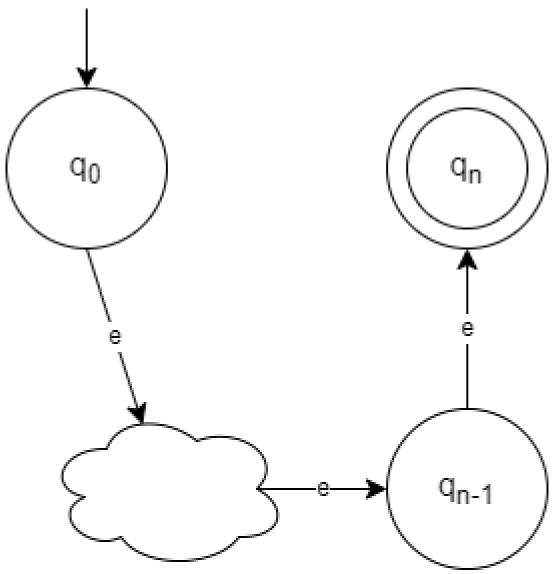
COT4210 Discrete Structures – Exam 1

Spring 2023

1. (10) E_n is the language that contains all and only strings that, in turn, contain nothing but the letter "**e**" some integer multiple of n times:

$$E_n = \{\mathbf{e}^k \mid k \text{ is a multiple of } n\}.$$

Show that this language is regular for any positive integer n. *(Hint: **You do not need induction**.)*



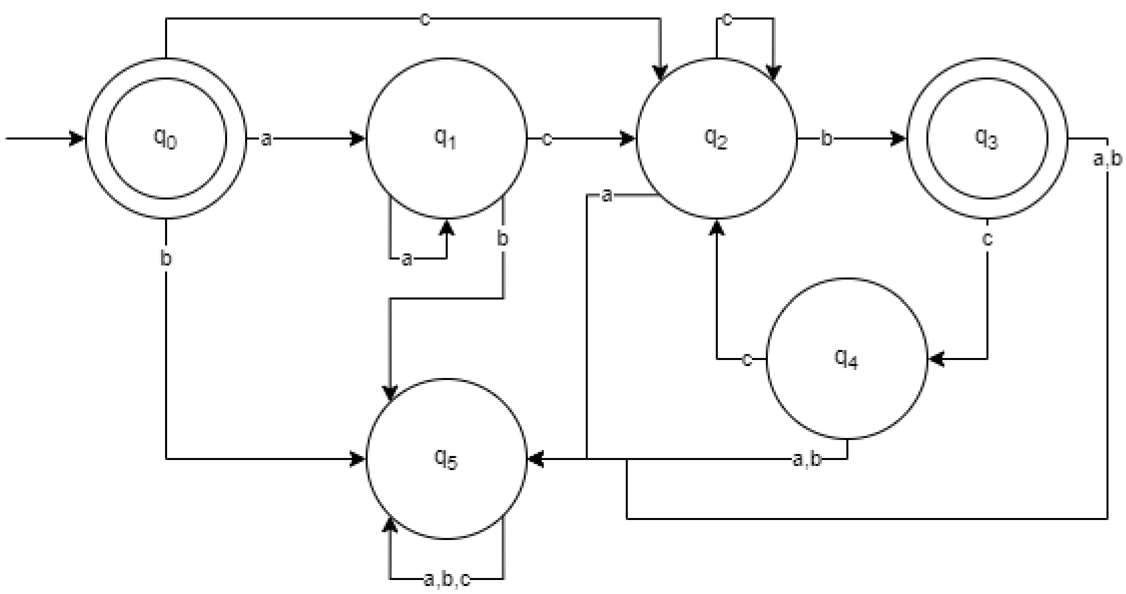
The above shows an NFA that proves language L where

- $Q = \{q_0, q_1, q_k\}$
- $F = \{q_k\}$
- $d(q_0, e) = \{q_1\}$
- $d(q_i, e) = \{q_{i+1}\}$ for all $i < k$
- $d(q_k, e) = \emptyset$

2. (15) Let *L* be the language over {**a**, **b**, **c**} accepting all strings so that:

1. No **b**'s occur before the first **c**.
2. No **a**'s occur after the first **c**.
3. The last symbol of the string is **b**.
4. Each **b** that is *not* the last symbol is immediately followed by at least two **c**'s.

