Exam Regex 2020

Vincent DAO - INFO 2

[Exercice 1](#_5mxr60yyfv4s)

[a)](#_hlpro8blhzcl)

[b)](#_exueinnp9lub)

[Exercice 2](#_b82qamyaqhxf)

[a)](#_aj1xnypbry09)

[b)](#_g4nd1ymlaklc)

[c)](#_a0813msok1d5)

[Exercice 3](#_4cvq5kfeqn5l)

[L1 :](#_m0lxzcaczuuq)

[L2 :](#_8dq304vzsptf)

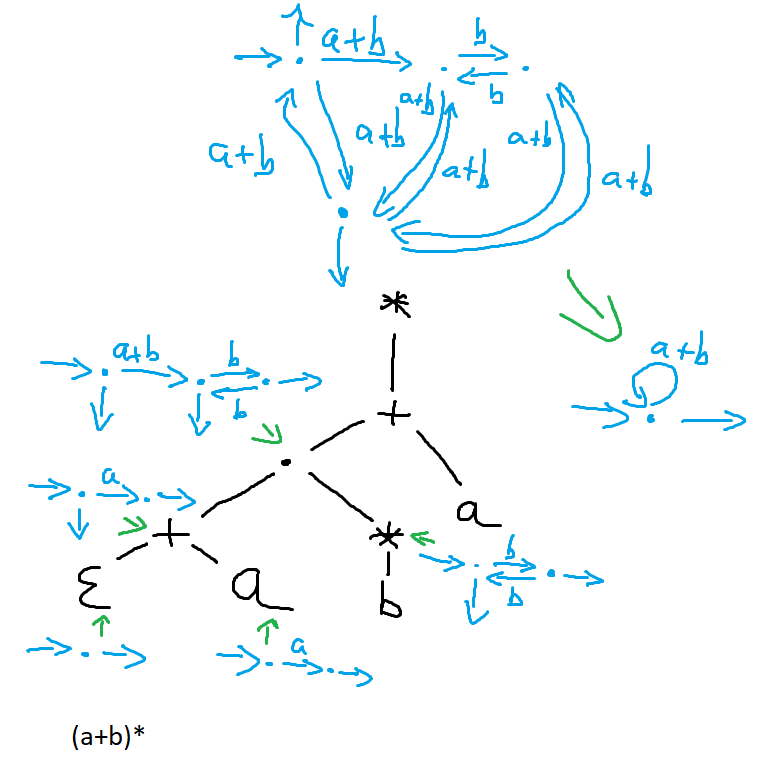
[L3 :](#_yvs3r1swkmg3)

[L4 :](#_t4kht2voc579)

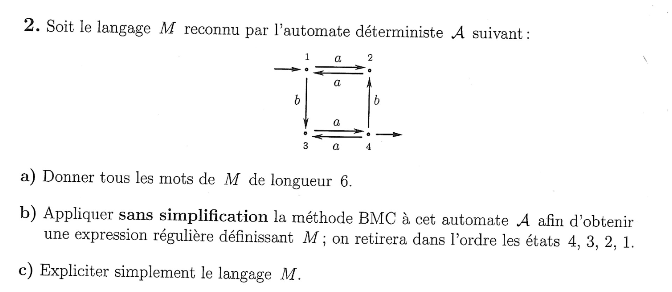
[L5 :](#_e7tehcmjwlvv)

# 

# Exercice 1a)



# Exercice 2

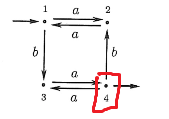


## a)

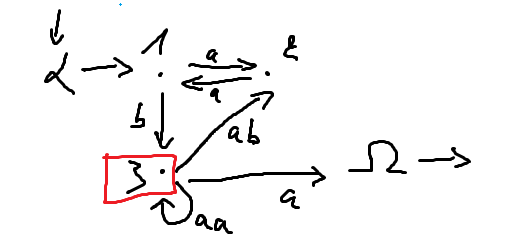
Aaaaba - aabaaa - baaaaa - bababa

## b)

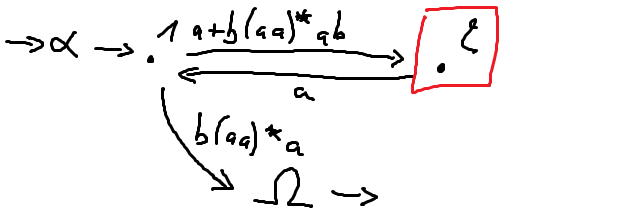
Starting point :



