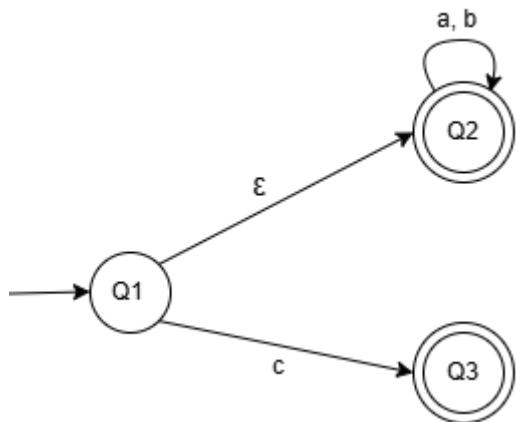
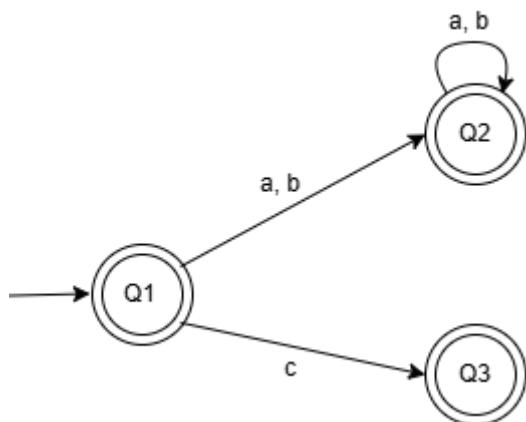


TEST .3

Expression Regulière : $c + (a+b)^*$



Suppression Epsilon-Transition :

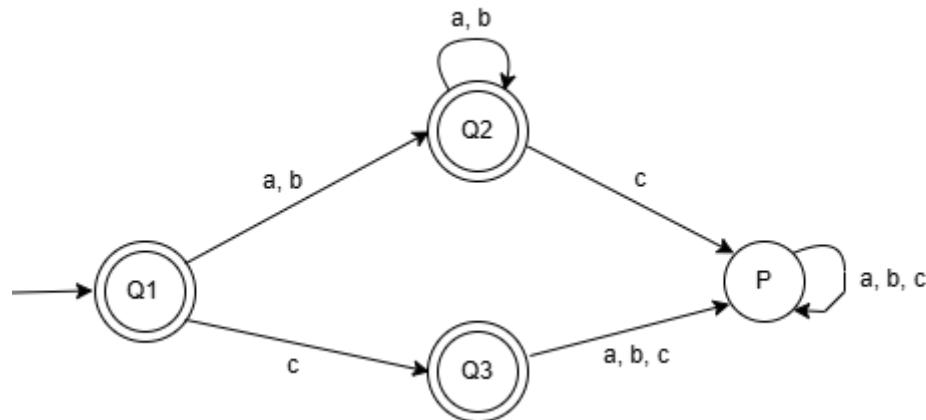


Q1 devient un état final car on a un chemin epsilon entre Q1 et Q2, et Q2 est final donc Q1 hérite du caractère final de Q2.

L'automate est un AFD, donc déjà déterministe.

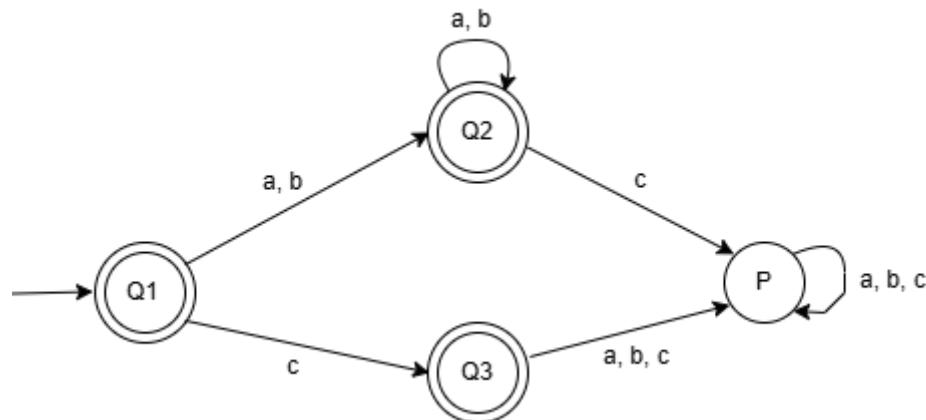
L'étape de la détermination est donc inutile puisqu'elle donnera le même automate.

