Automata – excercise 1.5 page 84.

(a) $L_a = \{w \,/\, w$ does not contain the substring ab $\}$.

let's look at L_a' ($L_a' = \overline{L_a}$):

$L_a' = \{w \,/\, w$ does contain the substring ab $\}$.

$= (a|b)^*.ab.(a|b)^*$.

we obtain:

$L_a = \overline{(a|b)^*.ab.(a|b)^*}$.

(b) $L_b = \{w \,/\, w$ does not contain the substring baba $\}$

similarly to our approach for L_a, we define $L_b' = \overline{L_b}$ as:

$L_b' = \{w \,/\, w$ does contain the substring baba $\}$

$= (a|b)^*.baba.(a|b)^*$

and we obtain:

$L_b = \overline{(a|b)^*.baba.(a|b)^*}$.

(c) $L_c = \{w \,/\, w$ contains neither ab nor ba $\}$

L_c can be seen as the intersection of $L_{c_1} = \{w \,/\, w$ does not contain ab $\}$ and $L_{c_2} = \{w \,/\, w$ does not contain ba $\}$.

Hence $=$ $L_c = L_{c_1} \cap L_{c_2}$.

Now for each of L_{c_1} and L_{c_2}, let's look at L_{c_1}' ($= \overline{L_{c_1}}$) and L_{c_2}' ($= \overline{L_{c_2}}$).

$L_{c_1}' = \{w \,/\, w$ does contain ab $\}$

$L_{c_2}' = \{w \,/\, w$ does contain ba $\}$.

Since $L_c = L_{c_1} \cap L_{c_2}$, $\overline{L_c} = \overline{L_{c_1}} \cup \overline{L_{c_2}}$.

We therefore proceed to describe $\overline{L_c}$.

$$\overline{L_c} = \underbrace{\left((a|b)^*.ab.(a|b)^*\right)}_{L_{c_1}' = \overline{L_{c_1}}} \cup \underbrace{\left((a|b)^*.ba.(a|b)^*\right)}_{L_{c_2}' = \overline{L_{c_2}}}$$

Now that we have a regular expression for $\overline{L_c}$, we easily obtain the regular expression of $L_c = \overline{\overline{L_c}}$:

$$L_c = \overline{\left(\left((a|b)^*.ab.(a|b)^*\right) \cup \left((a|b)^*.ba.(a|b)^*\right)\right)}$$

(d) $L_d = \{w \,/\, w$ is any string not in $a^*b^*\}$.

$\overline{L_d} = a^*b^*$, therefore $L_d = \overline{\overline{L_d}} = \overline{a^*b^*}$.

(e) $L_e = \{w \,/\, w$ is any string not in $(ab^+)^*\}$

$\overline{L_e} = (a.b^+)^*$, therefore $L_e = \overline{\overline{L_e}} = \overline{(ab^+)^*}$.

(f) $L_f = \{w \,/\, w$ is any string not in $a^* \cup b^*\}$.

$\overline{L_f} = a^* \cup b^*$, therefore $L_f = \overline{\overline{L_f}} = \overline{a^* \cup b^*}$.

(g) $L_g = \{ w \ / \ w$ is any string that does not contain exactly two a's $\}$

$\overline{L_g} = \{ w \ / \ w$ is any string that does contain exactly two a's $\}$
$= b^* a b^* a b^*$.

therefore $L_g = \overline{\overline{L_g}} = \overline{b^* a b^* a b^*}$.

(h) $L_h = \{ w \ / \ w$ is any string except a and b $\}$.

as a result L_h is the complement of the language L_h' ($= \overline{L_h}$) that contains only a and b.

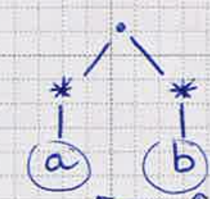
$L_h' = \{a, b\} = a \cup b$.

therefore $L_h = \overline{\overline{L_h}} = \overline{a \cup b}$.

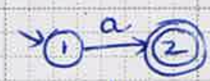
(d) $L_d = \overline{a^* b^*}$.

