COMPSCI 2AC3, Automata and Computability
Assignment 1 Solution, Winter 2024

Hassan Ashtiani, McMaster University

February 21, 2024

Notes. Your solutions **MUST be typeset in Latex** (refer to the first tutorial if you have missed it and not sure how to use Latex). Only upload a single pdf file as your solution to Avenue (avoid compressing your file). For drawing state machines use the the https://finsm.io/ website as discussed in the tutorial (also see the quick guide https://github.com/CSchank/finsm/wiki/QUICKSTART), and export the result to latex.

If you have questions about the assignment, post them in the dedicated Students Questions channel on MS Teams.

1. [**25 points**] Recall that a string consisting of only 0's and 1's can be thought as a number in base 2. For example, the decimal numbers 9 and 10 can be respectively represented by 1001 and 1010 in base 2. Let A be the set of all strings x over alphabet $\{0, 1\}$ whose first (i.e. left-most) symbol is 1, AND x is a base-2 representation of a number that is divisible by 3. For example, $11 \in A$, $110 \in A$, $1001 \in A$ but $10 \notin A$, $111 \notin A$, $1 \notin A$. Draw a DFA for A. Also, describe how your DFA works, and what each state represents (but you don't need to formally prove the correctness of your DFA).

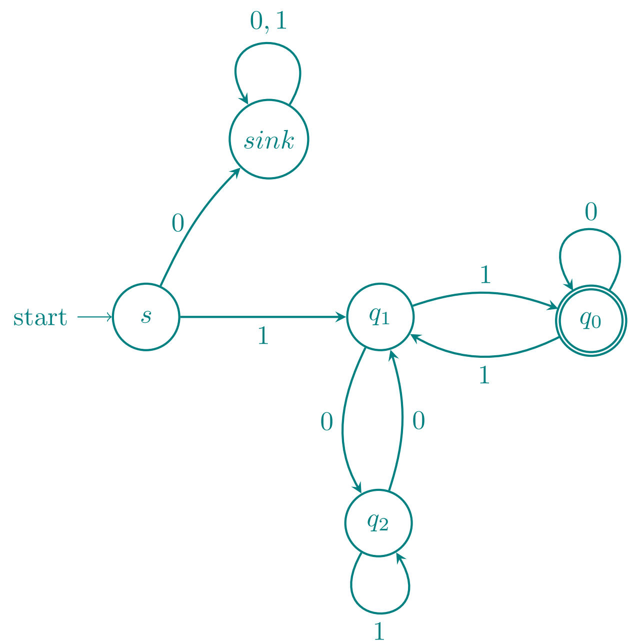
   The two conditions that need to be satisfied in our DFA:

   - The first symbol needs to be 1. So any stage reachable after reading a 0 needs to be **NOT** accepted. This means we can add a sink state.
   - The number needs to be a multiple of 3.

   Now the idea is that we can partition numbers based on their remainders by 3.

   Let q_0 be the state with the numbers with the format $3k$, q_1 be the state with the numbers with the format $3k + 1$, and q_2 be the state with the numbers with the format $3k + 2$. We will add transitions between the states based on how the number will change when we concatenate the number by the next symbol we read.

   Let's say we have a number $3k + 1$. Now when we read the next symbol 0, it means the number will be multiplied by 2, which gives us a $2 * (3k + 1) = 6k + 2 = 3k' + 2$. This means in the state $q1$, the transition function takes us to q_2 when reading a 0. Now let's consider the case where we are in $3k + 1$ and we read a 1, this means we are multiplying the number by 2, and also adding a 1. so this will give us $2 * (3k + 1) + 1 = 6k + 3 = 3k'$. This means by reading a 1 at state 1, the transition function takes us to q_0. The rest will be calculated similarly.



2. [**25 points**] Assume we have two NFAs, $N_1 = (Q_1, \Sigma, \Delta_1, S_1, F_1)$ and $N_2 = (Q_2, \Sigma, \Delta_2, S_2, F_2)$. Describe an NFA N_3 that satisfies $L(N_3) = L(N_1)L(N_2)$ (recall that for two sets A and B, their concatenation is defined by $AB = \{xy \mid x \in A, y \in B\}$). Include both a formal definition of N_3, and also an informal explanation of why it works (but you don't need to "prove" that it works). Note: in this question, we assume the NFAs don't have ε-transitions.

   Let $N_3 = (Q_1 \cup Q_2, \Sigma, \Delta_3, S_1, F_3)$ where

$$\Delta_3 = \Delta_1 \cup \Delta_2 \ \cup \bigcup_{a \in \Sigma} \{(q, a, q') | q \in F_1 \ \wedge \ q' \in Q_2 \ \wedge \ \exists q'' \in S_2 \ (q'', a, q') \in \Delta_2\}.$$

   and

$$F_3 = \begin{cases} F_2, & \text{if } S_2 \cap F_2 = \emptyset \\ F_1 \cup F_2 & \text{otherwise} \end{cases}$$

   This gives us the 'concatenation' of the two NFAs.

