Exam Regex 2020

Vincent DAO - INFO 2

Exercice 1

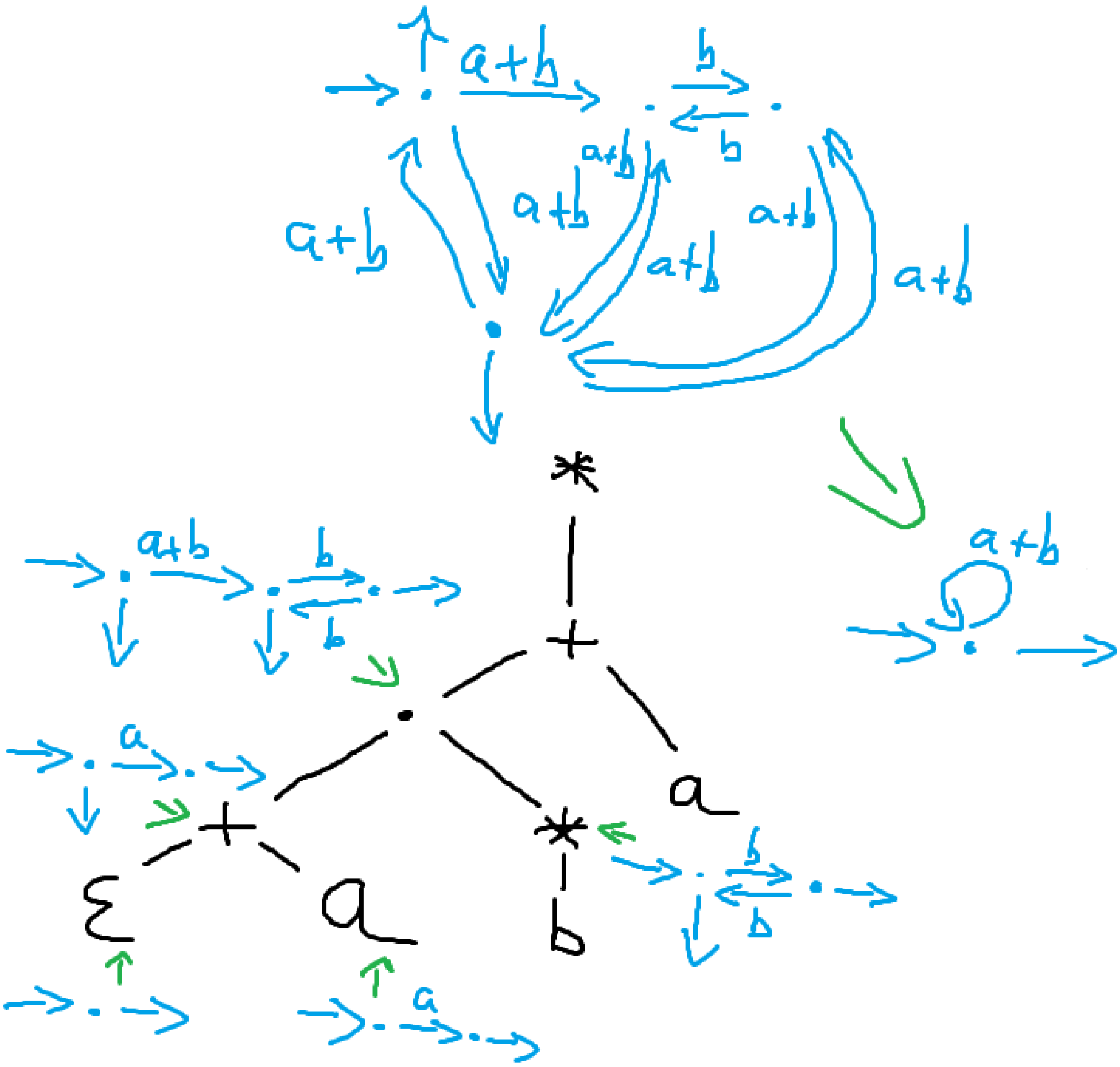
1. Soit le langage L défini par l'expression régulière e suivante :

$$e \;=\; (a + (\varepsilon + a)b^*)^*$$

a) Appliquer **sans simplification** la méthode de Thompson à l'expression e afin d'obtenir un automate fini reconnaissant L.

b) Donner une expression simple équivalente à e et l'expliquer.

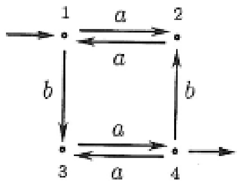
a)



(a+b)*

Exercice 2

2. Soit le langage M reconnu par l'automate déterministe $\mathcal{A}$ suivant :



a) Donner tous les mots de M de longueur 6.

b) Appliquer **sans simplification** la méthode BMC à cet automate $\mathcal{A}$ afin d'obtenir une expression régulière définissant M ; on retirera dans l'ordre les états 4, 3, 2, 1.