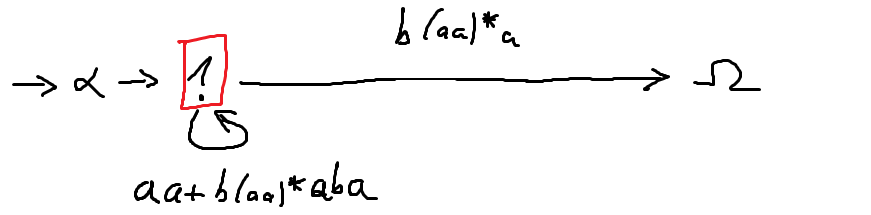
Step 1 : remove “4”



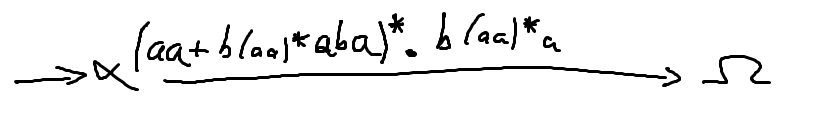
Step 2 : remove “3”



Step 3 : remove “2”



Final step : remove “1”

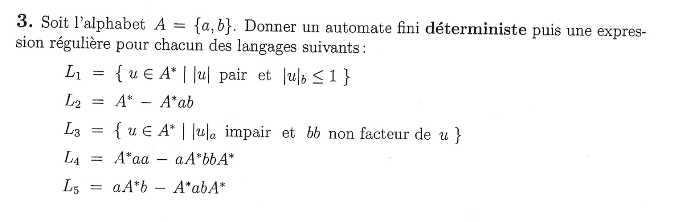


## c)

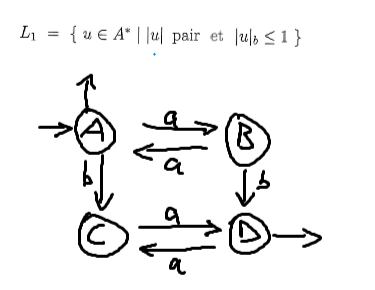
Tous les mots qui :

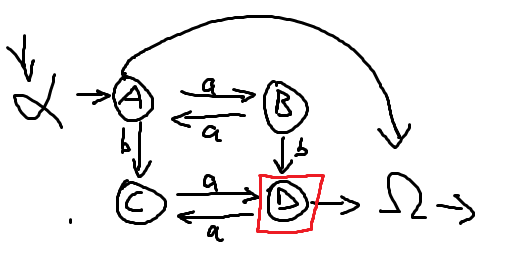
* N’ont pas 2b consécutifs ( bb non facteur de u )
* Ont un nombre pair de b et de a ( |u|b impair ET |u|a impair) donc |u| pair
* Où les b sont en position impair

