

Commentaires DS1

Remarques générales :

Présentation des copies : il y a encore quelque copie trop brouillonnes ou rédigées "à la va vite". Cela fait mauvaise impression ...

Les principales difficultés : l'énoncé était assez proche du cours, mais plusieurs notions ont posé problème :

- regexp : ici le + représentait le OU
- $\text{Pal}(\Sigma)$: l'ensemble des palindromes (avec l'alphabet Σ). Pour $\Sigma = \{a,b\}$: $\text{Pal}(\Sigma) = \{e, a, b, aa, bb, aba, bab, aaa, bbb \dots\}$. Ce langage n'est pas régulier (Q8).
- Σ^2 : l'ensemble des mots de longueur 2
- $(\Sigma^2)^*$: l'ensemble des mots de longueur paire
- $\text{numero}(X)$: codage en binaire de l'ensemble X

Des **confusions** trop fréquentes entre :

- automate et langage
- mot et expression régulière

Trop de points "bêtement" perdus en OCaml, sur des erreurs de "débutants" :

- confusion liste / tableau
- tentative de return à l'intérieur d'une boucle for
- tentative de modification de (variable, éléments d'une liste, champ non mutable)
- ...

Beaucoup d'algorithmes "trop compliqués" :

- perte de temps pendant le DS,
- parfois de perte de points,
- et c'est une mauvaise habitude de s'en satisfaire

Gestion du temps :

- l'exercice est souvent "bien payé" par rapport au problème. Eviter de ne pas le traiter, sans pour autant y consacrer trop de temps.
- c'est bien de vouloir se frotter à des questions "difficiles", mais il est beaucoup plus rentable de traiter correctement les questions "faciles".

Exercice 1 : on travaille sur un arbre n-aire en C

Q1. Renvoyer un pointeur vers une variable locale (allocation dans la pile) pose problème : l'adresse devient invalide après l'appel. On alloue dans le tas pour éviter ce problème.

Q2. Ne pas oublier de créer les noeuds et branches lorsqu'ils sont absents

Q3. Il est possible que certaines branches soient absentes

Q4. Un découpage en 2 fonctions est attendu : descente dans l'arbre puis comptage de tous les noeuds du sous-arbre

Q5. Ne pas oublier de détruire tous les noeuds créés.

Problème : plusieurs parties :

- 1- langage / automates autour des palindromes
- 2- détermination d'automates
- 3- simplification d'expressions régulières
- 4- automate des dérivés d'Antimirov

Q1-Q2 : OK Q3 : OK pour la construction. On attend une justification

Q4 : enregistrements en OCaml : de trop nombreuses erreurs ...

Q5 : $O(n)$ ne signifie pas grand chose si on ne sait pas ce qu'est n

Q6 : Aller jusqu'à l'indice $n/2 - 1$ est suffisant (pour $n = 5$: on compare les lettres (0-4) et (1-3) ; pour $n = 4$: on compare les lettres (0-3) et (1-2))

Q7 : une explication simple est attendue : "si l'alphabet n'a qu'une lettre, tous les mots sont des palindromes"

Q8 : lemme de l'étoile Q9 : $(\Sigma^2)^*$ souvent mal compris

Q11 - Q12 : questions assez difficiles

Q13-Q14-Q15 : pour la détermination, indiquer les ensembles d'états. Le résultat final est rarement juste

Q16 : on attend une justification

Q17 - Q18 : listes en OCaml : de trop nombreuses erreurs ...

Q19 à Q25 : une fois qu'on a compris $\text{numero}(X)$, les questions sont tout à fait abordables

Q30 à Q33 : plutôt facile, quelques cas particuliers posent problème

Q34 : faire l'appel récursif sur la racine peut provoquer une boucle infinie. Il faut faire les appels récursifs sur les sous-arbres

Q35-Q36 : on adapte les algorithmes de somme et produit matriciel mathématiques

Q37 : il s'agit de décrire les langages reconnus par des automates assez simples

Q44 à Q50 : difficile, avec beaucoup de notations à assimiler. Possible une fois qu'on a compris l'opération de dérivation