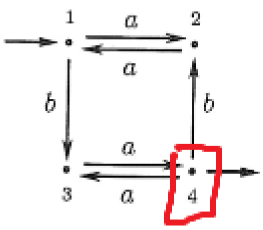
c) Expliciter simplement le langage M.

a)

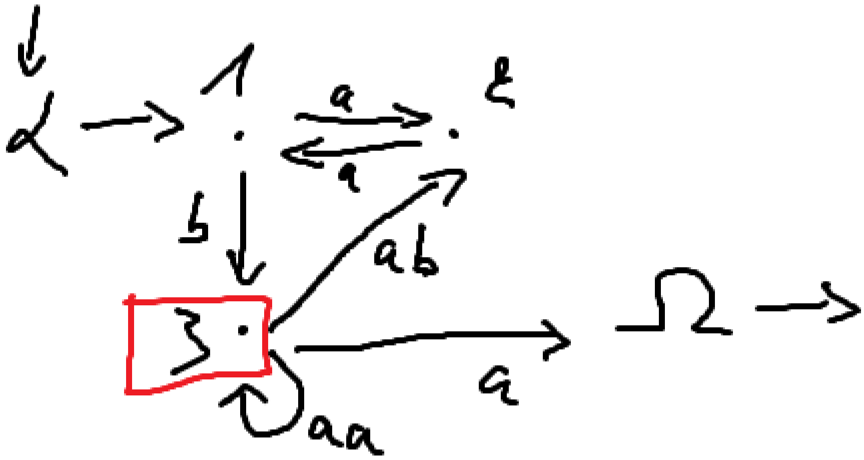
Aaaaba - aabaaa - baaaaa - bababa

b)

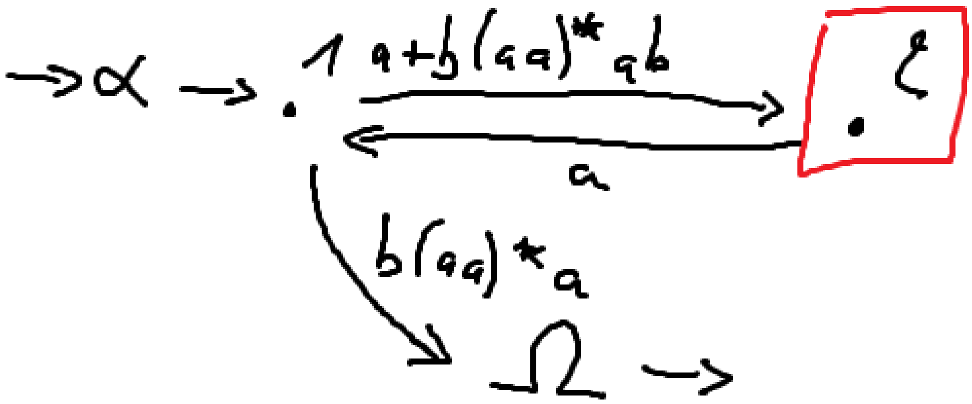
Starting point :



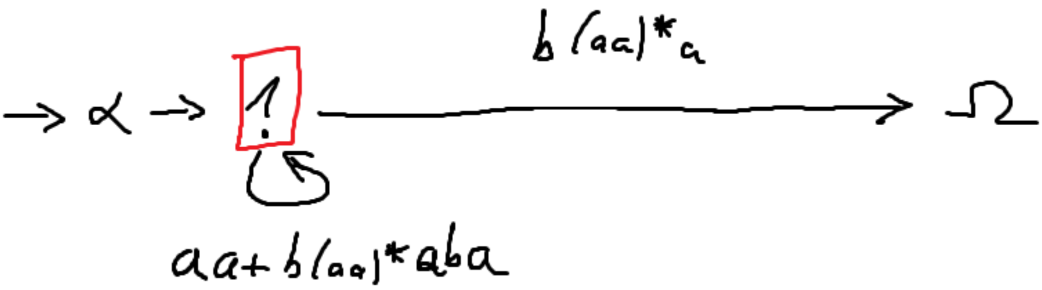
Step 1 : remove “4”



Step 2 : remove “3”



Step 3 : remove “2”



Final step : remove “1”

c)

Tous les mots qui :

- N’ont pas 2b consécutifs (bb non facteur de u)
- Ont un nombre pair de b et de a (|u|b impair ET |u|a impair) donc |u| pair
- Où les b sont en position impair

