ULC 214

SESSION 2002

Filière MP (groupes M/MP/MPI)
(Épreuve commune aux ENS de Paris, Lyon et Cachan)

Filières MP et PC (groupe I)
(Épreuve commune aux ENS de Paris et Lyon)

INFORMATIQUE

Corrigé de Jean-François Husson (J-Fr.Husson@wanadoo.fr) et M. Quercia (michel.quercia@prepas.org)

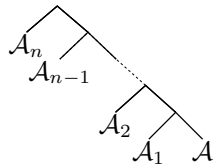
1. Introduction

Le principe d'induction structurelle donné pour les triplets d'arbres est peu compréhensible. Il fallait lire :

$$\big\{P(\mathbf{O},\mathbf{O},\mathbf{O}) \,\&\, \big[\forall a \in \mathbb{A}^3,\ (\forall b \in \mathbb{A}^3,\ a \rhd\!\rhd\!\rhd b \Longrightarrow P(b)) \Longrightarrow P(a)\big]\big\} \Longrightarrow (\forall a \in \mathbb{A}^3,\ P(a)).$$

Question 1.
Soit $\sqsupset$ un ordre strict. On doit montrer qu'il n'existe pas deux éléments (alors nécessairement distincts) x, y tels que $x \sqsupset y$ et $y \sqsupset x$. De fait, si l'on suppose $x \sqsupset y$ et $y \sqsupset x$ alors on obtient $x \sqsupset x$ par transitivité, ce qui contredit l'irréflexivité de $\sqsupset$.

Question 2.
De manière informelle, $R(\mathcal{A}, [\mathcal{A}_1; \ldots; \mathcal{A}_n])$ est l'arbre dessiné ci-contre. L'algorithme suivant implémente la fonction R :



$R(\mathcal{A}$**,***liste*$) =$
  si *liste* $=$ *[ ]* ***alors retourne*** $\mathcal{A}$
  sinon retourne $R(\mathsf{hd}(\textit{liste}) \cdot \mathcal{A},\ \mathsf{tl}(\textit{liste}))$
fin

2. Décomposition complète d'un nombre dans une base et suites de Goodstein

Question 1.
$4 = 3^{(3^0+0)} + 3^0 + 0, \qquad 83 = 3^{(3^{(3^0+0)}+3^0+0)} + 3^0 + 3^0 + 0.$

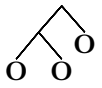
Question 2.
On suppose dans cette question et dans toute la suite du problème que $b \geqslant 2$.

La définition d'une décomposition complète donnée dans l'énoncé ne fait apparaître que le nombre 0, des additions et des exponentiations, donc si un nombre admet une décomposition complète, alors elle est bien de la forme demandée. Le but de la question est en fait de prouver que tout entier naturel n admet effectivement une décomposition complète. On raisonne par récurrence forte sur n. Si $n = 0$ alors n admet une décomposition complète : $0 = 0$. Supposons à présent que n est un entier naturel non nul et que tous les entiers de $\{0, \ldots, n-1\}$ admettent des décompositions complètes en base b. Soit p l'exposant maximal de n en base b ($p = \lfloor \ln n / \ln b \rfloor$) et $q = n - b^p$. On a $p, q \in \{0, \ldots, n-1\}$ donc p et q admettent des décompositions complètes en base b, puis $n = b^{\text{décomposition de } p} + (\text{décomposition de } q)$, ce qui prouve que n admet lui aussi une décomposition complète en base b.

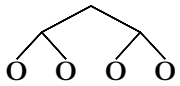
Partie 2. Décomposition complète d'un nombre dans une base et suites de Goodstein

Question 3.

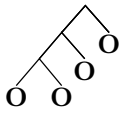
$1 = b^0 + 0$



$b = b^{(b^0+0)} + 0$



$b + 1 = b^{(b^0+0)} + b^0 + 0$



$b^b = b\hat{}(b^{(b^0+0)} + 0) + 0$

Définition formelle de arbre :

$$\mathsf{arbre}(b, 0) = \mathbf{O}$$
$$\mathsf{arbre}(b, n) = \mathsf{arbre}(b, p) \cdot \mathsf{arbre}(b, q) \quad \text{avec } (p, q) = \mathsf{dec1}(b, n), \text{ pour } n \geqslant 1.$$

La correction et la terminaison de cet algorithme sont évidentes.

Question 4.

Le code donné pour ***arbre*** *est mal parenthésé, il faut ignorer les parenthèses fermantes surnuméraires dans les affectations à la variable courant. On reproduit ci-dessous le code rectifié en numérotant les lignes.*

```
 1 arbre(b, n) =
 2     pile := [(0, [ ])]
 3     courant := (n, [ ])
 4     tant que pile ≠ [ ] faire
 5         k, liste_res := courant
 6         si k ≠ 0 alors
 7             m, r := dec1(b, k)
 8             pile := (r, liste_res) :: pile
 9             courant := (m, [ ])
10         sinon
11             q, l := hd(pile)
12             pile := tl(pile)
13             courant := (q, renv(liste_res) :: l)
14     fait
15     k, liste_res := courant
16     retourne (hd(liste_res))
17 fin
```

D'après la ligne *2*, *pile* est une liste de couples *(nombre,liste-1)*. D'après la ligne *3*, *courant* est un couple *(nombre,liste-2)*. D'après la ligne *5*, *liste_res* est une liste de type *liste-2*. D'après la ligne *8*, les types *liste-1* et *liste-2* sont égaux. Enfin, d'après la ligne *13*, *liste-2* est le type des listes d'arbres binaires.

En conclusion, *pile* contient une liste de couples *(nombre, liste d'arbres binaires)*, *courant* contient un couple de ce type, et *liste_res* contient une liste d'arbres binaires. On vérifie aisément que les types indiqués sont compatibles avec toutes les opérations codées dans ***arbre***, et l'on en déduit que ***arbre***(b, n) retourne un arbre binaire, sous réserve de terminaison.

Invariant de boucle : soit $[(k, l); (k_1, l_1); \ldots; (k_p, l_p)]$ la valeur de *courant* :: *pile* lors d'un passage en ligne *5*. Alors les nombres $k, k_1, \ldots, k_p$ sont des entiers naturels. De plus, soient les arbres binaires $\mathcal{A}_0, \ldots, \mathcal{A}_{p-1}$ définis par les relations :

$$\mathcal{A}_0 = R(\mathsf{arbre}(b, k), l), \quad \mathcal{A}_1 = R(\mathsf{arbre}(b, k_1),\ \mathcal{A}_0 :: l_1), \quad \ldots, \quad \mathcal{A}_{p-1} = R(\mathsf{arbre}(b, k_{p-1}),\ \mathcal{A}_{p-2} :: l_{p-1}).$$

Alors l'expression :

$$\mathcal{A}_{p-1} :: l_p = R(\mathsf{arbre}(b, k_{p-1}),\ R(\quad \ldots,\ R(\mathsf{arbre}(b, k_1),\ R(\mathsf{arbre}(b, k), l) :: l_1) :: \ldots \quad) :: l_{p-1}) :: l_p \qquad (*)$$

est constante.

Démonstration : le fait que $k, k_1, \ldots, k_p$ soient entiers naturels est immédiat. En ce qui concerne l'expression $(*)$, les instructions lignes *7,8,9* ont pour effet de remplacer dans $(*)$ l'arbre $R(\mathsf{arbre}(b, k), l)$ par :

$$\begin{aligned} R(\mathsf{arbre}(b, r),\ R(\mathsf{arbre}(b, m), [\,]) :: l) &= R(\mathsf{arbre}(b, r),\ \mathsf{arbre}(b, m) :: l) \\ &= R(\mathsf{arbre}(b, m) \cdot \mathsf{arbre}(b, r),\ l) \\ &= R(\mathsf{arbre}(b, k),\ l), \end{aligned}$$

tandis que les instructions lignes *11,12,13* ont pour effet de remplacer dans $(*)$ la liste $R(\mathsf{arbre}(b,k),l) :: l_1$ par :

$$\mathsf{renv}(l) :: l_1 = R(\mathbf{O},l) :: l_1 = R(\mathsf{arbre}(b,0),l) :: l_1 = R(\mathsf{arbre}(b,k),l) :: l_1.$$

Dans les deux cas, la valeur de $(*)$ est donc inchangée. ■

Contenu de *liste_res* **lorsque l'algorithme se termine :** la valeur de $(*)$ est celle déterminée lors du premier passage en ligne *5*, à savoir : $R(\mathsf{arbre}(b,n),[\,]) :: [\,] = [\mathsf{arbre}(b,n)]$. Au début de la dernière itération de la boucle ***tant que***, on a $\mathit{pile} = [(k_1,l_1)]$ et $\mathit{courant} = (0,l)$, et donc en ligne *15* on a :

$$\mathit{liste\_res} = \mathsf{renv}(l) :: l_1 = R(\mathsf{arbre}(b,0),l) :: l_1 = [\mathsf{arbre}(b,n)].$$

Ceci prouve que l'algorithme ***arbre***, s'il termine, retourne effectivement la valeur $\mathsf{arbre}(b,n)$.

Contenu de *liste_res* **dans les étapes de l'algorithme :** l'énoncé ne précisant pas en fonction de quoi il faut décrire ce contenu, on est fondé à répondre que, après chaque passage en ligne *5*, la variable *liste_res* contient une liste l d'arbres binaires telle que l'expression $(*)$ a pour valeur $[\mathsf{arbre}(b,n)]$.

Terminaison : la terminaison de l'algorithme ***arbre*** n'est pas demandée explicitement dans l'énoncé, mais elle peut être prouvée facilement en constatant que, avec les notations données dans de l'invariant de boucle, la quantité $2(k+k_1+\ldots+k_p)+p$ décroît strictement entre deux passages consécutifs en ligne *5*.

Question 5.

Algorithme :

erbra$(b,\mathcal{A}) =$
  si $\mathcal{A} = \mathbf{O}$ ***alors retourne*** 0
  sinon, *soient* $\mathcal{B}$ *et* $\mathcal{C}$ *tels que* $\mathcal{A} = \mathcal{B}\cdot\mathcal{C}$ *:* ***retourne*** $b^{\mathit{erbra}(\mathcal{B})} + \mathit{erbra}(\mathcal{C})$
fin

La correction et la terminaison sont immédiates. En ce qui concerne la complexité, si l'on note $T(\mathcal{A})$ le temps de calcul de ***erbra***$(\mathcal{A})$ compté en nombre d'additions et d'exponentiations et $N(\mathcal{A})$ la taille de $\mathcal{A}$, alors on a les relations :

$$T(\mathbf{O}) = 0, \quad T(\mathcal{B}\cdot\mathcal{C}) = T(\mathcal{B}) + T(\mathcal{C}) + 2,$$
$$N(\mathbf{O}) = 0, \quad N(\mathcal{B}\cdot\mathcal{C}) = N(\mathcal{B}) + N(\mathcal{C}) + 1.$$

D'où $T(\mathcal{A}) = 2N(\mathcal{A})$ par induction structurelle sur $\mathcal{A}$.

