TD LFA

02/04/2020

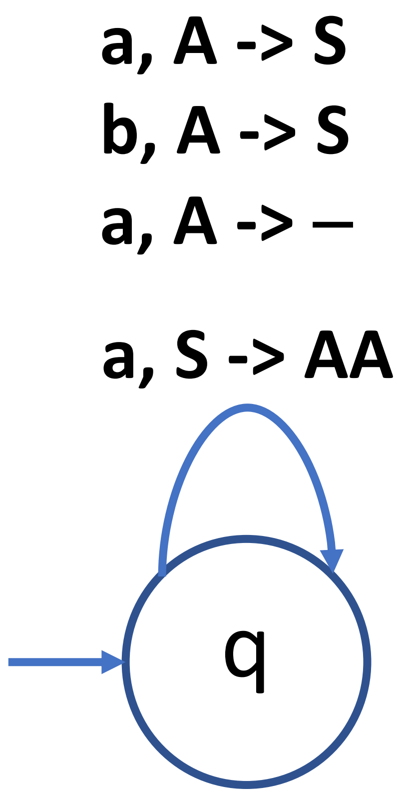
1. Construire un automate à pile qui accepte le langage engendré par la grammaire :

$N = \{S, A\}$

$T = \{a, b\}$

S

$$P = \left\{ \begin{array}{l} S \rightarrow aAA \\ A \rightarrow aS \mid bS \mid a \end{array} \right.$$



état	lecture	pile	nouvel état	empiler
q	a	S	q	AA
q	a	A	q	S
q	b	A	q	S
q	a	A	q	-

2. Sur l'alphabet $\Sigma = \{1, 2, +, =\}$, on considère l'ensemble des mots représentant une égalité numérique (vraie !). Par exemple :

- $1 + 1 = 2$
- $1 + 2 = 1 + 2$
- $1 + 2 + 1 = 2 + 2$

Montrer que ce langage est algébrique et construire un automate à pile qui l'accepte.

Une grammaire intuitive construirait l'égalité à partir du centre et vers les extrémités :

$S \rightarrow 1{+}S{+}1 \mid 2{+}S{+}2 \mid 1{=}1 \mid 2{=}2$

On se rend compte que cela ne suffit pas (exemple 1+1=2).

On l'améliore :

$S \rightarrow 1{+}S{+}1 \mid 2{+}S{+}2 \mid 1{=}1 \mid 2{=}2 \mid 1{+}1{+}S{+}2 \mid 2{+}S{+}1{+}1 \mid 1{+}1{=}2 \mid 2{=}1{+}1$

On se rend compte que cela ne suffit toujours pas (exemple 1+2+2=2+1+2)

Il faut changer d'idée !

On peut s’inspirer de la grammaire vu en TD n° 6, pour engendrer les mots ayant autant de «a» que de «b» !

Soient G (resp. D) les variables qui engendrent les mots ayant un de plus à gauche (resp. à droite).

$S \rightarrow 1+D \mid 2+D+1 \mid 2+S+2 \mid 1=1 \mid 2=2 \mid G+1 \mid 1+G+2$

$D \rightarrow S+1 \mid G+2 \mid 1=2$

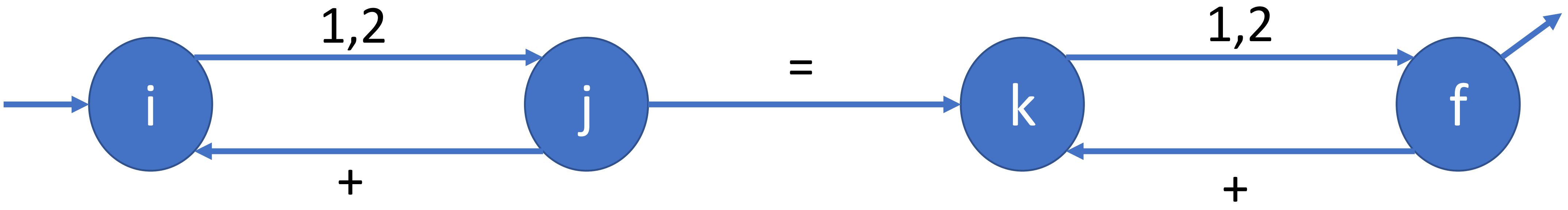
$G \rightarrow 1+S \mid 2+D \mid 2=1$

L’automate à pile doit vérifier deux choses :