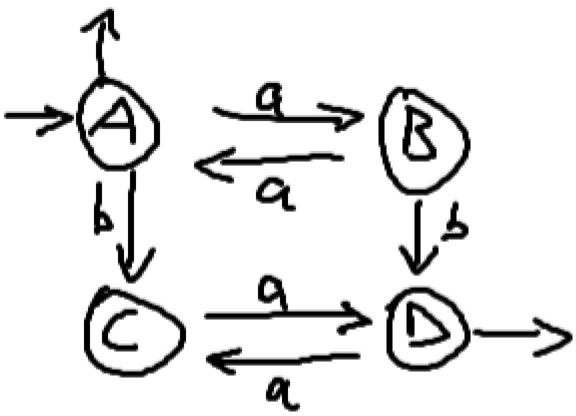
Exercice 3

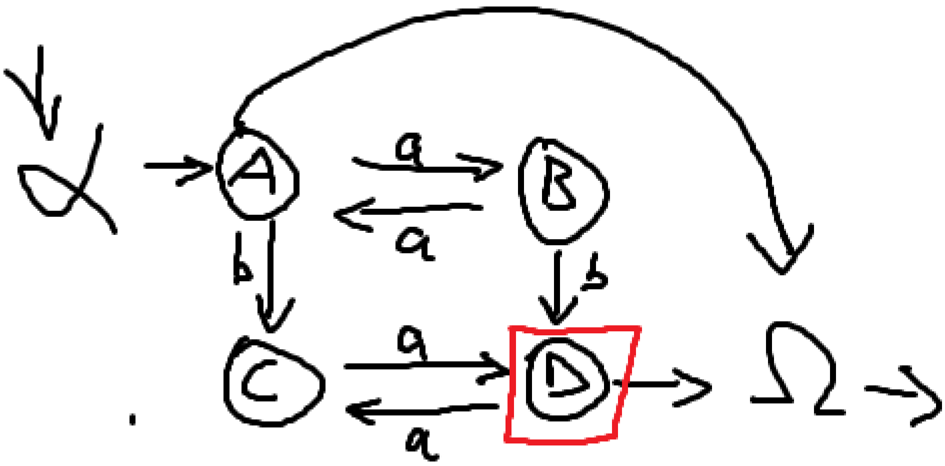
3. Soit l'alphabet $A = \{a, b\}$. Donner un automate fini **déterministe** puis une expression régulière pour chacun des langages suivants :

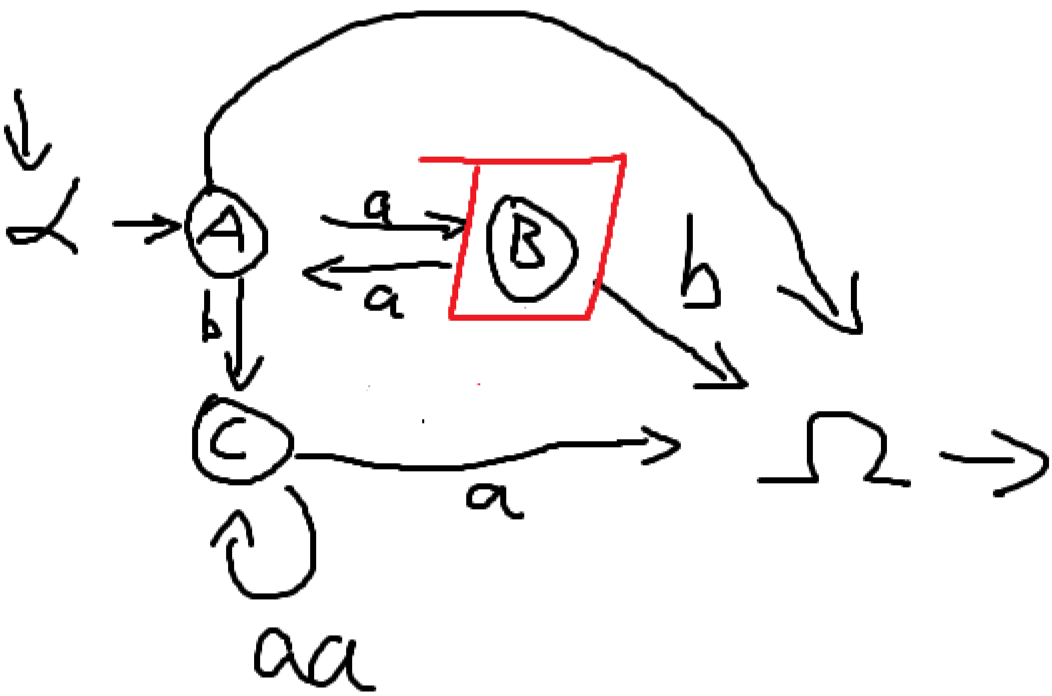
$$L_1 = \{ u \in A^* \mid |u| \text{ pair et } |u|_b \leq 1 \}$$
$$L_2 = A^* - A^*ab$$
$$L_3 = \{ u \in A^* \mid |u|_a \text{ impair et } bb \text{ non facteur de } u \}$$
$$L_4 = A^*aa - aA^*bbA^*$$
$$L_5 = aA^*b - A^*abA^*$$

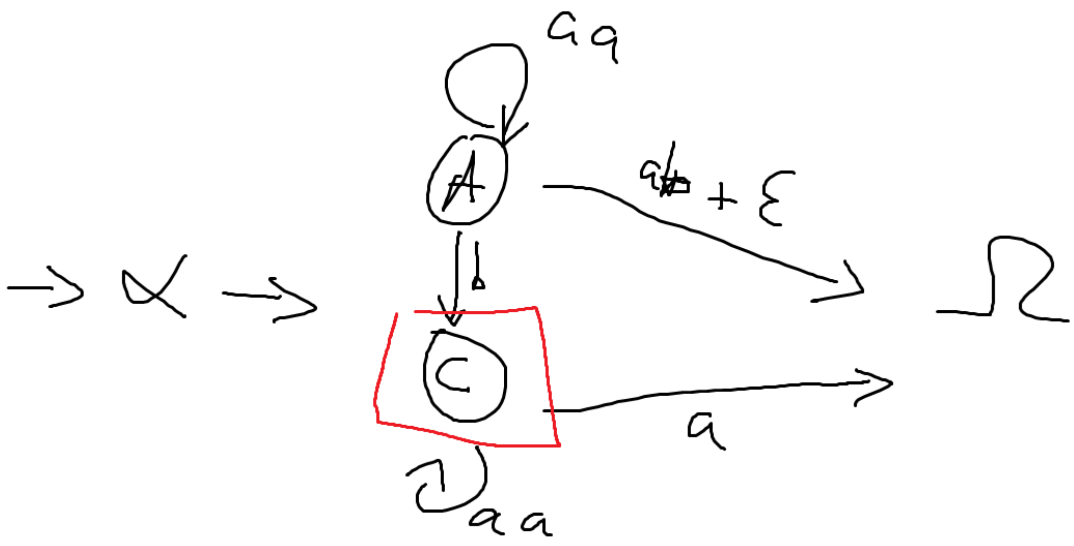
L1 :