Question 6.

Pour calculer g_4, on part de la décomposition complète de 83 en base 3 : $83 = 3^{(3^{(3^0+0)}+3^0+0)} + 3^0 + 3^0 + 0$, on remplace tous les 3 par des 4 dans cette expression, on évalue l'expression résultante et on retranche 1. D'où $g_4 = 4^{(4^{(4^0+0)}+4^0+0)} + 4^0 + 4^0 + 0 - 1 = 4^5 + 1 = 1025$.

Pour calculer g_2, on commence par inverser la relation de définition des suites de Goodstein :

$$g_{k+1} = \mathsf{erbra}(k+1,\mathsf{arbre}(k,g_k)) - 1 \iff g_k = \mathsf{erbra}(k,\mathsf{arbre}(k+1,g_{k+1}+1)).$$

On obtient donc g_k à partir de k et g_{k+1} en écrivant la décomposition complète de $g_{k+1}+1$ en base $k+1$, en remplaçant $k+1$ par k dans cette décomposition et en évaluant le résultat sous réserve que la formule obtenue soit une décomposition complète en base k valide.
D'où : $g_3 = 83 \Longrightarrow g_3 + 1 = 84 = 3^{(3^{(3^0+0)}+3^0+0)} + 3^{(3^0+0)} + 0 \Longrightarrow g_2 = 2^{(2^{(2^0+0)}+2^0+0)} + 2^{(2^0+0)} + 0 = 10$.

$g_k > 1000^{1000}$? Calculons quelques valeurs supplémentaires de la suite (g_k) :

$$g_5 = 5^{(5^{(5^0+0)}+5^0+0)} + 5^0 + 0 - 1 = 5^6,$$
$$g_6 = 6^{(6^{(6^0+0)}+6^0+0)} + 0 - 1 = 6^7 - 1 = 6^6 + 233279.$$

On montre alors par récurrence sur k que l'on a $g_k = k^k + h_k$ avec $233285 - k \leqslant h_k < 5\,k^k$ pour tout entier $k \in [\![6, 233285]\!]$. Cette hypothèse est vérifiée pour $k = 6$, et si elle l'est pour un certain entier $k < 233285$ alors la décomposition complète de g_k en base k est de la forme :

$$g_k = \underbrace{k^{(k^{(k^0+0)}+0)}}_{k^k} + \underbrace{k^{a_1} + \ldots + k^{a_p} + 0}_{h_k}$$

où $a_1, \ldots, a_p$ sont des entiers (complètement décomposés en base k) tels que $k \geqslant a_1 \geqslant \ldots \geqslant a_p \geqslant 0$ et $k > a_5$ si $p \geqslant 5$. Alors :

$$g_{k+1} = \underbrace{(k+1)^{((k+1)^{((k+1)^0+0)}+0)}}_{(k+1)^{k+1}} + \underbrace{(k+1)^{b_1} + \ldots + (k+1)^{b_p} + 0 - 1}_{h_{k+1}}$$

où b_i est l'entier obtenu à partir de a_i en remplaçant k par $k+1$ dans la décomposition complète de a_i en base k. On a donc $b_i = k+1$ si $a_i = k$ et $b_i = a_i$ sinon, d'où : $5(k+1)^{k+1} > h_{k+1} \geqslant h_k - 1 \geqslant 233285 - k - 1 = 233285 - (k+1)$. ■

Conclusion : on a $g_{999} < 6 \times 999^{999} < 1000^{1000} < g_{1000}$, donc g_{1000} est le premier entier de la suite (g_k) supérieur à 1000^{1000}.

Question 7.

Premier algorithme : l'énoncé autorise pour ce premier algorithme l'usage de l'addition des entiers naturels, mais ne précise pas si l'exponentiation effectuée dans l'algorithme *erbra*, question **5**, est aussi autorisée, ni si les opérations effectuées dans une implémentation (non explicitée) de dec1 sont elles aussi autorisées. On choisit ici d'interpréter naïvement la question :

$$\textit{addition\_arbres}(b, \mathcal{A}, \mathcal{B}) = \mathsf{arbre}(b,\ \mathsf{erbra}(b, \mathcal{A}) + \mathsf{erbra}(b, \mathcal{B})).$$

Deuxième algorithme : soient $u, v \in \mathbb{N}$ deux nombres entiers donnés par leurs arbres associés. Ces arbres fournissent des décompositions de u et v comme sommes de puissances de b :

$$u = b^{u_1} + \ldots + b^{u_p} + 0, \qquad v = b^{v_1} + \ldots + b^{v_q} + 0,$$

où $(u_1, \ldots, u_p)$ et $(v_1, \ldots, v_q)$ sont des suites décroissantes d'entiers naturels connus par leurs arbres associés. On peut calculer $u + v$ par additions successives d'une puissance de b à un nombre entier :

$$u + v = (\ldots((u + b^{v_1}) + b^{v_2}) + \ldots) + b^{v_q},$$

d'où l'algorithme :

addition$(b, \mathcal{U}, \mathcal{V})$ =
  si $\mathcal{V} = \mathbf{O}$ ***alors retourne*** $\mathcal{U}$
  sinon, *soient* $\mathcal{X}$ *et* $\mathcal{Y}$ *tels que* $\mathcal{V} = \mathcal{X} \cdot \mathcal{Y}$ *:* ***retourne*** *addition*$(b,$ *add_puiss*$(b, \mathcal{U}, \mathcal{X}),\ \mathcal{Y})$
fin

où *add_puiss* est un algorithme calculant $u + b^x$ pour u, x entiers naturels donnés sous forme d'arbres. En supposant *add_puiss* correctement implémentée, la correction et la terminaison de *addition* sont évidentes. *add_puiss* s'apparente à un algorithme d'insertion. Il s'agit d'insérer b^x au bon endroit dans la décomposition de u, en traitant à part le cas où l'on crée ainsi une suite de b termes égaux à b^x : il faut alors remplacer cette sous-suite par b^{x+1}, et propager la retenue ainsi créée vers le début de la décomposition. On aura donc besoin pour écrire *add_puiss* de savoir comparer deux exposants de b pour déterminer l'endroit où insérer b^x, et aussi de savoir calculer $x + 1$. Ce dernier problème peut être résolu par l'utilisation récursive de *add_puiss* puisque $x + 1 = x + b^0$. En reportant à plus tard l'écriture d'un algorithme de comparaison, on a donc l'algorithme suivant pour *add_puiss* :

```
1  add_puiss(b, U, X) =
2      si U = O alors retourne X · O
3      sinon, soient V et W tels que U = V · W :
4          comparer V et X
5          si V < X alors retourne X · U
6          si V > X alors retourne add_puiss(b, add_puiss(b, W, X), V)
7          si V = X alors :
8              l := [X; X];   n := 2;   a := W
9              tant que a est de la forme a = X · a' faire :
10                 l := X :: l;   n := n + 1;   a := a'
11             fait
12             si n < b alors retourne R(a, l)
13             sinon, retourne add_puiss(b, X, O) · a
14 fin
```

La terminaison et la correction de *add_puiss* s'établissent facilement par récurrence sur le nombre u associé au deuxième argument de *add_puiss*. En effet, les deux appels récursifs en ligne *6* et l'appel récursif en ligne *13* portent sur des arbres associés aux nombres w, $w + b^x$ et x et ces nombres sont strictement inférieurs à u dans les conditions où l'on passe par ces lignes.

Finalement, il ne reste plus qu'à écrire un algorithme comparant deux nombres u et v donnés par leurs arbres binaires associés. Le cas où l'un des nombres u ou v est nul est trivial. Dans le cas général, les arbres associés à u et v fournissent des décompositions de u et v comme sommes de puissances décroissantes de b :

$$u = b^{u_1} + \ldots + b^{u_p} + 0 = b^{u_1} + u', \qquad v = b^{v_1} + \ldots + b^{v_q} + 0 = b^{v_1} + v',$$

où u_1 et v_1 sont les exposants maximaux de u et v. Donc, si $u_1 < v_1$ on a $u < v$, si $u_1 > v_1$ on a $u > v$, et si $u_1 = v_1$ alors u et v sont dans le même ordre que u' et v'. L'algorithme ci-dessous implémente ce principe de comparaison, et retourne l'une des constantes symboliques sup, egal ou inf selon que $u > v$, $u = v$ ou $u < v$:

compare$(\mathcal{U}, \mathcal{V})$ =
  si $\mathcal{U} = \mathbf{O}$ *et* $\mathcal{V} = \mathbf{O}$ ***alors retourne*** egal
  sinon, si $\mathcal{U} = \mathbf{O}$ ***alors retourne*** inf
  sinon, si $\mathcal{V} = \mathbf{O}$ ***alors retourne*** sup
  sinon, *soient* $\mathcal{U}_1, \mathcal{U}_2, \mathcal{V}_1, \mathcal{V}_2$ *tels que* $\mathcal{U} = \mathcal{U}_1 \cdot \mathcal{U}_2$ *et* $\mathcal{V} = \mathcal{V}_1 \cdot \mathcal{V}_2$ *:*
    $r :=$ *compare*$(\mathcal{U}_1, \mathcal{V}_1)$
    si $r =$ egal ***alors retourne*** *compare*$(\mathcal{U}_2, \mathcal{V}_2)$
    sinon, retourne r
fin

Conclusion : le contrat est rempli, on a obtenu un algorithme réalisant l'addition de deux arbres binaires dans lequel la seule addition entre nombres entiers effectuée est l'incrémentation de la variable n dans la fonction *add_puiss*, ligne *10*, et la valeur de n est toujours inférieure ou égale à b (après incrémentation). Cet algorithme d'addition n'est pas le plus rapide qui puisse se concevoir, car la fusion des listes de puissances de b composant u et v dans *addition* est réalisée « par le haut », donc la complexité de cette fusion est $O(n^2)$ pour des arbres de taille $O(n)$, alors qu'on pourrait réaliser une fusion en temps $O(n)$ « en partant du bas ». Mais l'énoncé n'a pas demandé d'implémenter une addition de complexité optimale, et il y a encore beaucoup de questions à traiter…

Question 8.
Soient u, v deux entiers naturels donnés par leurs arbres associés. De même que pour l'addition, on ramène le calcul de l'arbre représentant uv à une série de calculs où le multiplicateur est une puissance de b. Ce cas simplifié de multiplication s'implémente par translation des exposants du multiplicande.

multiplie$(b, \mathcal{U}, \mathcal{V})$ =
  si $\mathcal{V} = \mathbf{O}$ ***alors retourne*** $\mathbf{O}$
  sinon, *soient* $\mathcal{X}$ *et* $\mathcal{Y}$ *tels que* $\mathcal{V} = \mathcal{X} \cdot \mathcal{Y}$ *:*
    retourne *addition*$(b,$ *translate*$(b, \mathcal{U}, \mathcal{X}),$ *multiplie*$(b, \mathcal{U}, \mathcal{Y}))$
fin

translate$(b, \mathcal{U}, \mathcal{X})$ =
  si $\mathcal{U} = \mathbf{O}$ ***alors retourne*** $\mathbf{O}$
  sinon, *soient* $\mathcal{V}$ *et* $\mathcal{W}$ *tels que* $\mathcal{U} = \mathcal{V} \cdot \mathcal{W}$ *:*
    retourne *addition*$(b, \mathcal{V}, \mathcal{X}) \cdot$ *translate*$(b, \mathcal{W}, \mathcal{X})$
fin

3. Un ordre sur les arbres

Question 1.
Remarquons déjà que si $\mathcal{A}$ et $\mathcal{B}$ sont des arbres binaires, alors la relation $\mathcal{A} \succ \mathcal{B}$ implique $\mathcal{A} \neq \mathbf{O}$.

