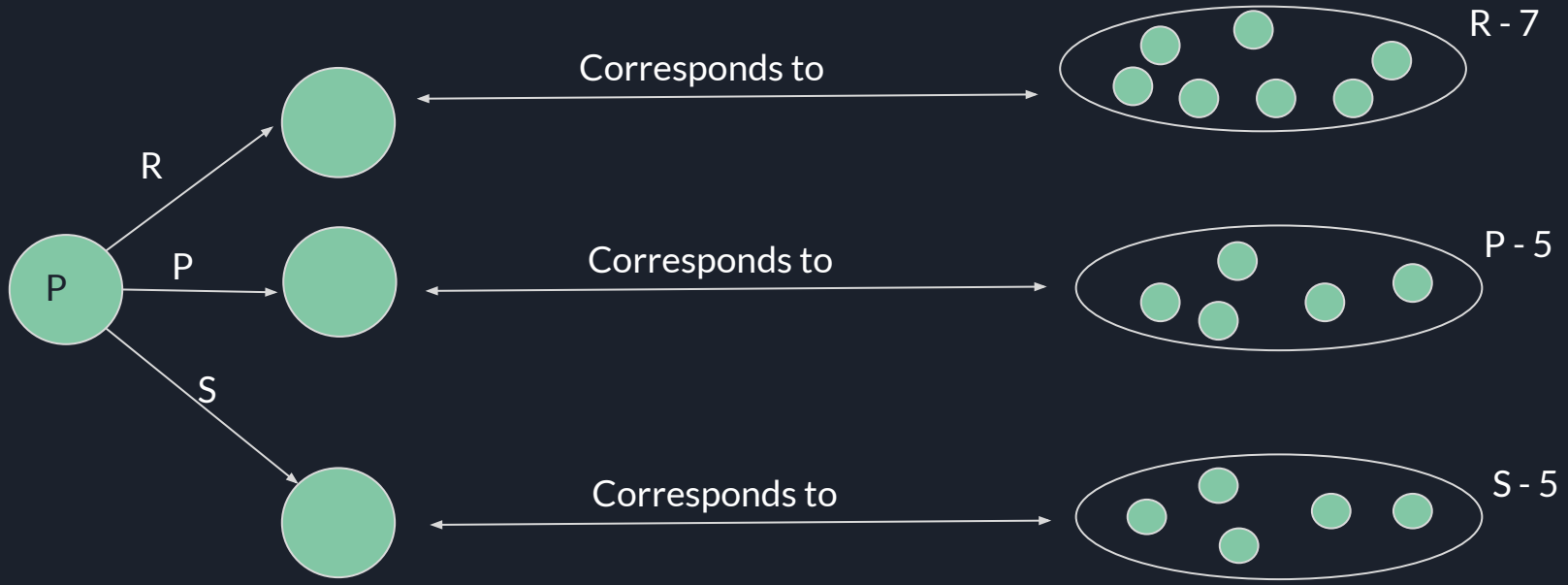
P

Corresponds to



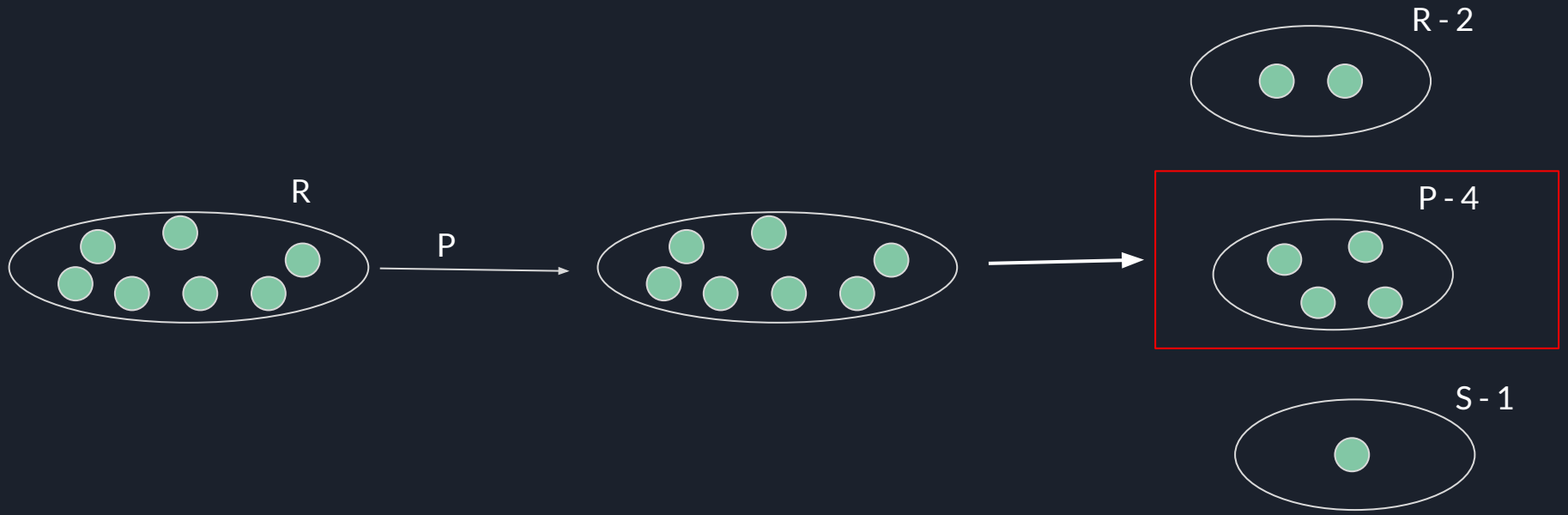
OUR DFA:

Transitions depend on the what opponent state was in the previous Round.



Given DFA:

