L3-INFO Algorithmique

UFR IM²AG
UNIVERSITÉ Grenoble Alpes

Correction du Quick 1

Écriture et preuve d'algorithmes *(8 points)*

Soient deux tableaux A indexé de 1 à n, et B indexé de 1 à p contenant des éléments de même type. On s'intéresse à vérifier si A est **inclus** dans B, autrement dit si tout élément présent dans A est présent dans B.

On se place bien entendu dans le cas où $n \leqslant p$.

Dans chacun des tableaux, on suppose que les éléments sont tous distincts deux à deux (il n'y a pas de doublon). On suppose également qu'on dispose d'un ordre total sur les éléments, et que les tableaux A et B sont triés (en ordre croissant).

1. Écrire un algorithme de coût au pire $n + p$ pour résoudre le problème posé.

Il faut profiter de l'ordre des éléments : si $B[j]$ est supérieur à $A[i]$, alors tous les éléments de B ayant un indice supérieur à j sont supérieurs à $B[j]$, et donc aussi supérieurs à $A[i]$. Inversement, si $A[i] = B[j]$, alors l'élément $A[i + 1]$ est forcément situé à droite de $B[j]$ dans B (s'il y est présent).
D'où l'algorithme suivant :

Inclusion(A[1..n], B[1..p])
i := 1
j := 1
pour i := 1 *to* n
    tant que $j \leqslant p$ *et* $A[i] > B[j]$
        j := j + 1
    si $A[i] \neq B[j]$
        return *False*
return *True*

2. Proposer un invariant pour cet algorithme. Il est recommandé d'illustrer cette réponse par un ou plusieurs schémas.

— Un premier invariant permet d'assurer que si l'algorithme répond $True$, alors effectivement $A \subseteq B$. Pour la boucle externe :

« Tous les éléments de A d'indice $< i$ sont présents dans B. »

Ceci assure qu'on ne poursuit la boucle que si les éléments précedents ont effectivement été trouvés ; en sortie de boucle, $i = n$ et donc on a bien trouvé tous les éléments de A dans B.

— Un second invariant assure que si l'algorithme répond $False$, alors effectivement un des éléments de A est absent de B. Pour la boucle interne :

« Tous les éléments de B d'indice $< j$ sont différents de $A[i]$. »

Ainsi, en sortie de la boucle interne, si $A[i] \neq B[j]$, alors tous les éléments de B d'indices $< j$ sont différents de $A[i]$ et $A[i] < B[j]$, donc tous les éléments d'indices $\geqslant j$ sont strictement supérieurs à $A[i]$.