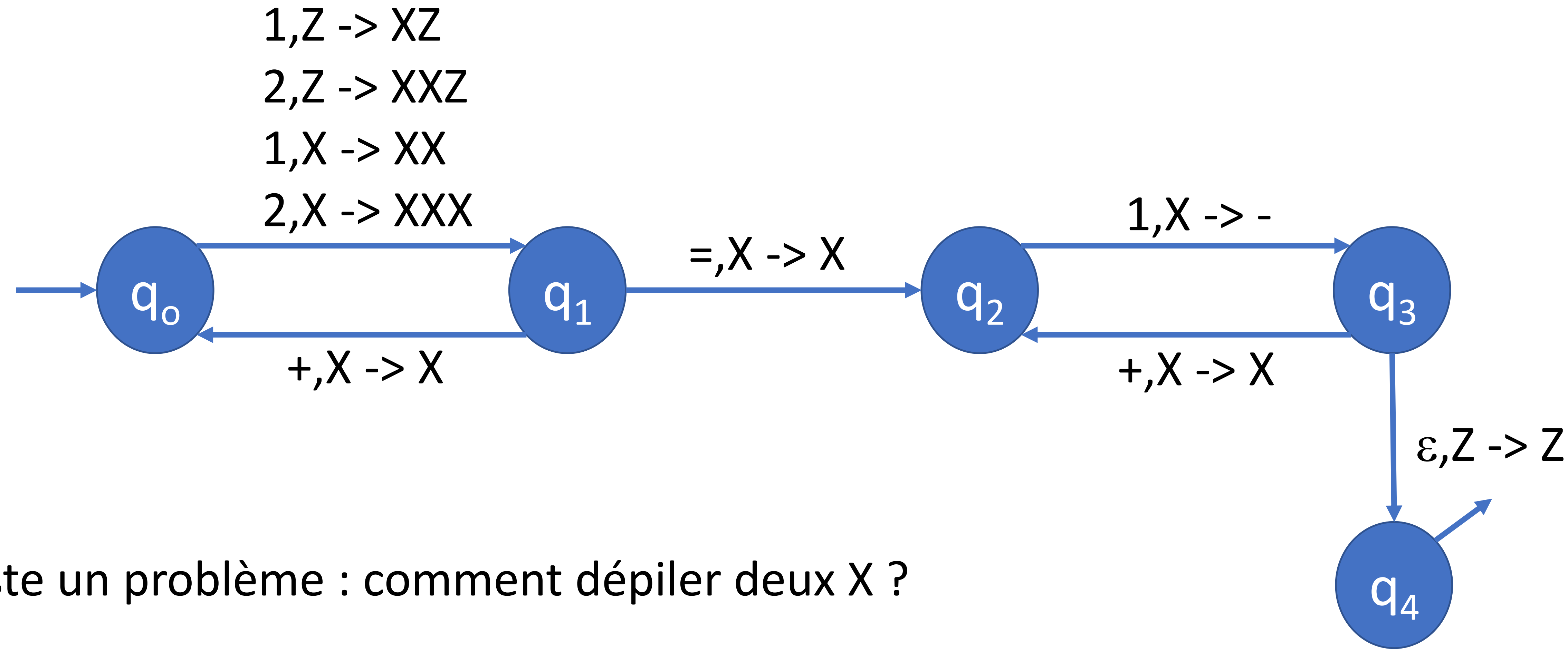
- syntaxe correcte
- somme correcte

Pour la syntaxe, la vérification est simple :

On doit avoir un chiffre suivi éventuellement d’un certain nombre de couples formés de + et chiffre, l’égal, et de nouveau un chiffre suivi éventuellement d’un certain nombre de couples formés de + et chiffre.