Since our move in previous step was P , apply P transitions to each of the elements in each of the subsets R,S,P



OUR DFA:

For the subset R

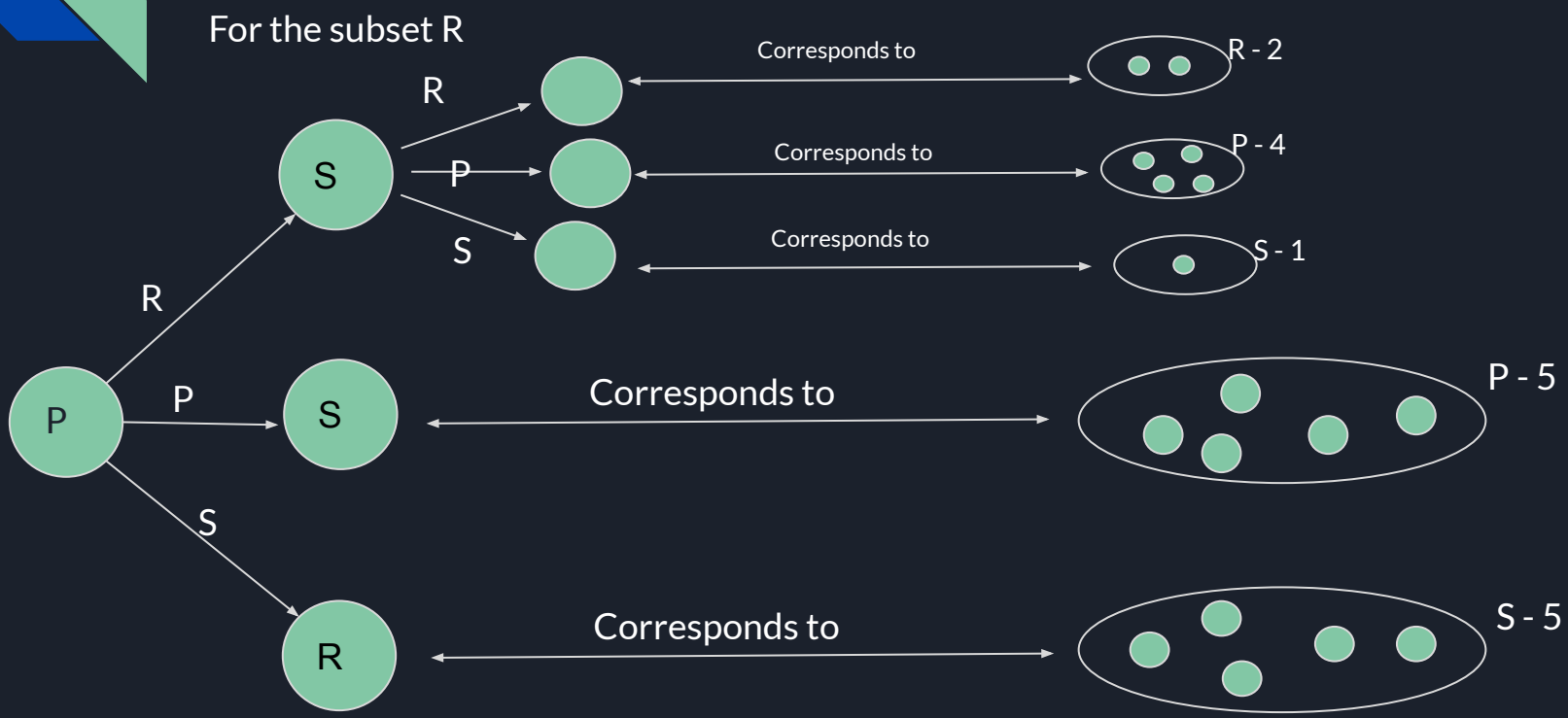
OUR DFA:

