Language and Automata, Assignment 1

Krzysztof Rudnicki
Student number: 307585

November 17, 2021

1.1 Regular expression

We are given following regular expression:

$$a^* + ba^*b + bba^*$$

1.2 Examples of accepted strings

1. ε
2. a
3. bab
4. bba
5. bb

1.3 Building NFA using Thompson construction algorithm

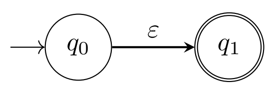

Figure 1.1: Operator 'a'

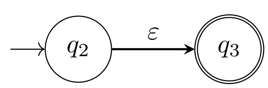

Figure 1.2: Operator 'b'

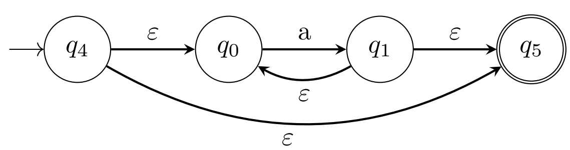

Figure 1.3: Operator 'a^*'

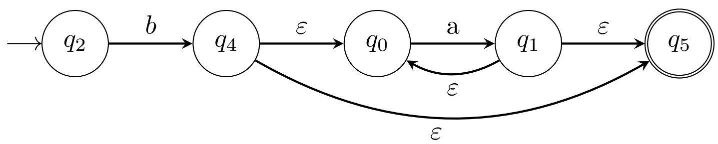

Figure 1.4: Operator 'ba^*'

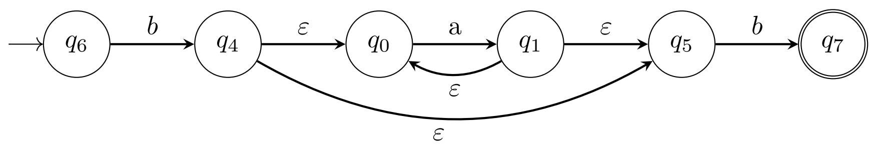

Figure 1.5: Operator 'ba^*b'

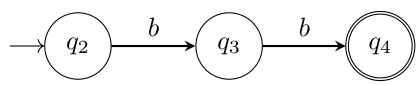

Figure 1.6: Operator 'bb'

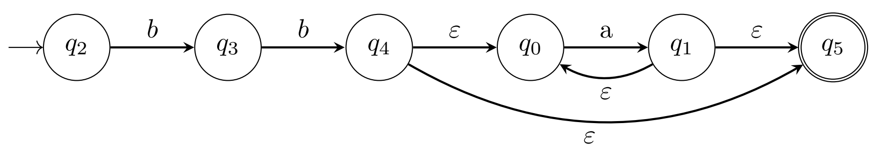

Figure 1.7: Operator 'bba^*'

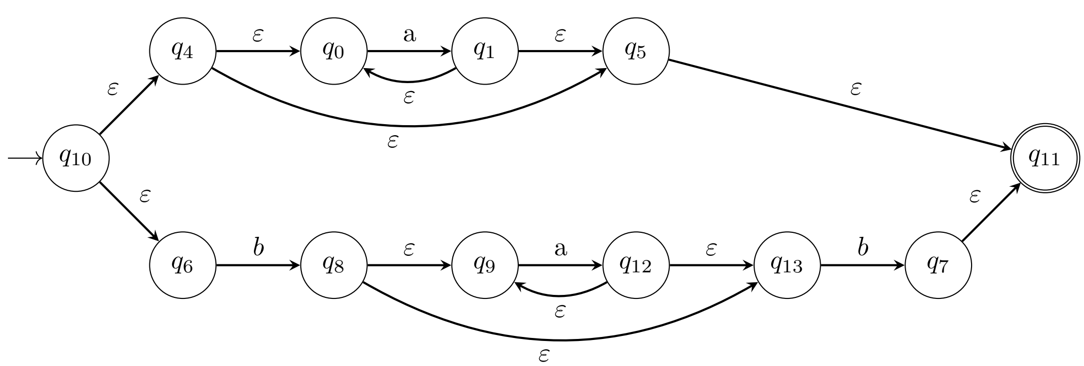

Figure 1.8: Operator '$a^* + ba^*b$'

