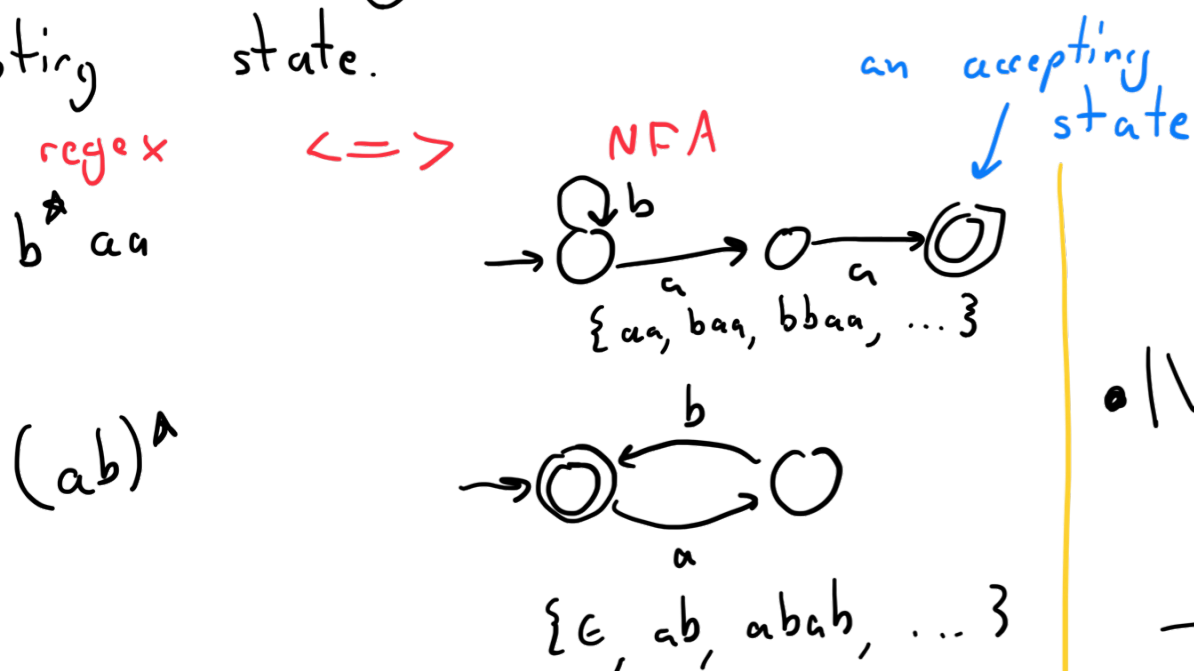
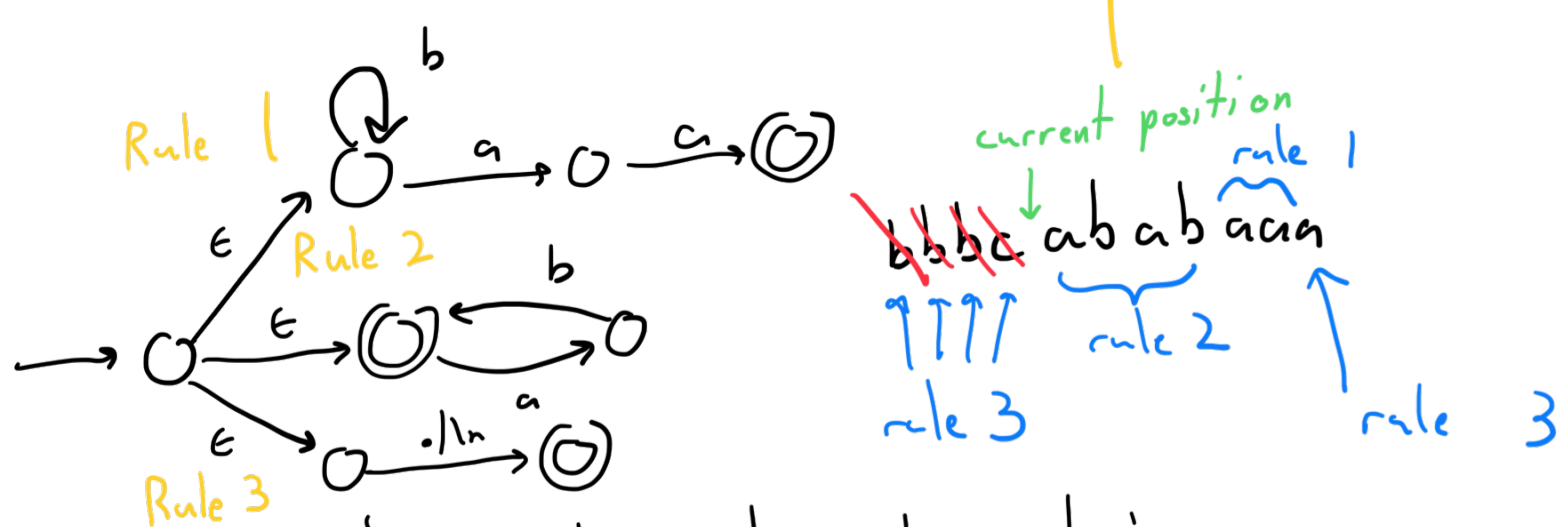
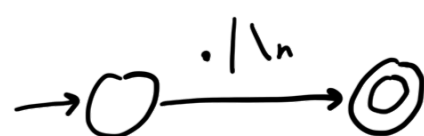


From each regex, Flex constructs an equivalent nondeterministic finite automaton (NFA) that accepts any string that the regex does.

- An NFA accepts a string if we can traverse the automaton, consume the entire string, and end on an accepting state.



$\cdot | \backslash n \{ ECHO; \}$



- Flex remembers the longest string consumed when we last visited an accepting state (greedy).
- Flex backtracks to the previous longest match after all branches of computation "die".
- Then Flex resets from the new current position in the input string.

Big Idea)

- Flex scans left-to-right & never skips characters.
- Flex will backtrack and give you back your consumed chars if all branches of computation die, up to the longest substring matched.
- Once a rule matches, those chars are never returned (unless we use lookahead ✓)

Tie-breaking. Rule selection order is:

1. Earliest starting position.

banana
match

2. Longest match

"foobar" ← gets matched on input "foobar"
"foo"

3. Earliest rule in the .l file

"foo" ← gets matched on input "foo"
[a-z]{3}