CSC341 – Lab 2B

Havin Lim, Carter English

February 3rd 2024

1 Interpretation

1.

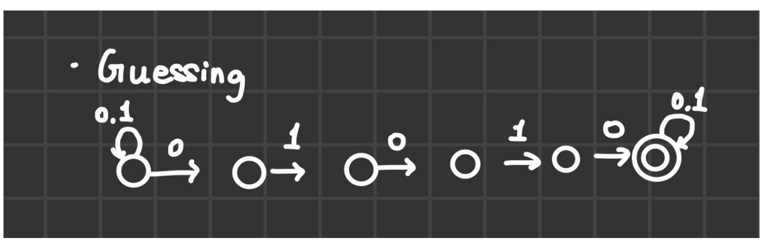
(a) A
(b) B
(c) C
(d) None

2.

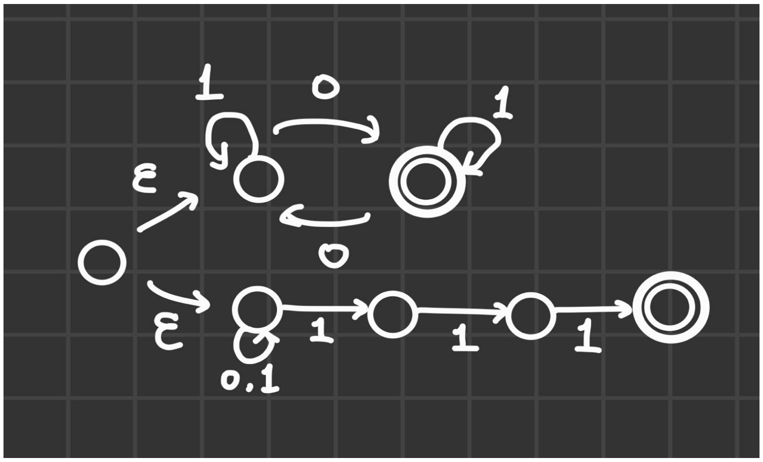
A : 01, 11, 101
B : {s $\in \Sigma^*$ | s starts with '0'}
C : {s $\in \Sigma^*$ | s only contains '10', repeatable}

2 Nondeterministic Design Patterns

1.



2.



3 Infinite Loops?

According to the formal definition of a NFA, the transition function takes epsilons and with epsilons, it is possible to create an infinite loop. The definition of an NFA does not contain anything that says computations need to be executable by a computer in finite time. Furthermore, by our definition of accepting a string, we accept a string if it can be written as a finite concatenation of elements from Σ_ϵ, so we don't need to consider infinite concatenations of elements from Σ_ϵ when doing computation.

4 Problem: All-NFAs

We first prove the former direction. Let L be an arbitrary regular language. Then it is recognized by some DFA $M = (Q, \Sigma, \delta, q_0, F)$. Then consider the all-NFA $N = (Q, \Sigma, \delta', q_0, F)$ where $\delta'(q, s) = \{\delta(q, s)\}$ for all $q \in Q, s \in \Sigma$.

Then if a string is in L, M accepts it, so the computation jn M passes through a deterministic sequence of states $\{q_0, q_1, ..., q_n\}$ where $q_n \in F$. Then when N performs computation on the string, the sequence of possible states at each step is $\{\{q_0\}, \{q_1\}, ..., \{q_n\}\}$. Thus the only possible final state is q_n, which is in F, so every possible final state is in F and thus N accepts the string. Conversely, if a string is not in L, M does not accept it, so the final state of the computation in M is q_n for some $q_n \notin F$. By the same logic as before, the set of final possible states of the computation in N is $\{q_n\}$, so there is a possible end state that is not in F and therefore N rejects the string. Therefore, N recognizes L, so because L was arbitrary every regular language is accepted by some all-NFA.

We now prove the reverse direction. Let L be an arbitrary language that is recognized by some all-NFA $N = (Q, \Sigma, \delta, q_0, F)$. Then let M be the DFA $(Q', \Sigma, \delta', E(\{q_0\}), F')$ where

$$Q' = \mathcal{P}(Q)$$

$$F' = \mathcal{P}(F)$$

$$\delta'(q, s) = \{E(\delta(q, s))\}.$$

Now for any string S in L, N accepts S. This means the sequence of possible states at each step of computation $\{\{q_{00}, q_{01}, ..., q_{0n_0}\}\{q_{10}, q_{11}, ..., q_{1n_1}\}, ..., \{q_{m0}, q_{m1}, ..., q_{mn_m}\}$ satisfies $\{q_{m0}, q_{m1}, ..., q_{mn_m}\} \subset F$. Therefore, when M does computation on S, it passes through the deterministic sequence of states $\{\{q_{00}, q_{01}, ..., q_{0n_0}\}\{q_{10}, q_{11}, ..., q_{1n_1}\}, ..., \{q_{m0}, q_{m1}, ..., q_{mn_m}\}$, so because $\{q_{m0}, q_{m1}, ..., q_{mn_m}\} \subset F$, it is in F' so we accept S. Conversely, if S is not in L then N does not accept S so $\{q_{m0}q_{m1}, ..., q_{mn_m}\} \not\subset F$, which means M also does not accept S because $\{q_{m0}, q_{m1}, ..., q_{mn_m}\}$ is not in $\mathcal{P}(F) = F'$. Therefore M recognizes L, so L is regular by definition.

Therefore, a language is regular if and only if it is accepted by some all-NFA.