M2 ATAL
X3IT030

Université de Nantes
2019/2020

Contrôle continu
Algorithmique et alignement de chaînes

Décembre 2019

Durée : 1h20. Documents de CM/TD autorisés. Le barème est indicatif. Présentation, clarté et orthographe seront pris en compte dans la note finale. Il est également important de bien justifier toutes vos réponses.

Exercice 1 (5 points, 20 min)

On veut appliquer l'algorithme KMP pour trouver toutes les occurrences du motif P dans le texte T où :

$P = barbara$
$T = barbaroux\ et\ barbara\ adorent\ la\ barbapapa$

On demande de :

1. Expliquer précisément, en français (pas de pseudo-code ici), les méthodes de calcul des paramètres nécessaires à l'exécution de l'algorithme KMP. Le but est de montrer que vous avez compris les démarches et les raisons des algorithmes.
2. Calculer à la main (sans montrer les détails du calcul) les paramètres nécessaires à l'application de l'algorithme KMP.
3. Appliquer (en donnant les détails de l'exécution) l'algorithme KMP sur l'exemple fourni, et calculer le nombre de comparaisons entre des éléments de P et des éléments de T effectuées.

Exercice 2 (5 points, 20min)

Calculer les occurrences approchées selon la distance "nombre de différences" de P dans T, où

$P = babar$
$T = babbarar$

et à 2 erreurs près, en utilisant l'algorithme qui privilégie les occurrences les plus courtes. Pour cela, il faut fournir, pour chaque occurrence trouvée, l'alignement entre P et T correspondant à l'occurrence.

Exercice 3 (10 points, 40min)

Dans cet exercice, on veut *extraire* de deux textes T_1 et T_2 (de longueurs respectives n_1 et n_2) leur *plus long motif commun* – le mot *extraire* signifie que le motif n'est pas connu à l'avance.

Un motif commun P entre deux textes T_1 et T_2 est une suite *consécutive* de caractères, que l'on trouve à la fois dans T_1 (par exemple $T_1[i..j]$ avec $1 \leq i \leq j \leq n_1$) et dans T_2 (par exemple $T[k..\ell]$ avec $1 \leq k \leq \ell \leq n_2$), et telle que $T_1[i..j] = T_2[k..\ell] = P$.

On appellera $PLMC(T_1, T_2)$ le *plus long* motif commun entre T_1 et T_2.

1. Supposons que $T_1 =$ `CAGCAA` et $T_2 =$ `GCAGCC`. Indiquer $PLMC(T_1, T_2)$.

On souhaite calculer $PLMC(T_1, T_2)$ en utilisant un arbre des suffixes généralisé (ou ASG) de T_1 et de T_2, auquel on ajoute quelques informations (on appellera donc cet arbre $ASG^+(T_1, T_2)$).
Plus précisément, pour chaque nœud interne v de $ASG^+(T_1, T_2)$, on ajoute les deux informations suivantes :
— un entier $l(v)$, qui est la longueur du mot représenté par le chemin qui mène de la racine à v ;
— un booléen $b(v)$, qui sera `Vrai` si le sous-arbre de racine v contient au moins une feuille provenant de chacun des deux textes T_1 et T_2, et `Faux` sinon.

2. Dessiner $ASG^+(T_1, T_2)$ dans le cas où $T_1 =$ `CAGCAA` et $T_2 =$ `GCAGCC`. On attend ici une représentation compacte mais, pour plus de lisibilité, on vous demande d'étiqueter les arêtes de $ASG^+(T_1, T_2)$ par des sous-séquences, et non pas par des couples d'entiers.
3. Indiquer clairement, dans $ASG^+(T_1, T_2)$, où se situe $PLMC(T_1, T_2)$.

On se place maintenant dans le cas général, donc on considère n'importe quels textes T_1 (de longueur n_1) et T_2 (de longueur n_2).

4. Donner, sous forme de notation de Landau, la taille de $ASG^+(T_1, T_2)$. Justifier.

On prétend que l'ajout des informations $l(v)$ et $b(v)$, pour tout nœud interne v, se fait en temps linéaire en la taille de l'ASG de T_1 et T_2.

5. Indiquer (en français, pas de pseudo-code) une méthode qui permet d'ajouter le booléen $b(v)$ à chaque nœud interne v de l'arbre. Justifier le fait que cet ajout se fait en temps linéaire en la taille de cet arbre.
6. Même question en ce qui concerne l'ajout de l'information $l(v)$.
7. Montrer que $PLMC(T_1, T_2)$ correspond toujours dans $ASG^+(T_1, T_2)$ à un chemin entre la racine et un *nœud interne*.
8. En déduire un algorithme qui permet de calculer $PLMC(T_1, T_2)$ pour deux textes T_1 et T_2, de longueurs n_1 et n_2. Plus précisément, cet algorithme devra retourner (a) les positions de départ, à la fois dans T_1 et dans T_2, de $PLMC(T_1, T_2)$, et (b) la longueur de $PLMC(T_1, T_2)$. Cet algorithme sera décrit en français (pas de pseudo-code), mais il devra être précis et détaillé.
9. Donner, en la justifiant, la complexité de l'algorithme proposé à la question précédente.