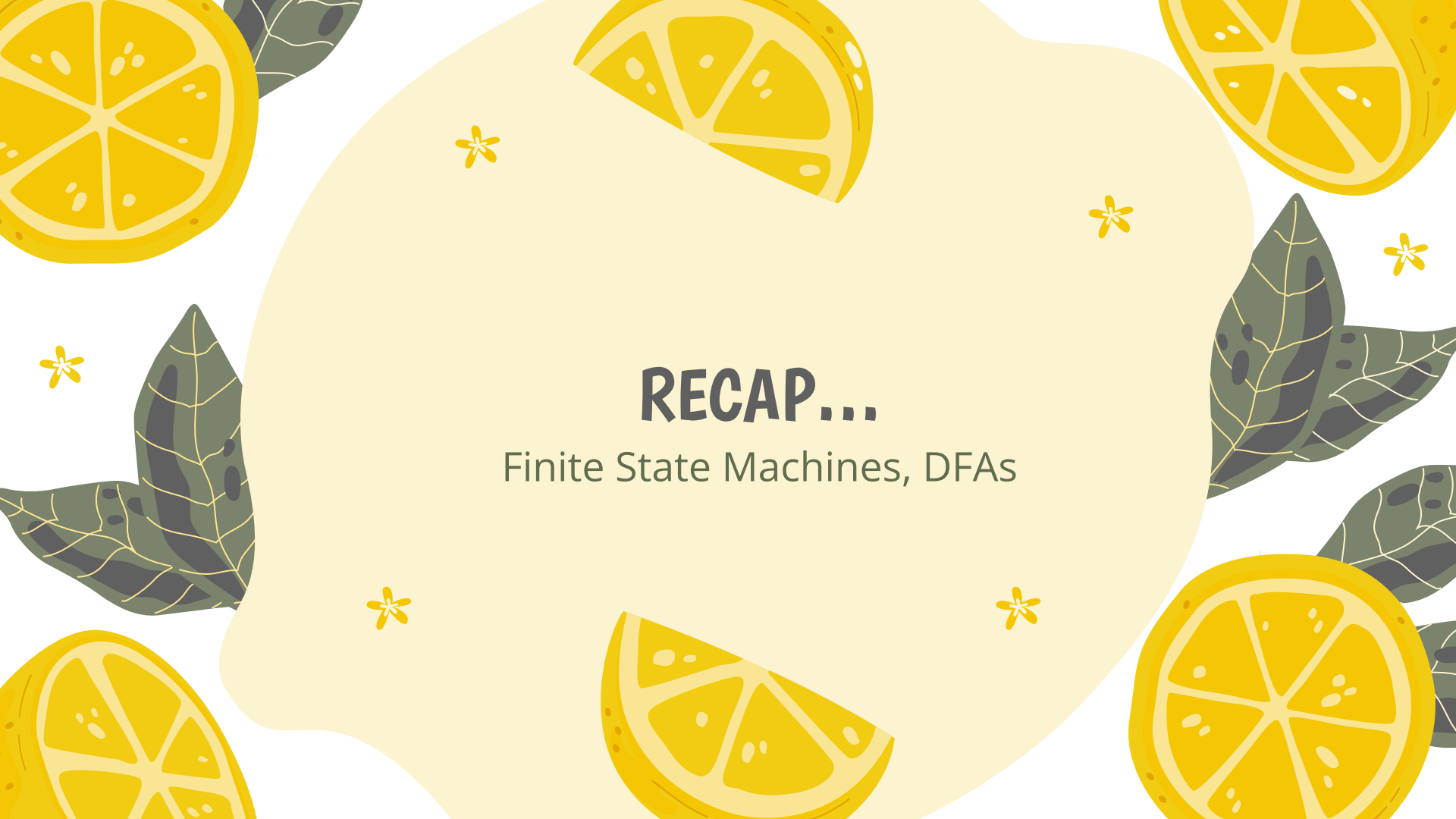
NFAs

CSE 311 25au

"when life gives you lemons...
nondeterministically choose the sweetest path"



RECAP...

