Contrôle continu
Algorithmique et alignement de chaînes

Novembre 2017

Durée : 2h. 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 (6 points)

1. Après avoir rappelé la différence entre corpus comparables et corpus parallèles, expliquer pourquoi l'alignement à partir de corpus comparables est plus complexe que l'alignement à partir de corpus parallèles.
2. Pourquoi l'approche dite "standard" ou encore "directe" à partir de corpus comparables est-elle peu appropriée pour les termes complexes ?
3. Existe-il un continuum entre corpus parallèles et corpus comparables ? (penser à motiver votre réponse)

Exercice 2 (3,5 points)

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 les paramètres nécessaires à l'application de l'algorithme KMP.
3. Appliquer 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 3 (3,5 points)

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 4 (7 points)

On propose une nouvelle méthode pour rechercher de façon exacte un motif P de longueur m dans un texte T de longueur n. Pour simplifier, on supposera que l'alphabet Σ est le suivant : $\Sigma = \{\texttt{0,1,2...9}\}$.
Dans ce cas, toute chaîne de k caractères exprimés sur Σ peut se voir comme un entier de longueur k. Par exemple, la chaîne $C =$`658042` peut être vue comme l'entier $n_C = 658042$.

Étant donné un motif P, on note n_P l'entier correspondant. De même, pour tout $0 \leq i \leq n-m$, on note n_i l'entier correspondant à $T[i \ldots i+m-1]$ (ici, T est donc indicé à partir de 0). Dans ce cas, P apparaît dans T à la position i *si et seulement si* $n_i = n_P$.

On suppose qu'on dispose d'une fonction `char_to_int(char c)` qui prend en entrée le caractère `c` et fournit en sortie le chiffre qui lui est associé, et que celle-ci s'exécute en temps constant.

1. Écrire en pseudo-code un algorithme `calculn_p` qui permet de calculer n_P *en temps* $O(m)$, et justifier sa complexité.
2. Indiquer la formule qui permet de calculer n_{i+1} à partir de n_i, $T[i]$ et $T[i+m]$.
3. En supposant que la valeur 10^{m-1} a été pré-calculée et stockée, quel est le temps d'exécution du calcul de la Question 2 ?
4. Quel est le temps d'exécution du calcul de la valeur 10^{m-1} ? Pourquoi ?
5. En déduire le temps total nécessaire à l'exécution de l'ensemble des calculs de $n_0, n_1 \ldots n_{n-m}$.

En s'appuyant sur les réponses précédentes, on peut en déduire un algorithme, qu'on appellera `RK`, dont la complexité en temps est en $O(n+m)$, et qui détermine les positions où un motif P apparaît dans un texte T.

6. Écrire en pseudo-code l'algorithme `RK`, et justifier sa complexité.

La méthode ci-dessus fonctionne bien car on a supposé que la taille σ de l'alphabet Σ est égale à 10.

7. Comment faire pour adapter cette méthode à n'importe quelle valeur de σ ? Détailler votre réponse.
8. Analyser la complexité de cette nouvelle méthode : (a) si on considère σ constant, puis (b) si on considère σ non constant.