Travaux Pratiques de Compilation n°1

Cours d'Informatique de l'École Centrale d'Électronique

—Ingénieur Quatrième Année—

L'objet de cette feuille est de mettre en place un programme ayant des fonctionnalités comparables à celles de la commande UNIX `grep` de recherche de motifs dans un texte.

Le système UNIX propose trois programmes de recherche fondés sur les expressions régulières : `grep`, `egrep` et `fgrep`. L'objectif de ces travaux pratiques concerne la mise en œuvre d'un programme de recherche de motifs dans un texte ayant un comportement voisin de celui de ces trois commandes standards. Pour cela, nous allons décomposer le développement du programme en plusieurs étapes :

- réalisation d'une bibliothèque de manipulation d'automates finis non déterministes ;
- développement d'un programme de construction d'un automate fini non déterministe à partir d'une expression régulière ;
- simulation d'un algorithme de reconnaissance de motifs par automate fini non déterministe.

▸ **Exercice 1.** (**automates finis non déterministes**) *On souhaite manipuler des expressions régulières. Pour cela, on demande de créer une bibliothèque de fonctions permettant de créer, manipuler et utiliser des automates finis non déterministes codant des expressions régulières.*

Ainsi, vous devrez tout d'abord choisir une structure de données pour représenter un automate fini non déterministe et puis vous devrez créer tout un ensemble de fonctions élémentaires de manipulations de cette structure. Ces fonctions élémentaires devront inclure les fonctions :

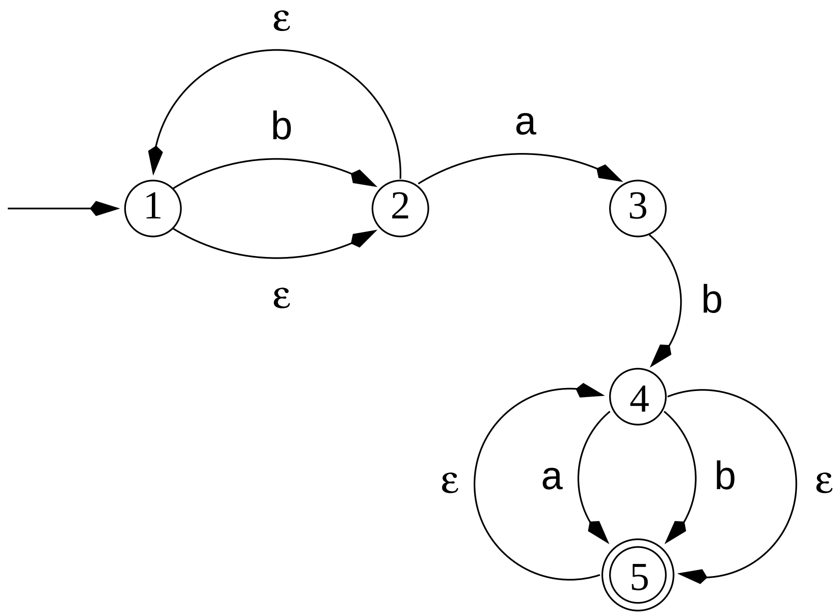
- *de création d'un automate reconnaissant une lettre ;*
- *de création d'un automate reconnaissant $L(r)L(s)$ à partir des automates reconnaissant respectivement $L(r)$ et $L(s)$;*
- *de création d'un automate reconnaissant $L(r) \cup L(s)$ à partir des automates reconnaissant respectivement $L(r)$ et $L(s)$;*
- *de création d'un automate reconnaissant $L(r)^*$ à partir de l'automate de $L(r)$;*
- *de création d'un automate reconnaissant $L(r)^+$ à partir de l'automate de $L(r)$;*
- *de création d'un automate reconnaissant $L(r)?$ (zéro ou une chaîne de $L(r)$) à partir de l'automate de $L(r)$.*

▸ **Exercice 2.** (**expressions régulières**) *À l'aide de la bibliothèque précédente, vous donnerez une fonction de création d'un automate fini non déterministe à partir d'une expression régulière donnée sous forme d'une chaîne de caractères.*

Pour cela, vous définirez une grammaire des langages réguliers que vous souhaitez manipuler (qui comprendra au moins les opérations décrites dans l'exercice précédent) et utiliserez une méthode d'analyse prédictive pour la création de l'automate fini non déterministe à partir de l'expression régulière d'entrée.

La fonction de construction de l'automate devra être construite à l'aide des outils standards de mise en place d'un analyseur lexical et d'un analyseur syntaxique.

À titre indicatif, le programme pourra par exemple traiter une chaîne ressemblant à "`b*ab(a|b)*`"*—correspondant à l'expression régulière* $b^*ab(a \mid b)^*$*—afin de créer l'automate :*



▸ **Exercice 3.** (**reconnaissance par automate**) *Enfin, vous devrez produire un programme de recherche de motifs dans un fichier, programme qui utilisera les fonctions développées précédemment. Afin de pouvoir faire "passer" un texte dans un automate, vous devrez concevoir une fonction qui lira le texte ligne par ligne et affichera la ligne si celle-ci contient un motif reconnu par l'automate construit.*

Pour la phase de reconnaissance, vous utiliserez une méthode visant à simuler la reconnaissance par automate fini non déterministe à l'aide de deux piles. L'idée est d'empiler au départ l'état initial dans la première pile, puis de calculer dans la deuxième pile, pour tous les états de la première pile, les ϵ*-fermetures des transitions par la lettre lue à partir de ces états. On recommence en échangeant le rôle des deux piles. En fin de lecture d'une ligne, la ligne contient un motif reconnu si la dernière pile mise à jour contient au moins un état final de l'automate.*