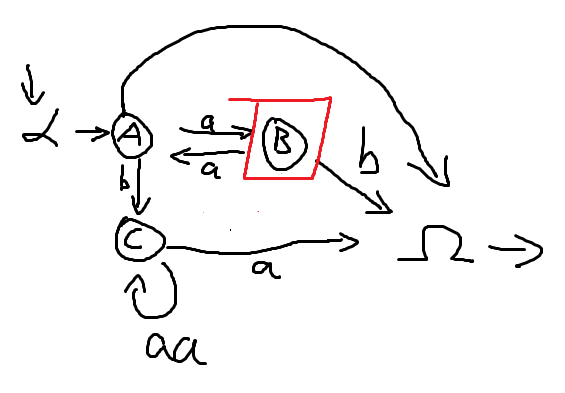
# Exercice 3

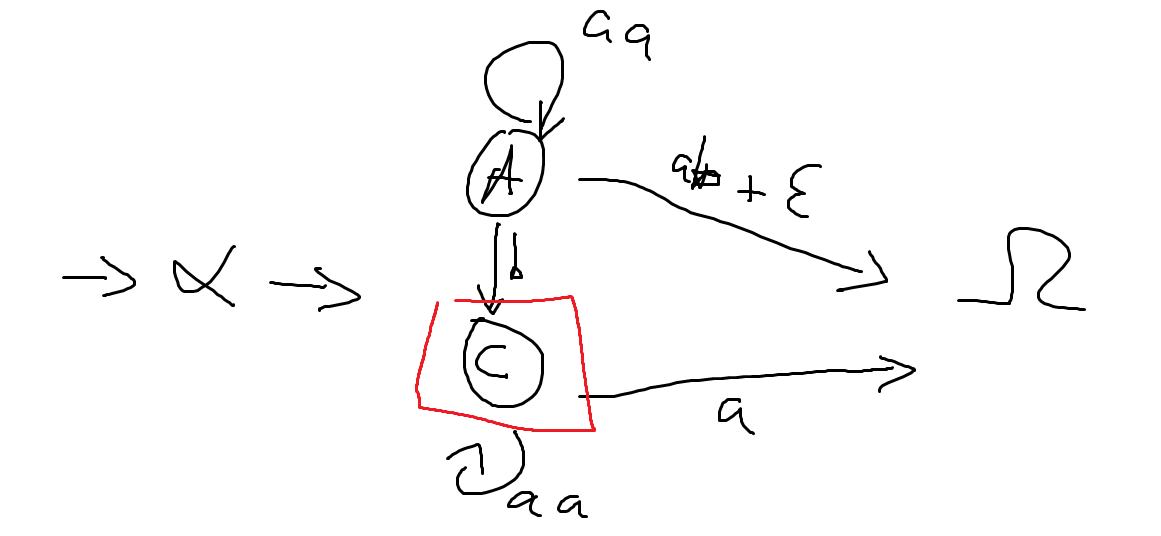


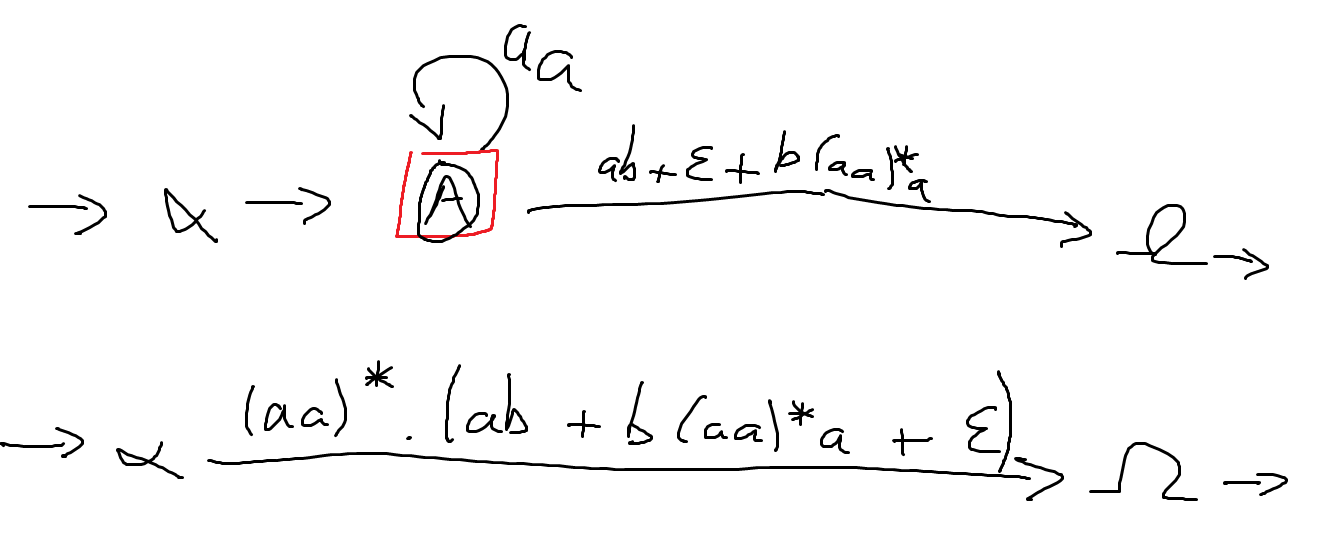
## L1 :