Irréflexivité : soit $\mathcal{A} \in \mathbb{A}$. On montre par récurrence sur la taille n de $\mathcal{A}$ que $\mathcal{A} \not\succ \mathcal{A}$. Pour $n = 0$ on a $\mathcal{A} = \mathbf{O}$ donc $\mathcal{A} \not\succ \mathcal{A}$ d'après la remarque précédente. Pour $n > 0$, on a $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$ où $\mathcal{A}_1$ et $\mathcal{A}_2$ sont deux arbres binaires de tailles inférieures à n. Par hypothèse de récurrence, $\mathcal{A}_1 \not\succ \mathcal{A}_1$ et $(\mathcal{A}_1 = \mathcal{A}_1,\ \mathcal{A}_2 \not\succ \mathcal{A}_2)$ donc $\mathcal{A} \not\succ \mathcal{A}$.

Transitivité : on montre de même la propriété $(\mathcal{A} \succ \mathcal{B},\ \mathcal{B} \succ \mathcal{C}) \Longrightarrow \mathcal{A} \succ \mathcal{C}$ par récurrence sur la taille de $\mathcal{C}$ (il y a 4 cas à examiner).

Question 2.

Soit $(\mathcal{A}_n)$ la suite d'arbres définie par les relations : $\mathcal{A}_0 = (\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O}$, $\mathcal{A}_{n+1} = \mathbf{O}\cdot\mathcal{A}_n$.
On a par construction $(\mathcal{A}_n \succ \mathcal{A}_{n+1}) \Longrightarrow (\mathcal{A}_{n+1} \succ \mathcal{A}_{n+2})$, et aussi $\mathcal{A}_0 \succ \mathcal{A}_1$, d'où $\mathcal{A}_n \succ \mathcal{A}_{n+1}$ pour tout entier n.

Question 3.

Le caractère « ordre strict » se démontre sans difficulté. En ce qui concerne la bonne fondation, supposons qu'il existe une suite infinie de couples $(a_n, b_n) \in A \times B$ strictement décroissante pour $>_A \times >_B$. Par définition de $>_A \times >_B$, la suite (a_n) est décroissante au sens large. Si elle prend une infinité de valeurs distinctes alors, quitte à se restreindre à une sous-suite, on peut supposer que tous les a_n sont distincts donc A contient une suite infinie strictement décroissante, ce qui est exclus. De même, si la suite (a_n) ne prend qu'un nombre fini de valeurs distinctes, alors à partir d'un certain rang n_0 tous les a_n sont égaux et la sous-suite $(b_n)_{n\geqslant n_0}$ est une suite infinie de B strictement décroissante, c'est impossible. Dans les deux cas on aboutit à une impossibilité, donc $A \times B$ ne contient pas de suite infinie strictement décroissante pour $>_A \times >_B$.

Question 4.

Remarque : traditionnellement, un alphabet est un ensemble fini non vide. Ici il faut autoriser les « alphabets » à être infinis, sinon la question de la bonne fondation de $(A, >_A)$ ne se pose pas.

En considérant que A est le sous-ensemble de A^{dec} constitué des mots à une seule lettre, l'ordre $>_{A^*}$ prolonge $>_A$. Donc si $(A^{\mathrm{dec}}, >_{A^*})$ est bien fondé, $(A, >_A)$ l'est aussi. Pour la réciproque, supposons $(A, >_A)$ bien fondé, $(A^{\mathrm{dec}}, >_{A^*})$ mal fondé, et soit (u_n) une suite infinie de mots décroissants strictement décroissante pour $>_{A^*}$. On en extrait une sous-suite (v_i) telle que v_i est de longueur supérieure ou égale à i et v_i, v_{i+1} ont les mêmes i premières lettres de la manière suivante :

1. Supposons que la suite des longueurs, $(|u_n|)$, est bornée. Alors (u_n) contient une sous-suite strictement décroissante de mots ayant tous même longueur $\ell > 0$, donc la restriction de $>_{A^*}$ à A^ℓ est un ordre mal fondé. Mais ceci est impossible d'après la question précédente, généralisée par récurrence sur ℓ à un produit cartésien de ℓ ordres bien fondés. Donc la suite $(|u_n|)$ est non bornée et, quitte à se restreindre à une sous-suite, on peut supposer $|u_n| \geqslant n$ pour tout entier $n \geqslant 1$.

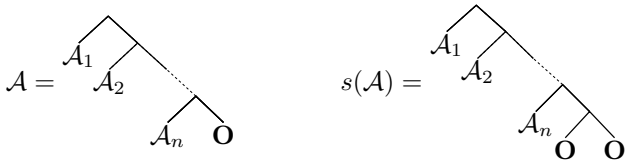
2. Soit $a_{n,1}$ la première lettre de u_n. Alors la suite $(a_{n,1})$ est une suite décroissante de lettres, elle ne prend qu'un nombre fini de valeurs puisque $>_A$ est un ordre bien fondé. Soit a_1 la plus petite de ces valeurs et n_1 un rang tel que $n \geqslant n_1 \Longrightarrow a_{n,1} = a_1$. On pose $v_1 = u_{n_1}$.

3. Soit $a_{n,2}$ la deuxième lettre de u_n, bien définie si $n \geqslant 2$. D'après le choix de n_1, la sous-suite $(a_{n,2})_{n>n_1}$ est décroissante donc il existe une lettre a_2 et un rang $n_2 > n_1$ tels que $n \geqslant n_2 \Longrightarrow a_{n,2} = a_2$. On pose $v_2 = u_{n_2}$.

4. Etc.

La suite (a_n) mise en évidence au cours de cette construction est une suite décroissante dans A car a_n, a_{n+1} sont deux lettres successives de v_{n+1}, mot décroissant. Donc cette suite ne prend qu'un nombre fini de valeurs, et il existe un rang p et une lettre a tels que $n \geqslant p \Longrightarrow a_n = a$. A partir de ce rang tous les v_n sont de la forme : $v_n = ma^{\ell_n}$ où $m = a_1 \ldots a_{p-1}$ et $\ell_n = |v_n| - p + 1$. Comme la suite (v_n) est strictement décroissante, la suite (ℓ_n) est une suite strictement décroissante d'entiers naturels, ce qui est impossible. On a ainsi prouvé que les propositions « $(A, >_A)$ est bien fondé » et « $(A^{\mathrm{dec}}, >_{A^*})$ est mal fondé » sont contradictoires. ■

Question 5.



Soient $\mathcal{A}_1, \ldots, \mathcal{A}_n$ des arbres binaires et $\mathcal{A}$, $s(\mathcal{A})$ les arbres dessinés ci-dessus ($s(\mathbf{O}) = \mathbf{O}\cdot\mathbf{O}$). On remarque que $\mathcal{A}$ est ordonné si et seulement $\mathcal{A}_1, \ldots, \mathcal{A}_n$ le sont et $\mathcal{A}_1 \succcurlyeq \ldots \succcurlyeq \mathcal{A}_n$. Dans ce cas, $s(\mathcal{A})$ est aussi un arbre ordonné et l'on a $s(\mathcal{A}) \succ \mathcal{A}$. On démontre de manière immédiate la propriété suivante :

que $\mathcal{A}$ soit ordonné ou non, pour tout arbre $\mathcal{B}$ on a $\mathcal{B} \succ \mathcal{A} \Longrightarrow \mathcal{B} \succcurlyeq s(\mathcal{A})$.

Donc $s(\mathcal{A})$ est le successeur de $\mathcal{A}$, aussi bien dans $\mathbb{A}$ que dans $\mathbb{O}$.

Question 6.

$\mathbb{O}_{\mathbf{O}} = \{\mathbf{O}\}$ donc la restriction de $\succ$ à $\mathbb{O}_{\mathbf{O}}$ est bien fondée. Supposons que pour tout arbre $\mathcal{A}$ de taille inférieure à n la restriction de $\succ$ à $\mathbb{O}_{\mathcal{A}}$ est bien fondée, et considérons un arbre $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$ de taille n. Si $\mathcal{B} \in \mathbb{O}_{\mathcal{A}}$ alors $\mathcal{B}$ est de la forme $\mathcal{B} = \mathcal{B}_1 \cdot (\mathcal{B}_2 \cdot (\ \ldots(\mathcal{B}_p \cdot \mathbf{O})\ldots\ ))$ avec $\mathcal{A}_1 \succcurlyeq \mathcal{B}_1 \succcurlyeq \ldots \succcurlyeq \mathcal{B}_p$, on lui associe le mot $\widetilde{\mathcal{B}} = \mathcal{B}_1 \ldots \mathcal{B}_p \in \mathbb{O}^{\text{dec}}_{\mathcal{A}_1}$. Cette association est injective, et on a clairement : $\mathcal{B} \succ \mathcal{C} \iff \widetilde{\mathcal{B}} \succ_{\mathbb{O}^*_{\mathcal{A}_1}} \widetilde{\mathcal{C}}$. $(\mathbb{O}_{\mathcal{A}_1}, \succ)$ est bien fondé par hypothèse de récurrence, donc $(\mathbb{O}^{\text{dec}}_{\mathcal{A}_1}, \succ_{\mathbb{O}^*_{\mathcal{A}_1}})$ ne contient pas de suite infinie strictement décroissante, et donc $(\mathbb{O}_{\mathcal{A}}, \succ)$ n'en contient pas non plus.

Question 7.

Si $(\mathbb{O}, \succ)$ contenait une suite $(\mathcal{A}_n)$ strictement décroissante, alors cette suite serait incluse dans $(\mathbb{O}_{\mathcal{A}_0}, \succ)$, en contradiction avec **6**.

Question 8.

