Option Informatique en Spé MP et MP*

Devoir surveillé du mardi 9 décembre 2003

Palindromes !

Résumé

Un mot est *palindrome* lorsqu'il se lit indifféremment de gauche à droite ou de droite à gauche. Le caractère palindrome est une forme de régularité connue depuis longtemps ; nous nous proposons d'en explorer quelques aspects.

Nous étudierons un algorithme construisant en temps linéaire la liste des préfixes palindromes d'un mot donné. Nous nous intéresserons ensuite au problème de la factorisation d'un mot en produit de palindromes.

Veuillez rédiger chaque partie sur une copie séparée.

Table des matières

1 Première approche

▶ Dans tout le texte, Σ désigne un alphabet (fini) contenant au moins deux lettres.

▶ Soit u un mot de longueur $n \geqslant 1$ sur Σ : $u = u_1 u_2 \dots u_n$. Soient i et j deux indices vérifiant $1 \leqslant i \leqslant j \leqslant n$; nous noterons $u[i..j]$ le mot $u_i u_{i+1} \dots u_{j-1} u_j$.

▶ Le miroir u^R du mot u est le mot $u_n u_{n-1} \dots u_2 u_1$. Le miroir du mot vide ε est lui-même. Remarquons que $(uv)^R = v^R u^R$. Soit L un langage ; nous noterons $L^R = \{u^R \mid u \in L\}$; il est bien connu que L^R est rationnel ssi L l'est.

▶ u est *palindrome* s'il est égal à son miroir ; un palindrome sera dit *trivial* s'il est de longueur 0 ou 1. Nous noterons Pal l'ensemble des palindromes. Remarquons qu'un palindrome de longueur paire est de la forme uu^R, tandis qu'un palindrome de longueur impaire est de la forme uxu^R où x est une lettre. Nous dirons qu'un palindrome est *pair* (resp. *impair*) lorsque sa longueur l'est.

Question 1 • Notons $p(n)$ le nombre de palindromes de longueur n sur l'alphabet Σ. Donnez une expression *simple* de $p(n)$.

Question 2 • Montrez que le langage Pal n'est pas rationnel.

Question 3 • Rédigez en Caml une fonction de signature

```
est_palindrome : string -> bool
```

spécifiée comme suit : `est_palindrome s` indique si la chaîne de caractères s est palindrome. Rappel : les caractères d'une chaîne sont indexés à partir de 0 ; l'accès au i-ième caractère de s se fait avec `s.[i]` ; la fonction `string_length` rend la longueur d'une chaîne. Remarques : vous n'avez pas le droit d'utiliser des listes ; et le nombre de comparaisons entre caractères de s doit être minimal.

▶ Dans les deux questions suivantes (qui ne doivent être traitées que par les 3/2), $\mathcal{Z}$ désigne l'ensemble des mots de la forme $vv^R w$, où v et w sont des mots sur Σ tous deux non vides.

Question 4 • Peut-on utiliser le lemme de l'étoile pour montrer que le langage $\mathcal{Z}$ est reconnaissable ?

Question 5 • Peut-on utiliser le lemme de l'étoile pour montrer que le langage $\mathcal{Z}$ n'est pas reconnaissable ?

Question 6 ⋆⋆⋆ • Soit $\mathcal{A}$ un automate fini à n états reconnaissant un langage rationnel L. Montrez que si L contient un palindrome, alors L contient un palindrome de longueur inférieure à $2n^2$. Indication : inspirez-vous de la preuve du lemme de l'étoile.

2 L'algorithme de Manacher

▶ Dans cette partie, nous étudions deux algorithmes déterminant tous les préfixes palindromes d'un mot u.

Question 7 • Proposez un algorithme naïf, de coût quadratique.

Question 8 • Rédigez en Caml une fonction de signature

```
préfixes_palindrome : string -> int list
```

spécifiée comme suit : `préfixes_palindrome s` dresse la liste des longueurs des préfixes palindromes de la chaîne de caractères s. Vous pourrez vous inspirer de la fonction rédigée à la question 3.

▶ Nous allons maintenant décrire un algorithme de coût linéaire.