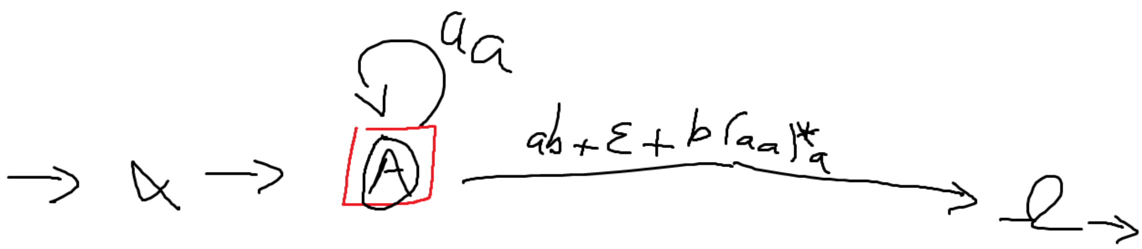
$$L_1 = \{ u \in A^* \mid |u| \text{ pair et } |u|_b \leq 1 \}$$

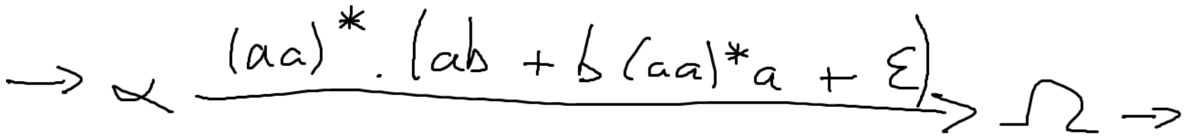



α
A
B
C
D
Ω
a
a
b
b
a
a


α
A
B
C
Ω
a
a
b
b
a
aa





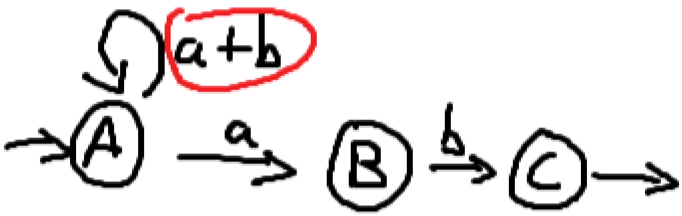


Expression : (aa) * . (ab + b (aa)* a + E)

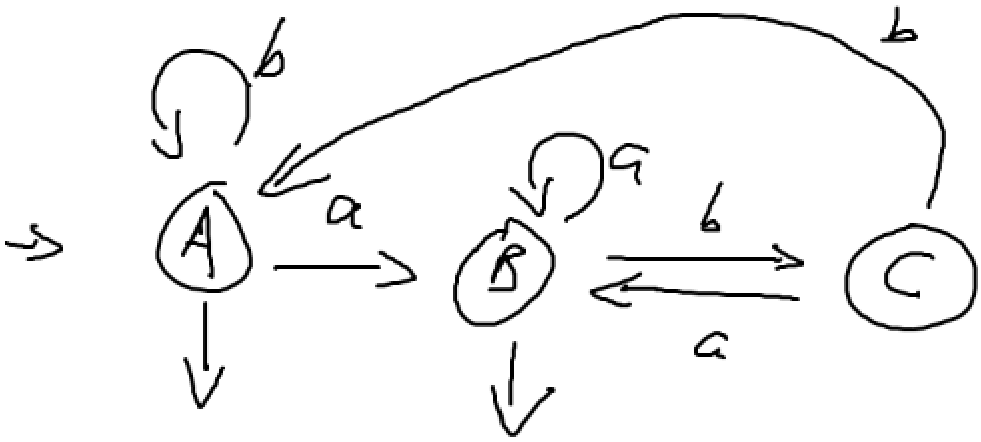
L2 :

