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.

An example (starting point)

```
N
```

```
D
```

```
{a, b}
```

```
{a}
```

```
{c}
```
Let \( P(A) \) be “There is an NFA whose language is the same as the language for \( A \).”

Base Cases:

\[ \emptyset \]

\[ \varepsilon \]

\[ a \ (a \in \Sigma) \]

Let \( R \) be a regex not covered by the base cases. By the exclusion rule, \( R = A \cup B \) or \( AB \) or \( A^* \) from some regexes \( A, B \).

Inductive Hypothesis: Suppose \( P(A) \) and \( P(B) \).

Inductive Step: **Case 2: \( AB \)**

Want a machine that accepts exactly strings matched by \( AB \).