▶ Soient u un mot et i une position dans ce mot : $1 \leqslant i \leqslant |u|$. Notons $\mathrm{Ri}(i)$ le plus grand j tel que $1 \leqslant i - j$, $i + j \leqslant |u|$ et $u[i+1..i+j]$ soit le miroir de $u[i-j..i-1]$. Ainsi, $u[i-j..i+j]$ est un palindrome impair, dont nous dirons qu'il est *centré* en i et de rayon $\mathrm{Ri}(i)$; remarquons que ceci n'a d'intérêt que si $j > 0$.

Question 9 • Soient i une position dans le mot u, et k un indice tel que $1 \leqslant k \leqslant \mathrm{Ri}(i)$ et $\mathrm{Ri}(i-k) < \mathrm{Ri}(i) - k$. Montrez que $\mathrm{Ri}(i+k) = \mathrm{Ri}(i-k)$.

Question 10 • Soient i une position dans le mot u, et k un indice tel que $1 \leqslant k \leqslant \mathrm{Ri}(i)$ et $\mathrm{Ri}(i-k) > \mathrm{Ri}(i) - k$. Montrez que $\mathrm{Ri}(i+k) = \mathrm{Ri}(i) - k$.

▶ Il y a un problème si $\mathrm{Ri}(i-k) = \mathrm{Ri}(i)-k$: dans ce cas, nous pouvons simplement affirmer que $\mathrm{Ri}(i+k)$ est au moins égal à $\mathrm{Ri}(i-k)$. Il convient ensuite de déterminer si le rayon du palindrome centré en $i+k$ est plus grand, par observation directe. Si l'égalité se produit pour k proche de $\mathrm{Ri}(i)$, ce n'est pas gênant ; en revanche, si elle se produit pour k proche de 1, nous ne pouvons pas bénéficier du «remplissage automatique» de la table Ri fourni par les résultats des deux questions précédentes. Nous allons examiner en détail le cas où $\mathrm{Ri}(i-1) = \mathrm{Ri}(i) - 1$; pour alléger, nous noterons $r = \mathrm{Ri}(i)$. Nous pouvons également supposer $i - r > 1$ et $i + r < |u|$, quitte à remplacer u par $\alpha u \beta$, où α et β sont deux lettres distinctes prises hors de Σ.

Question 11 • Montrez que $u[i-r..i+r] = (ab)^r a$, où $a = u_{i-r}$ et $b = u_{i-r+1}$.

Question 12 • Supposons $a = b$ et $u_{i+r+1} \neq a$: expliquez comment remplir les cases d'indice $i+1$ à $i+r$ du tableau Ri.

Question 13 • Supposons $a = b$ et $u_{i+r+1} = a$. Montrez l'existence d'un indice $q > i + r$ tel que les cases d'indice $i+1$ à q peuvent être remplies au prix d'une comparaison de caractères au plus par case.

Question 14 • Expliquez rapidement comment régler le cas $a \neq b$.

▶ L'analyse précédente nous donne les grandes idées d'une méthode de construction de la table des $\mathrm{Ri}(i)$ pour $1 \leqslant i \leqslant |u|$ en temps linéaire :

- nous initialisons $\mathrm{Ri}(1)$ avec la valeur 0 ;
- soit i le premier indice d'une case non remplie de $\mathrm{Ri}(i)$: par la méthode banale, nous calculons $r = \mathrm{Ri}(i)$; puis, si $r \geqslant 1$, nous appliquons les résultats des questions 9 et 10 tant que c'est possible ;
- si nous tombons sur le cas d'égalité, nous appliquons la technique esquissée aux questions 12, 13 et 14.

Question 15 • Appliquez cette méthode au mot $u = abaaaaabaa$; vous expliquerez soigneusement chaque étape.

▶ Notons $\mathrm{Rp}(i)$ le plus grand j tel que $1 \leqslant i-j+1$, $i+j \leqslant |u|$ et $u[i+1..i+j]$ soit le miroir de $u[i-j+1..i]$. Si $j > 0$, alors $u[i-j+1..i+j]$ est un palindrome pair ; nous dirons qu'il est *calé* sur i. Les $\mathrm{Rp}(i)$ peuvent être calculés en même temps que les $\mathrm{Ri}(i)$, en exploitant la même formule.

Question 16 • En déduire un algorithme calculant en temps linéaire tous les préfixes palindromes de u.