Finite State Machines, DFAs

Review: Finite State Machines

- Simple model of a computer!
- Takes an input and decides whether to accept it
- Finite!

Review: DFAs

- **Deterministic**:
  - Every state has exactly one transition for each valid character
  - For any input string, the DFA has one and only one possible computation path.
- **Finite**: Finite set of states

NFAs

Nondeterministic Finite Automaton

NFAs

Nondeterministic:

- A state may have ***0, 1, or multiple outgoing edges*** with the same character.
- ***ε-transitions***: The machine may move along an ε-edge without consuming input.
- ***Dead branches***: If a path cannot continue, that branch of computation stops.

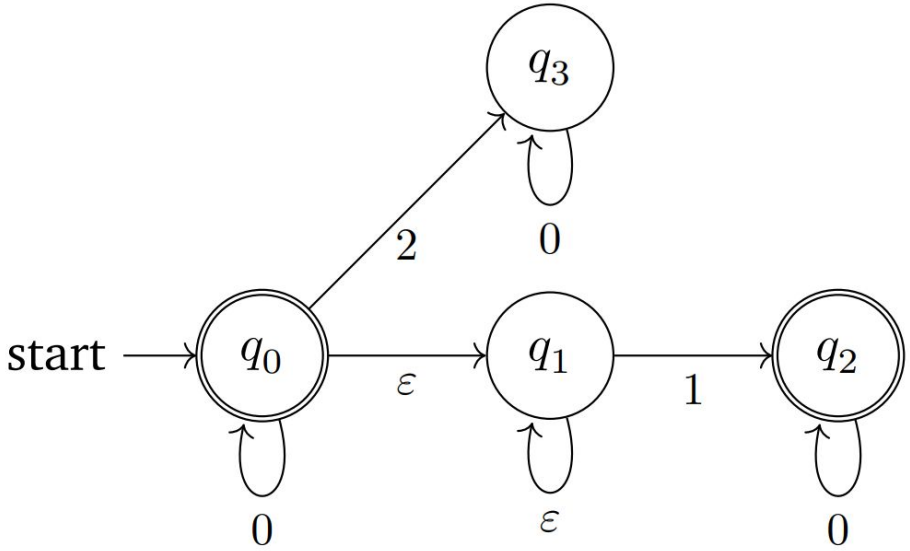
Finite: Finite set of states

💡 The machine explores all possible paths in parallel and accepts if one ends in a final state.

Let's Practice!

Nondeterministic Finite Automaton

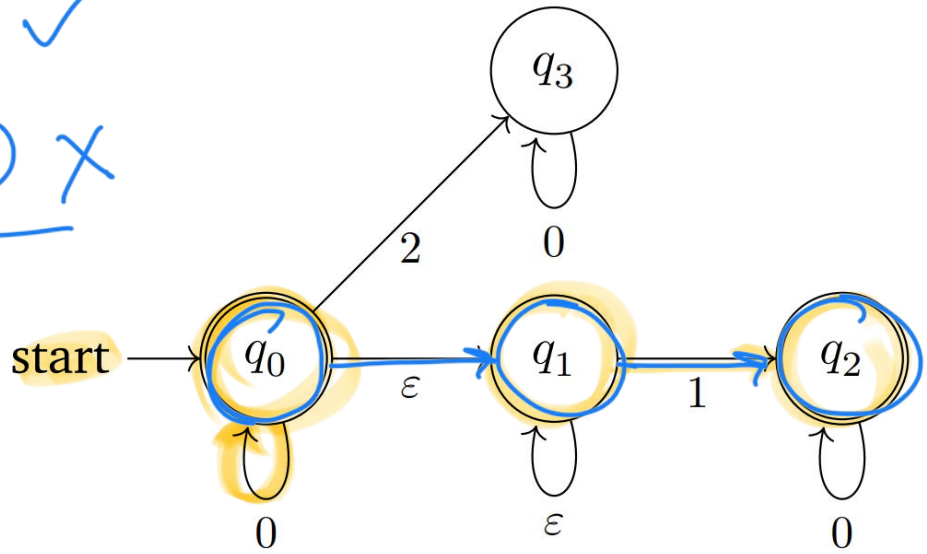
What language does this NFA accept?



What language does this NFA accept?

0 0 1 ✓

1 1 0 X



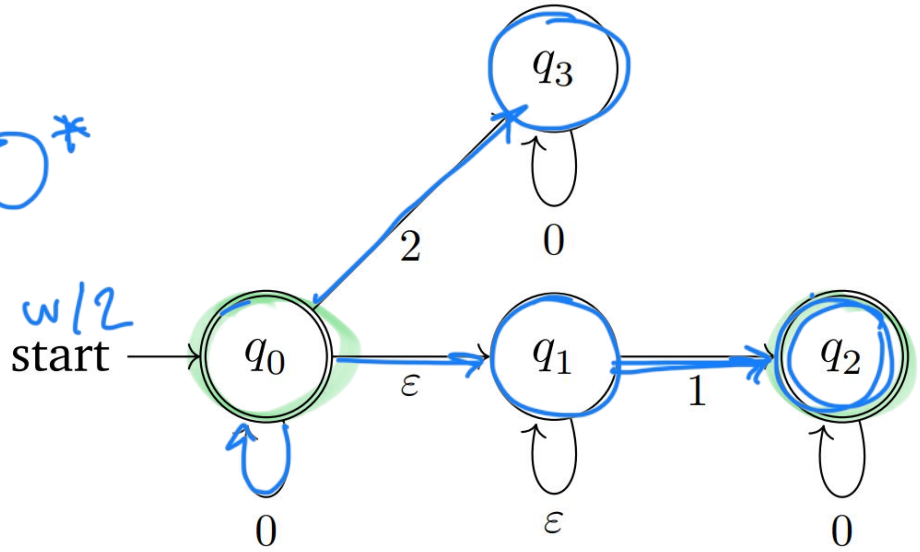
- states may have multiple (or no) options for a character
- ε allows for “jumps” without an additional character

What language does this NFA accept?