Choose any constructive method you wish, and demonstrate that *L* is regular. *You do not need an inductive proof, but you should explain how your construction accounts for each rule.*



The graph above shows a DFA that proves language L

- An empty string is accepted so q_0 is a final state.
- There can't be any b's before the first c so q_0 and q_1 have a b input that leads to a dead end.
- There can't be any a's after the first c so all states after q_1 have an a input that leads to a dead end.
- The last symbol must be b so q_3 is also a final state.
- Each b that is not the last symbol is immediately followed by at least two c's, so q_3 has a c input that leads to q_4 which has a c input that leads to q_2 and both q_3 and q_4 have a and b inputs that lead to a dead end which means b is always the final symbol.
- q_5 is a dead end so every input loops back into itself, if you've entered q_5 then the string is not accepted.

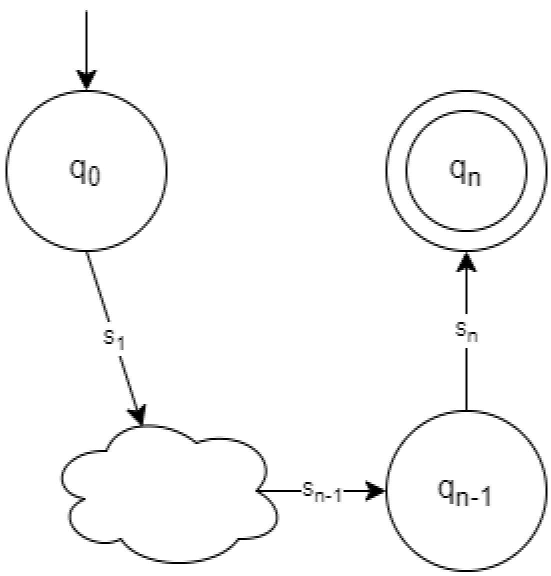
3. (15) A homomorphism on an alphabet is simply a function that gives a string for each symbol in that alphabet – for example, a homomorphism *h* on the binary alphabet might be defined so that *h*(**0**) = **ba** and *h*(**1**) = **edc**.

Homomorphisms can be extended to strings and languages in the straightforward way:

- If $s = s_1 s_2 s_3 \ldots s_n$ then $h(s) = h(s_1)\, h(s_2)\, h(s_3) \ldots h(s_n)$.
- If L is a language then $h(L) = \{ h(s) \mid s \text{ is in } L \}$.

Show that the class of regular languages is closed under homomorphism – that is, that for any regular language *L*, and any homomorphism *h* on its alphabet, *h*(*L*) defined as above is regular. *HINT: If your proof is very long at all, you are doing more than you need to.*

If a homomorphism can be defined as where L is a regular language, then its homomorphism is $h(L) = \{h(s) \mid s\ is\ in\ L\}$. We can create a general NFA that represents each $h(s)$ in $h(L)$ where $s = s_1 s_{2_{s_3}} \ldots s_n$ so that each transition concatenates a part of the string $h(s_i)$



- $Q = \{q_0, q_1, q_n\}$
- $F = \{q_n\}$
- $d(q_0, s_1) = \{q_1\}$
- $d(q_i, s_i) = \{q_{i+1}\}$ for all $i \leq n$

4. (10) Comments in certain types of computer programming languages appear between starting and ending strings such as **<#** and **#>**.

- A comment must begin with **<#** and end with **#>**.
- A comment may not contain **#>** inside the comment.
- A comment's starting and ending strings must be separate: **<##>** is a comment, **<#>** is not.
- For simplicity, assume that the alphabet is □ = {**a**, **b**, **<**, **>**, **#**}.