Question 17 ⋆⋆ • Proposez un algorithme qui, en temps linéaire, décide si un mot u est le produit de deux palindromes non triviaux ; en cas de réponse positive, l'algorithme donne une factorisation de u.

3 Factorisation en produit de palindromes pairs

▶ Un *palstar* est un produit de palindromes non triviaux. Voici un exemple de palstar, avec deux des factorisations possibles :

$$u = abaaabaaaaabbaa = (abaaba)(aa)(aabbaa) = (aba)(aba)(aaaa)(bb)(aa)$$

Un *palstar pair* est un produit de palindromes pairs.

▶ Soit u un palstar pair. Notons v le plus court préfixe palindrome pair de u apparaissant comme premier facteur dans une factorisation de u en produit de palindromes pairs ; notons également w le plus court préfixe palindrome pair de u. Clairement, w est préfixe de v ; si nous définissons n et p par $|v| = 2n$ et $|w| = 2p$, alors $1 \leqslant p \leqslant n$.

Question 18 • Montrez que l'hypothèse $n \geqslant 2p$ mène à une contradiction.

Question 19 • Montrez que l'hypothèse $p < n < 2p$ mène également à une contradiction.

Question 20 • Qu'en déduisez-vous ?

Question 21 • Décrivez un algorithme *glouton* qui, appliqué à un mot u de longueur paire :

- décide si u est un palstar pair ;
- en cas de réponse positive, donne une factorisation de u en produit de palindromes pairs.

4 Factorisation en produit de palindromes quelconques

► L'algorithme qui vient d'être décrit s'étend à la factorisation en palindromes quelconques, avec une complication que nous allons mettre en évidence.

Question 22 • Que se passe-t-il si nous appliquons l'algorithme glouton de la partie 3 au mot *abababb* ?

Question 23 • Que se passe-t-il si nous appliquons ce même algorithme au mot *aabaaaa* ?

► Notons v le plus court préfixe palindrome (pair ou impair) de u apparaissant comme premier facteur dans une factorisation de u en produit de palindromes de longueur quelconque. Par ailleurs, notons w le plus court préfixe palindrome (pair ou impair) de u. Notons $n = |v|$ et $p = |w|$; il est clair que $p \leqslant n$. Nous allons montrer que n est l'un des trois nombres p, $2p - 1$ ou $2p + 1$.

Question 24 • Montrez que $n = 2p$ mène à une contradiction.

Question 25 • Montrez que $n > 2p + 1$ mène à une contradiction.

Question 26 • Montrez que $p < n < 2p - 1$ mène à une contradiction ; vous distinguerez deux cas selon la parité de n.

► Soit u un mot de longueur $\ell \geqslant 2$. Pour $i \in [\![1, \ell]\!]$, définissons $\pi(i)$ comme suit : si $u[i..\ell]$ possède un préfixe palindrome, alors $\pi(i)$ est la longueur du plus court préfixe palindrome de $u[i..\ell]$; sinon, $\pi(i) = 0$.

Question 27 ⋆⋆ • Expliquez comment déduire le tableau π des tableaux Ri et Rp, pour un coût $\mathcal{O}(\ell)$.

► Pour $i \in [\![1, \ell]\!]$, définissons $\mathrm{pal}(i)$ comme suit : si $u[i..\ell]$ est un palstar, alors $\mathrm{pal}(i) = \mathrm{VRAI}$; sinon, $\mathrm{pal}(i) = \mathrm{FAUX}$. Il est avantageux de définir $\mathrm{pal}(\ell + 1) = \mathrm{VRAI}$.

Question 28 • Décrivez un algorithme qui construit le tableau pal, pour un coût $\mathcal{O}(\ell)$.

5 Complément de programme

► Dans la suite, l'alphabet utilisé est $\Sigma = \{0, 1\}$. Un langage L est *factoriel* si, dès que $u \in L$, tous les facteurs de u appartiennent aussi à L. Un langage L est *équilibré* si l'on a $\big||u|_0 - |v|_0\big| \leqslant 1$ quels que soient les mots u et v de L de même longueur.

Question 29 ⋆⋆⋆ • Montrez qu'un langage factoriel L sur l'alphabet Σ n'est pas équilibré ssi il existe un palindrome x tel que $0x0$ et $1x1$ appartiennent à L.

FIN