Feuille d'exercices – corrigé n°1

Fonctions récursives primitives

Les exercices de cette feuille sont en partie une reprise du cours. N'utiliser pour un exercice donné que les résultats des exercices qui précèdent.

Définitions. Posons $\mathscr{F} = \bigcup_{p\in\mathbb{N}^*} \mathbb{N}^{\mathbb{N}^p}$. L'ensemble des *fonctions récursives primitives* est le plus petit sous-ensemble de $\mathscr{F}$

i. qui contient les fonction *nulle* $\theta : \mathbb{N} \to \mathbb{N}$, la fonction *successeur* $\succ : \mathbb{N} \to \mathbb{N}$, les *projections* $p_k^i : \mathbb{N}^k \longrightarrow \mathbb{N}$ ($1 \leq i \leq k$) définies par $p_k^i(x_1,\dots,x_k) = x_i$.

ii. qui est clos par le *schéma de composition* :
si $h : \mathbb{N}^p \longrightarrow \mathbb{N}$, et $g_1,\dots,g_p : \mathbb{N}^n \longrightarrow \mathbb{N}$ sont récursives primitives, alors $f : \mathbb{N}^n \longrightarrow \mathbb{N}$ définie par $f(x_1,\dots,x_n) = h(g_1(x_1,\dots,x_n),\dots,g_p(x_1,\dots,x_n))$ est récursive primitive.

iii. et qui est clos par le *schéma de récurrence primitive* :
si $g : \mathbb{N}^p \longrightarrow \mathbb{N}$ et $h : \mathbb{N}^{p+2} \longrightarrow \mathbb{N}$ sont récursives primitives, alors $f : \mathbb{N}^{p+1} \longrightarrow \mathbb{N}$ est récursive primitive, f définie par :

$$\begin{aligned} f(a_1,\dots,a_p,0) &= g(a_1,\dots,a_p) \\ f(a_1,\dots,a_p,x+1) &= h(a_1,\dots,a_p,x,f(a_1,\dots,a_p,x)). \end{aligned}$$

Un prédicat P sur N^p, resp. un sous-ensemble E de N^p, est un *prédicat récursif primitif*, resp. *un sous-ensemble récursif primitif*, quand sa fonction caractéristique est récursive primitive.

Remarque : On peut prendre $p = 0$ dans le schéma de récurrence récursive primitive :
si $b \in \mathbb{N}$, si $h : \mathbb{N}^2 \longrightarrow \mathbb{N}$ est récursive primitive, alors $f : \mathbb{N} \longrightarrow \mathbb{N}$ est récursive primitive, f définie par :

$$\begin{aligned} f(0) &= b \\ f(x+1) &= h(x,f(x)). \end{aligned}$$

En effet il suffit d'ajouter un argument inutile : on définit une fonction auxiliaire $f' : \mathbb{N}^2 \to \mathbb{N}$ par récurrence primitive à partir de la fonction g constante égale à b, et de la fonction $h'(a,x,f(x)) = h(p_3^2(a,x,f(x)), p_3^3(a,x,f(x)))$. On a $f(x) = f'(x,x)$.

Exercice 1. Montrer que l'ensemble des fonctions récursives primitives est un ensemble dénombrable.

Exercice 2 (exemples, cas particuliers du schéma de récurrence récursive primitive).

1. Montrer que les fonctions constantes de $\mathbb{N}^k \to \mathbb{N}$, $k \in \mathbb{N}$, sont récursives primitives.
2. Montrer que les trois fonctions de $\mathbb{N}$ dans $\mathbb{N}$, $f : x \mapsto x+2$, $g : x \mapsto 2x$ et $h : x \mapsto 2x+1$ sont récursives primitives.
3. Montrer que l'addition, la multiplication, et l'exponentielle sont des fonctions récursives primitives.
4. Montrer que la fonction sg de $\mathbb{N}$ dans $\mathbb{N}$, qui à 0 associe 0 et qui à tous les autres entiers associe 1 ainsi que la fonction $\overline{sg}$ de $\mathbb{N}$ dans $\mathbb{N}$, qui à 0 associe 1 et qui à tous les autres entiers associe 0 sont récursives primitives (se servir de la récurrence primitive).
5. Montrer que l'ensemble des fonctions récursives primitives est clos sous le schéma de *définition par itération*, qui a une fonction g de $\mathbb{N}^p$ dans $\mathbb{N}$ et à une fonction h de $\mathbb{N}^{p+1}$ dans $\mathbb{N}$ associe la fonction f de $\mathbb{N}^{p+1}$ dans $\mathbb{N}$ définie par :

$$\begin{aligned} f(a_1,\dots,a_p,0) &= g(a_1,\dots,a_p) \\ f(a_1,\dots,a_p,x+1) &= h(a_1,\dots,a_p,f(a_1,\dots,a_p,x))). \end{aligned}$$

   Montrer ensuite que les fonctions introduites jusqu'à présent dans cet exercice se définissent à partir des fonctions de base et du schéma d'itération (on verra des exemples de fonctions qui utilisent naturellement pour leur définition un schéma récursif primitif qui n'est pas un schéma d'itération : le prédécesseur de l'exercice 4, la factorielle, ...).
6. Montrer que l'ensemble des fonctions récursives primitives est clos *par définition par cas* sur un prédicat récursif primitif : si g et h sont des fonctions récursives primitives de $\mathbb{N}^p$ dans $\mathbb{N}$, et P un prédicat récursif primitif sur N^p, alors la fonction f de $\mathbb{N}^p$ dans $\mathbb{N}$ définie ci-dessous est récursive primitive :

$$\text{si } P(a_1,\dots,a_p) \text{ alors } f(a_1,\dots,a_p) = g(a_1,\dots,a_p) \text{ sinon } f(a_1,\dots,a_p) = h(a_1,\dots,a_p).$$

Exercice 3 (somme et produit bornés). Montrer que si $f : \mathbb{N}^{p+1} \longrightarrow \mathbb{N}$ est récursive primitive, les fonctions g somme bornée de f, et h produit borné de f, toutes deux de $\mathbb{N}^{p+1} \longrightarrow \mathbb{N}$ définies par

$$g(a_1,\dots,a_p,x) = \sum_{i=0}^{x} f(a_1,\dots,a_p,i) \quad h(a_1,\dots,a_p,x) = \prod_{i=0}^{x} f(a_1,\dots,a_p,i)$$

sont récursives primitives.