Completion :

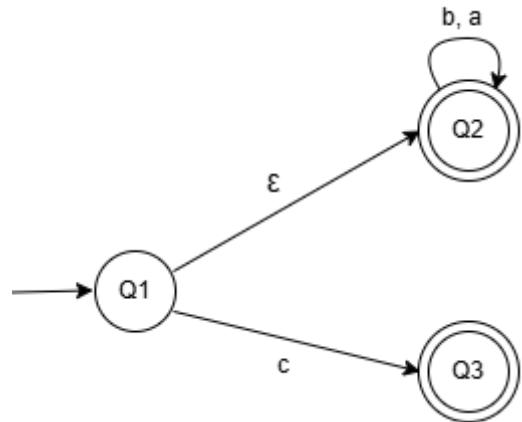


Minimisation :

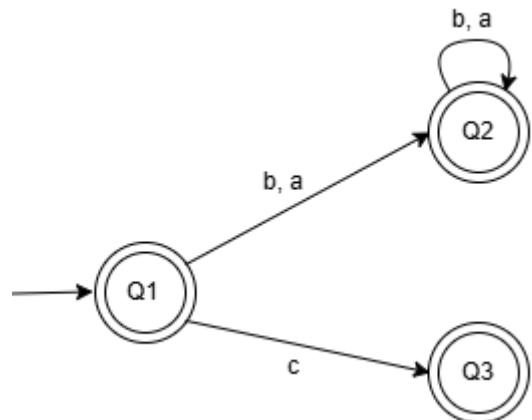
L'automate étant déjà très simple est déjà minimisé. Donc l'étape de la minimisation est inutile. Voici la version finale de l'automate :



Expression Régulière : $(b+a)^* + c$

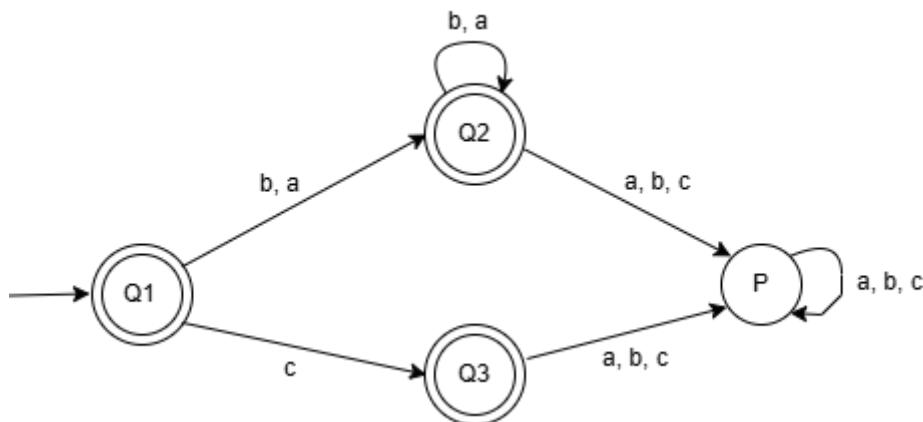


Suppression Epsilon-Transition :



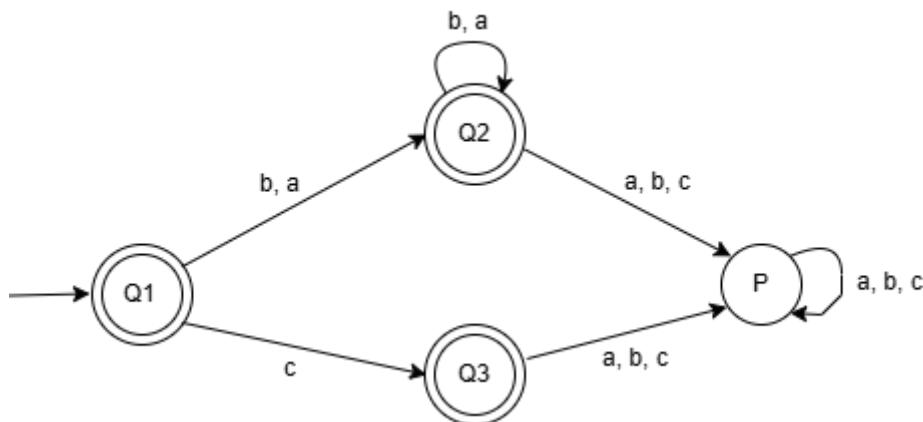
Q1 devient un état final car on a un chemin epsilon entre Q1 et Q2, et Q2 est final donc Q1 hérite du caractère final de Q2.

Completion :



Minimisation :

L'automate est déjà minimisé. Donc l'étape de la minimisation est inutile. Voici la version finale de l'automate :



$$c + (a+b)^* = (b+a)^* + c$$

OUI

Car les automates obtenus à partir de ces deux expressions régulières, après suppression chemin epsilon (quand il y en avait), déterminisation, complémentation et minimisation reconnaissent le même langage. Mise à part le nommage des états qui n'importe pas, les expressions régulières sont donc équivalents.