

MIAS 1°A, UE SM22 ; SM 2°A, module H7 ; STPI 2°A/IUP1 Miage

Informatique – Examen, 1^{ère} session (5 mai 2001) : des exemples de solutions

On ne donne ici que quelques unes des solutions possibles

1. Rang du dernier positif

Q1 Longueur : une séquence d'entiers \rightarrow un entier ≥ 0 {longueur de la séquence donnée}

$$(1) \text{Longueur}([\])=0 \quad (2) \text{Longueur}(e_0S)=1+\text{Lg}(S)$$

$$\text{RDP}(S \bullet e) = \text{si } e > 0 \text{ alors } 1+\text{Longueur}(S) \text{ sinon } \text{RDP}(S)$$

$$\text{RDP}(e_0S) = \text{soit } r = \text{RDP}(S)$$

$$\text{dans si } r \neq 0 \text{ alors } r+1$$

$$\text{sinon si } e > 0 \text{ alors } 1 \text{ sinon } 0$$

{On peut aussi écrire : soit $r = \text{RDP}(S)$ dans si $e > 0$ ou $r \neq 0$ alors $r+1$ sinon 0 }

2. A propos de pangrammes

Q2 **T2** comporte au moins un exemplaire de chacun des éléments du texte **T1** si et seulement si chaque élément de **T1** appartient à **T2** (fonction **Ap**) :

Ap : un caractère, un texte \rightarrow un booléen

{ $\text{Ap}(c, T)$ a la valeur vrai $\Leftrightarrow c \in T$ }

$$(1) \text{Ap}(c, [\]) = \text{faux}$$

$$(2) \text{Ap}(c_1, c_2_0T) = (c_1=c_2) \text{ ou } \text{Ap}(c_1, T)$$

$$(3) \text{AuMoinsUn}([c], T) = \text{Ap}(c, T)$$

$$(4) \text{AuMoinsUn}(c_0T_1, T_2) = \text{Ap}(c, T_2) \text{ et } \text{AuMoinsUn}(T_1, T_2)$$

{ T_1 non vide}

{dans l'équation (2), on peut mettre "ou alors" ; dans l'équation (4) on peut mettre "et puis"}

Une version sans fonction intermédiaire, fondée sur le fait que $\text{Ap}(c, T) = \text{AuMoinsUn}([c], T)$:

$$(1) \text{AuMoinsUn}([c], [\]) = \text{faux}$$

{on peut regrouper les cas (1) et (2)}

$$(2) \text{AuMoinsUn}(c_0T, [\]) = \text{faux}$$

{ T non vide}

$$(3) \text{AuMoinsUn}([c_1], c_2_0T) = (c_1=c_2) \text{ ou } \text{AuMoinsUn}([c_1], T)$$

{on peut mettre "ou alors"}

$$(4) \text{AuMoinsUn}(c_1_0T_1, c_2_0T_2) =$$

{ T_1 non vide}

$$\text{AuMoinsUn}([c_1], c_2_0T_2)$$

{on peut aussi écrire : $c_1 = c_2$ ou $\text{AuMoinsUn}([c_1], T_2)$ }

$$\text{et } \text{AuMoinsUn}(T_1, c_2_0T_2)$$

{on peut mettre "et puis"}

Pour réaliser la fonction **EstPangramme**, on applique la fonction **AuMoinsUn** à un texte formé des lettres (minuscules) de l'alphabet :

EstPangramme(T) : **AuMoinsUn**("abcdefghijklmnopqrstuvwxy", T)

3. Élagage d'un arbre

Q3 Analyse de **Élagage(A, n)** : si **n=1**, seule la racine est de niveau inférieur ou égal à 1. Si **n>1**, les nœuds de niveau supérieur à **n** dans **A** sont ceux de niveau supérieur à **n-1** dans ses sous-arbres.

$$(1) \text{Élagage}(\wedge, 1) = \wedge$$

{on peut regrouper les équations (1) et (2)}

$$(2) \text{Élagage}(\wedge, n+1) = \wedge$$

{ $n > 0$ }

$$(3) \text{Élagage}(/G, r, D, 1) = / \wedge, r, \wedge \backslash$$

$$(4) \text{Élagage}(/G, r, D, n+1) = / \text{Élagage}(G, n), r, \text{Élagage}(D, n) \backslash$$

{ $n > 0$ }

4. A propos de chemins d'accès aux nœuds d'un arbre binaire

Q4i (1) **RechParChemin**(\wedge , [$\]$) = <faux, [$\]$ >

{la valeur [$\]$ est un choix arbitraire}

$$(2) \text{RechParChemin}(\wedge, b_0C) = \langle \text{faux}, [\] \rangle$$

{on peut regrouper les équations (1) et (2)}

$$(3) \text{RechParChemin}(/G, r, D, [\]) = \langle \text{vrai}, r \rangle$$

$$(4) \text{RechParChemin}(/G, r, D, b_0C) = \text{RechParChemin}(\text{si } b = 0 \text{ alors } G \text{ sinon } D), C$$

Q4ii (1) **ChAccès**(E, \wedge) = <faux, [$\]$ >

{la valeur [$\]$ est un choix arbitraire}

$$(2) \text{ChAccès}(E, /G, r, D) =$$

$$\text{si } E = r \text{ alors } \langle \text{vrai}, [\] \rangle$$

$$\text{sinon soit } \langle TG, CG \rangle = \text{ChAccès}(E, G)$$

$$\text{dans si } TG \text{ alors } \langle \text{vrai}, 0_0CG \rangle$$

$$\text{sinon soit } \langle TD, CD \rangle = \text{ChAccès}(E, D)$$

$$\text{dans si } TD \text{ alors } \langle \text{vrai}, 1_0CD \rangle \text{ sinon } \langle \text{faux}, [\] \rangle$$

{on peut aussi écrire : dans $\langle TD, 1_0CD \rangle$ }

Une variante consiste à décomposer le problème en deux sous-problèmes : appartenance d'un élément à un arbre ; construction du chemin associé à un élément sachant qu'il appartient à l'arbre.