Exercice 4 (prédécesseur, comparaison).

1. Montrer que la fonction $pred : \mathbb{N} \longrightarrow \mathbb{N}$ qui vaut 0 en 0 et $n-1$ en $n > 0$ est récursive primitive.
2. Montrer que la fonction $\dot{-} : \mathbb{N}^2 \longrightarrow \mathbb{N}$ définie par « $x \dot{-} y = x - y$ si $x \geq y$, $x \dot{-} y = 0$ sinon », ainsi que la fonction : $x, y \mapsto |x - y|$ sont récursives primitives.
3. Montrer que les prédicats de comparaison $\leq, \geq, <, >, =, \neq$ sont récursifs primitifs.

Exercice 5 (prédicats récursifs primitifs, opérations booléennes).

1. Montrer que l'ensemble des prédicats récursifs primitifs d'arité quelconque est clos sous les opérations booléennes (conjonction, disjonction, négation). Par exemple on montrera que si les prédicats $P[x_1, \dots, x_p, y_1, \dots, y_n]$ et $Q[x'_1, \dots, x'_q, y_1, \dots, y_n]$ sont récursifs primitifs, alors le prédicat
$$P[x_1, \dots, x_p, y_1, \dots, y_n] \wedge Q[x'_1, \dots, x'_q, y_1, \dots, y_n]$$
est récursif primitif.
2. En déduire que l'ensemble des ensembles récursifs primitifs est clos par réunion, intersection et passage au complémentaire.

Exercice 6. Montrer que les sous-ensembles finis et cofinis des $\mathbb{N}^p$, $p \in \mathbb{N}$, sont récursifs primitifs.

Exercice 7 (minimisation bornée). On propose dans cet exercice une démonstration de la clôture de l'ensemble des fonctions récursives primitives sous le schéma de *minimisation bornée*, qui à un prédicat récursif primitif B sur $\mathbb{N}^{p+1}$ associe la fonction f de $\mathbb{N}^{p+1}$ dans $\mathbb{N}$ définie par :

$$\begin{array}{lcll} f(a_1, \dots, a_p, x) & = & \text{le plus petit entier } t \leq x \text{ tel que } B(a_1, \dots, a_p, t) & \text{s'il existe un tel entier,} \\ f(a_1, \dots, a_p, x) & = & 0 & \text{s'il n'existe pas de tel entier.} \end{array}$$

On note $f(a_1, \dots, a_p, x) = \mu t \leq x.B(a_1, \dots, a_p, t)$.

1. Soit un prédicat récursif primitif B sur $\mathbb{N}^{p+1}$, montrer que la fonction $b : \mathbb{N}^{p+1} \longrightarrow \mathbb{N}$ est récursive primitive, où b est définie par :
$$\begin{array}{lcll} b(a_1, \dots, a_p, x) & = & 0 & \text{s'il existe un entier } t \leq x \text{ tel que } B(a_1, \dots, a_p, t), \\ b(a_1, \dots, a_p, x) & = & 1 & \text{s'il n'existe pas de tel entier} \end{array}$$
(se servir de l'exercice 3).
2. En déduire que l'ensemble des fonctions récursives primitives est clos sous le schéma de minimisation borné.

Exercice 8 (quantifications bornées). Montrer que l'ensemble des prédicats récursifs primitifs est clos par quantification existentielle et universelle bornée (on peut utiliser la première question de l'exercice 7 ou directement l'exercice 3).

Exercice 9 (division euclidienne). Montrer que les fonctions $q : \mathbb{N}^2 \longrightarrow \mathbb{N}$ et $r : \mathbb{N}^2 \longrightarrow \mathbb{N}$, où $q(n, p)$ est le quotient et $r(n, p)$ le reste de la division de n par p sont des fonctions récursives primitives. En déduire que le prédicat binaire $\mid$ ($a \mid b$ signifie a est un diviseur de b) est récursif primitif.

Exercice 10 (nombres premiers). Soit $\mathrm{p} : \mathbb{N} \longrightarrow \mathbb{N}$ la fonction telle que $\mathrm{p}(n)$ soit le $n+1$-ième nombre premier.

1. Montrer que le prédicat «être premier» est récursif primitif.
2. Montrer que $\mathrm{p}(n+1) \leq \mathrm{p}(n)! + 1$ et que la fonction factorielle est récursive primitive.
3. Montrer que la fonction p est récursive primitive.

Exercice 11 (codage des couples et k-uplets). Soit α la bijection de Cantor $\mathbb{N} \times \mathbb{N}$ dans $\mathbb{N}$, définie par :

$$\alpha(n, p) = (\sum_{i=0}^{n+p} i) + p = \frac{(n+p+1)(n+p)}{2} + p$$

(faire un dessin).

