Problem Set I

Computing Models
April 23, 2023

Alon Filler[1]

1 Finite Deterministic Automatas

Given the automata $D = (Q^D, \{a, b\}, \delta^D, q_0^D, F^D)$

Problem 1.1.
Imagine a new automata $E = (Q^E, \{a, b\}, \delta^E, q_0^E, F^E)$ *s.t:*

- $Q^E = Q^D \cup \{q_0^E\}$
- $F^E = F^D$
- $\delta^E(q, \sigma) = \begin{cases} \delta^D(q, \sigma) & q \in Q^D \\ q_0^D & q = q_0^E, \sigma = a \\ q_0^E & q = q_0^E, \sigma = b \end{cases}$

Define:

- *L(A)*

Solution.
It would be non but rational to divide this construction into three divisions, each corresponding to a different set of circumstances recognised by the trasitions function.

One of those aforementioned circumstances is $q = q_0^E, \sigma = b$, the study of such case lead me to determine that for the character input of b, under the assumption that the current state is q_0^E, the state would lead back to itself, meaning that that an instance of $\{b\}^*$ at the beginning of the input would not affect the output of the automata. And hence $\{b\}^*$ should be imbued to the language L(A).

Another set of circumstances is $q = q_0^E, \sigma = a$, which implies the current state to be the one added to Q^D in order to craft Q^E, and that the input chracter is 'a'. Such circumstances appear to be digested by the automata to return q_0^D, the first state of the previous automata D. Accordingly, it would only be after the appearence of an 'a' character in the input that the state would be changed. And hence, $\{a\}$ must be added to the language L(A).

[1] With Σorer

The last of such circumstances addressed in δ^E appears to be $q \in Q^D$. For such case, the function would make the transition from the current state to the one returned by δ^D, accordingly, L(D) must be concatenated at the end of L(E)

Hence - I may declare that $L(A) = \{b\}^* \cdot \{a\} \cdot L(D)$. ■

Problem 1.2.
Consider the previous automata[2] $E = (Q^E, \{a, b\}, \delta^E, q_0^E, F^E)$
And a new automata $E' = (Q^E, \{a, b\}, \delta^E, q_0^D, F^E)$

Define:

- $L(E')$

Solution.
In the previous problem[3] I have explained in great detail the effects caused by the declaration of a new state q_0^E to be the initial state of E. Considering q_0^E not to be part of Q^D, it can be seen with vividness that δ^E, known as E''s transitions function would cease to return it as output as soon as it no longer is the current state. Therefore, knowing E' does not use q_0^E as its initial state it is cogent that it would never be returned. Accordingly, the only circumstances relevant to the transitions function would be $q \in Q^D$, and thus the addition of $\{b\}^* \cdot \{a\}$ to the beginning of L(E) must be undone for $L(E')$ to be precise. And hence $L(E') = L(D)$ ■

Problem 1.3.
Consider the previous automata[4] $E = (Q^E, \{a, b\}, \delta^E, q_0^E, F^E)$
And a new automata $E' = (Q^E, \{a, b\}, \delta^E, q_0^D, F^D \cup q_0^E)$

Define:

- $L(E')$

Solution.
Now that q_0^E is an accepting state, I should like for $L(E')$ to support it. Thus, anything that happens to follow $\{b\}^*$ is merely optional. And hence it might be defined as such: $L(E') = \{b\}^* \cdot \{\{\epsilon\} \cup (\{a\} \cdot L(D))\}$ ■

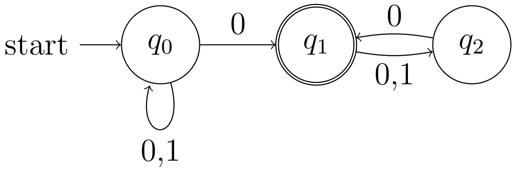
Consider the previous automata[5] $E = (Q^E, \{a, b\}, \delta^E, q_0^E, F^E)$

[2] Problem 1.1
[3] Problem 1.1
[4] Problem 1.2
[5] Problem 1.1

2 Finite Non Deterministic Automatas

Problem 2.1.
Build an automata to accept a language above $\{0, 1\}$ where each word is terminated with a character that follows a 0.