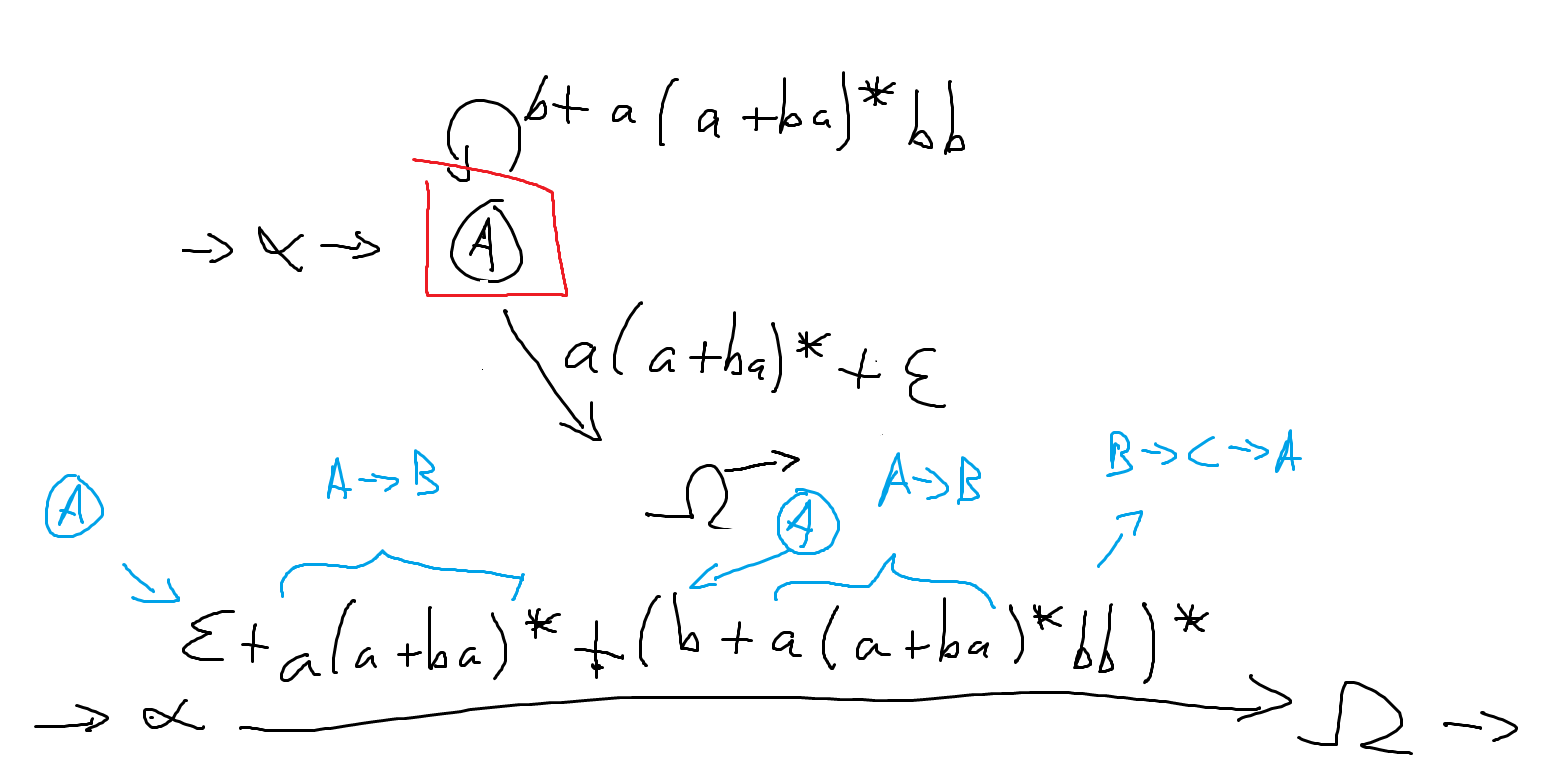




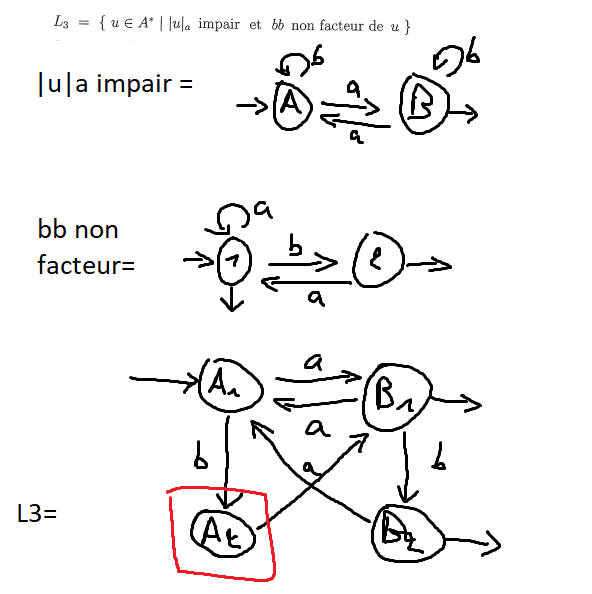