1. Vérifier que α est bien bijective et récursive primitive. Vérifier que α est croissante sur chacune de ses deux composantes.
2. Définir de façon récursive primitive les deux «projections» associées π_2^1 et π_2^2 vérifiant :
$$\alpha_2(\pi_2^1(c), \pi_2^2(c)) = c \quad \pi_2^1(\alpha(n, p)) = n \quad \pi_2^2(\alpha(n, p)) = p\,.$$

3. On définit par récurrence sur $k \leq 1$ les fonctions $\alpha_k : \mathbb{N}^p \to \mathbb{N}$ par :

$$\alpha_1(n) = n \quad \alpha_{k+1}(n_1, \ldots, n_{k+1}) = \alpha(n_1, \alpha_k(n_2, \ldots, n_{k+1})) \quad \alpha_2 = \alpha\,.$$

Montrer que, pour tout $k \geq 1$, α_k est une bijection récursive primitive de $\mathbb{N}^k \to \mathbb{N}$ et définir de façon récursive primitive les « projections » $\pi^i_k : \mathbb{N} \to \mathbb{N}$ associées. Vérifier que α_k est croissante sur chacune de ses composantes. On écrira aussi $\langle x_1, \ldots, x_k \rangle$ pour $\alpha_k(x_1, \ldots, x_k)$.

Exercice 12 (définitions par récurrences mutuelles). Utiliser la fonction α_k pour montrer que si les fonctions $g_1, \ldots, g_k : \mathbb{N}^n \to \mathbb{N}$, et $h_1, \ldots, h_k : \mathbb{N}^{n+k+1} \to \mathbb{N}$ sont récursives primitives, alors, les fonctions $f_1, \ldots, f_k$ définies ci-dessous sont récursives primitives (on écrit $\vec{a}$ pour $a_1, \ldots, a_n$).

$$\begin{array}{lcl} f_1(\vec{a}, 0) = g_1(\vec{a}) & & f_k(\vec{a}, 0) = g_k(\vec{a}) \\ f_1(\vec{a}, x+1) = h_1(\vec{a}, x, f_1(\vec{a}, x), \ldots, f_k(\vec{a}, x)) & \cdots & f_k(\vec{a}, x+1) = h_k(\vec{a}, x, f_1(\vec{a}, x), \ldots, f_k(\vec{a}, x)) \end{array}$$

Exercice 13 (un codage bijectif des suites finies). On obtient la fonction :: (notation infixe) en translatant α_2 de 1 :

$$\langle x, y \rangle = 1 + \alpha_2(x, y)$$

On obtient ainsi une fonction récursive primitive bijective de $\mathbb{N}^2 \to \mathbb{N}^*$. On appelle hd et tl les fonctions vérifiant :

$$\begin{array}{ll} \mathrm{hd}(0) = 0 & \mathrm{tl}(0) = 0 \\ \mathrm{hd}(\langle x, y \rangle) = x & \mathrm{tl}(\langle x, y \rangle) = y \end{array}$$

On définit une fonction $liste$ de l'ensemble $\mathscr{S}$ des suites finies d'entiers dans $\mathbb{N}$ de la façon suivante (on note $[a_0; \ldots; a_n] =_d liste((a_0, \ldots, a_n)))$:

$$\begin{array}{l} [\,] = 0 \\ [a_0; \ldots; a_n] = \langle a_0, [a_1; \ldots; a_n] \rangle \end{array}$$

Montrer que la fonction $liste$ est bijective, et que les fonctions hd et tl sont récursives primitives.

Solution - On montre par récurrence sur c qu'il existe une unique suite de code c (i.e. que la fonction $liste$ est bijective).
$c = 0$: La suite vide a pour code 0, et c'est la seule car $\langle x, y \rangle > 0$.
Supposons le résultat pour tout $d < c+1$. On sait qu'il existe un unique couple d'entiers (a, b) tel que $c+1 = \langle a, b \rangle$. Or $b \leq \alpha_2(a, b) < \langle a, b \rangle$, donc par hypothèse de récurrence, il existe une unique suite finie d'image b, soit $(u_1, \ldots, u_n)$, avec $n \geq 0$ (() si $n = 0$) . On en déduit que $c+1 = [a; u_1; \ldots; u_n]$. Supposons maintenant que $c+1$ soit l'image d'une suite finie, celle-ci est nécessairement non vide car $c+1 \neq 0$, soit $(v_1, \ldots, v_p)$. Alors, $\langle , \rangle$ étant une bijection, $v_0 = a$ et $[v_1; \ldots; v_p] = b$, donc $(v_1, \ldots, v_p) = (u_1, \ldots, u_n)$ par hypothèse de récurrence.

La fonction :: est récursive primitive (car α_2 l'est). Par minimisation bornée et clôture des prédicats récursifs primitifs par quantification bornée $\mathrm{hd}(c) = \mu x < c.\left[\exists y < c\left(\alpha_2(x, y) = c\right)\right]$ et $\mathrm{tl}(c) = \mu y < c.\left[\exists x < c\left(\alpha_2(x, y) = c\right)\right]$ sont récursives primitives.

Exercice 14 (récurrence sur la suite des valeurs). On va utiliser les entiers pour coder des ensembles définis par induction : termes, formules ... Pour cela nous aurons besoin d'une définition de fonction par récurrence qui fait appel, pour définir la fonction en n à un ou plusieurs entiers strictement plus petits que n, et non seulement au prédécesseur comme le schéma de récurrence primitive. C'est le schéma de définition par *récurrence primitive sur la suite des valeurs*.

1. Démontrer que l'ensemble des fonctions récursives primitives est clos par le schéma de récurrence sur la suite des valeurs suivant : si $g : \mathbb{N}^p \longrightarrow \mathbb{N}$ et $h : \mathbb{N}^{p+2} \longrightarrow \mathbb{N}$ sont récursives primitives, alors $f : \mathbb{N}^{p+1} \longrightarrow \mathbb{N}$ est récursive primitive, f définie par :

$$\begin{array}{l} f(a_1, \ldots, a_p, 0) = g(a_1, \ldots, a_p) \\ f(a_1, \ldots, a_p, x+1) = h\left(a_1, \ldots, a_p, x, \left[f(a_1, \ldots, a_p, x); \ldots; f(a_1, \ldots, a_p, 0)\right]\right). \end{array}$$

Solution - On suppose g, h et f comme dans l'énoncé. On définit par récurrence primitive la fonction $F(a_1, \ldots, a_p, x) = \left[f(a_1, \ldots, a_p, x); \ldots; f(a_1, \ldots, a_p, 0)\right]$

$$\begin{array}{l} F(a_1, \ldots, a_p, 0) = \langle g(a_1, \ldots, a_p), [\,] \rangle \\ F(a_1, \ldots, a_p, x+1) = \langle h\left(a_1, \ldots, a_p, x, F(a_1, \ldots, a_p, x)\right), F(a_1, \ldots, a_p, x) \rangle. \end{array}$$

qui est donc récursive primitive, et $f(a_1, \ldots, a_p, x) = \mathrm{hd}(F(a_1, \ldots, a_p, x))$ est donc récursive primitive.