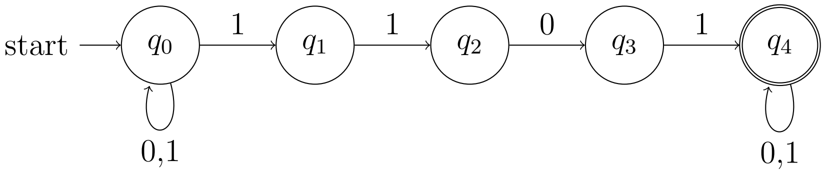
Solution.
The automata should have states, three in number, an initial one to which the automata would never appoint once departed from, one to acknowledge the preceding zero and one to halt upon receiving redundant input.



■

Problem 2.2.
Build an automata to accept a language above $\{0, 1\}$ where each word contains the string 1101.

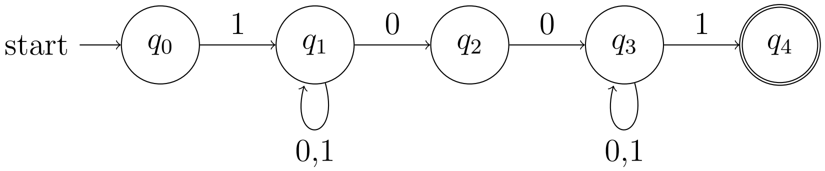
Solution.
The automata should have four states, each corresponding to a letter of the string 1101 and an additional one to process redundant input. Considering the automata not to be deterministic, it may be assumed that upon shifting from the initial state, the only characters to follow would be 101 and so on and so forth for the next characters.



■

Problem 2.3.
Build an automata to accept a language above $\{0, 1\}$ where each word begins and ends with a 1, and additionally contains the string 00.

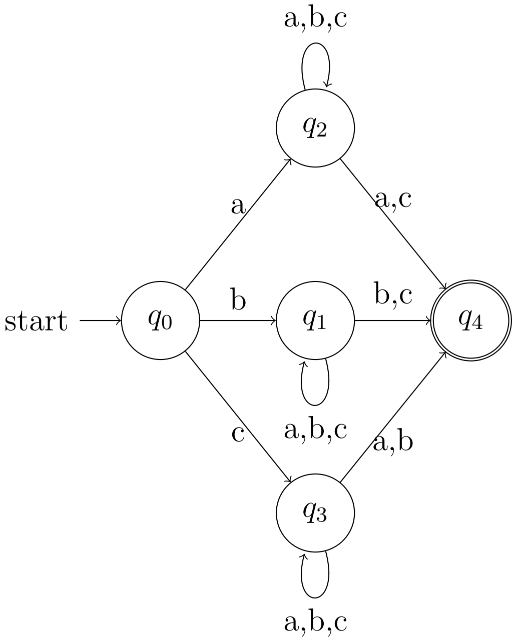
Solution.
The automata is to have six different states. An initial one, one to allow filtering words which begin-not with 1. A trap state for words which begin-not with 1 to fall into. A pair of states to indicate the presence of 00 and a state to indicate the termination of the string with 1.



■

Problem 2.4.
Build an automata to accept a language above $\{a, b, c\}$ where each word begins with a different characters than the one which terminates it.

Solution.
The automata is to have three different states, one to accomodate each possible beginning character, an additional initialisation state and a subsidiary 'sucess' state.

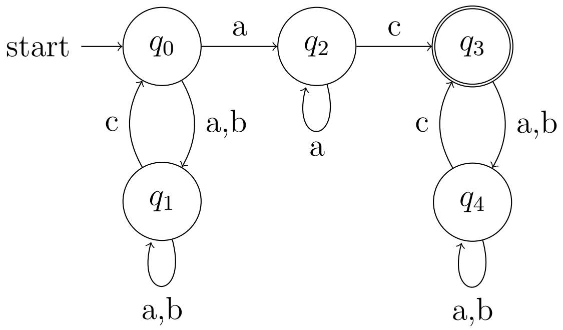


■

Problem 2.5.
Build an automata to accept the following language:

$$L = \begin{cases} k \geq 1, \\ \forall 1 \leq i \leq k : w_i \in \{a, b\}^+, \\ \exists 1 \leq i \leq k : w_i \in \{a\}^+ \end{cases}$$

Solution.
The automata requires a pair of states to indicate that the string of a's has been processed, an additional state to filter redundant parts of the input, an initial state and a state to filter redundant parts at the end of the latter.


start
q_0
a
q_2
c
q_3
c
a,b
a
c
a,b
q_1
q_4
a,b
a,b

■