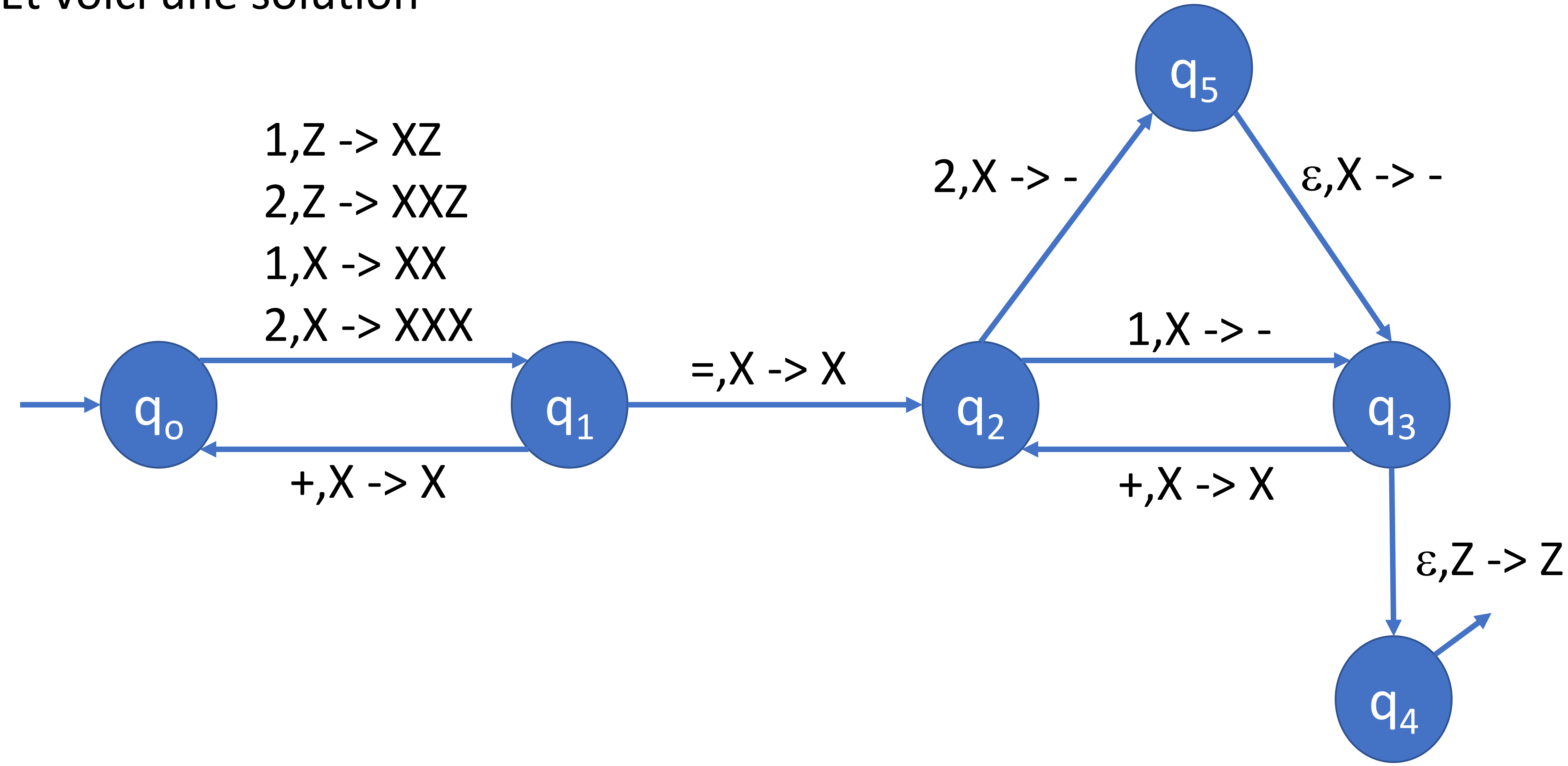


Pour compter et vérifier l'égalité, un méthode simple consiste à les compter en unaire.



Il reste un problème : comment dépiler deux X ?

Et voici une solution



état	lecture	pile	nouvel état	empiler
q_0	1	Z	q_1	XZ
q_0	2	Z	q_1	XXZ
q_0	1	X	q_1	XX
q_0	2	X	q_1	XXX
q_1	+	X	q_0	X
q_1	=	X	q_2	X
q_2	1	X	q_3	-
q_2	2	X	q_5	-
q_3	+	X	q_2	X
q_3	ε	Z	q_4	Z
q_5	ε	X	q_3	-

3. Soit $L = \{a^i b^j c^k \mid i < j < k\}$. Construire un automate à pile pour reconnaître le complémentaire de L.

Tout d'abord nous devons comprendre ce que c'est comme langage.

