

info623 : Théorie des langages, calculabilité
TD 1 : expressions régulières

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : <http://www.lama.univ-savoie.fr/~hyvernat/>

Exercice 1 : expressions régulières POSIX

Rappels : les motifs de fichiers d'un shell POSIX ("glob") sont plus simples que les expressions régulières, cf `$ man 7 glob` :

- "*x*" pour un caractère donné,
- "***" pour "n'importe quelle chaîne de caractères",
- "?" pour "un unique caractère (quelconque)",
- "[...]" pour "un caractère parmi ...",
- "[!...]" pour "un caractère qui n'est pas parmi ...".

Les expressions régulières POSIX étendues utilisent

- "*x*" pour un caractère donné,
- "." pour un caractère quelconque (sauf le saut de ligne),
- "[...]" pour "un caractère parmi ...",
- "[^...]" pour "un caractère qui n'est pas parmi ...".
- "*reg**" pour répéter l'expression *reg* un nombre arbitraire de fois,
- "*reg+*" pour répéter l'expression *reg* un nombre arbitraire de fois, mais au moins une fois,
- "*reg?*" pour la chaîne vide ou bien l'expression *reg*,
- "*reg|reg*" pour la première ou la deuxième expression,
- "*reg{m,n}*" pour l'expression *reg* répétée au moins *m* fois, et au plus *n fois*.

Les caractères "^" et "\$" dénotent le début et la fin d'une chaîne.

Pour tester une expression régulière POSIX étendue, vous pouvez utiliser la commande

```
$ grep --color -E '...'
```

Après chaque ligne que vous entrez, vous verrez :

- rien si la chaîne ne contient aucune sous-chaîne appartenant au langage de votre expression
- la ligne que vous avez entrée si elle contient une sous-chaîne appartenant au langage de votre expression. Une de ces sous-chaîne est affichée en couleur.

Question 1. Écrivez une expression régulière POSIX dont le langage associé correspond :

- aux chaînes se terminant par ".jpg",
- aux chaînes se terminant par ".jpg" ou ".jpeg",
- aux chaînes se terminant par ".jpg", mais sans prendre en compte les majuscules. Autrement dit, "image-1.Jpg" est dans le langage associé,
- aux chaînes se terminant par ".jpg", ".gif" ou ".png", sans prendre les majuscules en compte.

Pouvez-vous écrire les motifs shell correspondants ?

Question 2. Écrivez une expression régulière POSIX dont le langage associé correspond aux chaînes de la forme "projet-bak-*n*.tgz", où "*n*" est un nombre positif.

Pouvez-vous écrire le motif shell correspondant ?

Question 3. Écrivez une expression régulière POSIX correspondant à une chaîne qui ne contient aucun symbole "\"" non échappé. Autrement dit, chaque "\"" doit être précédé d'un "\\".

Question 4. Écrivez une expression régulière correspondant aux nombres flottants du langage C.

Question 5. Les motifs étendus de bash¹ autorisent le motif “!(...|...|...)” pour dénoter une chaîne qui ne correspond à aucun des motifs “...”.

En supposant que le répertoire courant contienne uniquement les fichiers `test.jpg`, `test.jpeg`, `test-1.gif`, `test-1.1.gif` et `test-1.2.gif`, qu’affichent les commandes suivantes :

- `$ ls *!(????)`
- `$ ls *!(.jpeg|.jpg)`
- `$ ls *!(gif)`

Exercice 2 : Expressions régulières de Kleene

Rappel : les expressions régulières de Kleene ne peuvent utiliser que les symboles

- “*s*” pour un symbole de l’alphabet,
- “ \emptyset ” (parfois noté \emptyset) pour l’expression dénotant le langage vide,
- “1” (parfois noté ε) pour l’expression dénotant la chaîne vide (notée ε),
- “+” pour une disjonction
- “.” ou une simple juxtaposition pour concaténer des expressions,
- “_” pour l’étoile de Kleene.

Question 1. Donnez des expressions régulières sur l’alphabet $\Sigma = \{a\}$ pour les langages suivants :

- les mots de longueur multiple de trois,
- les mots non-vides de longueur multiple de trois,
- les mots de longueur multiple de trois ou de longueur multiple de cinq,
- les mots dont la longueur peut-être obtenue en ajoutant des 3 et des 5.

Question 2. Écrivez une expression régulière sur l’alphabet $\Sigma = \{a, b, c\}$ dont le langage associé est exactement l’ensemble des mots où *a* est toujours suivi de *b* et *b* est toujours suivi de *a*, sauf éventuellement pour le dernier symbole du mot.

Question 3. Écrivez une expression régulière sur l’alphabet $\Sigma = \{a, b\}$ dont le langage associé est exactement l’ensemble des mots qui ne contiennent jamais deux *a* consécutifs.

Question 4. Écrivez une expression régulière sur l’alphabet $\Sigma = \{a, b, c\}$ dont le langage associé est exactement l’ensemble des mots qui ne contiennent ni deux *a* consécutifs ni deux *b* consécutifs. (Difficile...)

Question 5. En regardant les mots des langages associés, montrez que les expressions régulières suivantes sont équivalentes :

- “ R^* ” et “ $(R^*)^*$ ” ;
- “ $(R_1R_2)^*$ ” et “ $1 + R_1(R_2R_1)^*R_2$ ” ;
- “ $(R_1 + R_2)^*$ ” et “ $(R_1^*R_2)^*R_1^*$ ” ;
- “ R^* ” et “ $(1 + R)(RR)^*$ ”.

Question 6. Rappelez la définition inductive (vue en cours) utilisée pour tester si une expression régulière contient le mot vide ε .

Est-ce que les expressions régulières suivantes contiennent le mot vide ε ?

- $(a + ba^*)^* + b(a + (b + aba)^*)^*$
- $(1 + b)(aa^* + bb^*a)^*$
- $(1 + a)(1 + b)(1 + c)(1 + d)(e + f)$
- $(a + (b + (c + d)^*)^*)^*$

Question 7. Donnez une définition inductive similaire pour tester si le langage d’une expression régulière est vide.

¹ pour les activer, il faut peut-être lancer la commande `$ shopt -s extglob`