

**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.

**Exercice 2.**

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

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

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

**Exercice 3.**

- Les règles de construction de Thompson transforment une expression régulière  $R1$  en un automate d'états finis  $N1$ . Proposer des règles analogues de construction d'automates d'états finis non déterministes pour les opérateurs suivants :
  - $R1^+$
  - $R1?$  (dont la signification est  $R1|\epsilon$ )
- 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  $\epsilon$ , 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.