Ella

Worksheet #3
Non-Deterministic Finite Automata

1. What extra symbol is added to NFA alphabets?

The empty string ϵ

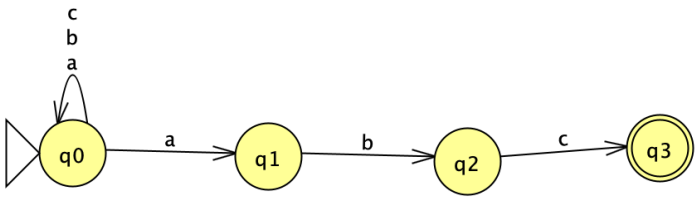
2. Are there any languages that NFAs can accept that no DFA can accept? If so, give an example. If not, then describe why not.

No, because a language can be accepted by a DFA if and only if it can be accepted by an NFA, and any NFA can be converted to a DFA.

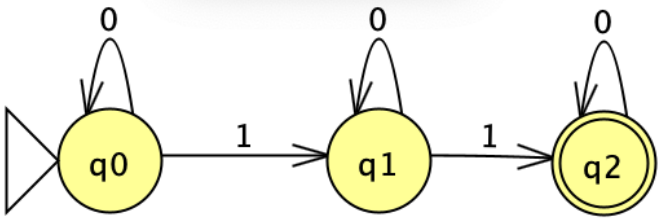
3. Carefully describe the differences between DFA transition functions and NFA transition functions.

DFA transition function results are a single state, while NFA transition function results are a subset of length 0+ of all of the sets. Each state that you feed into the NFA transition function is a subset from Q', where Q' is the powerset of Q, rather than the single state that is fed into the transition function of DFAs.

4. Draw a diagram for an NFA that accepts the language A = {w: w ends with the symbols abc} when using Σ = {a, b, c}.



5. Draw an NFA diagram for an NFA that accepts the language A = {w: w contains exactly two 1s} when using Σ = {0, 1}.



6. Suppose you have a regular language, L, that is accepted by a DFA that has n states. Further suppose you have a string of length 10n that is accepted by the DFA. Give a high-level overview of a proof that there must be strings longer than 10n that are accepted by the DFA.

> We know that we have a really long string compared to the memory of our DFA (the number of states represents the memory of the machine). We have a relatively small amount of memory with respect to the length of the input string that is accepted, so we know we have to reuse the memory. We know we wouldn't be able to keep track of each of the letters separately, so we must be reusing memory. Reusing memory means going back to some other state, so part of the computation is getting redone (there's a loop or cycle). It is very important that the accepted string overruns the capacity of the machine. If we have a loop, we can go through it more times, so any strings longer than 10n could be accepted if they were to repeat any cycles. We can make it shorter by just not going through the loop and directly to an accept state.

7. Given an alphabet of $\Sigma = \{a, b, c\}$, give five example strings in the language described by the regular expression: $a^*(bc)^*cc$

1. cc
2. aaaacc
3. bccc
4. abcbccc

5. bcbcbcbccc