

Finite automata and formal languages

Assignment 2

DIT323 (Finite automata and formal languages)
at Gothenburg University

Sebastian Pålsson

Last updated: **February 09, 2024**

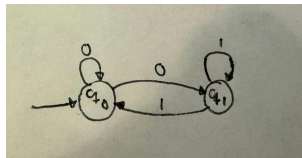
1

Let $\Sigma = \{0, 1\}$

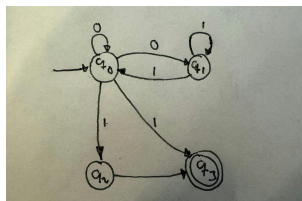
Let $R = (0 + 01^*)^*(\varepsilon + 1)1(\varepsilon + 0 + 1)^*$

Let's construct a NFA for R:

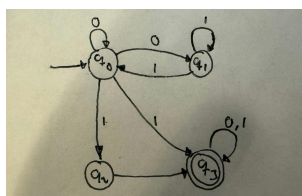
1. Construct a NFA for $(0 + 01^*)^*$



2. Add $(\varepsilon + 1)1$ to the NFA



3. Add $(\varepsilon + 0 + 1)^*$ to the NFA



The NFA for R is the thus following:

1	0	1
2	$\rightarrow q_0$	$\{q_0, q_1\} \{q_2\}$
3	q_1	$\emptyset \{q_0, q_1\}$
4	q_2	$\emptyset q_3$
5	$* q_3$	q_3

2

The regular expression is: $R = 1(1 + 01)^+$

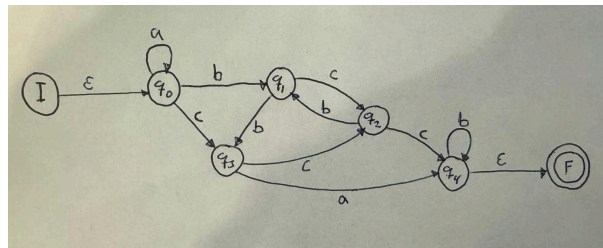
- The “1” in the beginning makes it so that the string must start with a 1.
- The “+” makes it so $|w| \geq 2$
- $(1 + 01)$ makes it so it always ends with a 1 and two zeroes never appear after one another.

3

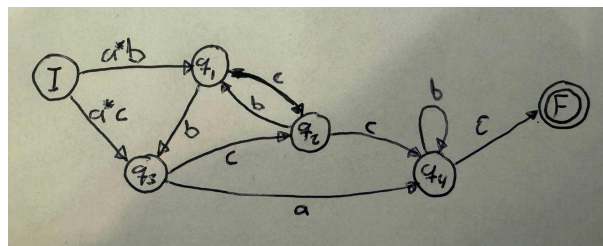
4

We will solve this problem by removing each node one by one and construct the regular expression in steps by writing subexpressions for each edge in the NFA.

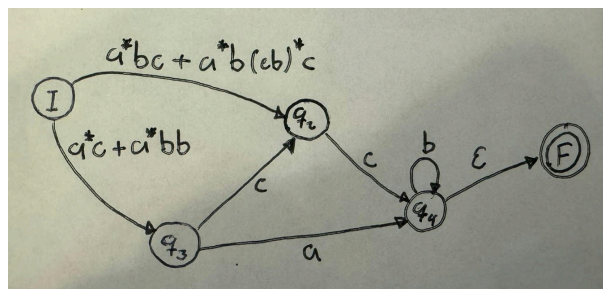
We start by adding node I and F that stands for the initial and final state of the NFA.



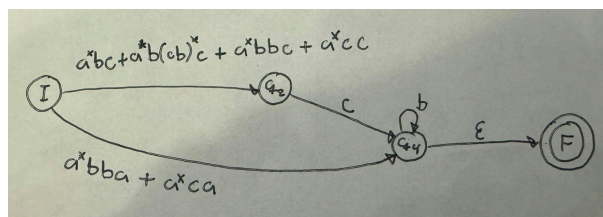
Then we remove node q_0 and add the subexpressions to the new edges in the NFA.



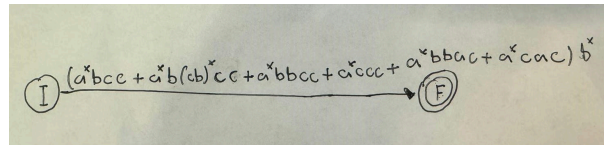
Then we remove node q_1 and do the same thing.



Then we remove node q_3 and do the same thing.



Finally we remove node q_2 and q_4 and do the same thing.



We get the following regular expression:

$$(a^*bcc + a^*b(cb)^*cc + a^*bbcc + a^*ccc + a^*bbac + a^*cac)b^*$$

We can remove the first term a^*bcc because this is a subset of the second term $a^*b(cb)^*cc$. Thus we get the final regular expression:

$$(a^*b(cb)^*cc + a^*bbcc + a^*ccc + a^*bbac + a^*cac)b^*$$