On peut définir ce langage comme l'union de plusieurs langages :

$L_1 = \{a^i b^j c^k \mid i \geq j\}$

$L_2 = \{a^i b^j c^k \mid j \geq k\}$

$L_3 = \{w \in (a+b+c)^* \mid w \notin a^*b^*c^*\}$

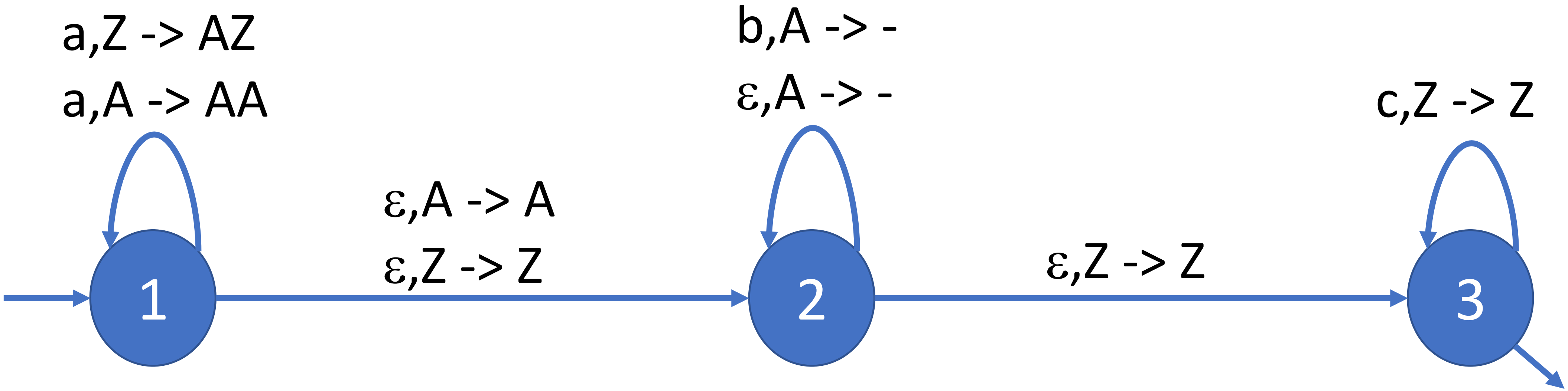
$L = L_1 \cup L_2 \cup L_3$

On peut traiter séparément ces langages.

$L_1 = \{a^i b^j c^k \mid i \geq j\}$

Alors soit $L_{11} = \{a^i b^j \mid i \geq j\}$ et $L_{12} = c^*$ et ainsi $L_1 = L_{11}.L_{12}$