2. Montrer que la fonction nthl qui à l et i associe la suite codée par l à partie du $i+1$-ième élément (0 sinon), et la fonction nth qui à l et i associe le $i+1$-ème élément de la suite codée par l (0 sinon), sont récursives primitives.

Solution - On peut définir par récurrence primitive la fonction nthl :

$$\begin{array}{lcl} \mathrm{nthl}(l, 0) = l & & \mathrm{nth}(l, 0) = 0 \\ \mathrm{nthl}(l, i+1) = \mathrm{tl}(\mathrm{nthl}(l, i)) & \text{puis} & \mathrm{nth}(l, i+1) = \mathrm{hd}(\mathrm{nthl}(l, i)) \end{array}$$

3. Montrer que si $g : \mathbb{N}^p \longrightarrow \mathbb{N}$, $h : \mathbb{N}^{p+k+1} \longrightarrow \mathbb{N}$ sont récursives primitives, et si $p_1, \ldots, p_k : \mathbb{N} \longrightarrow \mathbb{N}$ sont des fonctions récursives primitives vérifiant chacune :

$$\forall x \in \mathbb{N}\ p_i(x) \leq x$$

alors $f : \mathbb{N}^{p+1} \longrightarrow \mathbb{N}$, définie par :

$$\begin{array}{l} f(a_1, \ldots, a_p, 0) = g(a_1, \ldots, a_p) \\ f(a_1, \ldots, a_p, x+1) = h\big(a_1, \ldots, a_p, x, f(a_1, \ldots, a_p, p_1(x)), \ldots, f(a_1, \ldots, a_p, p_k(x))\big) \end{array}$$

est récursive primitive. Ce schéma est indépendant du codage des suites. Mais il peut-être vu comme un cas particulier de la récurrence sur la suite des valeurs : la fonction au rang x peut dépendre seulement d'un nombre fixe des valeurs de la fonction en les y avant x, et non de toutes ces valeurs.

Solution - voir fonctions élémentaires

Exercice 15 (récurrence sur les listes).

1. Montrer que l'ensemble des fonctions récursives primitives est clos par le schéma de récurrence sur les listes (préciser pourquoi f est bien définie) : si $g : \mathbb{N}^p \longrightarrow \mathbb{N}$ et $h : \mathbb{N}^{p+3} \longrightarrow \mathbb{N}$ sont récursives primitives, alors $f : \mathbb{N}^{p+1} \longrightarrow \mathbb{N}$ est récursive primitive, f définie par

$$\begin{aligned} &f(a_1,\ldots,a_p,[\,]) = g(a_1,\ldots,a_p)\\ &f(a_1,\ldots,a_p,x :: l) = h\big(a_1,\ldots,a_p,x,l,f(a_1,\ldots,a_p,l)\big)\,. \end{aligned}$$

et l'utiliser pour montrer que la fonction mem fonction caractéristique de l'appartenance d'un entier n à une suite codée par l, la fonction @ vérifiant que $l\,@\,l'$ est le code de la concaténation des suites codées par l et l', la fonction $length$ qui à un entier l associe la longueur de la suite codée par l, sont récursives primitives.

Solution - Supposons maintenant g, h et f comme dans l'énoncé. la fonction f est bien définie car tout entier est soit 0 soit s'écrit $\langle x,l\rangle$ d'une unique façon. On peut définir f ainsi ($\vec{a} = a_1,\ldots,a_p$) :

$$\begin{aligned} &f(\vec{a},0) = g(\vec{a})\\ &f(\vec{a},n+1) = h\big(\vec{a},\mathrm{hd}(n+1),\mathrm{tl}(n+1),f(\vec{a},\mathrm{tl}(n+1))\big) \end{aligned}$$

ou encore, comme $n \dot{-} \mathrm{tl}(n+1) < n+1$ et $\mathrm{tl}(n+1) \leq n$:

$$\begin{aligned} &f(\vec{a},0) = g(\vec{a})\\ &f(\vec{a},n+1) = h\big(\vec{a},\mathrm{hd}(n+1),\mathrm{tl}(n+1),\mathrm{nth}([f(\vec{a},n);\ldots;f(\vec{a},0)],n \dot{-} \mathrm{tl}(n+1))\big)\,. \end{aligned}$$

La fonction f est donc récursive primitive par récurrence sur la suite des valeurs définie en 14, composition et car hd, tl, $\dot{-}$ et nth sont récursives primitives. On en déduit que les fonctions mem et @ sont récursives primitives en les définissant par récurrence sur les listes :

$$\left\{\begin{array}{lcll} mem(a,[\,]) & = & 0 & \\ mem(a,b::l) & = & mem(a::l) & \text{si } a \neq b\\ & = & 0 & \text{sinon} \end{array}\right. \qquad \left\{\begin{array}{lcl} []@l' & = & l'\\ (a::l)@l' & = & a::(l@l') \end{array}\right.$$

2. Montrer que si $f : \mathbb{N}^{p+1} \to \mathbb{N}$ est récursive primitive, alors la fonction map_f qui à l codant $(u_i)_{i\leq n}$ associe $map_f(l)$ codant $\big(f(a_1,\ldots,a_p,u_i)\big)_{i\leq n}$ est récursive primitive.
Montrer que la fonction $concat$ qui à un entier l codant une suite de suites $\big((u_i)_{i\leq n_i}\big)_{i\leq p}$ associe l'entier $concat(l)$ codant la suite des entiers de chaque suite $(u_i)_{i\leq n_i}$ dans le même ordre est récursive primitive.
Montrer que la fonction $subst$ qui à trois entiers l, k, v, associe le code la suite obtenue en remplaçant dans la suite codée par l toutes les occurrences de v par les entiers de la suite codée par k est récursive primitive (on peut se servir des deux fonctions précédentes).
Solution - A FAIRE

On peut utiliser la décomposition en nombre premiers pour un autre codage (non bijectif) des suites (voir feuille sur les fonctions élémentaires).