Expression : ( aa) \* . ( ab + b ( aa )\* a + E)

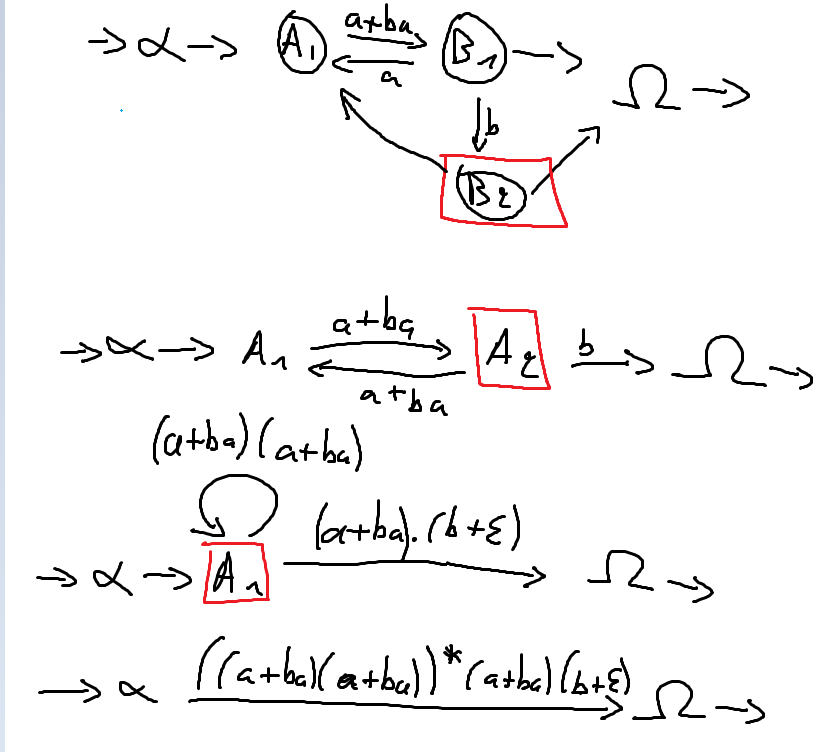
## L2 :



