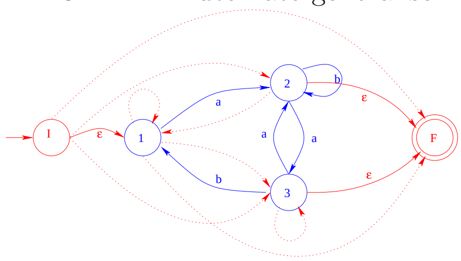
A.3 Corrigés

- $\boxed{\text{n}^o \text{ 1, p 19}}$

  Le point important est de bien déterminer l'ensemble des combinaisons à considérer. À partir de l'automate initial, on passe à l'automate généralisé représenté à la figure A.1 (arcs non passants représentés par des traits pointillés).

FIG. A.1 – Automate généralisé

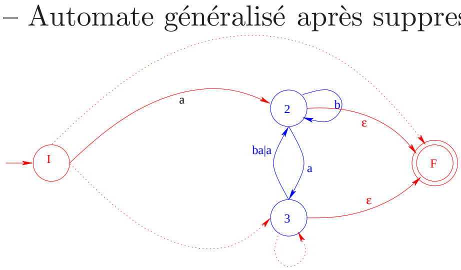


La réduction consiste donc à éliminer successivement les états de l'automate, en ne laissant que I et F. Commençons par la suppression de l'état 1. Le tableau suivant liste la totalité des triplets de la forme $(q_i, 1, q_j)$; pour chaque triplet, on construit l'étiquette de la transition (q_i, q_j) qui reconnaît le même langage. La troisième colonne donne la forme simplifiée de l'expression rationnelle[3].

I 1 I	*pas à considérer, I n'a pas d'arcs entrants*		
I 1 2	$\varepsilon\ \emptyset^*$	$a \mid \emptyset$	a
I 1 3	$\varepsilon\ \emptyset^*$	$\emptyset \mid \emptyset$	$\emptyset$
I 1 F	$\varepsilon\ \emptyset^*$	$\emptyset \mid \emptyset$	$\emptyset$
2 1 I	*pas à considérer, I n'a pas d'arcs entrants*		
2 1 2	$\emptyset\ \emptyset^*$	$a \mid b$	b
2 1 3	$\emptyset\ \emptyset^*$	$\emptyset \mid a$	a
2 1 F	$\emptyset\ \emptyset^*$	$\emptyset \mid \varepsilon$	ε
3 1 I	*pas à considérer, I n'a pas d'arcs entrants*		
3 1 2	$b\ \emptyset^*$	$a \mid a$	$ba \vert a$
3 1 3	$b\ \emptyset^*$	$\emptyset \mid \emptyset$	$\emptyset$
3 1 F	$b\ \emptyset^*$	$\emptyset \mid \varepsilon$	ε

Cette table donne directement le nouvel automate généralisé, débarrassé de l'état 1, mais reconnaissant le même langage (figure A.2).

FIG. A.2 – Automate généralisé après suppression de 1



On recommence alors pour l'état 2, le nombre de combinaisons à considérer est bien sûr

[3]Les définitions courantes de la sémantique du langage des expressions rationnelles donnent aisément, pour r une expression rationnelle quelconque : $\emptyset r = \emptyset$, $r\emptyset = \emptyset$, $\emptyset^* = \varepsilon$, et enfin $r\varepsilon = \varepsilon r = r$.

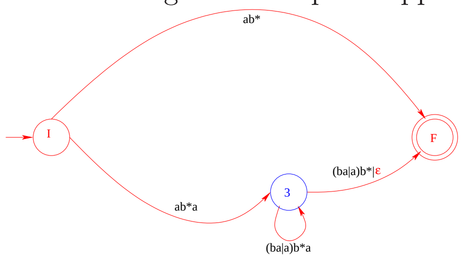
beaucoup plus réduit.

I 2 3	$a\ b^*$	$a \mid \emptyset$	ab^*a
I 2 F	$a\ b^*$	$\varepsilon \mid \emptyset$	ab^*
3 2 3	$ba\|a\ b^*$	$a \mid \emptyset$	$(ba\|a)b^*a$
3 2 F	$ba\|a\ b^*$	$\varepsilon \mid \varepsilon$	$(ba\|a)b^*\|\varepsilon$

TAB. A.1 – Les triplets à considérer impliquant l'état 2

L'automate résultant est représenté à la figure A.3.

FIG. A.3 – Automate généralisé après suppression de 2



Il ne reste plus qu'à supprimer l'état 3, ce qui se fait en appliquant directement la règle illustrée plus haut. L'expression résultante est :

$$\left(ab^*a\,((ba|a)b^*a)^*\,((ba|a)b^*|\varepsilon)\ \Big|\, ab^*\right)$$

– $\boxed{\text{n}^o\ 2,\ \text{p}\ 19}$

(sauf erreur)

$\mathcal{A}_1 \cap \mathcal{A}_2$	a	b	c
$\rightarrow$ 1A	2A		4D
2A		EB	
$\leftarrow$ 4D	2C		
3B			4B
2C			
4B		4C	
4C	2A		