Exercice 16 (récurrence avec substitution de paramètre). On appelle *schéma de récurrence avec substitution de paramètre* le schéma de récurrence suivant (énoncé ici avec un seul paramètre) qui étant données g, $\gamma : \mathbb{N} \longrightarrow \mathbb{N}$ et $h : \mathbb{N}^3 \longrightarrow \mathbb{N}$ définit $f : \mathbb{N}^2 \longrightarrow \mathbb{N}$.

$$\begin{aligned} &f(a,0) = g(a)\\ &f(a,x+1) = h(a,x,f(\gamma(a),x))\,. \end{aligned}$$

Intuitivement, ce schéma conserve le fait d'être calculable (le calcul termine puisque la variable de récurrence décroit). On veut montrer que l'ensemble des fonctions récursives primitives est clos sous ce schéma. On suppose dans la suite g, γ et h récursives primitives, et f définie comme ci-dessus. On note $\gamma^p(x) = \underbrace{\gamma \circ \cdots \circ \gamma}_{p}(x)$ ($\gamma^0 = \lambda x.x$).

1. Montrer que la fonction F définie ci-dessous est récursive primitive

$$\begin{aligned} &F(p,a,0) = g\big(\gamma^p(a)\big)\\ &F(p,a,x+1) = h\big(\gamma^{p\dot{-}(x+1)}(a),x,F(p,a,x)\big)\,. \end{aligned}$$

2. Montrer que :

$$\forall x,a,p \in \mathbb{N}\ \big(x \leq p \to F(p,a,x) = f(\gamma^{p\dot{-}x}(a),x)\big)$$

et en déduire que f est récursive primitive.

3. Application : montrer que la fonction $inc : \mathbb{N}^2 \to \mathbb{N}$ qui à i et $l = [a_0;\ldots;a_i;\ldots;a_n]$ associe $inc(i,l) = [a_0;\ldots;a_i+1;\ldots;a_n]$ quand $i \leq n$, $inc(i,l) = l$ sinon, est récursive primitive.

Exercice 17 (récurrence double sans imbrication). On sait que la récurrence double ne conserve pas en général le fait d'être récursif primitif (la fonction d'Ackermann est définie par récurrence double). On peut montrer que s'il n'y a pas *imbrication* des appels récursifs dans la récurrence double, alors celle-ci reste "primitive récursive".

On suppose que a et b sont des entiers, que h est une fonction récursive primitive à 4 arguments, que h_2 est une fonction récursive primitive à 2 arguments.

Le schémas qui suivent ne comportent pas de paramètres, ce qui simplifie les notations, les démonstrations étant essentiellement les mêmes en présence de paramètres.

1. Montrer que la fonction f définie par :

$$\begin{array}{rcl} f(0,y) & = & a\,, \\ f(x+1,0) & = & b\,, \\ f(x+1,y+1) & = & h(x,y,f(x,y),f(x+1,y))\,. \end{array}$$

est récursive primitive (on peut utiliser le codage des couples et la "récurrence sur la suite des valeurs").

Solution - On note α une bijection de $\mathbb{N}^2$ dans $\mathbb{N}$ vérifiant pour tous x et y que $\alpha(x,y)<\alpha(x+1,y)$ et $\alpha(x,y)<\alpha(x,y+1)$ (telle que celle donnée en cours), et π_1, π_2 les projections associées.
On suppose que a et b sont des entiers, que h est une fonction récursive primitive à 4 arguments. La fonction f est définie par :

$$\begin{array}{rcl} f(0,y) & = & a\,, \\ f(x+1,0) & = & b\,, \\ f(x+1,y+1) & = & h(x,y,f(x,y),f(x+1,y))\,. \end{array}$$

On montre que ce schéma est récursif primitif en utilisant que l'appel récursif se fait sur des valeurs de f antérieures dans l'ordre de l'énumération par α^{-1} des éléments de $\mathbb{N}^2$ (récurrence sur la suite des valeurs, voir question 3 de l'exercice 14).
Posons $\tilde{f}(z)=f(\pi_1(z),\pi_2(z))$, donc $f=\tilde{f}\circ\alpha$. On a :

$$\begin{array}{rcll} \tilde{f}(0) & = & a & \\ \tilde{f}(z+1) & = & a & \text{si } \pi_2(z+1)=0 \\ & = & b & \text{si } \pi_1(z+1)=0 \\ & = & h(\pi_1(z+1)\dot{-}1,\pi_2(z+1)\dot{-}1,\tilde{f}(\gamma_1(z+1)),\tilde{f}(\gamma_2(z+1))) & \text{sinon, avec} \\ \gamma_1(c) & = & \alpha(\pi_1(c)\dot{-}1,\pi_2(c)\dot{-}1) & \\ \gamma_2(c) & = & \alpha(\pi_1(c),\pi_2(c)\dot{-}1) & \end{array}$$

La fonction $\tilde{f}$ est récursive primitive par récurrence sur la suite des valeurs car si $\pi_1(c)\geq 1$ et $\pi_2(c\geq 1$, $\gamma_1(c)<c$, et $\gamma_2(c)<c$.
La fonction $f=\tilde{f}\circ\alpha$ est donc récursive primitive.

