

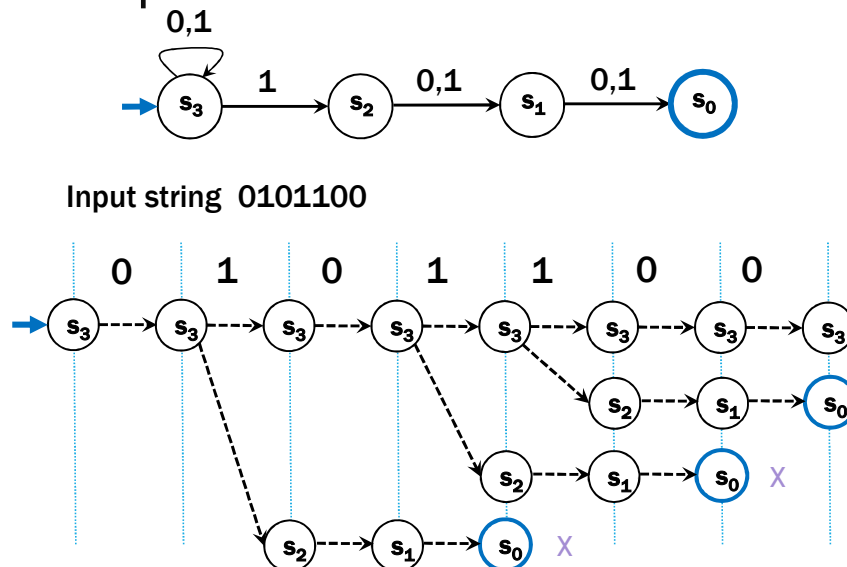
NFA that recognizes "binary strings with a 1 in the third position from the end"

"Perfect Guesser": The NFA has input x , and whenever there is a choice of what to do, it **magically** guesses a transition that will eventually lead to acceptance (if one exists)

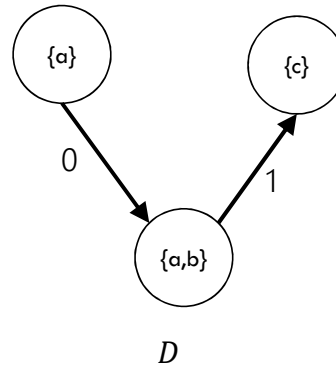
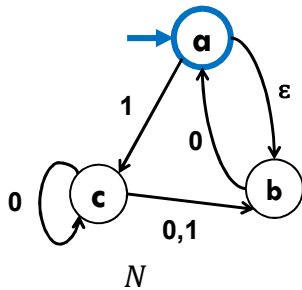
Perfect guesser view makes this easier.

Design an NFA for the language in the title.

Parallel Exploration view of an NFA



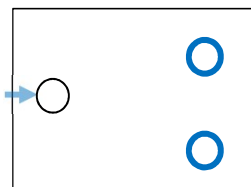
An example (starting point)



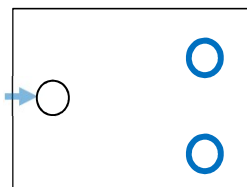
Let $P(A)$ be "There is an NFA whose language is the same as the language for A ."

Inductive Hypothesis: Let A, B be arbitrary regular expressions. Suppose $P(A)$ and $P(B)$.

Inductive Step: **Case AB**



N_A



N_B

Want a machine that accepts exactly strings matched by AB .