– $\boxed{\text{n}^o\ 3,\ \text{p}\ 19}$

Automate initial

	0	1
$\rightarrow$ A	B	A
$\leftarrow$ B	B	A

Automate généralisé

	X	A	B	Y
X	/	ε	$\emptyset$	$\emptyset$
A	/	1	0	$\emptyset$
B	/	1	0	ε
Y	/	/	/	/

Suppression de A

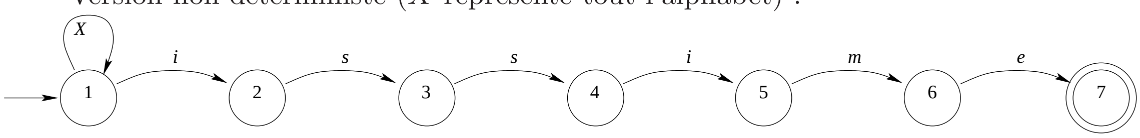
	X	B	Y
X	/	1*0	$\emptyset$
B	/	11*0|0	ε
Y	/	/	/

Suppression de B

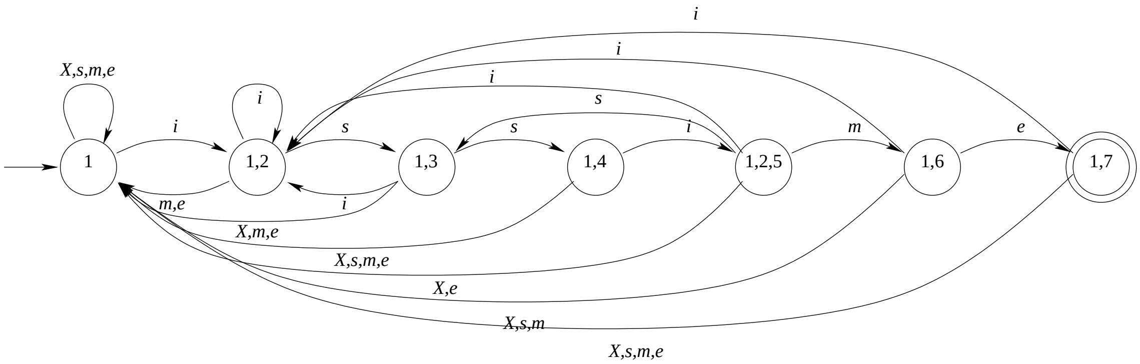
	X	Y
X	/	1*0(11*0|0)*
Y	/	/

– $\boxed{\text{n}^o\ 4,\ \text{p}\ 19}$

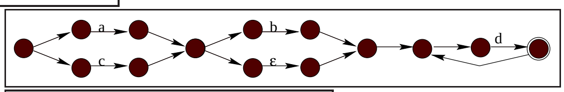
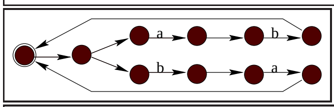
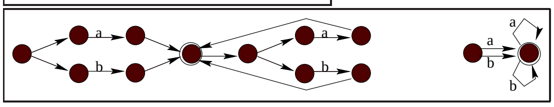
Version non déterministe (X représente tout l'alphabet) :



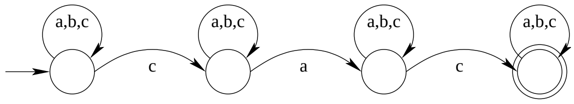
Version déterministe (et aussi complète) (X représente l'alphabet privé de i, m, s, e) :



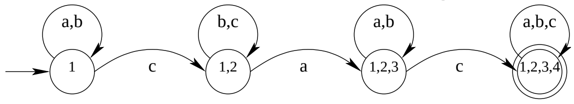
On peut vérifier facilement que cette version est minimale : chaque itération de l'algorithme sépare l'état le plus à droite de la classe de tous les autres.
Le suffixe u à reconnaître doit donner lieu à un chemin complet, de longueur égale à $|u|$. La reconnaissance du reste du mot, quant à elle, ne "coûte" qu'un état. Le nombre minimal d'états pour reconnaître un langage de la forme X^*u est donc $|u| + 1$.

- $\boxed{\text{n}^o \text{ 5, p 19}}$ à voir en séance
- $\boxed{\text{n}^o \text{ 1, p 20}}$ (1[re] série)
  - 
  - 
  - 

  (Transitions non étiquetées : ε-transitions.)
- $\boxed{\text{n}^o \text{ 2, p 20}}$ Le plus facile est d'envisager un automate non déterministe, construit autour du chemin *cac*. À chaque état, n'importe quelle lettre peut s'intercaler. Cela donne :



L'expression rationnelle correspondant est : $(a|b|c)^*c(a|b|c)^*a(a|b|c)^*c(a|b|c)^*$.
On peut très facilement déterminiser cet automate, l'algorithme donne :



Ce qui inspire l'expression rationnelle : $(a|b)^*c(b|c)^*a(a|b)^*c(a|b|c)^*$.

- $\boxed{\text{n}^o \text{ 3, p 20}}$ à voir en séance
- $\boxed{\text{n}^o \text{ 4, p 20}}$ La réponse est : oui, c'est légitime ! En effet, grâce au théorème de Kleene, on sait que l'on dispose d'une grammaire régulière équivalente à toute expression rationnelle. Il suffit de remplacer la partie droite β qui est sous forme d'une expression rationnelle par l'axiome d'une grammaire régulière qui engendre exactement le même langage que β. Attention, il faut noter que ces expressions rationnelles produisent (en général) un langage sur $(X \cup V)^*$ (et non sur X^*).
  Une grammaire qui reconnaît l'expression (construite en passant par l'automate) :
  $E \to Det\,X$; $X \to A\,X \mid N\,Y$; $Y \to A\,Y \mid Rel\,Y \mid \varepsilon$.
  La règle initiale s'écrit alors tout simplement $NP \to E$.