2. (généralisation, difficile) Montrer que la fonction f définie par :

$$\begin{array}{rcl} f(0,y) & = & a\,, \\ f(x+1,0) & = & b\,, \\ f(x+1,y+1) & = & h(x,y,f(x,h_2(x,y)),f(x+1,y))\,. \end{array}$$

est récursive primitive. On peut procéder comme à la question précédente, mais en cherchant pour h_2 donné une nouvelle fonction α' de codage des couples (injective mais non nécessairement bijective) vérifiant :

$$\alpha'(x+1,y)<\alpha'(x+1,y+1) \text{ et } \alpha'(x,h_2(x,y))<\alpha'(x+1,y+1)\,.$$

Solution - (généralisation : indications). On suppose que a et b sont des entiers, que h_1 est une fonction récursive primitive à 4 arguments, que h_2 est une fonction récursive primitive à 2 arguments. Montrons que la fonction f définie par :

$$\begin{array}{rcl} f(0,y) & = & a\,, \\ f(x+1,0) & = & b\,, \\ f(x+1,y+1) & = & h_1(x,y,f(x,h_2(x,y)),f(x+1,y))\,. \end{array}$$

est récursive primitive.
On procède comme à la question précédente. On cherche une nouvelle fonction α' de codage des éléments de N^2 vérifiant cette fois :

$$\alpha'(x+1,y)<\alpha'(x+1,y+1) \text{ et } \alpha'(x,h_2(x,y))<\alpha'(x+1,y+1)$$

Soit h_1' définie par $h_1'(z,u,v)=h_1(\pi_1(z),\pi_2(z),u,v)$. Soit $\tilde{h}_2$ définie par $\tilde{h}_2(z)=h_2(\pi_1(z),\pi_2(z))$. Soit h_2' définie par $h_2'(z)=\sum_{i=0}^{z}(\tilde{h}_2(i)+1)$ (h_2' est une fonction strictement croissante récursive primitive qui majore h_2)).
Soit h_2'' la fonction récursive primitive définie en itérant x fois h_2' sur z :

$$\begin{array}{rcl} h_2''(0,z) & = & z\,, \\ h_2''(x+1,z) & = & h_2'(x,h_2'''(x,z))\,, \end{array} \qquad \text{(autrement dit } \; h_2''(x,z)=\underbrace{h_2'\circ\cdots\circ h_2'}_{x}(z)\,)$$

Soit α' définie par $\alpha'(x,y) = \alpha(x, h''_2(x, \alpha(x,y)))$. Soient $\pi'_1 = \pi_1$ et π'_2 définie par :

$$\pi'_2(z) = \mu y \leq z.\ h''_2(\pi_1(z), \alpha(\pi_1(z), y)) = \pi_2(z)\ .$$

Ces 3 fonctions sont récursives primitives. Comme h''_2 et α sont strictement croissantes sur leur deuxième entrée, α' est injective, et on a $\pi'_1(\alpha'(x,y)) = x$ et $\pi'_2(\alpha'(x,y)) = y$.
Bien entendu α' n'est pas nécessairement bijective mais,

$$\forall z (\pi'_2(z) \neq 0 \Rightarrow z = \alpha'(\pi'_1(z), \pi'_2(z))\ .$$

Il suffit ensuite démontrer les deux inégalités demandées pour α', puis on procède comme pour la première question.
Montrons maintenant les deux inégalités dont nous avons besoin :

$$\alpha'(x+1,y) < \alpha'(x+1,y+1) \ \text{ et } \ \alpha'(x, h_2(x,y)) < \alpha'(x+1,y+1)$$

Voyons la première. On sait que $\alpha(x+1,y) < \alpha(x+1,y+1)$, donc, $\lambda y.h''_2(x,y)$ étant croissante,

$$h''_2(x+1, \alpha(x+1,y)) < h''_2(x+1, \alpha(x+1,y+1))\ ,$$

donc $\lambda y.\alpha(x,y)$ étant croissante,

$$\alpha(x+1, h''_2(x+1, \alpha(x+1,y))) < \alpha(x+1, h''_2(x+1, \alpha(x+1,y+1)))\ ,$$

ce qui est l'inégalité cherchée.
Voyons la deuxième. On compare

$$\alpha'(x, h_2(x,y)) = \alpha(x, h''_2(x, h_2(x,y))) \quad \text{et} \quad \alpha'(x+1,y+1) = \alpha(x+1, h''_2(x+1, \alpha(x+1,y+1)))$$

Pour avoir le résultat cherché, vu les propriétés de croissance de α sur ces deux composantes, il suffit de montrer :

$$h''_2(x, h_2(x,y)) < h''_2(x+1, \alpha(x+1,y+1))$$

Comme h'_2 majore $\tilde{h}_2$, on a

$$h''_2(x, h_2(x,y)) < h''_2(x, h'_2(\alpha(x,y)) = \underbrace{h'_2 \circ \cdots \circ h'_2}_{x+1}(\alpha(x,y))$$

or

$$h''_2(x+1, \alpha(x+1,y+1)) = \underbrace{h'_2 \circ \cdots \circ h'_2}_{x+1}(\alpha(x+1,y+1))$$

Et on a le résultat cherché, par croissance de h'_2.
Soient γ_0, γ_1 et γ_2 définies par $\gamma_0(z) = \alpha'(\pi'_1(z) \dot{-} 1, \pi'_2(z) \dot{-} 1)$, $\gamma_1(z) = \alpha'(\pi'_1(z) \dot{-} 1, h_2(\gamma_0(z)))$, $\gamma_2(z) = \alpha'(\pi'_1(z), \pi'_2(z) \dot{-} 1)$.
Si $\pi'_2(z) \neq 0$, on déduit des inégalités précédentes que $\gamma_1(z) < z$ et $\gamma_2(z) < z$.
On définit ensuite $\tilde{f}(z) = f(\pi'_1(z), \pi'_2(z))$ par récurrence sur la suite des valeur comme à la question précédente ce qui montrer que $\tilde{f}$ est récursive primitive, et donc $f = \tilde{f} \circ \alpha'$ également.
Et de la même façon, on en déduit que $\tilde{f}$ puis $f = \tilde{f} \circ \alpha'$ sont récursives primitives.