On démontre par récurrence sur n que l'on a les propriétés :

$$\mathsf{arbre}(b, n) \in \mathbb{O}, \qquad \text{et} \qquad \forall\ m < n,\ \mathsf{arbre}(b, n) \succ \mathsf{arbre}(b, m).$$

Le cas $n = 0$ est trivial. Soit $n > 0$ tel que ces propriétés sont vérifiées pour tous les entiers inférieurs à n. On a $n = b^x + y$ où x est l'exposant maximal de n, donc $\mathsf{arbre}(b, n) = \mathsf{arbre}(b, x) \cdot \mathsf{arbre}(b, y)$. Par hypothèse de récurrence, $\mathsf{arbre}(b, x)$ et $\mathsf{arbre}(b, y)$ sont des arbres ordonnés, et si $\mathsf{arbre}(b, y) = \mathsf{arbre}(b, z) \cdot \mathsf{arbre}(b, t)$ alors z est l'exposant maximal de y donc $z \leqslant x$, d'où $\mathsf{arbre}(b, x) \succcurlyeq \mathsf{arbre}(b, z)$ par hypothèse de récurrence encore. Ceci prouve que $\mathsf{arbre}(b, n) \in \mathbb{O}$. Ensuite, si $m < n$ on a soit $m = 0$ d'où $\mathsf{arbre}(b, n) \succ \mathsf{arbre}(b, m) = \mathbf{O}$, soit $m = b^{x'} + y'$ où x' est l'exposant maximal de m, donc $x' < x$ ou $(x' = x,\ y' < y)$, ce qui implique, par hypothèse de récurrence toujours, $\mathsf{arbre}(b, x) \succ \mathsf{arbre}(b, x')$ ou $(\mathsf{arbre}(b, x) = \mathsf{arbre}(b, x')$ et $\mathsf{arbre}(b, y) \succ \mathsf{arbre}(b, y'))$, donc $\mathsf{arbre}(b, n) \succ \mathsf{arbre}(b, m)$ dans les deux cas.

Remarque : l'équivalence entre $>_{\mathbb{N}}$ et $\succ_{\mathbb{O}}$ restreinte aux arbres binaires représentant des entiers en base b a déjà été prouvée, par un raisonnement similaire, dans la partie **2**, question **7**.

Question 9.

Supposons qu'il existe une suite (g_k) de Goodstein infinie. On a $\mathsf{arbre}(k, g_k) = \mathsf{arbre}(k+1, g_{k+1}+1)$ par définition, et $\mathsf{arbre}(k+1, g_{k+1}+1) \succ \mathsf{arbre}(k+1, g_{k+1})$ d'après la question précédente. Donc la suite $(\mathsf{arbre}(k, g_k))$ est strictement décroissante dans $(\mathbb{O}, \succ)$, c'est impossible.

Remarque : voyons à quel rang la suite de Goodstein telle que $g_3 = 83$ atteint zéro. On a successivement :

$$g_6 = 6^7 - 1 = 5 \cdot 6^6 + \overline{555555}^6, \quad g_7 = 5 \cdot 7^7 + \overline{555554}^7, \quad \ldots, \quad g_{11} = 5 \cdot 11^{11} + \overline{555550}^{11}.$$

Dans la suite, on note K l'expression r^r où r est le rang de l'entier de Goodstein considéré. On voit facilement que :

$$g_{k-1} = 5K + \overline{abcde0}^{k-1} \implies g_{2k-1} = 5K + \overline{abcd(e-1)0}^{2k-1} \implies \ldots \implies g_{2^e k-1} = 5K + \overline{abcd00}^{2^e k-1}.$$

En particulier, $g_{383} = 5 \cdot 383^{383} + \overline{555500}^{383}$ est le premier terme g_k avec $k \geqslant 6$ dont l'écriture en base k se termine par deux zéros. Continuons :

$$g_{k-1} = 5K + \overline{abcd00}^{k-1} \implies g_{k2^k-1} = 5K + \overline{abc(d-1)00}^{k2^k-1}.$$

Donc le prochain terme après g_{383} dont l'écriture en base k se termine par deux zéros est :

$$g_{384 \cdot 2^{384}-1} = 5(384 \cdot 2^{384} - 1)^{384 \cdot 2^{384}-1} + \overline{555400}^{384 \cdot 2^{384}-1}.$$

Le premier entier k tel que l'écriture de g_k en base k se termine par trois zéros est défini par :

$$a = 384 \cdot 2^{384}, \quad b = a2^a, \quad c = b2^b, \quad d = c2^c, \quad e = d2^d, \quad k = e - 1.$$

Pour cette valeur de k, on a $g_k = 5K + \overline{555000}^k$. Il ne reste plus qu'à grignoter patiemment les trois 5 qui restent, avant d'attaquer le premier des cinq K...

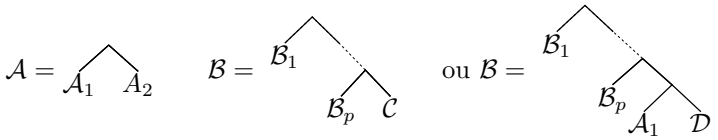
4. Un autre ordre sur les arbres

Question 1.

Lemme : soient $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$. Les arbres $\mathcal{B}$ tels que $\mathcal{A} \blacktriangleright \mathcal{B}$ sont les arbres de l'une des formes suivantes :

(1) $\mathcal{B} = \mathcal{B}_1 \cdot (\ \ldots(\mathcal{B}_p \cdot \mathcal{C})\ldots\ )$ avec $p \in \mathbb{N}$, $\mathcal{A}_1 \blacktriangleright \mathcal{B}_i$ et $\mathcal{A}_2 \not\blacktriangleright \mathcal{C}$

(2) $\mathcal{B} = \mathcal{B}_1 \cdot (\ \ldots(\mathcal{B}_p \cdot (\mathcal{A}_1 \cdot \mathcal{D}))\ldots\ )$ avec $p \in \mathbb{N}$, $\mathcal{A}_1 \blacktriangleright \mathcal{B}_i$ et $\mathcal{A}_2 \blacktriangleright \mathcal{D}$.



Démonstration : on constate par récurrence sur p que si $\mathcal{B}$ est donné par (1) ou par (2) alors on a bien $\mathcal{A} \blacktriangleright \mathcal{B}$. On démontre la réciproque par récurrence sur la taille n de $\mathcal{B}$. Pour $n = 0$, on a $\mathcal{B} = \mathbf{O}$ qui est de la forme (1) avec $p = 0$ et $\mathcal{C} = \mathbf{O}$. Si $n > 0$ alors $\mathcal{B} = \mathcal{B}_1 \cdot \mathcal{B}'$ et l'on a trois possibilités.

1. $\mathcal{A}_1 \blacktriangleright \mathcal{B}_1$ et $\mathcal{A} \blacktriangleright \mathcal{B}'$: alors $\mathcal{B}'$ est de l'une des formes (1) ou (2), et $\mathcal{B}$ aussi.
2. $\mathcal{A}_1 = \mathcal{B}_1$ et $\mathcal{A}_2 \blacktriangleright \mathcal{B}'$: alors $\mathcal{B}$ est de la forme (2) avec $p = 0$ et $\mathcal{D} = \mathcal{B}'$.
3. $\mathcal{A}_2 \not\blacktriangleright \mathcal{B}$: alors $\mathcal{B}$ est de la forme (1) avec $p = 0$ et $\mathcal{C} = \mathcal{B}$. ■

Transitivité de $\blacktriangleright$: on montre par récurrence sur n que si $\mathcal{A}, \mathcal{B}, \mathcal{E}$ sont des arbres dont la somme des tailles est inférieure ou égale à n tels que $\mathcal{A} \blacktriangleright \mathcal{B}$ et $\mathcal{E} \blacktriangleright \mathcal{A}$, alors $\mathcal{E} \blacktriangleright \mathcal{B}$. Pour $n = 0$ il n'y a rien à démontrer. Pour $n > 0$, on écrit $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$, $\mathcal{E} = \mathcal{E}_1 \cdot \mathcal{E}_2$ et l'on décompose $\mathcal{B}$ selon l'une des formes (1) ou (2) du lemme. Il y a cinq cas à examiner, et dans chaque cas on peut appliquer la propriété de transitivité aux arbres considérés car la somme des tailles de ces arbres est inférieure à n.

1. $\mathcal{E}_1 \blacktriangleright \mathcal{A}_1$, $\mathcal{E} \blacktriangleright \mathcal{A}_2$ et $\mathcal{B}$ est de la forme (1) : alors $\mathcal{E}_1 \blacktriangleright \mathcal{A}_1 \blacktriangleright \mathcal{B}_i$ et $\mathcal{E} \blacktriangleright \mathcal{A}_2 \not\blacktriangleright \mathcal{C}$. Donc $\mathcal{E}_1 \blacktriangleright \mathcal{B}_i$ et $\mathcal{E} \blacktriangleright \mathcal{C}$, d'où $\mathcal{E} \blacktriangleright \mathcal{B}$ par applications répétées de la règle 1 de la définition de $\blacktriangleright$.
2. $\mathcal{E}_1 \blacktriangleright \mathcal{A}_1$, $\mathcal{E} \blacktriangleright \mathcal{A}_2$ et $\mathcal{B}$ est de la forme (2) : idem (transitivité et itération de la règle 1).
3. $\mathcal{E}_1 = \mathcal{A}_1$, $\mathcal{E}_2 \blacktriangleright \mathcal{A}_2$ et $\mathcal{B}$ est de la forme (1) : alors $\mathcal{E}_1 \blacktriangleright \mathcal{B}_i$ et $\mathcal{E}_2 \blacktriangleright \mathcal{C}$, donc $\mathcal{B}$ est aussi de la forme (1) vis à vis de $\mathcal{E}$.
4. $\mathcal{E}_1 = \mathcal{A}_1$, $\mathcal{E}_2 \blacktriangleright \mathcal{A}_2$ et $\mathcal{B}$ est de la forme (2) : ici $\mathcal{B}$ est de la forme (2) vis à vis de $\mathcal{E}$.
5. $\mathcal{E}_2 \not\blacktriangleright \mathcal{A}$: alors $\mathcal{E}_2 \not\blacktriangleright \mathcal{A} \blacktriangleright \mathcal{B}$ d'où $\mathcal{E}_2 \blacktriangleright \mathcal{B}$ et $\mathcal{E} \blacktriangleright \mathcal{B}$. ■

Irréflexivité : montrons par récurrence sur n que si $\mathcal{A}$ est un arbre de taille inférieure ou égale à n, alors $\mathcal{A} \not\blacktriangleright \mathcal{A}$. C'est évident si $n = 0$. Pour $n > 0$, considérons un éventuel arbre $\mathcal{A}$ de taille n tel que $\mathcal{A} \blacktriangleright \mathcal{A}$. On écrit $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$, donc $\mathcal{A}_1 \cdot \mathcal{A}_2 \blacktriangleright \mathcal{A}_1 \cdot \mathcal{A}_2$, et parmi les trois possibilités données dans la définition de $\blacktriangleright$, les deux premières sont par hypothèse de récurrence exclues car $\mathcal{A}_1 \not\blacktriangleright \mathcal{A}_1$ et $\mathcal{A}_2 \not\blacktriangleright \mathcal{A}_2$. On a aussi $\mathcal{A}_2 \neq \mathcal{A}$, donc il reste une seule possibilité : $\mathcal{A}_2 \blacktriangleright \mathcal{A}$. Mais on a $\mathcal{A} \blacktriangleright \mathcal{A}_2$ d'après la propriété 3 de la définition de $\blacktriangleright$, donc $\mathcal{A}_2 \blacktriangleright \mathcal{A}_2$ par transitivité, ce qui est impossible. ■

Ordre total : Soient $\mathcal{A}, \mathcal{B}$ deux arbres binaires, on montre par récurrence sur la somme n de leurs tailles que les relations $\mathcal{A} \not\blacktriangleright \mathcal{B}$ et $\mathcal{B} \not\blacktriangleright \mathcal{A}$ impliquent $\mathcal{A} = \mathcal{B}$, le cas $n = 0$ étant comme d'habitude trivial. Pour $n > 0$ un des deux arbres est non nul, donc l'autre aussi car $\mathcal{X} \blacktriangleright \mathbf{O}$ pour tout arbre $\mathcal{X}$ non nul. On écrit $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$, $\mathcal{B} = \mathcal{B}_1 \cdot \mathcal{B}_2$, et l'on a les six faits suivants :

(1) $\mathcal{A}_1 \not\blacktriangleright \mathcal{B}_1$ ou $\mathcal{A} \not\blacktriangleright \mathcal{B}_2$, (2) $\mathcal{A}_1 \neq \mathcal{B}_1$ ou $\mathcal{A}_2 \not\blacktriangleright \mathcal{B}_2$, (3) $\mathcal{A}_2 \not\blacktriangleright \mathcal{B}$,

(4) $\mathcal{B}_1 \not\blacktriangleright \mathcal{A}_1$ ou $\mathcal{B} \not\blacktriangleright \mathcal{A}_2$, (5) $\mathcal{B}_1 \neq \mathcal{A}_1$ ou $\mathcal{B}_2 \not\blacktriangleright \mathcal{A}_2$, (6) $\mathcal{B}_2 \not\blacktriangleright \mathcal{A}$.