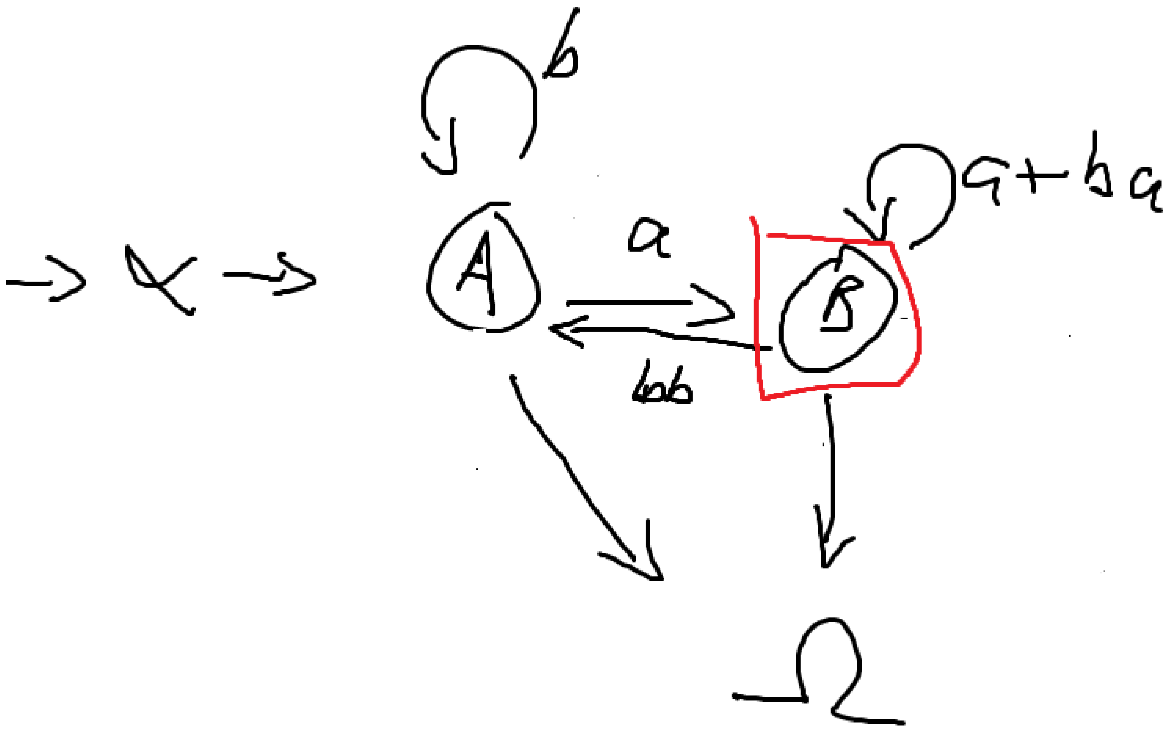
$L_2 = A^* - A^*ab \longrightarrow$ Tous les mots qui ne finissent PAS par ab

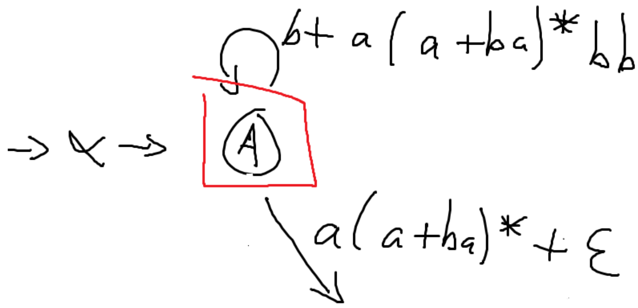
$A^*ab =$

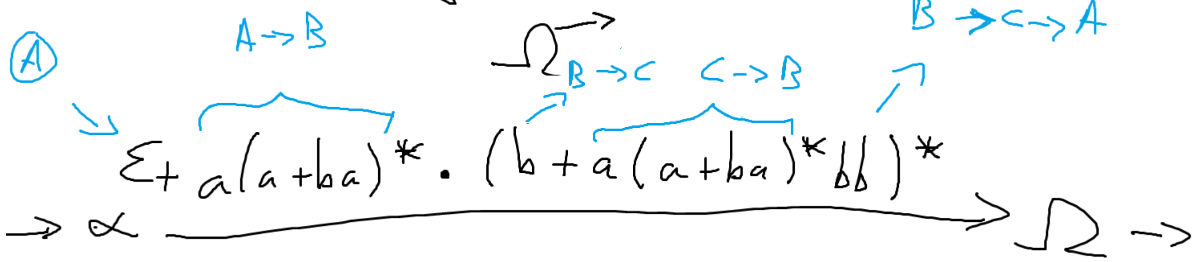


changer pour sorties pour avoir $\overline{A * ab}$




b
a+ba
α
A
a
B
bb
Ω

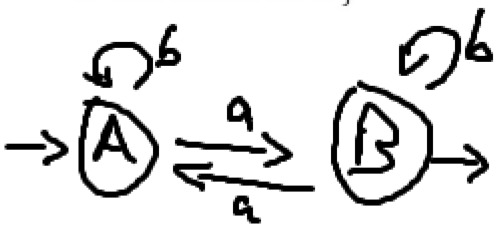

b+a(a+ba)*bb
α
A
a(a+ba)*+ε
Ω


A
A->B
B->C
C->B
B->C->A
ε+a(a+ba)*·(b+a(a+ba)*bb)*
α
Ω

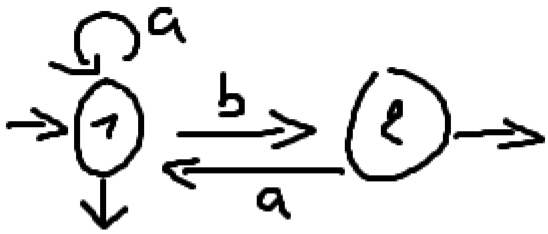
L3 :