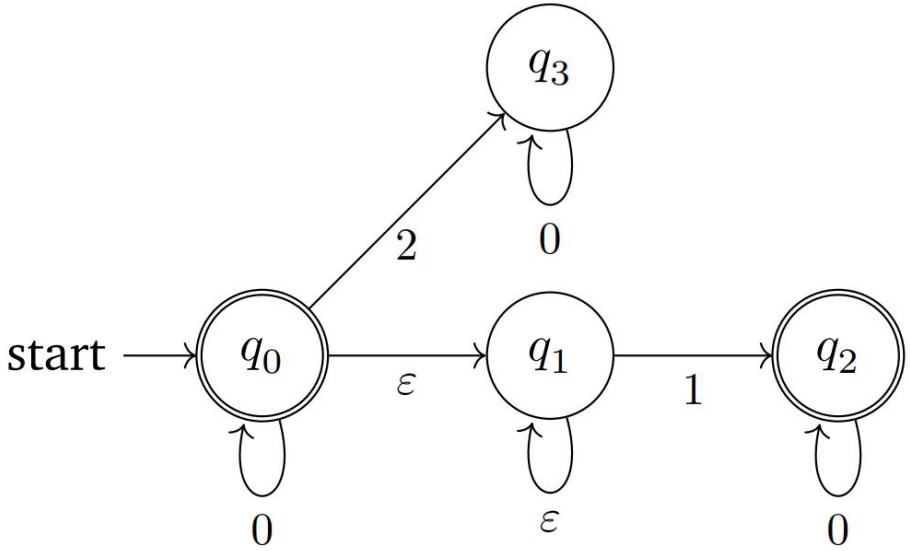
✓ 0^*

✓ 0^*10^*

X anything w/ 2



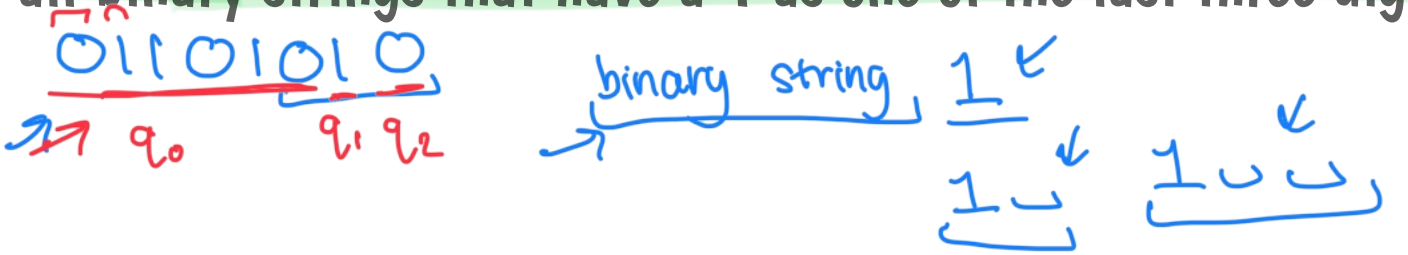
What language does this NFA accept?

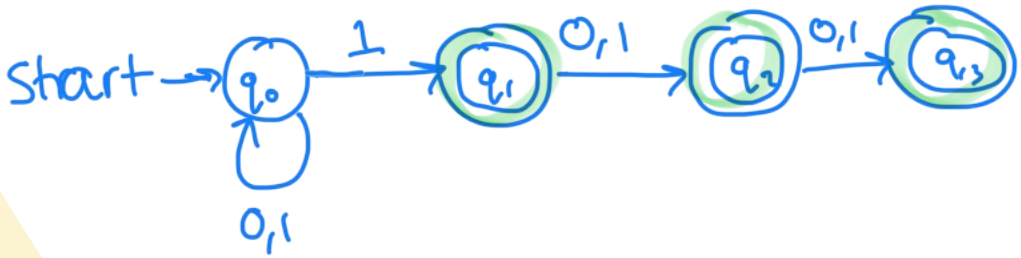


All strings of only 0's and 1's, not containing more than one 1.

Create an NFA for the language:

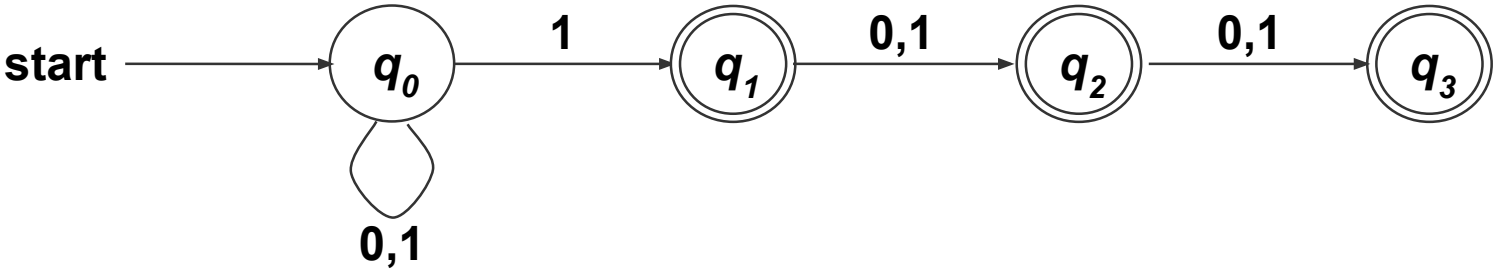
"all binary strings that have a 1 as one of the last three digits".