5. A propos de sous-séquences

Q5i

- (1) $\text{LePréfixe}([e], 1) = [e]$ *{le cas [e], k+1 est impossible du fait de la pré-condition}*
 (2) $\text{LePréfixe}(e_0S, 1) = [e]$ *{S non vide}*
 (4) $\text{LePréfixe}(e_0S, k+1) = e_0\text{LePréfixe}(S, k)$ *{S non vide et k > 1}*
{on peut regrouper les équations 1 et 2}

Q5ii

— Soit $\text{NbSSC}(S, k)$ le nombre de sous-séquences de S de longueur k . Soit n la longueur de S .

Si $k > n$, $\text{NbSSC}(S, k) = 0$. Sinon, soit p la position dans S du premier élément d'une sous-séquence de longueur k . La plus grande valeur possible pour p , soit g , est telle que $g+k-1 = n$; donc $g = n-k+1$. Il y a exactement une sous-séquence connexe de longueur k pour chaque position p de $[1..g]$. Par conséquent, $\text{NbSSC}(S, k) = n-k+1$.

— Pour formuler les équations de récurrence, on applique le principe suivant :

Soit E l'ensemble des sous-séquences connexes de longueur k d'une séquence non vide e_0S de longueur $n+1$. Si $k > n+1$, E est vide. Sinon E est formé d'une part du préfixe de longueur k et d'autre part de l'ensemble E' des sous-séquences connexes de longueur k de S qui est de longueur n .

- (1) $\text{LesSousSéqC}([e], 1, k) =$ si $k = 1$ alors $[[e]]$ sinon $[]$
 (2) $\text{LesSousSéqC}(e_0S, n+1, k) =$ *{n > 0, S non vide}*
 si $k > n+1$ alors $[]$ sinon $\text{LePréfixe}(e_0S, k) \circ \text{LesSousSéqC}(S, n, k)$

Q5iii

— Soit une séquence S de longueur 5, $S = [a, b, c, d, e]$. Pour énumérer les sous-séquences de longueur 3, on considère d'abord celles commençant par a , puis celles commençant par b , etc. On obtient par exemple :

$[a, b, c], [a, b, d], [a, b, e], [a, c, d], [a, c, e], [a, d, e], [b, c, d], [b, c, e], [b, d, e], [c, d, e]$.

Il y en a 10, ce qui correspond au nombre de manières de prendre 3 éléments parmi 5.

— Soit $\text{NbSS}(S, k)$ le nombre de sous-séquences de S de longueur k . Soit n la longueur de S . Si $k > n$, $\text{NbSSC}(S, k) = 0$. Sinon, il y a autant de sous-séquence de longueur k que de manières de choisir k éléments parmi n : $\text{NbSSC}(S, k) = C_{n,k} = n! / (k! (n-k)!)$.

— Pour formuler les équations de récurrence, on applique le principe suivant :

Soit E l'ensemble des sous-séquences de longueur k d'une séquence non vide e_0S . E est décomposé en deux parties disjointes E' et E'' : E' comporte toutes les sous-séquences qui commencent par e ; E'' comporte toutes celles qui ne commencent pas par e :

— Tout élément de E' est de la forme e_0s' où s' est une sous-séquence de S de longueur $k-1$: pour construire E' , on commence par construire les sous-séquences de S de longueur $k-1$, puis on ajoute e au début de chacune des sous-séquences ainsi construites.

— E'' est l'ensemble des sous-séquences de S de longueur k .

On introduit ainsi une fonction intermédiaire nommée AjG :

AjG : un Élément, une séquence de (SéqE non vides) \rightarrow une séquence de (SéqE non vides)
{AjG(e, SS) est déduite de SS en ajoutant e à gauche de chacune des séquences de SS.}
 $\text{AjG}(e, []) = [.]$

- (1) $\text{LesSousSéq}([e], 1) = [[e]]$
 (2) $\text{LesSousSéq}([e], k+1) = []$ *{k > 0}*
 (3) $\text{LesSousSéq}(e_0S, 1) = [[e]] \& \text{LesSousSéq}(S, 1)$ *{n > 0 et S non vide}*
 (4) $\text{LesSousSéq}(e_0S, k+1) =$ *{k > 0 et S non vide}*
 $\text{AjG}(e, \text{LesSousSéq}(S, k)) \& \text{LesSousSéq}(S, k+1)$
 (5) $\text{AjG}(e, []) = [.]$
 (6) $\text{AjG}(e, s_0SS) = (e_0s)\circ\text{AjG}(e, SS)$

Variante : Une autre solution consiste à construire d'abord l'ensemble des sous-séquences de la séquence donnée (voir cours), puis à extraire de la séquence de séquences ainsi obtenues, les séquences de longueur k .