Fonction d'Ackermann La fonction d'Ackermann est définie par récurrence double avec imbrication des appels récursifs :

$$\begin{array}{ll} \mathrm{Ack}(0,x) = x+2 & \mathrm{Ack}(n+2,0) = 1 \\ \mathrm{Ack}(1,0) = 0 & \mathrm{Ack}(n+1,x+1) = \mathrm{Ack}(n, \mathrm{Ack}(n+1,x)) \end{array}$$

La fonction $x \mapsto \mathrm{Ack}(n,x)$ est notée Ack_n. La fonction Ack_{n+1} est obtenue en itérant la fonction Ack_n :

$$\mathrm{Ack}_{n+1}(x+1) = \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x))\ .$$

Toutes les fonctions Ack_n sont récursives primitives, strictement croissantes, et cette croissance est de plus en plus rapide (voir exercice suivant).

$$\mathrm{Ack}_1(x) = 2x,\ \mathrm{Ack}_2(x) = 2^x,\ \mathrm{Ack}_3(x) = 2^{2^{\cdot^{\cdot^{\cdot^2}}}}\Big\}\, x,\ \ldots$$

On dit qu'une fonction $f : \mathbb{N} \longrightarrow \mathbb{N}$ *domine* une fonction $g : \mathbb{N}^p \longrightarrow \mathbb{N}$ si f est supérieure à g à partir d'un certain rang, i.e.

$$\exists K \in \mathbb{N}\ \forall \vec{x} \in \mathbb{N}^p\ g(\overline{x}) \leq f(\sup(\overline{x}, K))$$

On va montrer que pour toute fonction récursive primitive f, il existe n tel que Ack_n domine f, ce dont on déduit que *la fonction d'Ackermann n'est pas récursive primitive.*

Une hiérarchie des fonctions récursives primitives On définit une suite $\mathscr{C}_n$ d'ensembles de fonctions récursives primitives tous clos par composition, les fonctions de $\mathscr{C}_n$ étant les fonctions qui utilisent des suites d'au plus n schémas de récurrence primitive imbriqués. En voici une définition par induction :

(i) $\mathscr{C}_0$ est la clôture par composition de l'ensemble des fonctions de bases : constantes, projections, et successeur ;

(ii) $\mathscr{C}_{n+1}$ est la clôture par composition de la réunion de $\mathscr{C}_n$ et des fonctions obtenues par une seule occurrence du schéma de récurrence primitive à partir des fonctions de $\mathscr{C}_n$.

Il est clair que $\mathscr{C}_n \subset \mathscr{C}_{n+1}$ et que $\bigcup_{i=0}^{\infty} \mathscr{C}_n$ est l'ensemble de toutes les fonctions récursives primitives. On vérifie facilement que $\mathrm{Ack}_n \in \mathscr{C}_n$. On peut maintenant préciser l'énoncé du paragraphe précédent.

Proposition 1. Si $f \in \mathscr{C}_n$, alors Ack_{n+1} domine f, en particulier $\mathrm{Ack}_{n+1} \notin \mathscr{C}_n$.

La hiérarchie des $\mathscr{C}_n$ est donc stricte, puisque $\mathrm{Ack}_{n+1} \in \mathscr{C}_{n+1}$, mais $\mathrm{Ack}_{n+1} \notin \mathscr{C}_n$. Les démonstrations, en particulier celle de la proposition précédente, sont détaillées dans l'exercice qui suit.

Exercice 18 (fonction d'Ackermann).

1. Vérifier qu'il existe bien une et une seule fonction de $\mathbb{N}^2 \to \mathbb{N}$ vérifiant les équations de la fonction d'Ackermann. Calculez explicitement les premières valeurs de Ack, par exemple $\{\mathrm{Ack}(n,x) \mid 0 \le n \le 3,\ 0 \le x \le 3\}$, et donner un argument informel pour la calculabilité (au sens intuitif) de Ack, c'est-à-dire la raison pour laquelle la suite des appels récursifs termine (on peut utiliser l'ordre lexicographique sur les couples).

   Solution - La fonction est définie par récurrence ordinale sur ω^2. Informellement : par décroissance stricte dans ω^2 (ordre lexicographique sur $\mathbb{N}^2$) des valeurs appels récursifs $((n+1,x+1) > (n+1,x)$ et $(n+1,x+1) > (n,\mathrm{Ack}(n+1,x))$, la suite de ceux ci est finie et le calcul termine.

2. Montrer que
$$\forall n \in \mathbb{N}\ \forall x > 0\ \mathrm{Ack}_{n+1}(x) = \underbrace{\mathrm{Ack}_n \circ \cdots \circ \mathrm{Ack}_n}_{x}(\mathrm{Ack}_{n+1}(0))$$
   et vérifier les expressions des fonctions Ack_1, Ack_2 et Ack_3 données ci-dessus.

   Solution - La dernière clause de la définition par récurrence de la fonction d'Ackermann s'écrit :
$$\mathrm{Ack}_{n+1}(x+1) = \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x))\ . \qquad (*)$$
   Le résultat demandé se déduit par récurrence sur x.

3. Vérifiez que chacune des fonctions Ack_n est récursive primitive, et donnez en une définition dont vous montrerez qu'elle utilise exactement n instances du schéma de définition par itération (voir polycopié). On peut remarquer que les n schémas de récurrences sont imbriqués.