For the subset R

OUR DFA:

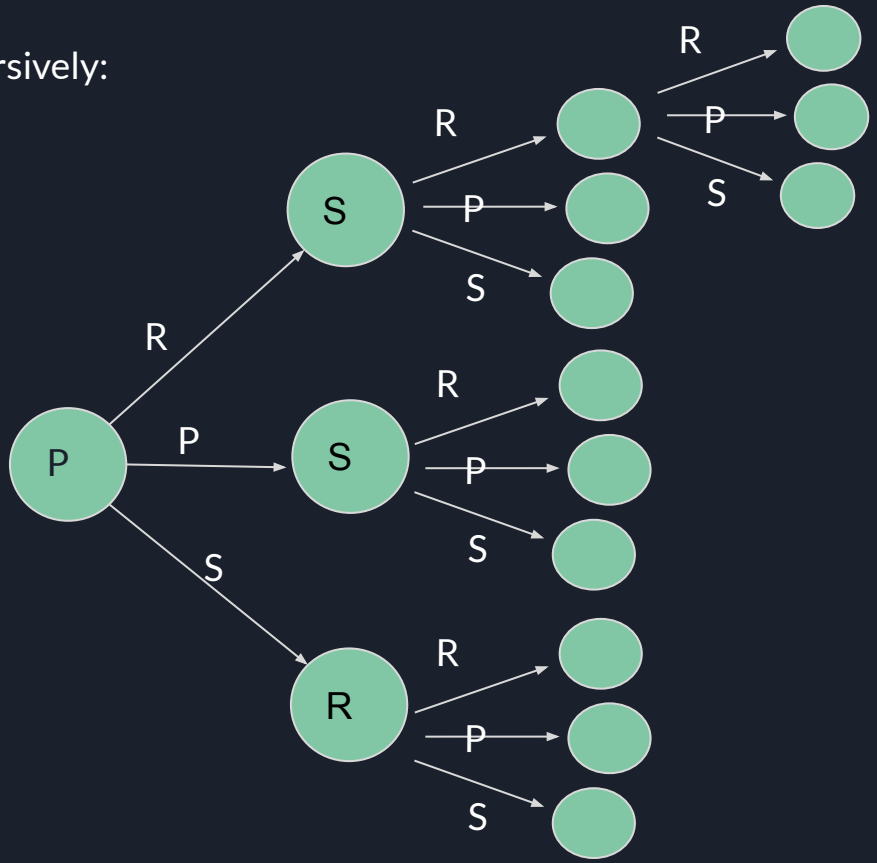
For the subset R

OUR DFA:


