



NFA has lambda transition

Example, string ends with 01 or 10

Right Linear Grammar to NFA:

Each non-terminal symbol is a state.

Each terminal symbol is a transition.

For production rule A -> x, create accepting state f and have delta (a, x, f).

Example:

|  |  |
| --- | --- |
| S → aS S → bB S → cC S → ε B → bB B → cC B → ε C → cC **C → ε** | https://swaminathanj.github.io/fsm/images/rgastarbstarcstar10.png |

Regular Expression to NFA:

**The process of converting a regular expression into an ∈-NFA is as follows:**

1. Create a single start state for the automaton, and mark it as the initial state.
2. For each character in the regular expression, create a new state and add an edge between the previous state and the new state, with the character as the label.
3. For each operator in the regular expression (such as “\*” for zero or more, “+” for one or more, and “?” for zero or one), create new states and add the appropriate edges to represent the operator.
4. Mark the final state as the accepting state, which is the state that is reached when the regular expression is fully matched.



**Step 1: Convert the given NFA to its equivalent transition table**

**Step 2: Create the DFA’s start state**

**Step 3: Create the DFA’s transition table**

**Step 4: Create the DFA’s final states**

**Step 5: Simplify the DFA**