$L_3 = \{ u \in A^* \mid |u|_a \text{ impair et } bb \text{ non facteur de } u \}$

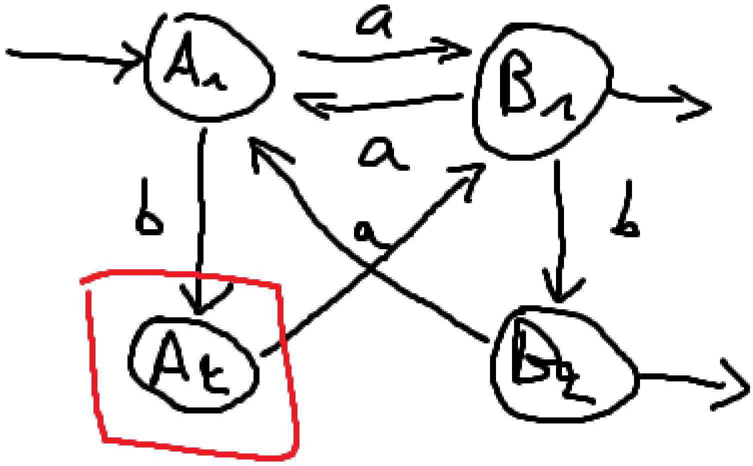
|u|a impair =



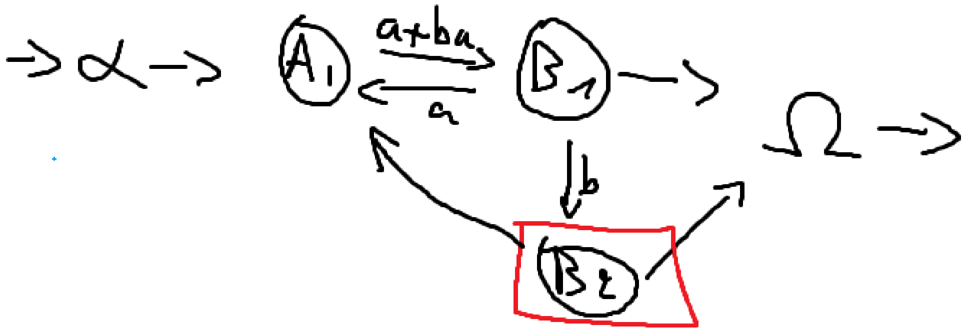
bb non facteur=

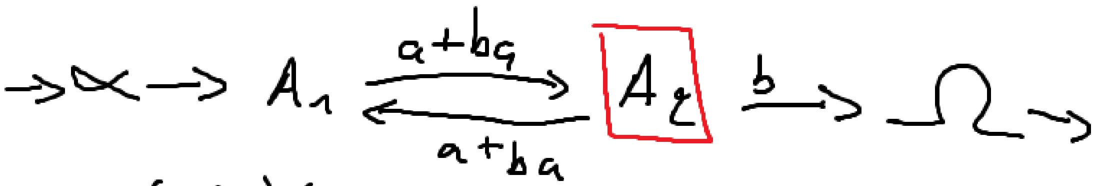


L3=

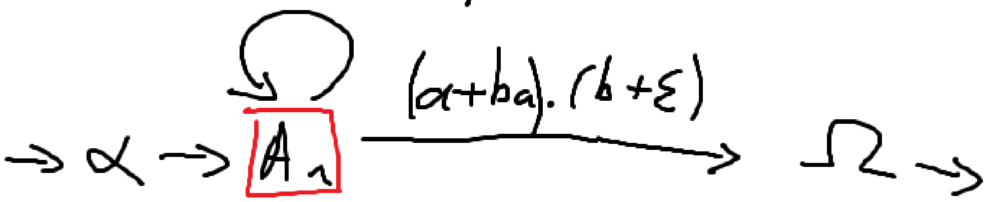


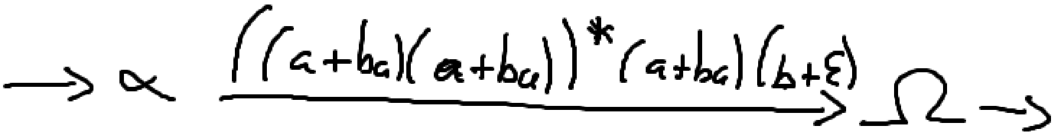
BMC en commençant par retirer A2





$(a+ba)(a+ba)$

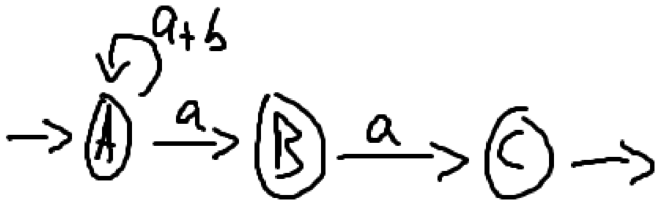




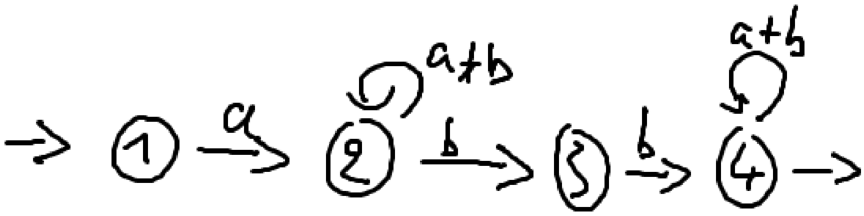
L4 :