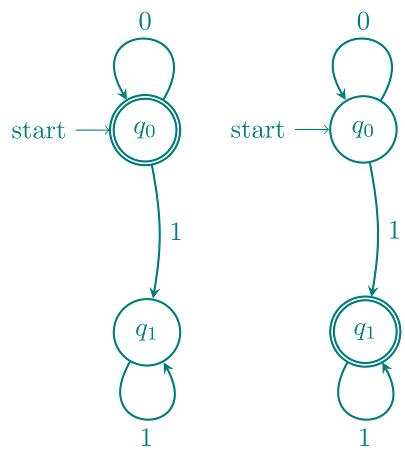
   - We select the start states of N_1 (S_1) as the start states for N_3. This ensures that the concatenation 'starts' with strings recognized by N_1.
   - The first part of the concatenated string 'must' be accepted by N_1 and the second part by N_2 for the overall string to be accepted by N_3. Now, we will have two cases, if none of the start states of N_2 is a final state, we only need to put $F_3 := F_2$. Otherwise, since this means that the empty string ϵ is accepted by N_2, we will have to make sure that anything accepted by N_2(concatenated by ϵ) is accepted in N_3. This means we need to put $F_3 = F_1 \cup F_2$
   - Δ_3 by construction keeps all the transitions 'inside' the two NFAs intact, and it also adds a path from any final state q of N_1 to states (immediately) following the start states of N_2. If we were allowed to use ϵ-transitions, we could have added ϵ-transitions between final states of N_1 and start states of N_2 instead. We are in a sense merging the states of F_1 with start states of N_2.

     (Reading the mathematical formula above: if there is a transition in Δ_2 from any start state of N_2, then we should add transitions from all the final states of N_1 to that state.)

- Let's say we have a string $s_1s_2s_3s_4s_5s_6s_7$ on the tape such that $s_1s_2s_3 \in L(N_1)$ and $s_4s_5s_6s_7 \in L(N_2)$. Now starting at a start state in S_1 and reading $s_1s_2s_3$ we will get to one of the final states of N_1 using the transitions in Δ_1. Now we should virtually be in the start states of N_2 (before reading s_4) and that's why in the construction of Δ_3 we are looking at the states reachable by start states and adding a transition from our final state to those reachable from start states of N_2. Note that even though we are not using the start states of N_1 we do NOT need to remove them from the set of states. In fact we have to keep them since there might be other transitions coming in to start states from other states of N_2. Now we read the rest of $s_4s_5s_6s_7$ and we will eventually end up in a final state of N_2, and that's why we will have F_2 as the set of our final states.

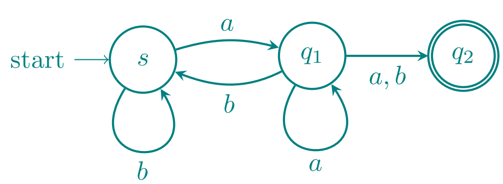
3. [**25 points**] Is the following statement true? "Let $N_1 = (Q, \Sigma, \Delta, S, F)$ be any non-deterministic finite state machine. Let $N_2 = (Q, \Sigma, \Delta, S, Q \setminus F)$. Then $L(N_1) =\sim L(N_2)$". If you think it is true, then prove it. Otherwise, provide a counterexample.

   The statement above is **NOT** true. A counter-example would be as follows:

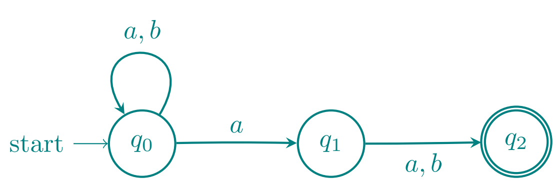


   The NFA on the left accepts $L_1 = \{$all the words consisting of all 0s$\}$ Now, by swapping the final and non-final states, we get the NFA on the right, which only accepts $L_2 = \{$all the (non-empty) words consisting of all 1s $\}$. But $L_1 \neq\sim L_2$(Since the word 10 is not accepted in L_1, but also not accepted in L_2).

4. (a) [**5 points**] Let $A = \{x \in \{a, b\}^* : |x| \geq 2$ and the 2nd symbol of x from the right is an $a\}$. Draw an NFA for A with 3 states (no proof is required).



   Another possible NFA would be:



(b) [**20 points**] Use the subset construction game to create a DFA for A out of its NFA. Label each state with subsets of the states of the NFA, and show your work.

The solution is based on the first NFA presented above. (Although, any DFA constructed from any other equivalent 3-state NFA is also acceptable.)

Let's start by constructing all subsets of the states:

$$\emptyset,\ \{s\}, \{q_1\}, \{q_2\}, \{s, q_1\}, \{s, q_2\}, \{q_1, q_2\}, \{s, q_1, q_2\}$$

The deterministic automaton is going to be as follows:

start at	a	b
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{s\}$	$\{q_1\}$	$\{s\}$
$\{q_1\}$	$\{q_1, q_2\}$	$\{s, q_2\}$
$\{q_2\}F$	$\emptyset$	$\emptyset$
$\{s, q_1\}$	$\{q_1, q_2\}$	$\{s, q_2\}$
$\{s, q_2\}F$	$\{q_1\}$	$\{s\}$
$\{q_1, q_2\}F$	$\{q_1, q_2\}$	$\{s, q_2\}$
$\{s, q_1, q_2\}F$	$\{q_1, q_2\}$	$\{s, q_2\}$

Following a, b-transitions of state $\{s\}$, we can see that the states $\emptyset, \{q_2\}, \{s, q_1\}, \{s, q_1, q_2\}$ cannot be reached and hence we might as well remove them. We are going to be left with:

start at	a	b
$\rightarrow \{s\}$	$\{q_1\}$	$\{s\}$
$\{q_1\}$	$\{q_1, q_2\}$	$\{s, q_2\}$
$\{s, q_2\}F$	$\{q_1\}$	$\{s\}$
$\{q_1, q_2\}F$	$\{q_1, q_2\}$	$\{s, q_2\}$

which gives us the DFA below: