Projet grep

L'objectif de ce projet est de programmer une version rudimentaire mais utilisable de la commande grep (voir son fonctionnement dans le TP3). Pour ce faire, vous utiliserez le langage de programmation de votre choix parmi C et Ocaml et les notions vues dans le cours sur les automates finis. Ce projet est facultatif, il n'a pas de date de rendu : vous pourrez me montrer vos avancées et me poser vos questions quand vous le souhaitez.

Dans un premier temps au moins, on acceptera les limitations suivantes (vous êtes libres de lever au fur et à mesure ces restrictions et d'enrichir la diversité des expressions régulières étendues que votre programme acceptera une fois qu'il fonctionnera dans les conditions suivantes) :

- Le programme prendra en entrée (en plus de l'expression régulière) un éventuel fichier passé en argument ou l'entrée standard sinon.
- Le programme traitera le flux d'entrée ligne par ligne et décidera pour chacune si la ligne dans son ensemble est une instance de l'expression régulière choisie : les lignes acceptées seront affichées sur la sortie standard.
- Les seuls opérateurs permettant de construire une expression régulière étendue seront dans un premier temps : la concaténation, qu'on écrira explictement avec le symbole `@`, l'union (représentée par |), l'étoile (représentée par `*`) et le "zéro ou une fois ce caractère" (représenté par `?`).
- Pour construire une expression régulière étendue, on n'aura le droit dans un premier temps que d'utiliser les opérateurs ci-dessus, les caractères ASCII (sauf ceux réservés, voir ci-dessous) et le point dont la sémantique sera "n'importe quel caractère est autorisé à cet emplacement".
- Les caractères parmi `@ | * ? .` seront réservés et ne pourront donc jamais être matchés (autrement dit, on n'implémentera pas dans un premier temps la possibilité de rechercher le caractère '?' en l'échappant via `\?` comme c'est le cas lorqu'on utilise grep).
- L'expression régulière étendue donnée au progamme sera écrite en notation postfixe.

Par exemple, en supposant que le programme écrit a été compilé vers un exécutable du nom de mygrep, on devrait avoir les correspondances suivantes (au sens où, pour chaque ligne du tableau, les deux commandes devraient produire le même résultat) :

Avec grep	Avec mygrep
`grep -E '^ab*$' entree.txt`	`mygrep 'ab*@' entree.txt`
`grep -E '^(ab)*$' entree.txt`	`mygrep 'ab@*' entree.txt`
`grep -E '(ab)*c$' entree.txt`	`mygrep '.*ab@*@c@' entree.txt`
`grep -E '([ab].)*'`	`mygrep '.*ab|.@*.*@@'`