$$L_4 = A^*aa - aA^*bbA^*$$

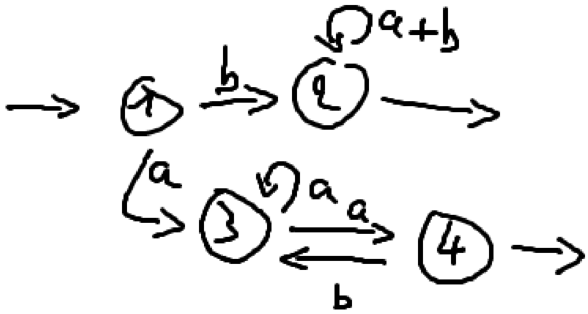
A*aa=



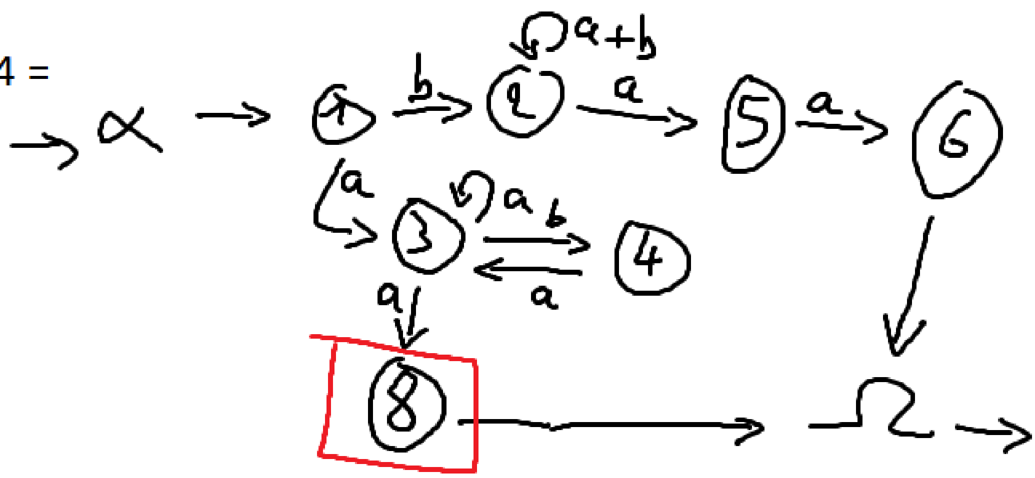
aA*bbA*=

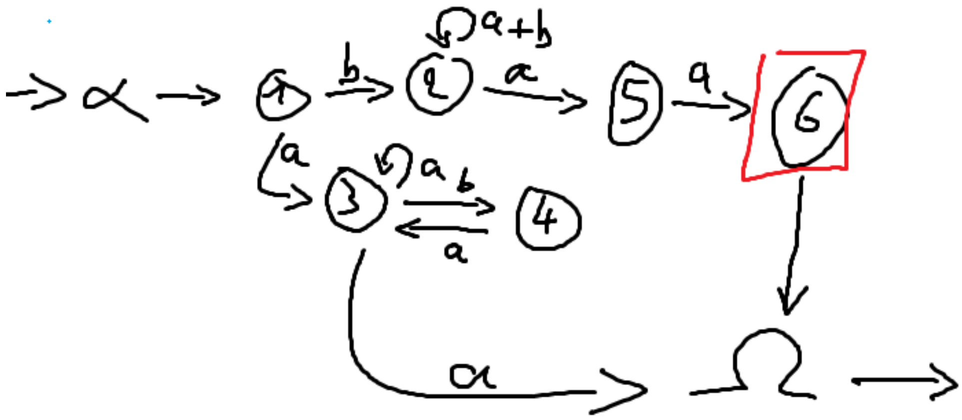


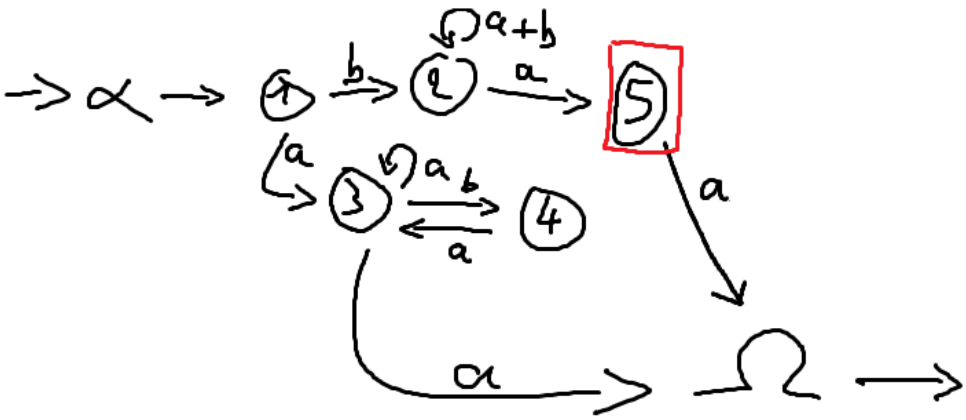
$\overline{\text{aA*bbA*}}$=

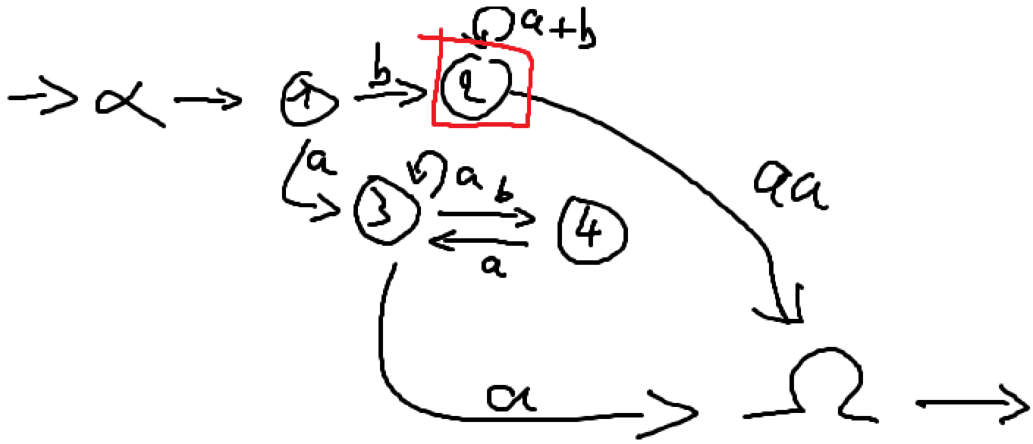


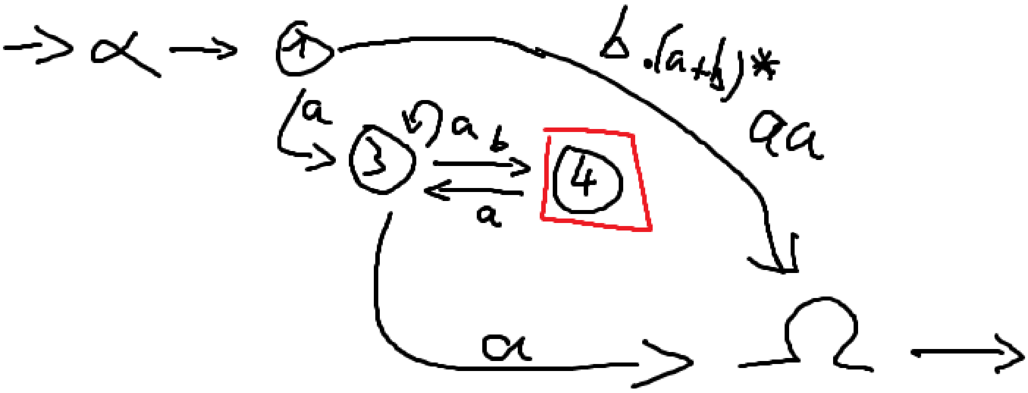
L4 =

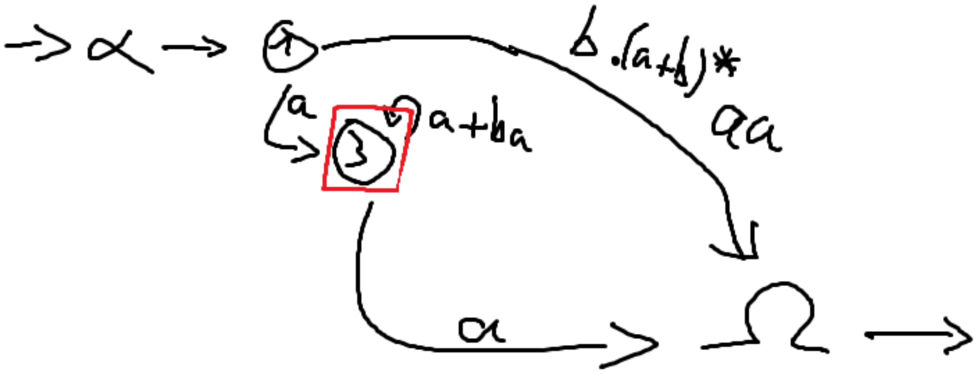



a+b
b
a
5
a
6
a
a
b
4
a
α


a+b
b
a
5
a
a
b
4
a
a
α







$$\to \alpha \to 1 \xrightarrow{(b.(a+b)^{*}aa)+(a.(a+ba)^{*}a)} \Omega \to$$

$$\to \alpha \xrightarrow{(b.(a+b)^{*}aa)+(a.(a+ba)^{*}a)} \Omega \to$$

L5 :

$$L_5 = aA^*b - A^*abA^*$$

Résultat = langage vide car si on commence par a et finit par b, il y aura forcément une instance de ab.