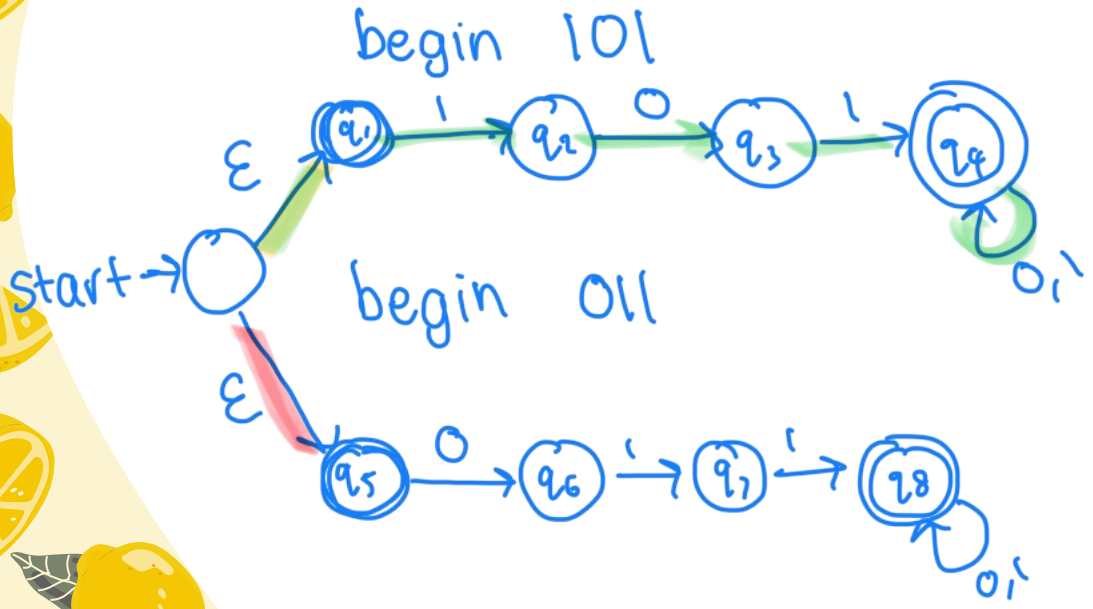
Create an NFA for the language:

"all binary strings that have a 1 as one of the last three digits".



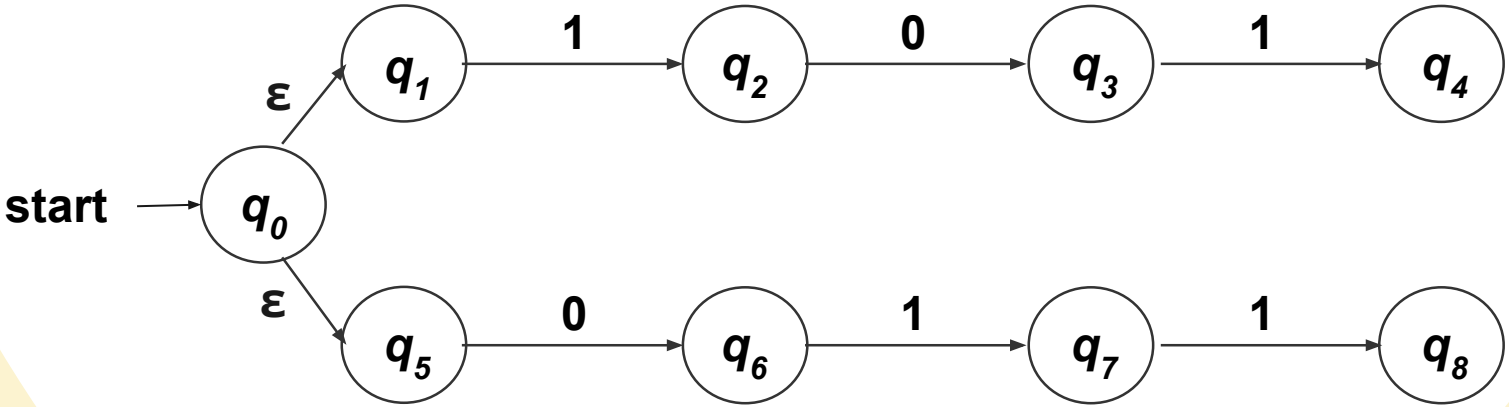
Create an NFA for the language:

"all binary strings that begin with 101 OR 011".