D'après l'hypothèse de récurrence, tous les couples d'arbres apparaissant dans ces négations sont comparables. (3) et (4) impliquent $\mathcal{A}_1 \not\blacktriangleright \mathcal{B}_1$, (1) et (6) impliquent $\mathcal{B}_1 \not\blacktriangleright \mathcal{A}_1$, donc $\mathcal{A}_1 = \mathcal{B}_1$ puis en reportant dans (2) et (5) : $\mathcal{A}_2 = \mathcal{B}_2$, soit finalement $\mathcal{A} = \mathcal{B}$. ■

Question 2.

On montre récurrence sur n que si $\mathcal{A}$ est de taille inférieure ou égale à n et $\mathcal{B}$ est un sous-arbre strict de $\mathcal{A}$, alors $\mathcal{A} \blacktriangleright \mathcal{B}$. Si $n = 0$ il n'y a rien à démontrer. Si $n > 0$, soit $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$. On a déjà remarqué que $\mathcal{A} \blacktriangleright \mathcal{A}_2$, donc si $\mathcal{B}$ est un sous-arbre de $\mathcal{A}_2$, on a $\mathcal{A} \blacktriangleright \mathcal{A}_2 \not\blacktriangleright \mathcal{B}$ par hypothèse de récurrence. De même, si $\mathcal{B}$ est un sous-arbre de $\mathcal{A}_1$, alors on a $\mathcal{A}_1 \not\blacktriangleright \mathcal{B}$ par hypothèse de récurrence, donc il reste à prouver que l'on a $\mathcal{A} \blacktriangleright \mathcal{A}_1$. Écrivons $\mathcal{A}_1 = \mathcal{A}_{11} \cdot (\ \ldots(\mathcal{A}_{1p} \cdot \mathbf{O})\ldots\ )$. Alors $\mathcal{A}_1 \blacktriangleright \mathcal{A}_{1i}$ pour tout i, donc $\mathcal{A}_1$ est de la forme (1) vis à vis de $\mathcal{A}$, ce qui prouve que $\mathcal{A} \blacktriangleright \mathcal{A}_1$. ■

Question 3.

Proposition : soient $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$ et $\mathcal{B} = \mathcal{B}_1 \cdot \mathcal{B}_2$ deux arbres binaires non nuls. Alors on a :

$$\mathcal{A}_1 \blacktriangleright \mathcal{B}_1 \Longrightarrow (\mathcal{A} \blacktriangleright \mathcal{B} \Longleftrightarrow \mathcal{A} \ \blacktriangleright \mathcal{B}_2),$$
$$\mathcal{A}_1 = \mathcal{B}_1 \Longrightarrow (\mathcal{A} \blacktriangleright \mathcal{B} \Longleftrightarrow \mathcal{A}_2 \blacktriangleright \mathcal{B}_2),$$
$$\mathcal{A}_1 \blacktriangleleft \mathcal{B}_1 \Longrightarrow (\mathcal{A} \blacktriangleright \mathcal{B} \Longleftrightarrow \mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot \ \mathcal{B} \ ).$$

Démonstration : si $\mathcal{A}_1 \blacktriangleright \mathcal{B}_1$ alors $\mathcal{A} \blacktriangleright \mathcal{B}_2 \Longrightarrow \mathcal{A} \blacktriangleright \mathcal{B}$ d'après la règle 1, et $\mathcal{A} \not\blacktriangleright \mathcal{B}_2 \Longrightarrow \mathcal{A}_2 \not\blacktriangleright \mathcal{B}_2 \Longrightarrow A \not\blacktriangleright \mathcal{B}$ car aucune règle ne s'applique à ce cas. Le cas $\mathcal{A}_1 = \mathcal{B}_1$ se traite de même, et le cas $\mathcal{A}_1 \blacktriangleleft \mathcal{B}_1$ est une réécriture de la règle 3. ■

On peut donc décider de la règle éventuelle à appliquer pour comparer $\mathcal{A}$ et $\mathcal{B}$ dès que l'on connaît l'ordre de classement de $\mathcal{A}_1$ et $\mathcal{B}_1$. Ceci donne l'algorithme suivant :

compare$(\mathcal{A}, \mathcal{B})$ =
  si $\mathcal{A} = \mathbf{O}$ *et* $\mathcal{B} = \mathbf{O}$ ***alors retourne*** egal
  sinon, si $\mathcal{A} = \mathbf{O}$ ***alors retourne*** inf
  sinon, si $\mathcal{B} = \mathbf{O}$ ***alors retourne*** sup
  sinon, *soient* $\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}_1, \mathcal{B}_2$ *tels que* $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$ *et* $\mathcal{B} = \mathcal{B}_1 \cdot \mathcal{B}_2$ *:*
    $r :=$ *compare*$(\mathcal{A}_1, \mathcal{B}_1)$
    si $r =$ sup ***alors retourne*** *compare*$(\mathcal{A}, \mathcal{B}_2)$
    sinon, si $r =$ egal ***alors retourne*** *compare*$(\mathcal{A}_2, \mathcal{B}_2)$
    sinon, retourne compare$(\mathcal{A}_2, \mathcal{B})$
fin

triangle_noir$(\mathcal{A}, \mathcal{B})$ = (*compare*$(\mathcal{A}, \mathcal{B})$ = sup)

La terminaison de *compare* s'établit par récurrence sur la somme des tailles des arbres considérés. En ce qui concerne la complexité, on montre également par récurrence sur la somme des tailles a, b de $\mathcal{A}, \mathcal{B}$ que le nombre d'appels à *compare* effectués pour comparer $\mathcal{A}$ et $\mathcal{B}$ est majoré par $ab + 1$.

Question 4.

Soit $\mathcal{B}$ un arbre binaire quelconque, on pose $s(\mathcal{B}) = \mathbf{O} \cdot \mathcal{B}$ et on montre que $s(\mathcal{B})$ est le successeur de $\mathcal{B}$ pour $\blacktriangleright$. Déjà on a clairement $s(\mathcal{B}) \blacktriangleright \mathcal{B}$, donc il reste à prouver l'implication :

$$\forall\ \mathcal{A} \in \mathbb{A}, \forall\ \mathcal{B} \in \mathbb{A},\ \mathcal{A} \blacktriangleright \mathcal{B} \Longrightarrow \mathcal{A} \blacktriangleright\!\!\!\!\cdot\ \mathbf{O} \cdot \mathcal{B}.$$

On démontre cette propriété ... par récurrence sur la taille n de $\mathcal{A}$. Pour $n = 0$ il n'y a rien à démontrer. Si l'implication est vraie pour tous arbres $\mathcal{A}, \mathcal{B}$ tels que $\mathcal{A}$ est de taille inférieure à n, considérons un arbre $\mathcal{A}$ de taille n tel que $\mathcal{A} \blacktriangleright \mathcal{B}$, arbre que l'on écrit $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$:

si $\mathcal{A}_1 \neq \mathbf{O}$, alors $\mathcal{A} \blacktriangleright \mathbf{O} \cdot \mathcal{B}$ d'après la propriété 1 de la définition de $\blacktriangleright$.
si $\mathcal{A}_1 = \mathbf{O}$ et $\mathcal{B} = \mathbf{O}$, alors $\mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot\ \mathbf{O}$ donc $\mathcal{A} \blacktriangleright\!\!\!\!\cdot\ \mathbf{O} \cdot \mathbf{O}$.
si $\mathcal{A}_1 = \mathbf{O}$ et $\mathcal{B} = \mathcal{B}_1 \cdot \mathcal{B}_2$ avec $\mathcal{B}_1 \neq \mathbf{O}$, on a $\mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot\ \mathcal{B}$ (cas 3 de la définition de $\blacktriangleright$), d'où $\mathcal{A} \blacktriangleright\!\!\!\!\cdot\ \mathbf{O} \cdot \mathcal{B}$.
si $\mathcal{A}_1 = \mathbf{O}$ et $\mathcal{B} = \mathbf{O} \cdot \mathcal{B}_2$, alors on a $\mathcal{A}_2 \blacktriangleright \mathcal{B}_2$ (cas 2 de la définition de $\blacktriangleright$) ou $\mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot\ \mathcal{B}$ (cas 3) :
  si $\mathcal{A}_2 \blacktriangleright \mathcal{B}_2$ alors $\mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot\ \mathbf{O} \cdot \mathcal{B}_2$ par hypothèse de récurrence, d'où $\mathcal{A} \blacktriangleright\!\!\!\!\cdot\ \mathbf{O} \cdot \mathcal{B}$.
  si $\mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot\ \mathcal{B}$ alors $\mathcal{A} = \mathbf{O} \cdot \mathcal{A}_2 \blacktriangleright\!\!\!\!\cdot\ \mathbf{O} \cdot \mathcal{B}$. ■

En conclusion : $s_{\blacktriangleright}(\mathcal{B}) = \mathbf{O} \cdot \mathcal{B}$.

Question 5.