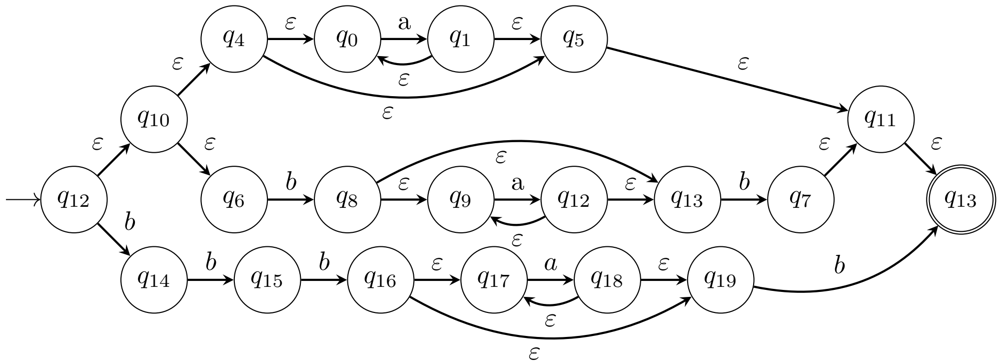

Figure 1.9: Operator '$a^* + ba^*b + bba^*$'

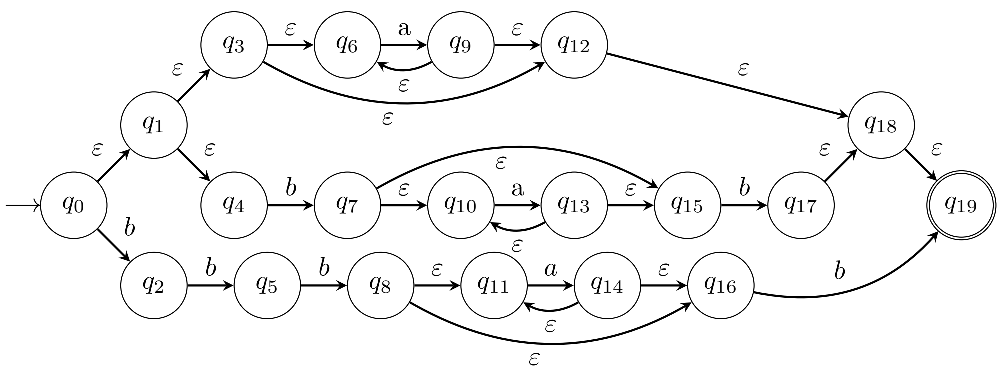

Figure 1.10: Operator '$a^* + ba^*b + bba^*$' - changed names of states

1.4 Transforming NFA into DFA using subset algorithm

I will use ϵ_{cl} instead of ϵ-closure for brevity sake. Final state - $\underline{q_{19}}$ was marked with an *underline* and so did all the states of DFA that contain it.

$$A = \epsilon_{cl}(q_0) = (q_0, q_1, q_3, q_4, q_6, q_{12}, q_{18}, \underline{q_{19}}) = \underline{A}$$

$$\epsilon_{cl}(move(\underline{A}, a)) = (\epsilon_{cl}(move(q_0, q_1, q_3, q_4, q_6, q_{12}, q_{18}, \underline{q_{19}}, a))) = \epsilon_{cl}(q_9) = (q_9, q_{12}, q_{18}, \underline{q_{19}}) = \underline{B}$$

$$A = \epsilon_{cl}(move(\underline{A}, b)) = \epsilon_{cl}(move(q_0, q_1, q_3, q_4, q_6, q_{12}, q_{18}, \underline{q_{19}}, b))) = \epsilon_{cl}(q_2, q_7) = (q_{10}, q_{15}) = C$$

$$\epsilon_{cl}(move(\underline{B}, a)) = \epsilon_{cl}(move(q_9, q_{12}, q_{18}, \underline{q_{19}}), a) = \emptyset$$

$$\epsilon_{cl}(move(\underline{B}, b)) = \epsilon_{cl}(move(q_9, q_{12}, q_{18}, \underline{q_{19}}), b) = \emptyset$$