Create an NFA for the language:

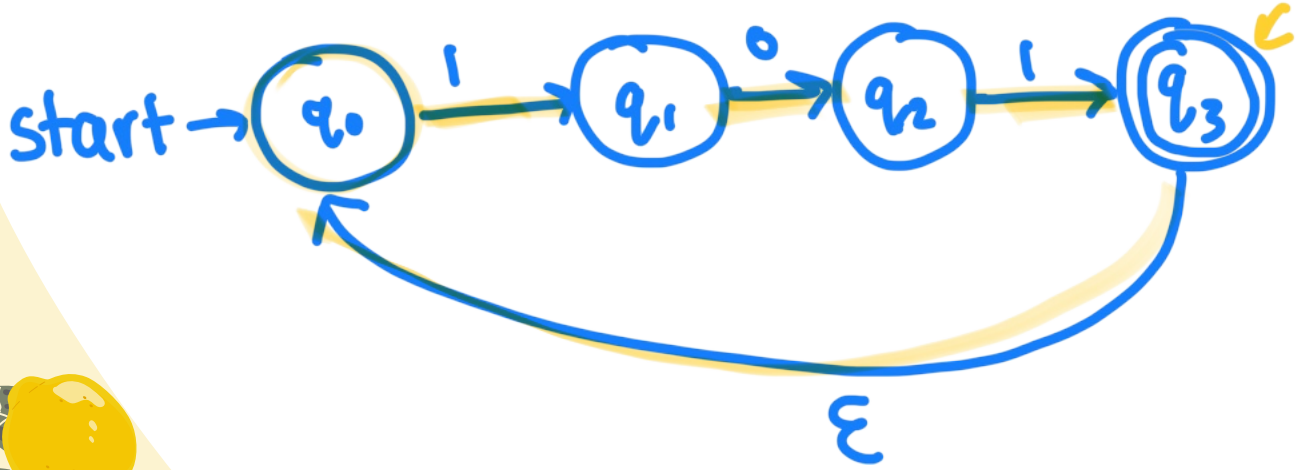
"all binary strings that begin with 101 OR 011".



Create an NFA for the language:

"all strings composed of 0 or more occurrences of 101 only".

