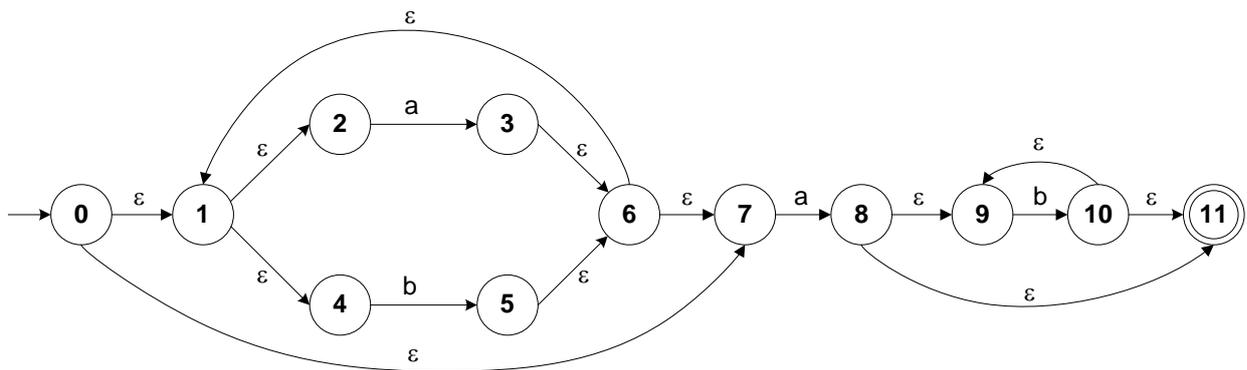


Exercice 1.

- Considérer l'expression régulière suivante : $(a|b)^* ab^*$
- a) Construire un automate d'états finis non déterministe pour l'expression régulière précédente en utilisant la construction de Thompson.
- b) Transformer cet automate d'états finis non déterministe en un automate d'états finis déterministe.
- c) Minimiser le nombre d'états de l'automate obtenu.

Solution :

- a) Construction d'AFN en utilisant la construction de Thompson pour l'ER $(a|b)^* ab^*$:

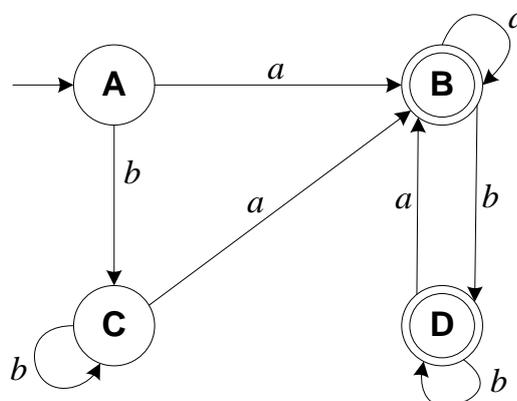


- b) Transformation AFN → AFD :

- Etat initial = ϵ -fermeture(0) = {0,1,2,4,7}

Groupe	Etat	a	B
{0,1,2,4,7}	A	B	C
{1,2,3,4,6,7,8,9,11}	B	B	D
{1,2,4,5,6,7}	C	B	C
{1,2,4,5,6,7,9,10,11}	D	B	D

- Les états d'acceptation : { B, D }

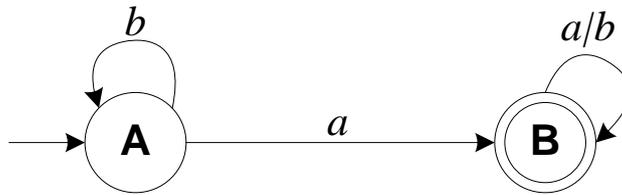


c) Minimisation :

1) $\Pi = \{A,C\} , \{B,D\}$

2) $\Pi = \{A,C\} , \{B,D\}$

Etat	a	b
A	B	A
B	B	B



AFD minimal pour l'ER $(a|b)^*ab^*$

Exercice 2.

- Les entités lexicales d'un mini-langage de programmation sont les suivantes :

Mots-clés	begin , end , if , then , else
Identificateurs	chaînes composées d'une lettre suivie de zéro ou plusieurs lettres ou chiffres
Constantes	Chaînes composées d'un chiffre suivi de zero ou plusieurs chiffres
Opérateurs	< , <= , = , <> , > , >= , + , -

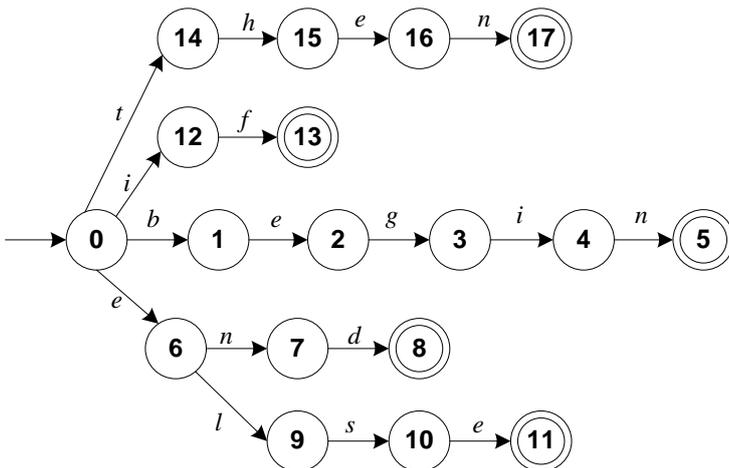
- Donner les expressions régulières qui décrivent ces entités lexicales.
- Construire un automate fini non déterministe pour ces expressions.