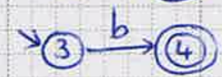
let's first build the NFA of $a^* b^*$:

$a^* b^* =$

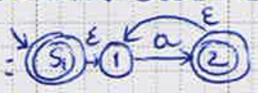


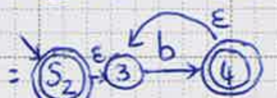
building blocks.

a: 

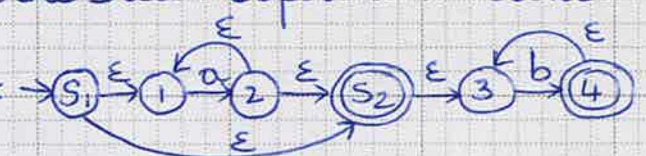
b: 

next steps (following the order of operations as shown in the tree):

a^*: 

b^*: 

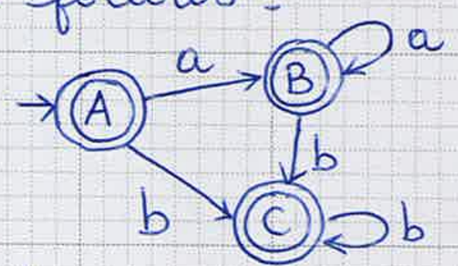
and the last operation: concatenation

$a^* b^*$: 

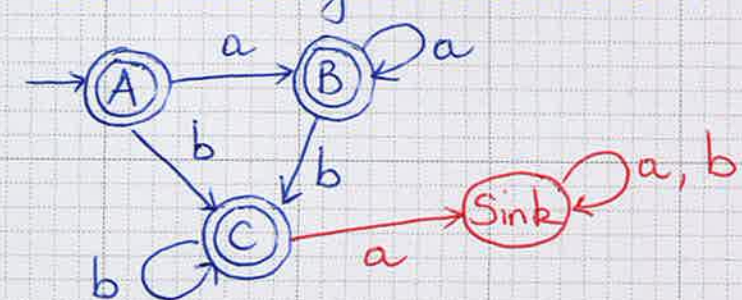
now, before to get $\overline{a^* b^*}$, we need to make the above NFA for $a^* b^*$ deterministic.

δ	a	b
{S1, S2, 3} A	{2, 1, S2, 3} B	{4, 3} C
{2, 1, S2, 3} B	{2, 1, S2, 3} B	{4, 3} C
{4, 3} C	Sink	{4, 3} C
Sink	Sink	Sink

the DFA corresponding to the previous NFA for $a^* b^*$ is as follows :

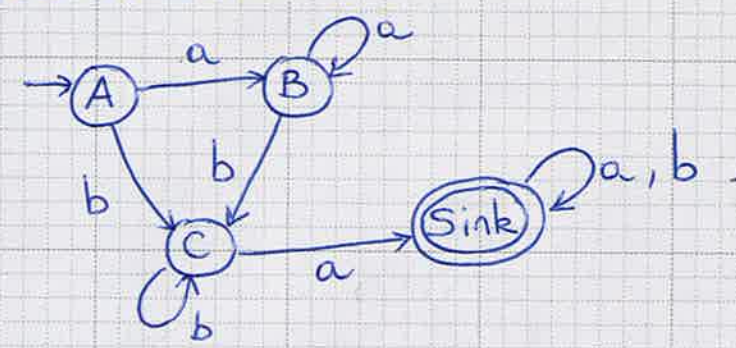


this DFA is not complete = we complete it with a Sink as shown above in the table and on the diagram below.



Now that we have obtained a <u>complete</u> <u>DFA</u> for $a^* b^*$, we can get <u>its</u> complement (i.e., ~~the~~ a DFA for $\overline{a^* b^*}$) by simply swapping final and non-final states.