Le procédé de construction des $\mathcal{A}_i$ est clairement valide, et il fournit une suite infinie strictement décroissante. Par ailleurs la suite des longueurs des $\mathcal{A}_i$ est croissante au sens large. Enfin, $\mathbf{O}$ n'ayant pas de minorant ne peut appartenir à une suite infinie strictement décroissante, donc $\mathcal{A}_i \neq \mathbf{O}$. Écrivons $\mathcal{A}_i = \mathcal{B}_i \cdot \mathcal{C}_i$. Parmi les trois possibilités pour avoir $\mathcal{B}_i \cdot \mathcal{C}_i \blacktriangleright \mathcal{B}_{i+1} \cdot \mathcal{C}_{i+1}$ la troisième est exclue, car :

si $\mathcal{C}_i \blacktriangleright \mathcal{A}_{i+1}$ alors $\mathcal{A}_{i-1} \blacktriangleright \mathcal{A}_i \blacktriangleright \mathcal{C}_i \blacktriangleright \mathcal{A}_{i+1}$ donc on peut remplacer dans la suite $(\mathcal{A}_n)$ l'arbre $\mathcal{A}_i$ par $\mathcal{C}_i$ sans perdre la décroissance stricte, en contradiction avec le caractère minimal de $\mathcal{A}_i$.

si $\mathcal{C}_i = \mathcal{A}_{i+1}$ alors $\mathcal{A}_{i+1}$ a une taille strictement inférieure à $\mathcal{A}_i$, c'est impossible.

On en déduit que la comparaison de $\mathcal{A}_i$ avec $\mathcal{A}_{i+1}$ se fait toujours sur les cas 1 ou 2, donc la suite $(\mathcal{B}_i)$ est décroissante au sens large. Si elle prend une infinité de valeurs distinctes, alors on peut en extraire une suite strictement décroissante $(\mathcal{B}_{i_k})$, donc la suite $(\mathcal{A}_0, \ldots, \mathcal{A}_{i_0-1}, \mathcal{B}_{i_0}, \mathcal{B}_{i_1}, \ldots)$ est une suite infinie strictement décroissante contredisant la pertinence du choix de $\mathcal{A}_{i_0}$. Si au contraire la suite $(\mathcal{B}_i)$ ne prend qu'un nombre fini de valeurs distinctes, alors à partir d'un certain rang seul le cas 2 s'applique, la suite $(\mathcal{C}_i)$ est strictement décroissante à partir de ce rang, et on conclut là aussi qu'un des arbres $\mathcal{A}_i$ a été mal choisi.

Question 6.

On démontre par récurrence sur n que si $\mathcal{A}$ et $\mathcal{B}$ sont des arbres ordonnés de tailles inférieures ou égales à n tels que $\mathcal{A} \succ \mathcal{B}$ alors $\mathcal{A} \blacktriangleright \mathcal{B}$. Il n'y a rien à démontrer si $n = 0$, et c'est évident dans le cas général si $\mathcal{B} = \mathbf{O}$. Pour $n > 0$ et $\mathcal{B} \neq \mathbf{O}$ on a aussi $\mathcal{A} \neq \mathbf{O}$. Écrivons $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$ et $\mathcal{B} = \mathcal{B}_1 \cdot (\ \ldots(\mathcal{B}_p \cdot \mathbf{O})\ldots\ ) = \mathcal{B}_1 \cdot \mathcal{C}$:

si $\mathcal{A}_1 = \mathcal{B}_1$ alors $\mathcal{A}_2 \succ \mathcal{C}$, donc par hypothèse de récurrence $\mathcal{A}_2 \blacktriangleright \mathcal{C}$, et l'on a bien $\mathcal{A} \blacktriangleright \mathcal{B}$.

sinon $\mathcal{A}_1 \succ \mathcal{B}_1$, donc $\mathcal{A}_1 \succ \mathcal{B}_i$ pour tout i puisque $\mathcal{B}$ est ordonné. On a alors, par hypothèse de récurrence, $\mathcal{A}_1 \blacktriangleright \mathcal{B}_i$ pour tout i, ce qui prouve que $\mathcal{A} \blacktriangleright \mathcal{B}$ d'après le lemme vu en question **1**.

Question 7.

C'est évident.

Question 8.

On peut démontrer assez facilement que si ψ existe, alors :

$$\psi((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O}) = (\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O} \quad \text{et} \quad \psi(((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O}) = (\mathbf{O}\cdot(\mathbf{O}\cdot\mathbf{O}))\cdot\mathbf{O},$$

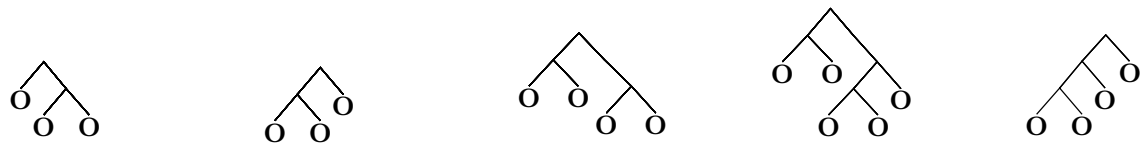
mais le calcul de $\psi((((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O})$ est plus délicat, et la démonstration de l'existence de ψ nécessite d'élever le niveau du discours. Nous allons donc ici présenter brièvement la théorie des ordinaux de Cantor et en déduire une construction explicite de ψ.

Nombres ordinaux

Soit ω un symbole abstrait. On convient de représenter un arbre ordonné $\mathcal{A}$ par une expression formelle en ω, $\mathsf{expr}(\mathcal{A})$, où la fonction expr est définie récursivement par :

$$\mathsf{expr}(\mathbf{O}) = 0, \qquad \mathsf{expr}(\mathcal{A}_1 \cdot \mathcal{A}_2) = \omega^{\mathsf{expr}(\mathcal{A}_1)} + \mathsf{expr}(\mathcal{A}_2).$$

Cette définition est copiée sur celle de la fonction erbra donnée dans la partie **1**, question **5** ; on a juste remplacé la base b utilisée dans cette question par la base abstraite ω. Par analogie avec la décomposition complète d'un entier en base b, on convient de simplifier ω^0 en 1, $\omega^a + \ldots + \omega^a$ en $k\omega^a$ où $k \in \mathbb{N}$ est le nombre de ω^a, et de ne pas écrire le zéro final dans une somme de plusieurs termes. Les expressions formelles obtenues sont appelées *nombres ordinaux*. La figure ci-dessous présente quelques arbres ordonnés et les ordinaux associés.



2 ω $\omega + 1$ 2ω ω^ω

La relation d'ordre $\succ$ entre arbres ordonnés induit une relation d'ordre entre nombres ordinaux, aussi notée $\succ$. On a donc :

$$0 \prec 1 \prec 2 \prec \ldots \prec 1000^{1000} \prec \ldots \prec \omega \prec \omega + 1 \prec \ldots \prec \omega^2 \prec \ldots \prec \omega^3 \prec \ldots \prec \omega^\omega \prec \ldots \prec \omega^{\omega^\omega} \prec \ldots$$

Ainsi, ω apparaît comme un ordinal infini, au sens où il est plus grand que tout nombre entier, de même que $\omega+1$, ω^ω, etc. En fait ω est le plus petit ordinal infini, et l'arbre ordonné associé à un ordinal est la décomposition

complète de cet ordinal en base ω. Il existe dans la théorie des ensembles de Cantor d'autres ordinaux n'ayant pas de décomposition complète en base ω, par exemple :

$$\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}$$

où le « nombre » d'exposants est égal à ω, mais nous n'en auront pas besoin ici. Dans la suite de cet exposé, le terme « ordinal » désigne donc en fait un « ordinal représentable par un arbre ordonné », et nous conviendrons de confondre un nombre entier k avec l'ordinal fini correspondant.

Décomposition euclidienne d'un ordinal infini

Soit $a = \omega^{a_1} + \ldots + \omega^{a_n} + k$ un ordinal écrit comme une somme de puissances décroissantes de ω et d'un entier k. L'exposant maximal de a est indéfini si $a = 0$, nul si $a = k$, et égal à a_n si a est un ordinal infini. En termes d'arbres ordonnés, si $\mathcal{A}$ est l'arbre ordonné associé à a, alors l'arbre ordonné associé à l'exposant maximal de a est la branche gauche de $\mathcal{A}$ lorsque $\mathcal{A} \neq \mathbf{O}$. Si a admet un exposant maximal a_1 et si $a_1 \succ 0$, alors le sur-exposant maximal de a est par définition l'exposant maximal de a_1. Si a_1 est indéfini ou si $a_1 = 0$, c'est-à-dire si a est un ordinal fini, alors a n'a pas de sur-exposant maximal.

Soit $a = \omega^{a_1} + \ldots + \omega^{a_n} + k$ un ordinal infini et b son sur-exposant maximal. Comme la suite (a_i) est décroissante pour $\succ$, il existe un entier $p \in [\![1, n]\!]$ tel que $a_1 \succcurlyeq \ldots \succcurlyeq a_p \succcurlyeq \omega^b \succ a_{p+1}$. Pour $i \leqslant p$, écrivons $a_i = \omega^b + a'_i$. La décomposition euclidienne de a est :

$$a = \omega^{\omega^b} q + r \qquad \text{avec } q = \omega^{a'_1} + \ldots + \omega^{a'_p} \text{ et } r = \omega^{a_{p+1}} + \ldots + \omega^{a_n} + k.$$

Dans cette décomposition, la multiplication de ω^{ω^b} par q est à interpréter formellement, on additionne ω^b à chaque exposant de la décomposition de q. Cette décomposition n'est pas due à Euclide, mais nous l'avons appelée ainsi par analogie avec la division euclidienne des entiers. En effet, on a par construction $0 \preccurlyeq r \prec \omega^{\omega^b}$. Remarquons aussi que nous avons $0 \prec q$, et le sur-exposant maximal de q, s'il existe, est inférieur ou égal à b.

En termes d'arbres ordonnés, on obtient la décomposition euclidienne d'un arbre $\mathcal{A} \succcurlyeq (\mathbf{O} \cdot \mathbf{O}) \cdot \mathbf{O}$ de la manière suivante :

1. On écrit $\mathcal{A} = \mathcal{A}_1 \cdot (\ \ldots(\mathcal{A}_n \cdot (\mathbf{O} \cdot (\ \ldots(\mathbf{O} \cdot \mathbf{O})\ldots\ )))\ldots\ )$ avec $\mathcal{A}_i \succ \mathbf{O}$.
2. On décompose $\mathcal{A}_1 = \mathcal{B} \cdot \mathcal{A}'_1,\ \ldots,\ \mathcal{A}_p = \mathcal{B} \cdot \mathcal{A}'_p$ où p est le plus grand entier tel que $\mathcal{A}_p$ ait même branche gauche que $\mathcal{A}_1$.
3. $\mathcal{B}$ est l'arbre ordonné associé à b, l'arbre ordonné associé à q est $\mathcal{Q} = \mathcal{A}'_1 \cdot (\ \ldots(\mathcal{A}'_p \cdot \mathbf{O})\ldots\ )$ et l'arbre ordonné associé à r est le reste de $\mathcal{A}$: $\mathcal{R} = \mathcal{A}_{p+1} \cdot (\ \ldots(\mathcal{A}_n \cdot (\mathbf{O} \cdot (\ \ldots(\mathbf{O} \cdot \mathbf{O})\ldots\ )))\ldots\ )$.

