

Analyse lexicale : expressions régulières, automates d'états finis

Exercice 1

Soit $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$ l'alphabet des chiffres arabes. En utilisant des expressions régulières étendues, donner des expressions rationnelles dénotant les langages suivants :

- Les séquences non vides de 1
Correction : 1^+
- Les chiffres pairs (2 expressions possibles)
Correction : $[24680]$ et $[\wedge 13579]$
- Les nombres pairs (les zéros superflus sont autorisés)
Correction : $[0 - 9]^*[24680]$
- Les nombres ne se terminant pas par 2
Correction : $[0 - 9]^*[\wedge 2]$
- Les nombres sans les zéros superflus
Correction : $[\wedge 0][0 - 9]^*[0]$

Exercice 2

Sur l'alphabet $\{a, b\}$, donner, lorsque c'est possible, une expression rationnelle et un automate fini pour chacun des langages suivants. Pour les automates, on précisera si la solution proposée est déterministe ou pas, et si elle ne l'est pas on essaiera de la modifier pour que l'automate devienne déterministe.

Correction : Seules les expressions rationnelles sont indiquées ci-dessous. Il n'y a pas (pas encore) de correction pour les automates.

1. $L = \{\text{ensemble des mots } w \text{ commençant par } ab\}$;
Correction : $ab(a|b)^*$
2. $L = \{\text{ensemble des mots } w \text{ se terminant par } bb\}$;
Correction : $(a|b)^*bb$
3. $L = \{\text{ensemble des mots } w \text{ commençant par } ab \text{ et se terminant par } bb\}$;
Correction : $ab(a|b)^*bb|abb$
point délicat : le cas abb
4. $L = \{\text{ensemble des mots } w \text{ contenant au moins trois occurrences successives de } a\}$;
Correction : $(a|b)^*aaa(a|b)^*$
5. $L = \{\text{ensemble des mots } w \text{ ne commençant pas par } ba\}$;
Correction : $(bb|a)(a|b)^*|b|\epsilon$
points délicats : les cas a , b et ϵ
6. $L = \{\text{ensemble des mots } w \text{ de longueur paire, y compris le mot de longueur } 0\}$;
Correction : $(aa|ab|ba|bb)^*$
7. $L = \{\text{ensemble des mots } w \text{ ne contenant pas deux } a \text{ consécutifs}\}$;
Correction : $(ab|b)^*(a|\epsilon)$
point délicat : le cas du a tout seul

8. $L = \{\text{ensemble des mots } w \text{ ne contenant ni deux } a \text{ consécutifs ni deux } b \text{ consécutifs}\}$;

Correction : $(ab)^*(a|\epsilon)|(ba)^*(b|\epsilon)$

points délicats : les cas a et b tout seul

9. $L = \{\text{ensemble des mots } w \text{ de la forme } w = a^n b^n\}$;

Correction : pas de solution mais “exprimable” à l’aide d’une grammaire que l’on verra plus tard

10. $L = \{\text{ensemble des mots } w \text{ contenant autant de } a \text{ que de } b\}$;

Correction : pas de solution mais “exprimable” à l’aide d’une grammaire

11. $L = \{\text{ensemble des mots } w \text{ de la forme } w = a^n b^p\}$;

Correction : a^*b^*

Exercice 3

On s’intéresse ici à quelques expressions régulières associées à des langages que l’on trouve en informatique :

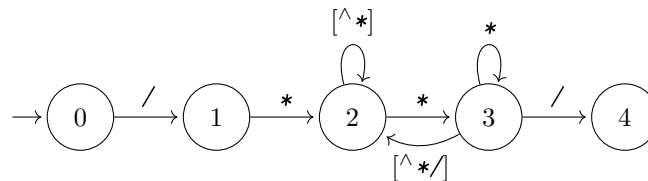
Soit A l’alphabet formé des caractères ascii visibles, on utilisera les notations usuelles pour les caractères alpha numériques $[a - z]$, $[A - Z]$, $[0 - 9]$ et $[\text{^}xyz]$ pour exclure des caractères.

Donner des expressions régulières étendues, dénotant les langages suivants et proposer un automate pour reconnaître le même langage :

- Les commentaires qui débutent par $/*$ et se terminent à la première apparition de la séquence $*/$ (attention à ne pas confondre le caractère d’entrée $*$ et le symbole utilisé pour faire l’étoile d’une expression régulière)

Correction : $/*([\text{^} *][\text{^} /])^*+/*$

Une petite attention à avoir sur le fait que l’on demande en interne que le caractère $*$ soit suivi par autre chose que le caractère $/$ mais il ne faut pas exclure une fin de la forme $*/$.