Solution :

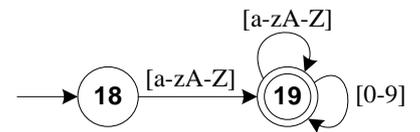
- Expressions régulières décrivant les entités lexicales:

- Mots-clés : **begin | end | if | then | else**
- Identificateurs : **$[A-Za-z]([A-Za-z0-9])^*$**
- Constantes : **$[0-9]^+$**
- Opérateurs : **$< | <= | = | <> | > | >= | + | -$**

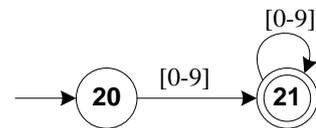
- Construire un automate à états finis pour ces expressions.



AF pour les mots clés

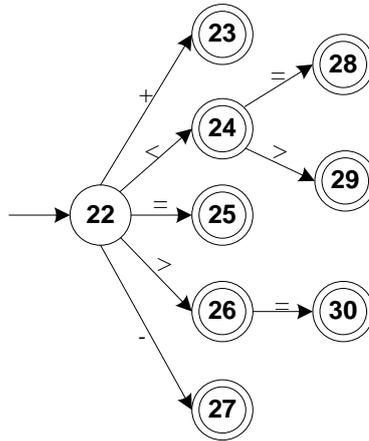


AF pour les identificateurs



AF pour les constantes entiers

AF pour les opérateurs

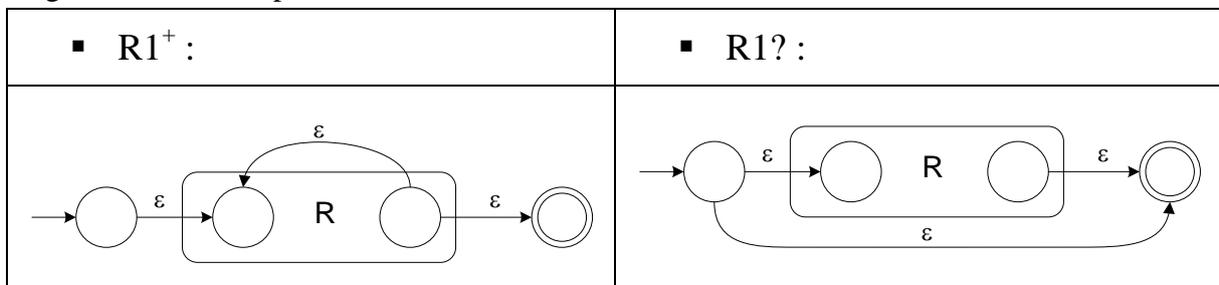


Exercice 3.

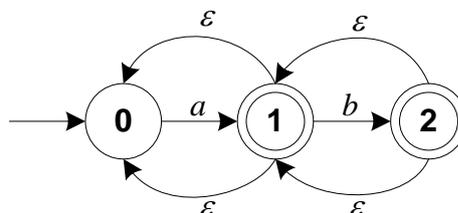
- Les règles de construction de Thompson transforment une expression régulière **R1** en un automate d'états finis **N1**. Proposer des règles analogue de construction d'automates d'états finis non déterministes pour les opérateurs suivants :
 - **R1⁺**
 - **R1?** (dont la signification est **R1|ε**)
- On modifie la règle de Thompson de construction de l'automate pour l'expression **R*** en ne rajoutant pas un état initial et un état final (on rajoute deux transitions étiquetées ε, l'une de l'état final de l'automate de R vers l'état initial ce cet automate et l'autre de l'état initial vers l'état final). Cette règle est-elle toujours valable pour l'expression **R***? Dans le cas général (composition des règles), la modification proposée affecte-elle la validité des constructions ? Donner un exemple concis pour justifier votre réponse.

Solution :

- Règles de Thomson pour **R1⁺** et **R1?** :



- Contre exemple : L'ER **a*b***: avec la nouvelle règle on aura l'AFN suivant :



Contradiction : on peut voir par exemple que la chaîne **ababab** peut être reconnu par cet AFN alors qu'elle n'appartient pas au langage engendré par l'ER d'origine: **a*b***.