$$\epsilon_{cl}(move(C, a)) = \epsilon_{cl}(move(q_{10}, q_{15}), a) = \epsilon_{cl}(q_{13}) = (q_{15}) = D$$

$$\epsilon_{cl}(move(C, b)) = \epsilon_{cl}(move(q_{10}, q_{15}), b) = \epsilon_{cl}(q_{17}) = (q_{18}, \underline{q_{19}}) = \underline{E}$$

$$\epsilon_{cl}(move(D, a)) = \epsilon_{cl}(move((q_{15}), a) = \epsilon_{cl}(\emptyset) = \emptyset$$

$$\epsilon_{cl}(move(D, b)) = \epsilon_{cl}(move((q_{15}), b) = \epsilon_{cl}(q_{17}) = (q_{18}, \underline{q_{19}}) = \underline{E}$$

$$\epsilon_{cl}(move(\underline{E}, a)) = \epsilon_{cl}(move((q_{18}, \underline{q_{19}}), a) = \epsilon_{cl}(\emptyset) = \emptyset$$

$$\epsilon_{cl}(move(\underline{E}, b)) = \epsilon_{cl}(move((q_{18}, \underline{q_{19}}), b) = \epsilon_{cl}(\emptyset) = \emptyset$$

1.4.1 State table

State	a	b
$\underline{A}$	$\underline{B}$	C
$\underline{B}$	$\emptyset$	$\emptyset$
C	D	$\underline{E}$
D	$\emptyset$	$\underline{E}$
$\underline{E}$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

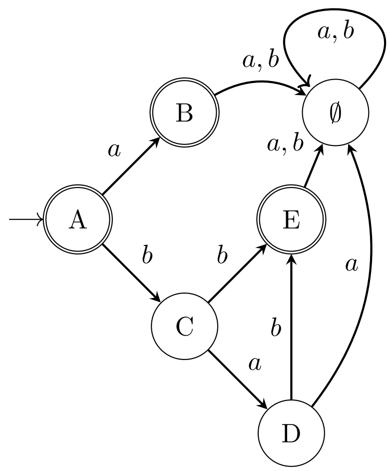

Figure 1.11: DFA graph before minimalization

1.5 Constructing minimal state DFA

$\underline{A}$						
$\underline{B}$	x_1					
C	x_1	x_1				
D	x_1	x_1	x_2			
$\underline{E}$	x_1	x_1	x_1	x_1		
$\emptyset$	x_1	x_1	x_2	x_2	x_1	
	$\underline{A}$	$\underline{B}$	C	D	$\underline{E}$	$\emptyset$

1. First I marked (with x_1) all the pairs in which at least one of them were final state:
$$([\underline{A}, \emptyset], [\underline{A}, \underline{E}], [\underline{A}, D], [\underline{A}, C], [\underline{A}, \underline{B}])$$
$$([\underline{B}, \emptyset], ([\underline{B}, \underline{E}], ([\underline{B}, D], ([\underline{B}, C])$$
$$([\underline{E}, \emptyset], [\underline{E}, C], [\underline{E}, D])$$

2. We are left with the pairs:
$$([\emptyset, C], [\emptyset, D], [D, C])$$
For pair: $[\emptyset, C]$ C goes to final state $\underline{E}$ on transition 'b' therefore we mark it with x_2 For pair: $[\emptyset, D]$ D goes to final state $\underline{E}$ on transition 'b' therefore we mark it with x_2 For pair: $[D, C]$ both C and D go to final state $\underline{E}$ on transition 'b' therefore we mark it with x_2

No states could be minimized! Therefore our final minimal state DFA looks like this:

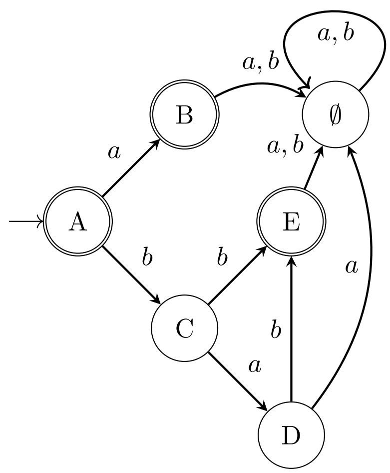

Figure 1.12: DFA graph after minimalization