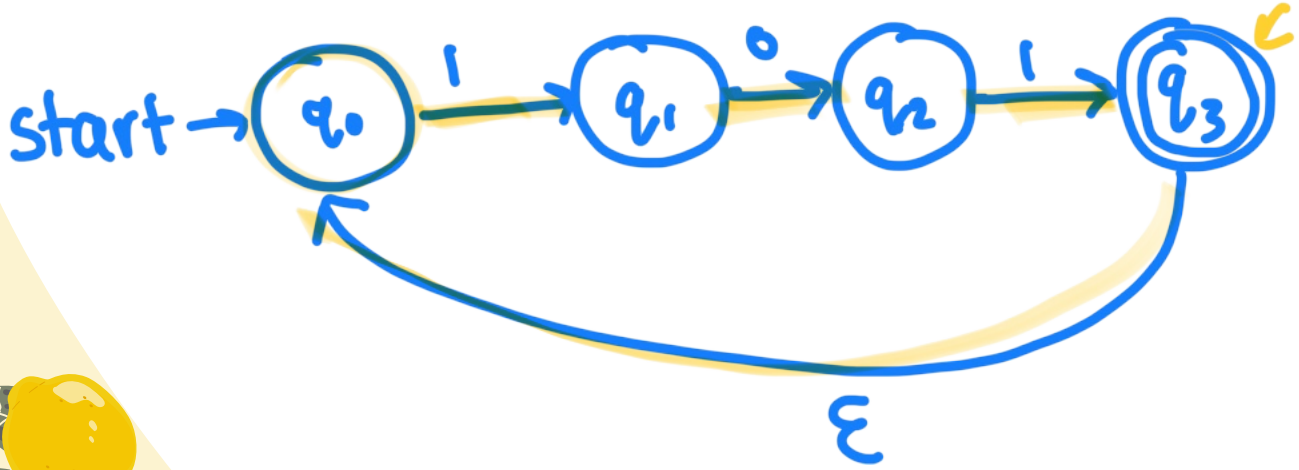
"" "101" "101101101" (101)*



Create an NFA for the language:

"all strings composed of 0 or more occurrences of 101 only".

"" "101" "101101101" (101)*

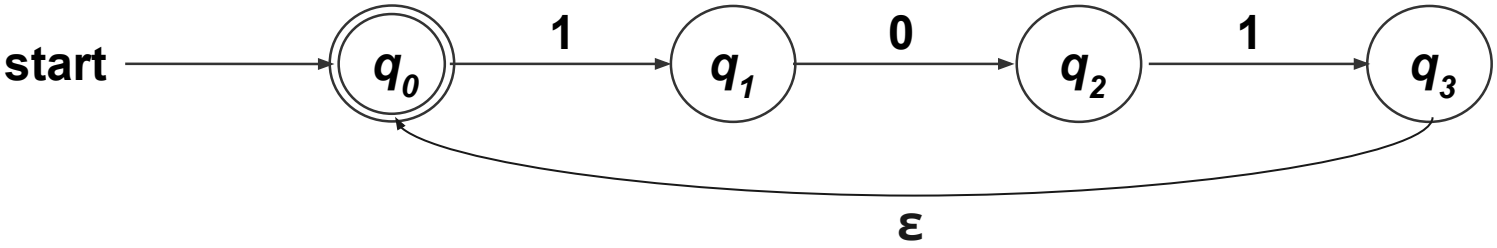


Create an NFA for the language:

"all strings composed of 1 or more occurrences of 101 only".

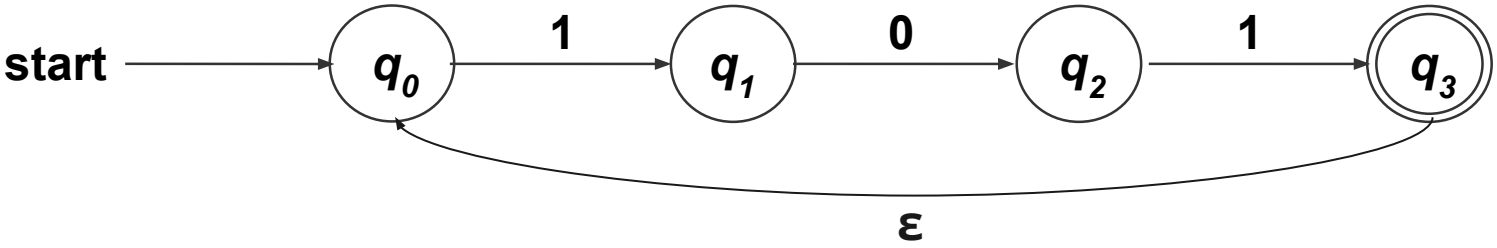
Create an NFA for the language:

"all strings composed of 0 or more occurrences of 101 only".



Create an NFA for the language:

"all strings composed of 1 or more occurrences of 101 only".



⚠️ Remember ⚠️

- Just like DFAs, NFAs must still have one start state, and may have multiple final states
- Build small NFAs for simple parts, then connect them.
  - the ε-transitions are great for “gluing” together parts
- Can be helpful to think about the regex before building NFA
  - ε-transition for a loop: (...)* in regex
  - ε-transition to go between options: ...|...|... in regex
- If the machine needs to “try multiple options,” can add multiple outgoing edges

Good luck!

and happy thanksgiving! :))