For the subset R
R
P
S
Corresponds to
R - 2
Corresponds to
P - 4
Corresponds to
S - 1
R
P
S
P
Corresponds to
P - 5
S
R
Corresponds to
S - 5

OUR DFA:

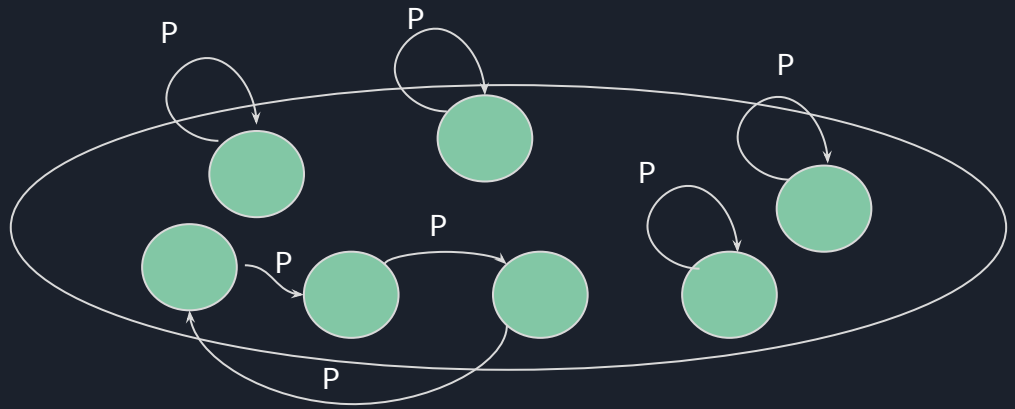
Continue Recursively:



Where Should I Stop?

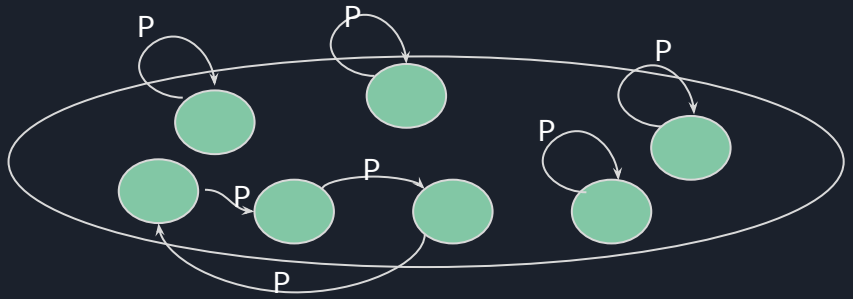
Idea 1: somewhere along the lane , I will endup at a stage where the subset has only 1 node in it , and thereafter i will be able to win all the rounds.

But , what if this happens:



Where Should I Stop?

But , what if this happens:



The set on applying transition P , keeps looping in itself, i.e idea 1 doesnt work.

Idea 2: keep track of what all subsets you have already seen previously , and if you have seen some subset previously , simply point it to its corresponding state in output DFA.

Idea 2 contd....

Since there are only 2^n subsets , it definitely have to endup in a loop somewhere or the other. We detect that and proceed accordingly.

fps.txt

I Couldnt think of any testcase manually ,
Any testcase i thought of was getting 99.99 in my approach ,
So i used random generator , and ran my programme on 100 - 200 testcases , to obtain a fps that gave minimum score with my code.
(clearly no restrictions of this sort in assignment were given :))

Progress since prefinal

I implemented my full code in prefinal itself, But i had the following idea:

I was only using a maximum of available states , and at each step i was not considering the fact that we should also get a higher score in next step. (i.e if say i went with a state where my dfa wins lesser states in current round , but higher wins in next rounds

I.e in few cases:

(DFA against 2nd maximum + DFA against corresponding 1st maximum in next round > DFA against 1st maximum + DFA against corresponding 1st maximum in next round)

I tried to code up my logic for this, But it gave a very bad score. So i ended up submitting the prefinal version of the code.