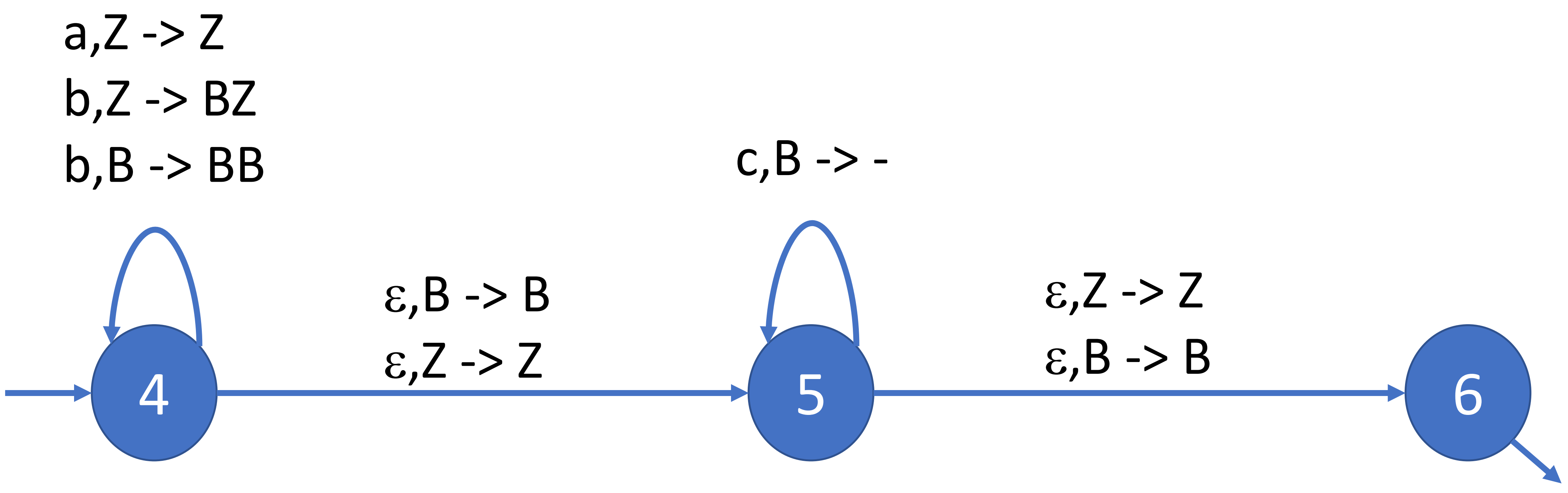
Ceci nous permet de donner facilement un automate à pile :



$L_2 = \{a^i b^j c^k \mid j \geq k\}$

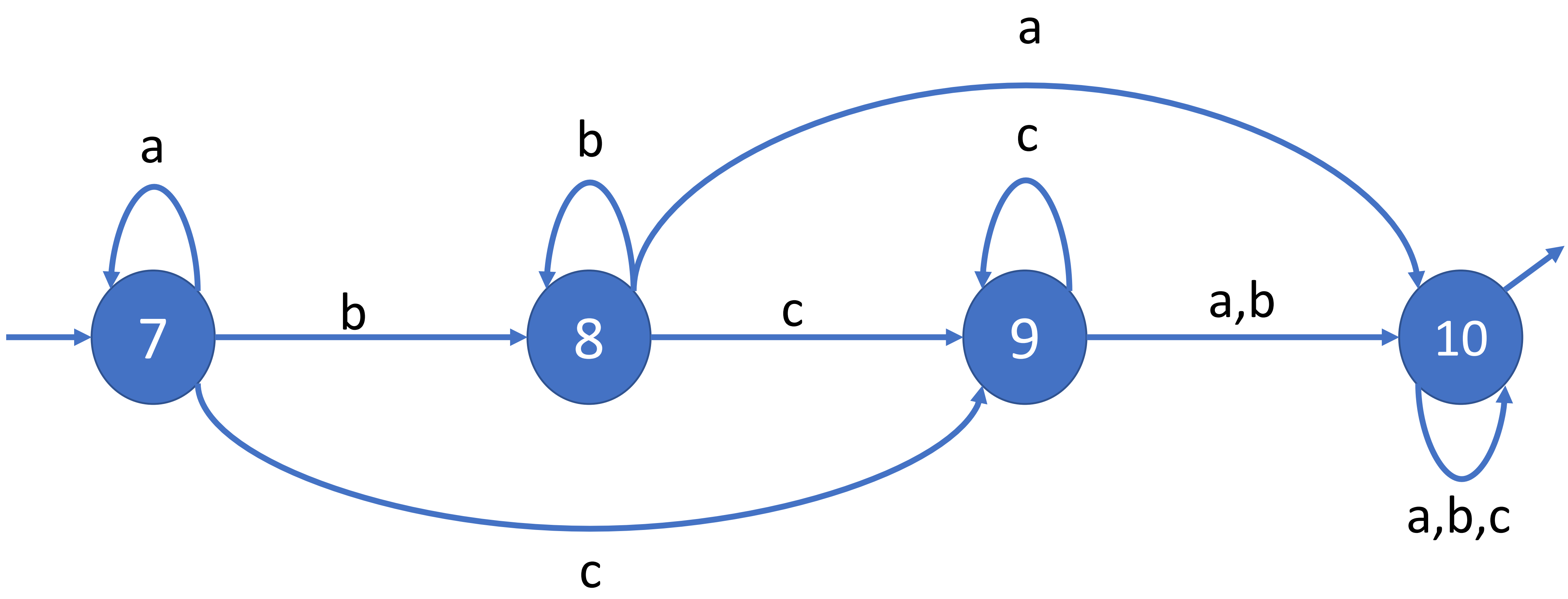
Alors soit $L_{22} = \{b^j c^k \mid j \geq k\}$ et $L_{21} = a^*$ et ainsi $L_2 = L_{21}.L_{22}$