Concaténation de deux arbres

Soient $\mathcal{A}, \mathcal{B} \in \mathbb{A}$. On construit un nouvel arbre noté $\mathcal{A} :: \mathcal{B}$ en remplaçant dans $\mathcal{A}$ le $\mathbf{O}$ le plus à droite par $\mathcal{B}$. Formellement, $\mathcal{A} :: \mathcal{B}$ est défini par les relations :

$$\mathbf{O} :: \mathcal{B} = \mathcal{B}, \qquad (\mathcal{A}_1 \cdot \mathcal{A}_2) :: \mathcal{B} = \mathcal{A}_1 \cdot (\mathcal{A}_2 :: \mathcal{B}).$$

On montre facilement par récurrence sur la taille de $\mathcal{B}$ que si $\mathcal{A}_1 \blacktriangleright \mathcal{A}_2$ alors $\mathcal{A}_1 :: \mathcal{B} \blacktriangleright \mathcal{A}_2 :: \mathcal{B}$.

La fonction ψ

Soit $\mathcal{A} \in \mathbb{O}$ et a l'ordinal représenté par $\mathcal{A}$. On note $\psi(\mathcal{A})$ l'arbre défini récursivement par les relations suivantes :

Si a est un ordinal fini alors $\psi(\mathcal{A}) = \mathcal{A}$.

Si a est un ordinal infini, soit $a = \omega^{\omega^b} q + r$ sa décomposition euclidienne. On note b', q'' les ordinaux suivants :
si b est un ordinal fini alors $b' = b + 1$, sinon $b' = b$;
si q est un ordinal fini alors $q'' = q - 1$, sinon $q'' = q$.
Soient $\mathcal{B}', \mathcal{Q}'', \mathcal{R}$ les arbres ordonnés associés à b', q'', r. Alors $\psi(\mathcal{A}) = \psi(\mathcal{R}) :: (\psi(\mathcal{B}') \cdot \psi(\mathcal{Q}''))$.

Proposition : la fonction ψ est bien définie et elle réalise un isomorphisme entre les ordres $(\mathbb{O}, \succ)$ et $(\mathbb{A}, \blacktriangleright)$.

La démonstration est scindée en plusieurs étapes.

1. Bonne définition de ψ : si a est un ordinal fini alors $\psi(\mathcal{A})$ est bien défini. Si a est infini, alors les arbres $\mathcal{B}'$, $\mathcal{Q}''$ et $\mathcal{R}$ ont strictement moins de nœuds que $\mathcal{A}$ et un arbre ayant zéro ou un nœud représente un ordinal fini. Donc, par récurrence sur la taille de $\mathcal{A}$, $\psi(\mathcal{A})$ est bien défini.

2. Forme de $\psi(\mathcal{A})$: on montre par une récurrence immédiate sur la taille de $\mathcal{A}$ que si $\mathcal{A} \neq \mathbf{O}$ alors $\psi(\mathcal{A}) \neq \mathbf{O}$, et $\psi(\mathcal{A})$ est de la forme $\psi(\mathcal{C}_1)\cdot(\ \ldots(\psi(\mathcal{C}_k)\cdot\mathbf{O})\ldots\ )$ où chaque $\mathcal{C}_i$ est un arbre ordonné de taille inférieure à celle de $\mathcal{A}$ tel que $C_i = \mathbf{O}$ si $\mathcal{A}$ représente un ordinal fini, et $\mathcal{C}_i \preccurlyeq \mathcal{B}'$ si $\mathcal{A}$ représente un ordinal infini avec les notations de la définition de ψ.

3. Croissance de ψ : on montre par récurrence sur n que si $\mathcal{A}_1$ et $\mathcal{A}_2$ sont deux arbres ordonnés de tailles inférieures ou égales à n tels que $\mathcal{A}_1 \succ \mathcal{A}_2$, alors $\psi(\mathcal{A}_1) \blacktriangleright \psi(\mathcal{A}_2)$. Notons a_1, a_2 les ordinaux associés à $\mathcal{A}_1, \mathcal{A}_2$ et $a_1 = \omega^{\omega^{b_1}}q_1 + r_1$, $a_2 = \omega^{\omega^{b_2}}q_2 + r_2$ les décompositions euclidiennes éventuelles de a_1 et a_2. Pour $n = 0$ il n'y a rien à prouver. Si a_1 est fini alors a_2 l'est aussi donc $\psi(\mathcal{A}_1) = \mathcal{A}_1 \blacktriangleright \mathcal{A}_2 = \psi(\mathcal{A}_2)$ car $\blacktriangleright$ et $\succ$ coïncident sur $\mathbb{O}$. Si a_1 est infini et a_2 est fini, alors comme $b'_1 \neq 0$, on a : $(\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O} \blacktriangleleft \psi(\mathcal{B}'_1)\cdot\mathbf{O} \blacktriangleleft \psi(\mathcal{B}'_1)\cdot\psi(\mathcal{Q}''_1) \blacktriangleleft \psi(\mathcal{A}_1)$, et $(\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O}$ majore strictement pour $\blacktriangleright$ tout arbre ordonné représentant un ordinal fini, d'où $\psi(\mathcal{A}_1) \blacktriangleright \psi(\mathcal{A}_2)$. Supposons à présent a_1 et a_2 infinis. On distingue trois cas.

Cas $b_1 \succ b_2$: alors $b'_1 \succ b'_2$ donc $\psi(\mathcal{B}'_1) \blacktriangleright \psi(\mathcal{B}'_2)$ par hypothèse de récurrence. $\psi(\mathcal{A}_2)$ est de la forme $\psi(\mathcal{C}_1)\cdot(\ \ldots(\psi(\mathcal{C}_k)\cdot\mathbf{O})\ldots\ )$ où $\mathcal{C}_i \preccurlyeq \mathcal{B}'_2$ pour tout i, d'où $\mathcal{C}_i \prec \mathcal{B}'_1$ par transitivité, et $\psi(\mathcal{C}_i) \blacktriangleleft \psi(\mathcal{B}'_1)$ par hypothèse de récurrence. On en déduit $\psi(\mathcal{A}_2) \blacktriangleleft \psi(\mathcal{B}'_1)\cdot\mathbf{O} \blacktriangleleft \psi(\mathcal{A}_1)$.

Cas $b_1 = b_2$, $q_1 \succ q_2$: alors $q''_1 \succ q''_2$ donc $\psi(\mathcal{Q}''_1) \blacktriangleright \psi(\mathcal{Q}''_2)$ par hypothèse de récurrence. On a alors $\psi(\mathcal{B}'_1)\cdot\psi(\mathcal{Q}''_1) \blacktriangleright \psi(\mathcal{B}'_2)\cdot\psi(\mathcal{Q}''_2)$, puis $\psi(\mathcal{B}'_1)\cdot\psi(\mathcal{Q}''_1) \blacktriangleright \psi(\mathcal{R}_2) :: (\psi(\mathcal{B}'_2)\cdot\psi(\mathcal{Q}''_2))$ d'après la forme de $\mathcal{R}_2$ et le fait que $r_2 \prec \omega^{\omega^{b_2}}$, d'où $\psi(\mathcal{A}_1) \blacktriangleright \psi(\mathcal{B}'_1)\cdot\psi(\mathcal{Q}''_1) \blacktriangleright \psi(\mathcal{A}_2)$.

Cas $b_1 = b_2$, $q_1 = q_2$, $r_1 \succ r_2$: alors $\psi(\mathcal{R}_1) \blacktriangleright \psi(\mathcal{R}_2)$ par hypothèse de récurrence, et l'on en déduit, d'après la propriété de croissance de la concaténation : $\psi(\mathcal{R}_1) :: (\psi(\mathcal{B}'_1)\cdot\psi(\mathcal{Q}''_1)) \blacktriangleright \psi(\mathcal{R}_2) :: (\psi(\mathcal{B}'_2)\cdot\psi(\mathcal{Q}''_2))$.

4. Bijectivité de ψ : on sait déjà que ψ est injective, puisque strictement croissante entre deux ensembles totalement ordonnés. Il reste à prouver que tout arbre $\mathcal{X} \in \mathbb{A}$ admet un antécédent par ψ. On procède par récurrence sur la taille de $\mathcal{X}$. Si $\mathcal{X}$ est l'arbre ordonné associé à un ordinal fini alors $\mathcal{X} = \psi(\mathcal{X})$. Sinon, soit $\mathcal{X} = \mathcal{X}_1\cdot(\ \ldots(\mathcal{X}_n\cdot\mathbf{O})\ldots\ )$, et soit p le premier indice tel que $\mathcal{X}_p = \max(\mathcal{X}_1, \ldots, \mathcal{X}_n)$. On écrit $\mathcal{X} = \mathcal{U} :: (\mathcal{X}_1\cdot\mathcal{V})$ et $\mathcal{U}, \mathcal{X}_1, \mathcal{V}$ ont des tailles inférieures à celle de $\mathcal{X}$, donc il existe des arbres ordonnés $\mathcal{R}, \mathcal{C}, \mathcal{D}$ tels que $\mathcal{U} = \psi(\mathcal{R})$, $\mathcal{X}_1 = \psi(\mathcal{C})$ et $\mathcal{V} = \psi(\mathcal{D})$. Soient r, c, d les ordinaux associés. Avec les notations de la définition de ψ, il existe des ordinaux b, q tels que $c = b'$ et $d = q''$ car $c \succ 0$, $\mathcal{X}$ ne représentant pas un ordinal fini. Par choix de p et d'après la forme des arbres $\psi(\mathcal{R})$ et $\psi(\mathcal{D}) = \psi(\mathcal{Q}'')$, b majore strictement le sur-exposant maximal de r s'il existe, et b majore au sens large le sur-exposant maximal de q s'il existe, donc $\omega^{\omega^b}q + r$ est une décomposition euclidienne valide. Soit a l'ordinal ayant cette décomposition et $\mathcal{A}$ l'arbre ordonné associé à a, on a $\mathcal{X} = \psi(\mathcal{A})$. ■

Réponse à la question posée

La fonction ψ construite répond aux conditions de l'énoncé, et c'est la seule. On a :

$$\psi((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O}) = (\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O},$$
$$\psi(((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O}) = (\mathbf{O}\cdot(\mathbf{O}\cdot\mathbf{O}))\cdot\mathbf{O},$$
$$\psi((((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O}) = (((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O})\cdot\mathbf{O}.$$

Démonstration : puisque ψ est un isomorphisme entre $(\mathbb{O}, \succ)$ et $(\mathbb{A}, \blacktriangleright)$, ψ commute avec les fonctions « successeur » et « borne supérieure » et l'on a aussi $\psi(\mathbf{O}) = \mathbf{O}$ par définition, donc ψ est une solution au problème posé. On obtient les valeurs de $\psi((\mathbf{O}\cdot\mathbf{O})\cdot\mathbf{O}))$, etc. par application de la définition de ψ.

