

Option Informatique en Spé MP et MP*

Complexité des mots de longueur k sur un alphabet à k lettres (d'après Florence Levé et Patrick Séébold)

Complexité d'un mot

► Soit $k \geq 2$. On note $M_k = \frac{k(k+1)}{2}$.

► Soit A un alphabet contenant k lettres. Pour tout mot w sur l'alphabet A , on note $C(w)$ le nombre de facteurs de w autres que ε . Par exemple, avec $w = abaa$, on aura $C(w) = 8$ puisque les facteurs de w autres que ε sont $a, b, aa, ab, ba, aba, baa$ et $abaa$. On notera bien que le facteur a , qui possède trois occurrences dans w , n'est compté qu'une seule fois!

Question 1 Soit w un mot de longueur k , sur un alphabet à k lettres. Prouvez l'inégalité $C(w) \geq k$, et caractérisez les mots w pour lesquels $C(w) = k$.

Question 2 Soit w un mot de longueur k , sur un alphabet à k lettres. Prouvez l'inégalité $C(w) \leq M_k$, et caractérisez les mots w pour lesquels $C(w) = M_k$.

► Nous dirons qu'un naturel $j \in \llbracket k, M_k \rrbracket$ est k -réalisable s'il existe au moins un mot w de longueur k , sur un alphabet à k lettres, vérifiant $C(w) = j$.

Question 3 Montrez que $2k - 1$ est k -réalisable.

Question 4 Montrez qu'aucun élément de $\llbracket k + 1, 2k - 2 \rrbracket$ n'est k -réalisable.

Question 5 Dressez la liste des naturels k -réalisables, pour $k \in \llbracket 1, 4 \rrbracket$.

Un peu de programmation

► On se propose de déterminer expérimentalement la liste des naturels k -réalisables, pour quelques valeurs de k .

Question 6 Rédigez en Caml une fonction :

```
intervalle : int -> int -> int list
```

spécifiée comme suit : `intervalle p q` construit la liste des éléments de l'intervalle $\llbracket p, q \rrbracket$. Cette liste est vide si $p > q$.

Question 7 Rédigez en Caml une fonction :

```
flat : 'a list list -> 'a list
```

spécifiée comme suit : `flat l` «met à plat» la liste de listes ℓ . Par exemple, `flat [[5;2];[7];[];[9;2;3]]` construit la liste `[5;2;7;9;2;3]`.

Question 8 Rédigez en Caml une fonction :

```
ajoute_devant : 'a -> 'a list list -> 'a list list
```

spécifiée comme suit : `ajoute_devant x l` ajoute x en tête de chaque membre de la liste de listes ℓ . Par exemple, `ajoute_devant 9 [[5;2];[7];[];[9;2;3]]` construit la liste `[[9;5;2];[9;7];[9];[9;9;2;3]]`.

Question 9 Rédigez en Caml une fonction :

```
allonge_avec : 'a list -> 'a list list -> 'a list list
```

spécifiée comme suit : `allonge_avec l1 l2` construit la liste de toutes les listes obtenues en ajoutant un élément de la liste ℓ_1 en tête de l'un des membres de la liste de listes ℓ_2 . La longueur de la liste résultat sera égale au produit des longueurs des listes ℓ_1 et ℓ_2 . Par exemple, `allonge_avec [9;3] [[5;2];[7];[]]` construit la liste `[[9;5;2];[9;7];[9];[3;5;2];[3;7];[3]]`. L'ordre des éléments dans le résultat n'est pas imposé.

Question 10 Rédigez en Caml une fonction :

```
uniq = 'a list -> 'a list
```

spécifiée comme suit: `uniq l` élimine les «doublons» de la liste ℓ . Par exemple, `uniq [6;2;0;6;3;2]` rend la liste `[6;2;0;3]` (à l'ordre près).

► Les trois fonctions suivantes servent à calculer respectivement la liste des préfixes, la liste des suffixes et la complexité d'un mot w , représenté par la liste (éventuellement vide) de ses lettres. Par exemple, le mot 101330 sur l'alphabet $\{0, 1, 2, 3\}$ sera représenté par la liste `[1;0;1;3;3;0]`.

Question 11 Rédigez en Caml une fonction :

```
prefixes : 'a list -> 'a list list
```

spécifiée comme suit: `prefixes l` construit la liste des préfixes de la liste ℓ . Par exemple, `prefixes [3;1;7]` construit la liste de listes `[[]; [3]; [3;1]; [3;1;7]]`. Le résultat peut aussi être présenté dans l'ordre inverse.

Question 12 Rédigez en Caml une fonction :

```
facteurs : 'a list -> 'a list list
```

spécifiée comme suit: `facteurs l` construit la liste des facteurs de la liste ℓ . Par exemple, `facteurs [3;1;7]` construit la liste de listes `[[]; [3]; [1]; [7]; [3;1]; [1;7]; [3;1;7]]`. L'ordre des éléments dans le résultat n'est pas imposé.

Question 13 Rédigez en Caml une fonction :

```
complexite : 'a list -> int
```

spécifiée comme suit: `complexite l` calcule le nombre de facteurs de la liste ℓ , autres que le mot vide.

Question 14 Rédigez en Caml une fonction :

```
liste_realisables : int -> int list
```

spécifiée comme suit: `liste_realisables k` construit la liste des naturels k -réalisables. On présentera cette liste classée en ordre croissant, en utilisant la fonction `sort` de la bibliothèque `sort`.

La conjecture de Kása

► Notons b_k le plus petit naturel tel que tout élément de $[[b_k, M_k]]$ soit k -réalisable. Z. KÁSA a conjecturé une formule donnant la valeur de b_k en fonction de k . Cette formule a été prouvée récemment par Florence LEVÉ et Patrick SÉÉBOLD. Elle pourrait faire l'objet d'un autre TD!

FIN