Ceci nous permet de donner facilement un automate à pile :

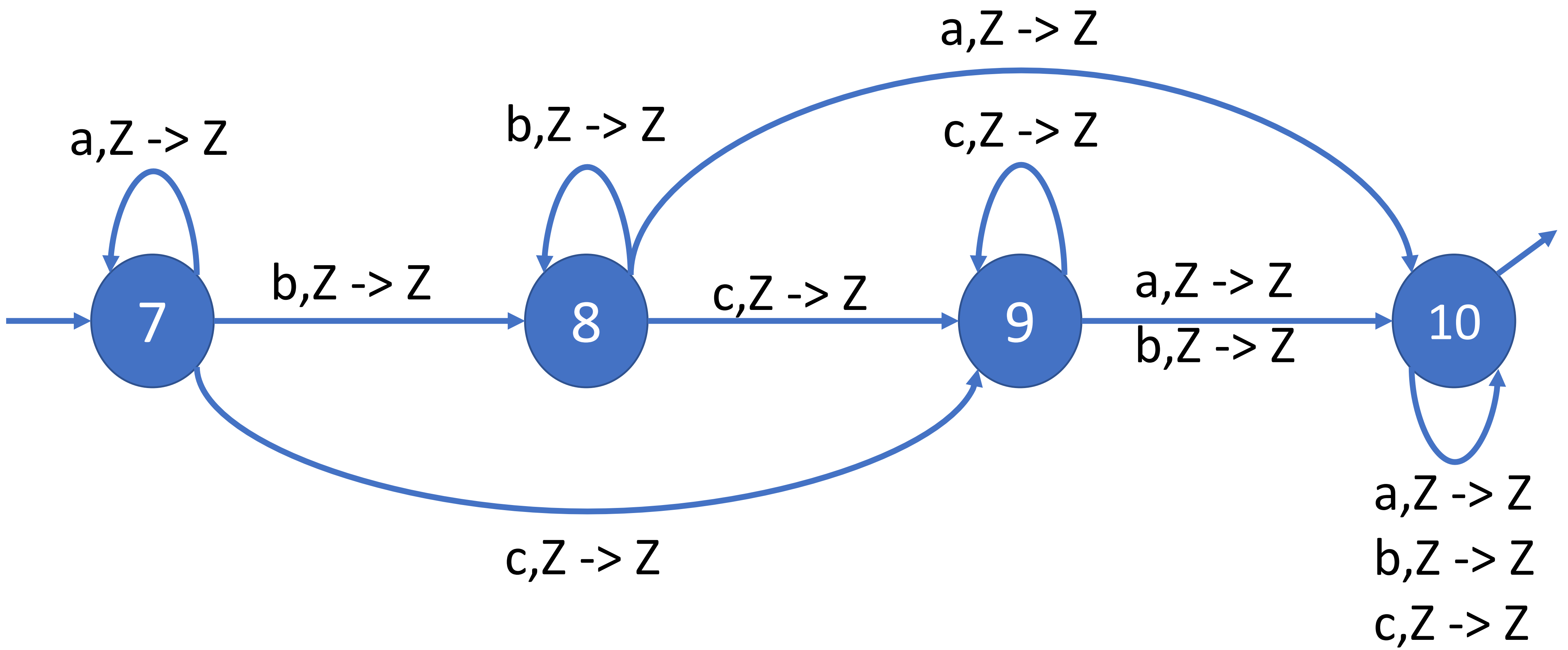


$$L_3 = \{w \in (a+b+c)^* \mid w \notin a^*b^*c^*\}$$

C'est un langage rationnel !!!



Et voici l’automate fini transformé en automate à pile :



Il suffit de cobiner les trois automates