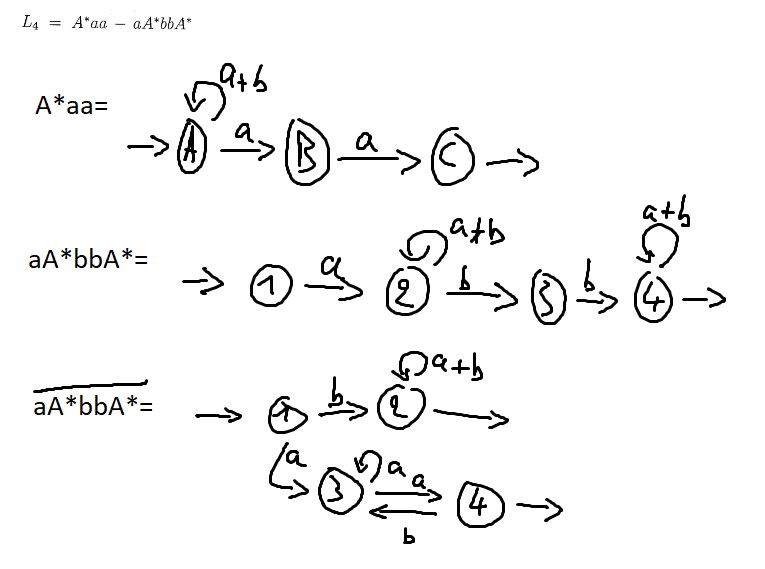
## L3 :