Give a regular expression that accepts comments.

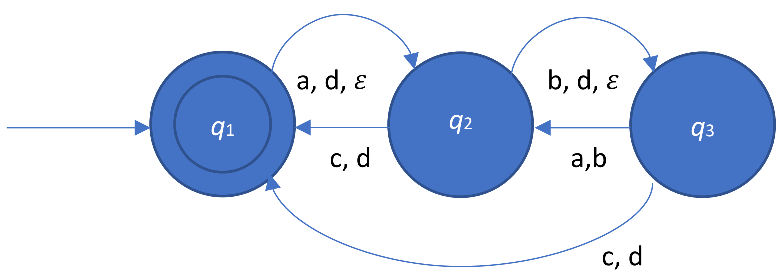
Resulting regular expression: <#(#|(a|b|<)*)*#>

5. (15) Let $\Sigma = \{a, b, \#\}$ and $L = \{\, w \mid w$ can be written as $t\#s\#t^R$ with $s, t \in \{a, b\}^* \}$. Show that L is not regular.

Assume this language is regular with p as the pumping length and $s = t^p\#s^p\#(t^R)^p$

- Then $s = xyz$ with xy^iz in p for all nonnegative integers i, with $|y| > 0$ and $|xy| \leq p$
- Since $|xy| \leq p$, x and y must be entirely in function t^p with the remaining within z
- $xyyz$ will only change t and not t^R
- Since t^Rwill no longer equal the reverse of t, the language will be violated and therefore by contradiction the language is not regular.

6. (15) Using the procedure demonstrated in class and in the textbook, convert this NFA to a DFA.