We obtain :

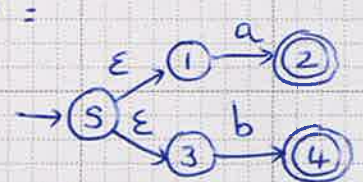


(h) $L_h = \overline{a \cup b}$

let's first look at the NFA of $a \cup b$:

We obtain :

$a \cup b$:

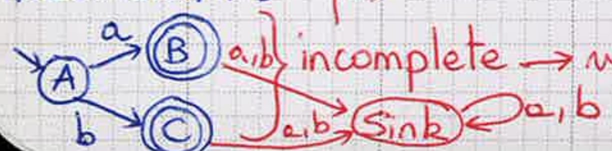


before to take the complement of the above NFA, we first need to make it a DFA and then complete it if necessary.

to go from an NFA to a DFA, we create the transition table of the reachable states:

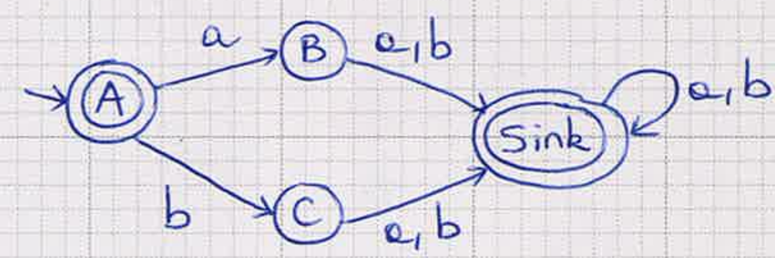
δ	a	b
$\{S,1,3\}_A$	$\{2\}$ ~~A~~ B	$\{4\}_C$
$\{2\}$ B	Sink	Sink
$\{4\}$ C	Sink	Sink
Sink	Sink	Sink.

new DFA :



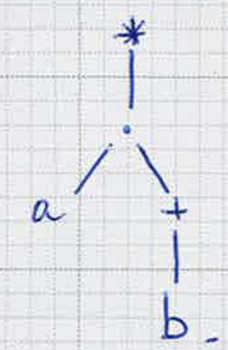
incomplete → we add a Sink

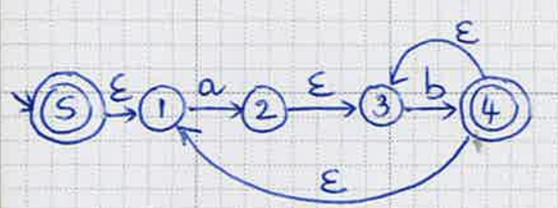
Now that we have a DFA for $a \cup b$ that is complete, we can obtain a DFA for $\overline{a \cup b}$ as follows :



(e) $L_e = \overline{(ab^+)^*}$.

let's first create an NFA for $(ab^+)^*$:

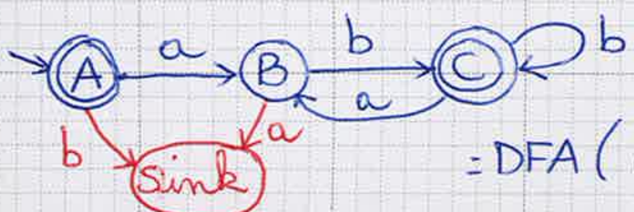




$= NFA((ab^+)^*)$.

NFA → DFA: + we complete the DFA

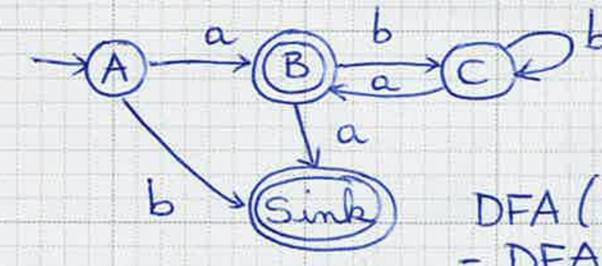
δ	a	b
$\{\underline{S},1\}_A$	$\{2,3\}_B$	Sink
$\{2,3\}_B$	Sink	$\{4,3,1\}_C$
$\{\underline{4},3,1\}_C$	$\{2,3\}_B$	$\{4,3,1\}_C$
Sink	Sink	Sink



$= DFA((ab^+)^*)$

We take the complement of $DFA((ab^+)^*)$ by swapping final and non-final states.

Note: the reason why we can do that is because the DFA of $(ab^+)^*$ we created is complete.



$DFA(\overline{(ab^+)^*})$

$= DFA(L_e)$.