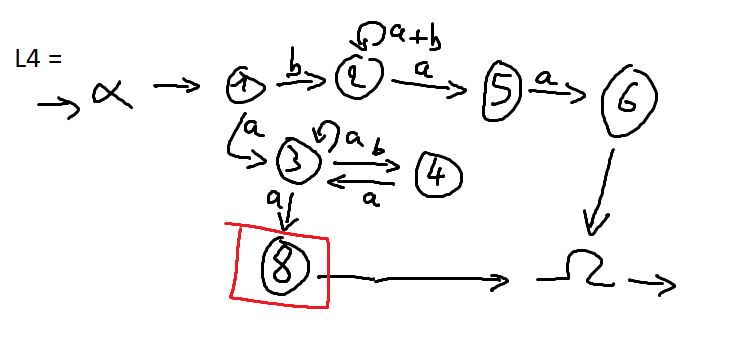


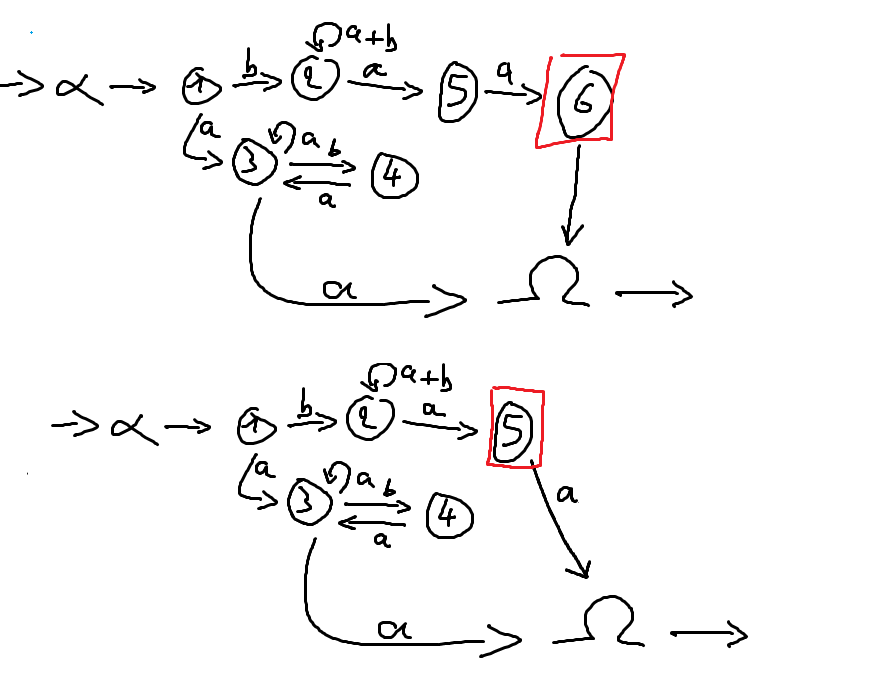
BMC en commençant par retirer A2

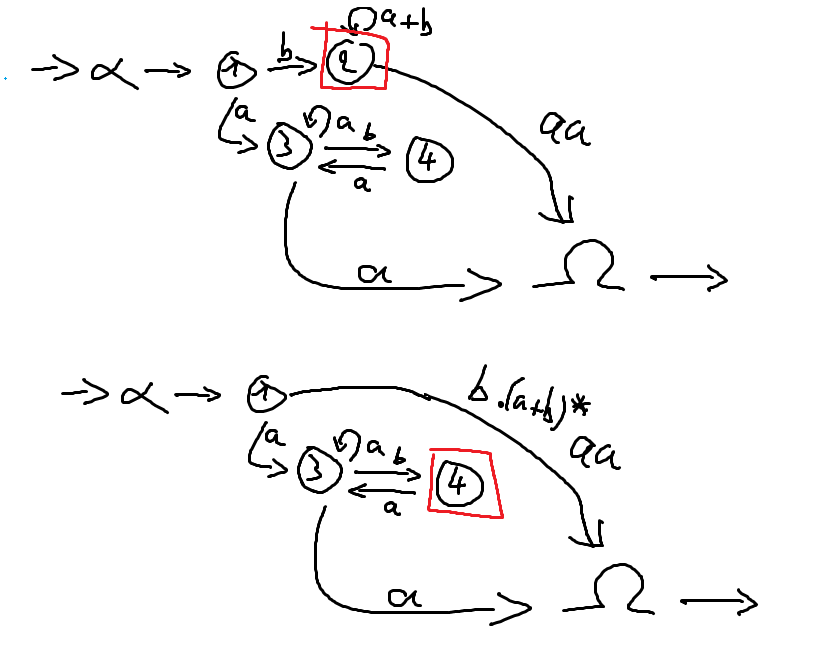


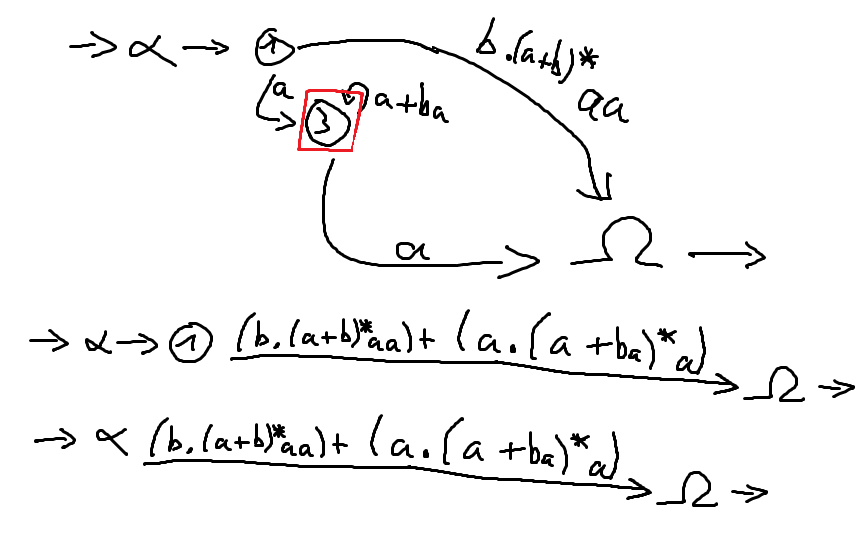
## L4 :











## L5 :



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