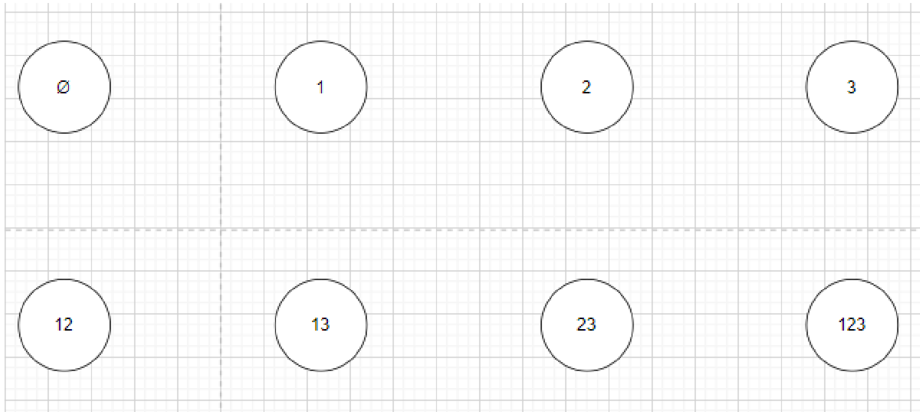


$Q_D = P(Q_N)$

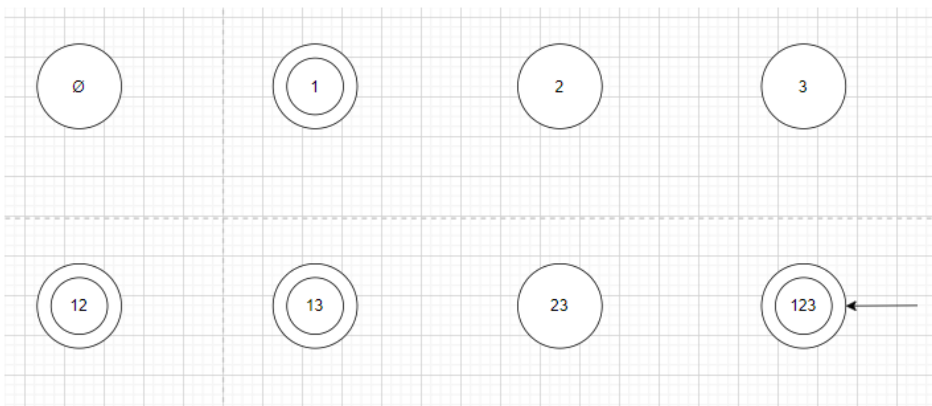
$q_{0D} = E(\{q_{0N}\})$

$F_D = \{R \,\epsilon\, Q_D | R \cap F_N \neq \emptyset\}$

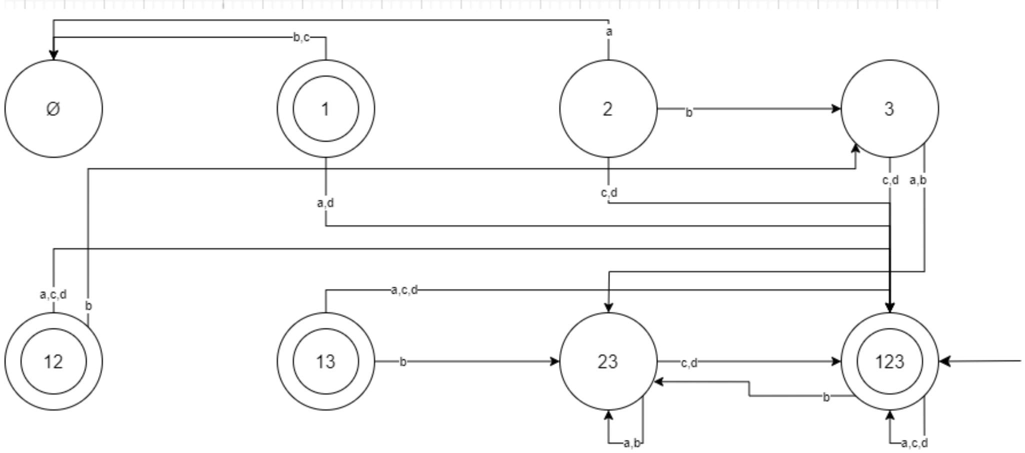
$\delta_D(R, a) = \{q \,\epsilon\, Q_N |\, q \,\epsilon\, E\big(\delta_N(r, a)\big)\, for\ some\ r \,\epsilon\, R\}$