- Les chaînes de caractères qui débutent et terminent par le caractère `"`. Une chaîne de caractères ne contient pas de retour à la ligne (on prend la convention que le caractère **LF** est le caractère ASCII pour le retour à la ligne). Pour inclure dans la chaîne le caractère guillemet ou des retours à la ligne, on utilise un codage avec deux caractères, le caractère `\` qui est appelé caractère d’échappement suivi d’un autre caractère distinctif. Ainsi `\"` représente un caractère guillemet dans la chaîne, `\n` un retour à la ligne, et `\\` pour inclure le caractère `\` dans la chaîne.

Correction : $"([\text{^} LF\backslash] | \backslash["n\backslash])^*"$

- Lorsqu’on entre un mot de passe, il est souvent vérifié qu’il contient une lettre minuscule, une majuscule, un chiffre et un caractère spécial comme `&`, `$`, `@`, `,`... également une longueur minimale est requise.

Par contre l’ordre dans lequel ces éléments apparaissent est libre ce qui rend l’écriture d’une expression régulière complexe.

On peut par contre raisonner de manière plus directe avec des automates, si on pense chaque état comme représentant quels sont parmi les critères requis ceux qui sont déjà vérifiés.

Dans l’état initial, aucun des critères n’est vérifié, par contre dès qu’on entre une minuscule on peut passer à l’état où le critère minuscule est satisfait et si on lit ensuite une majuscule, on passera dans l’état où les deux critères majuscule et minuscule sont satisfaits. On peut

aussi arriver au même état (mais par un autre chemin) si la majuscule apparaît avant la minuscule.

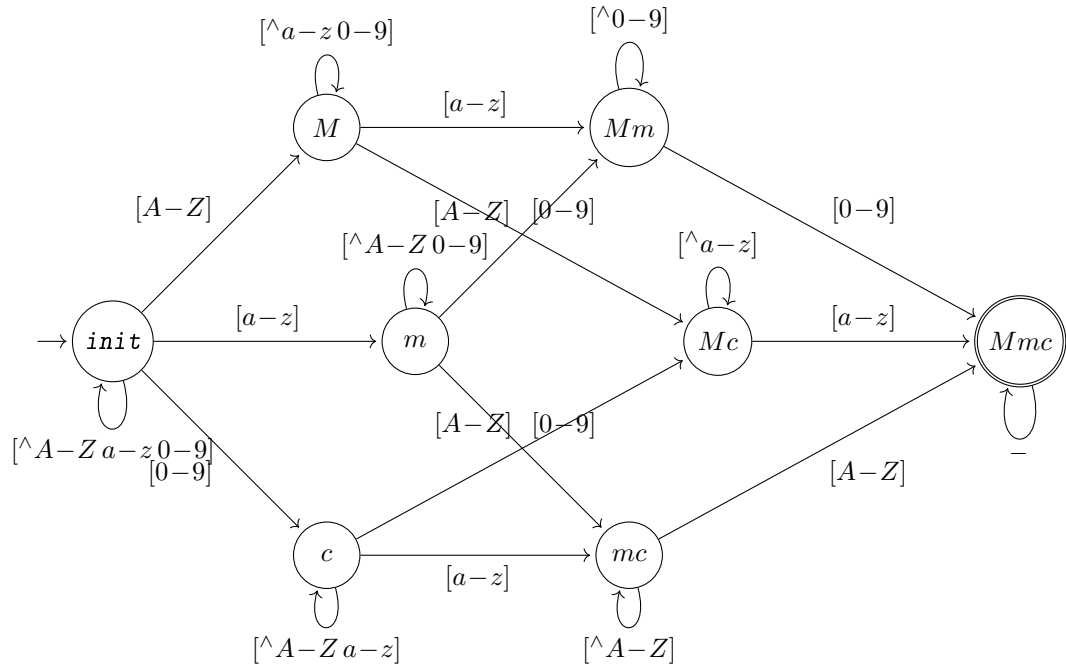
On s'intéresse ici à seulement trois critères : majuscule, minuscule et chiffre

- Combien faut-il d'états pour représenter toutes les situations possibles de respect des 3 critères ?
- Quel sont les états terminaux ?
- Construire l'automate correspondant

Correction :

(a) Il y a 3 critères donc $2^3 = 8$ états

(b) Un seul état terminal dans lequel les trois critères ont été vérifiés



(c)