Supposons qu'il existe une autre solution $\psi_1 \neq \psi$ et considérons $\mathbb{E} = \{\mathcal{A} \in \mathbb{O} \text{ tq } \psi_1(\mathcal{A}) \neq \psi(\mathcal{A})\}$. On démontre par l'absurde que $\mathbb{E}$ n'a pas de plus petit élément pour l'ordre $\succ$: si $\mathcal{A} = \min_{\succ}(\mathbb{E})$ alors pour tout arbre $\mathcal{B} \in \mathbb{O}$ tel que $\mathcal{B} \prec \mathcal{A}$, on a $\psi_1(\mathcal{B}) = \psi(\mathcal{B})$, d'où $\psi_1(s_{\succ}(\mathcal{B})) = s_{\blacktriangleright}(\psi_1(\mathcal{B})) = s_{\blacktriangleright}(\psi(\mathcal{B})) = \psi(s_{\succ}(\mathcal{B}))$. On en déduit :

$$\forall\ \mathcal{B} \in \mathbb{O},\ s_{\succ}(\mathcal{B}) \preccurlyeq \mathcal{A} \Longrightarrow \psi_1(s_{\succ}(\mathcal{B})) = \psi(s_{\succ}(\mathcal{B})),$$

et ceci est contradictoire avec la compatibilité de ψ et ψ_1 avec la borne supérieure, car $\mathcal{A}$ est la borne supérieure des arbres $s_\succ(\mathcal{B})$ considérés. L'hypothèse $\mathcal{A} = \min_\succ(\mathbb{E})$ est donc absurde, comme annoncé. Mais alors on obtient à nouveau une contradiction, car $\mathbb{E}$ est un ensemble non vide sans plus petit élément, donc on peut construire de proche en proche une suite strictement décroissante d'éléments de $\mathbb{E}$, en contradiction avec la bonne fondation de $(\mathbb{O}, \succ)$. Tout compte fait l'existence d'une fonction $\psi_1 \neq \psi$ est impossible. ■

Les auteurs de ce corrigé s'excusent d'avoir mis en œuvre une machinerie aussi lourde pour répondre à cette question, mais ils ont séché dessus pendant plusieurs jours et n'ont pas trouvé de moyen plus simple pour la résoudre.

5. Un ordre sur les mots

Coquille de l'énoncé. Lire : $\varphi : \mathbb{N} \longrightarrow \mathbb{N}$ *strictement croissante.*

Question 1.

$(a) \Longrightarrow (b)$: on peut prendre $> = \sqsupset$, donc $(E, >)$ est bien fondé. Supposons que E contient une antichaîne infinie. On extrait de cette antichaîne une suite (x_i) sans répétitions, et on prolonge $>$ en posant :

$$(x \sqsupset y) \Longleftrightarrow (x > y) \text{ ou (il existe } i > j \text{ tels que } x \geqslant x_j \text{ et } x_i \geqslant y).$$

Vérifions que $\sqsupset$ est un ordre strict. Si x, y, z sont tels que $x \sqsupset y$ et $y \sqsupset z$ alors il y a 4 cas :

$x > y$ et $y > z$: alors $x > z$ donc $x \sqsupset z$.

$x > y$, $y \geqslant x_j$, $x_i \geqslant z$, $i > j$: alors $x \geqslant x_j$ et $x_i \geqslant z$ donc $x \sqsupset z$.

$x \geqslant x_j$, $x_i \geqslant y$, $y > z$, $i > j$: idem.

$x \geqslant x_j$, $x_i \geqslant y$, $y \geqslant x_\ell$, $x_k \geqslant z$, $i > j$, $k > \ell$: alors $x_i \geqslant x_\ell$ donc $k > \ell = i > j$ et $x \geqslant x_j$, $x_k \geqslant z$, d'où $x \sqsupset z$.

En prenant $z = x$ dans la discussion précédente, on constate immédiatement que les relations $x \sqsupset y$ et $y \sqsupset x$ sont contradictoires, donc $\sqsupset$ est bien un ordre strict prolongeant $>$. Pourtant $\sqsupset$ est mal fondé car la suite (x_i) est strictement décroissante pour cet ordre, on a obtenu la contradiction cherchée.

$(b) \Longrightarrow (c)$: on raisonne encore par l'absurde en supposant qu'il existe une mauvaise suite, (x_i). On pose $x^0 = x_0$. Comme x_0 n'est majoré par aucun x_i tel que $i > 0$, on peut partitionner la suite $(x_i)_{i \geqslant 1}$ en deux sous-suites : la sous-suite constituée des x_i tels que $x_i < x_0$, et la sous-suite constituée des autres x_i, qui sont incomparables à x_0. L'une de ces deux sous-suites est infinie, notons la (x_i^1) et soit x^1 son premier terme. On partitionne alors la sous-suite $(x_i^1)_{i \geqslant 1}$ en éléments inférieurs à x^1 et en éléments incomparables à x^1, et on nomme (x_i^2) celle qui est infinie ou l'une des deux si elles sont toutes deux infinies, et x^2 le premier terme de cette suite. Et ainsi de suite. On a ainsi constitué une suite (x^i) telle que pour $i < j$, soit $x^i > x^j$, soit x^i et x^j sont incomparables, et le cas qui a lieu est indépendant de j. Puisque $(E, >)$ est bien fondé, la suite (x^i) ne contient aucune sous-suite strictement décroissante, donc il existe une infinité d'indices i tels que x^i et x^{i+1} sont incomparables. L'ensemble des x^i correspondants à ces indices constitue une antichaîne infinie.

$(c) \Longrightarrow (a)$: s'il existe un ordre $\sqsupset$ mal fondé prolongeant $>$, soit (x_i) une suite infinie strictement décroissante pour cet ordre. Elle est bonne pour $>$, donc il existe $i < j$ tels que $x_i \leqslant x_j$. Comme $x_i \sqsupset x_j$, on a aussi $x_i \neq x_j$, d'où $x_i < x_j$ puis $x_i \sqsubset x_j$, contradiction.

$(b) \Longrightarrow (d)$: même raisonnement que $(b) \Longrightarrow (c)$, en partitionnant la suite (x_i^k) en trois sous-suites selon que $x_i^k < x^k$, $x_i^k \geqslant x^k$ et x_i^k est incomparable à x^k.

$(d) \Longrightarrow (c)$: évident.

Question 2.

L'énoncé est confus. Disons que A est un alphabet fini ou infini ordonné, que α désigne un mot non vide dans la règle $\alpha \succcurlyeq \varepsilon$, et que α et β désignent des mots quelconques dans les deux règles suivantes. Disons aussi que ça commence à bien faire tous ces symboles de relation d'ordre plus bizarres les uns que les autres et impossibles à calligraphier calmement au bout de 4h de réflexion, et qu'il est temps que le problème se termine.

Proposition : soient $\alpha = a_1 \dots a_n$ et $\beta = b_1 \dots b_p \in A^*$. Alors :

$$(\alpha \succcurlyeq \beta) \iff (\text{il existe } 1 \leqslant i_1 < \dots < i_p \leqslant n \text{ tels que } a_{i_1} \geqslant_A b_1, \dots, a_{i_p} \geqslant_A b_p)$$
(condition vérifiée par convention si $p = 0$).

Démonstration : par récurrence sur n. On note $\alpha \succcurlyeq \beta$ le deuxième membre de l'équivalence annoncée. Pour $n = 0$, les énoncés $\alpha \succcurlyeq \beta$ et $\alpha \succcurlyeq \beta$ sont équivalents d'après la convention adoptée lorsque $\beta = \varepsilon$. Pour $n > 0$, si $\alpha \succcurlyeq \beta$ alors on a l'un des cinq cas suivants où a, b désignent des lettres et α', β' des mots :

(1) $\alpha = \beta$,
(2) $\alpha = a$ et $\beta = b$ avec $a >_A b$,
(3) $\beta = \varepsilon$,
(4) $\alpha = a\alpha'$, $\beta = b\beta'$ avec $a \geqslant_A b$ et $\alpha' \succcurlyeq \beta'$,
(5) $\alpha = a\alpha'$ avec $\alpha' \succcurlyeq \beta$.

Dans chaque cas on obtient $\alpha \succcurlyeq \beta$, soit directement, soit en appliquant l'hypothèse de récurrence à α'. Supposons à présent $\alpha \succcurlyeq \beta$ et soit $(i_1, \dots, i_p)$ une suite strictement croissante d'indices tels que $a_{i_j} \geqslant_A b_j$. Si $\alpha \neq \beta$ et $\beta \neq \varepsilon$, il faut montrer que l'un des cas (2), (4) ou (5) a lieu. Lorsque $n = 1$ on a $p = 1$ puisque $\beta \neq \varepsilon$, donc $a_1 \geqslant_A b_1$ et $a_1 \neq b_1$ puisque $\alpha \neq \beta$, d'où (1). Lorsque $n > 1$ et $i_1 = 1$, on a $\alpha = a_1\alpha'$ et $\beta = b_1\beta'$ avec $a_1 \geqslant_A b_1$ et $\alpha' \succcurlyeq \beta'$, donc $\alpha' \succcurlyeq \beta'$ par hypothèse de récurrence, d'où (4). Enfin, lorsque $n > 1$ et $i_1 > 1$ on a $\alpha = a_1\alpha'$ et $\alpha' \succcurlyeq \beta$, donc $\alpha' \succcurlyeq \beta$ par hypothèse de récurrence, d'où (5). ■

$(A^*, \succcurlyeq)$ **est un bel ordre** $\implies$ $(A, >_A)$ **est un bel ordre :** car toute suite de lettres bonne pour $\succcurlyeq_A$ est bonne pour $>_A$. ■

$(A, >_A)$ **est un bel ordre** $\implies$ $(A^*, \succcurlyeq)$ **est un bel ordre :** s'il existe une suite infinie (α_n) de mots mauvaise pour $\succcurlyeq$ alors, comme au **4.5**, on peut se ramener au cas où α_n est un mot de longueur minimale parmi ceux tels que $(\alpha_0, \dots, \alpha_n)$ est le début d'une suite mauvaise. Aucun des mots α_n n'est vide puisque $\alpha_n \not\preccurlyeq \alpha_{n+1}$, on note a_n la première lettre de α_n et $\alpha_n = a_n\beta_n$. En procédant de même qu'à la question 1, on peut extraire une suite (a_{n_k}) telle que l'une des trois relations a lieu :

1. $\forall\ k < \ell$, $a_{n_k} >_A a_{n_\ell}$,
2. $\forall\ k < \ell$, $a_{n_k} \leqslant_A a_{n_\ell}$,
3. $\forall\ k < \ell$, a_{n_k} et a_{n_ℓ} sont incomparables.

Les cas 1. et 3. sont exclus car $(A, >_A)$ est un bel ordre. Dans le cas 2, si la suite (β_{n_k}) est bonne alors il existe $k < \ell$ tels que $\beta_{n_k} \preccurlyeq \beta_{n_\ell}$, donc $n_k < n_\ell$ et $\alpha_{n_k} \preccurlyeq \alpha_{n_\ell}$, ce qui est absurde puisque (α_n) est mauvaise. Reste le cas où (β_{n_k}) est mauvaise : alors la suite $(\alpha_0, \dots, \alpha_{n_0-1}, \beta_{n_0}, \beta_{n_1}, \dots)$ est elle aussi mauvaise (car $\alpha_i \not\preccurlyeq \alpha_j \implies \alpha_i \not\preccurlyeq \beta_j$), en contradiction avec le choix de α_{n_0}. ■

fin du corrigé