Ø
1
2
3
12
13
23
123

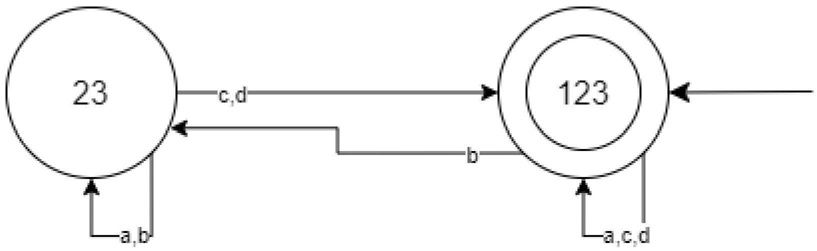


Ø
1
2
3
12
13
23
123

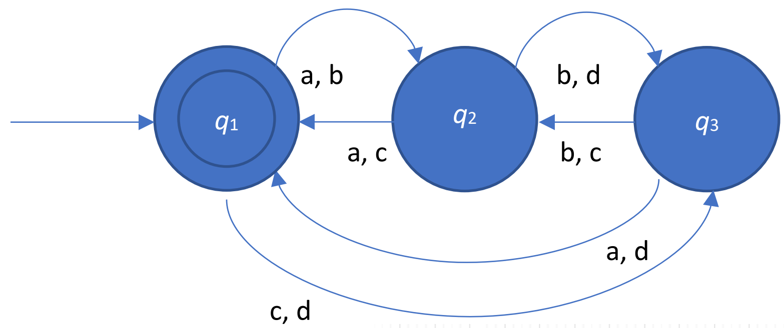


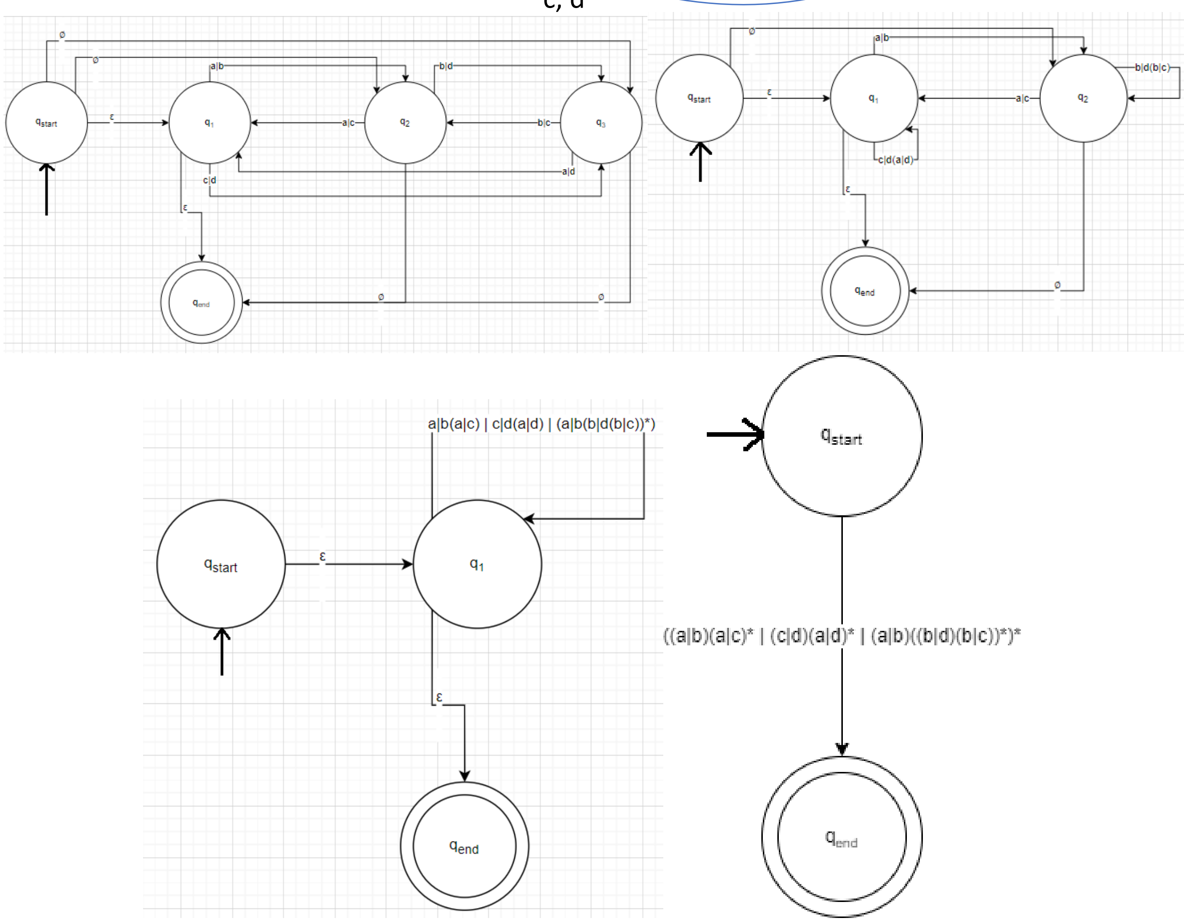
Ø
1
2
3
12
13
23
123
a
b,c
b
c,d
a,b
a,d
c,d
a,c,d
b
a,c,d
b
c,d
b
a,b
a,c,d

7. (5) Reduce the DFA resulting from problem 6.



8. (15) Convert this DFA to a regular expression using a GNFA.





Resulting regular expression: ((a|b)(a|c)* | (c|d)(a|d)* | (a|b)((b|d)(b|c))*)*