   Solution - Par récurrence sur n. On a $\mathrm{Ack}_0(x) = \mathrm{s} \circ \mathrm{s} \circ p_1^1$, et Ack_{n+1} se définit par itération à partir de Ack_n d'après $(*)$.

4. Montrer que si $(n,x) \neq (1,0)$, $\mathrm{Ack}_n(x) > x$, en particulier :
$$\forall n \ge 2 \forall x \in \mathbb{N}\ \mathrm{Ack}_n(x) > x\ ; \quad \forall n \in \mathbb{N} \forall x \in \mathbb{N}\ \mathrm{Ack}_n(x) \ge x\ .$$
   Solution - Le résultat est vrai pour $\mathrm{Ack}_0(x) = x+2$ et $\mathrm{Ack}_1(x) = 2x$. On Montre que $\forall x \in \mathbb{N}\ Ack_n(x) > x$ pour $n \ge 2$ par récurrence sur n puis sur x. Le résultat est vrai pour $Ack_2(x) = 2^x$. Supposons le pour $n \ge 2$. On a alors pour tout $x \in \mathbb{N}$, $\mathrm{Ack}_{n+1}(x+1) = \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x)) > \mathrm{Ack}_{n+1}(x)$. On en déduit par récurrence sur x que $\mathrm{Ack}_{n+1}(x) \ge x + \mathrm{Ack}_{n+1}(0)$. Or pour $n \ge 2$, $\mathrm{Ack}_n(0) = 1$ par définition.

5. En déduire que pour tout entier n, Ack_n est strictement croissante.

   Solution - Pour $n \ge 1$, $\mathrm{Ack}_{n+1}(x+1) = \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x)) > \mathrm{Ack}_{n+1}(x)$ d'après la question précédente, et le résultat est évident pour Ack_0 et Ack_1.

6. Déduire de la question **4** que, à partir de 2, Ack est croissante au sens large sur son premier argument, le second étant fixé :
$$\forall x \ge 2\ \forall n \in \mathbb{N}\ \mathrm{Ack}(n,x) \le \mathrm{Ack}(n+1,x)\ .$$
   Solution - D'après la question **4** $x+1 \le \mathrm{Ack}(n+1,x)$, pour $x \ge 1$, donc par croissance de Ack_n, $\mathrm{Ack}(n,x+1) \le \mathrm{Ack}(n,\mathrm{Ack}(n+1,x)) = \mathrm{Ack}(n+1,x+1)$.

On pose pour k entier, $\mathrm{Ack}_n^k = \underbrace{\mathrm{Ack}_n \circ \cdots \circ \mathrm{Ack}_n}_{k}$.

7. Montrer que : $\forall n,k \in \mathbb{N}\ \mathrm{Ack}_n^k \in \mathscr{C}_n$

   Solution - Récurrence immédiate sur n pour Ack_n, d'où le résultat pour Ack_n^k, $\mathscr{C}_n$ étant clos par composition.

8. Montrer que $\forall n,k,x \in \mathbb{N}\ \mathrm{Ack}_n^k(x) \le \mathrm{Ack}_{n+1}(x+k)$.

   Solution - En itérant k fois la clause récursive de la définition, $\mathrm{Ack}_{n+1}(x+k) = \mathrm{Ack}_n^k(\mathrm{Ack}_{n+1}(x))$, or $x \le \mathrm{Ack}_{n+1}(x)$ (question **4**), et Ack_n est croissante (question **5**), d'où le résultat.

9. Montrer que :

$$\text{si } f \in \mathscr{C}_n, \text{ alors } \exists k \in \mathbb{N} \operatorname{Ack}_n^k \text{ domine } f.$$

Solution - Par récurrence double d'abord sur n, et pour chaque f par induction sur la définition de $\mathscr{C}_n$.

Fonctions de base : chacune des fonctions de base est dominée par Ack_0 (dès le rang 0).

Composition : si $f \in \mathscr{C}_n$, $f = h \circ (g_1, \dots, g_p)$ avec $g_i : \mathbb{N}^n \to \mathbb{N}$ et $h, g_1, \dots, g_p \in \mathscr{C}_n$. Alors On a par hypothèse d'induction $p+1$ des entiers k_i tels que h est dominée par $\operatorname{Ack}_n^{k_0}$ et g_i par $\operatorname{Ack}_n^{k_i}$, c'est-à-dire que l'on a $p+1$ entiers $K_0, K_1, \dots, K_p$ tels que :

$$h(\overline{y}) \leq \operatorname{Ack}_n^{k_0}(\sup(\overline{y}, K_0)\ ; \dots g_i(\overline{x}) \leq \operatorname{Ack}_n^{k_i}(\sup(\overline{x}, K_i)) \dots$$

On pose $K = \sup(K_0, K_1, \dots, K_p)$, $k = \sup(1, k_1, \dots, k_p)$. Comme Ack_n est croissante, $g_i(\overline{x}) \leq \operatorname{Ack}_n^k(\sup(\overline{x}, K))$. Comme $y \leq \operatorname{Ack}_n(y)$ et $\operatorname{Ack}_n^{k-1}$ est croissante en tant que composée de fonctions croissantes (c'est l'identité si $k = 1$),

$$h(g_1(\sup(\overline{x}, K)), \dots, g_k(\sup(\overline{x}, K))) \leq \operatorname{Ack}_n^{k_0}(\sup(g_1(\sup(\overline{x}, K)), \dots, g_k(\sup(\overline{x}, K)), K))$$

et comme $\operatorname{Ack}_n^{k_0}$ est croissante,

$$f(\sup \overline{x}, K)) = h(g_1(\sup(\overline{x}, K)), \dots, g_k(\sup(\overline{x}, K))) \leq \operatorname{Ack}_n^{k_0}(\sup(\operatorname{Ack}_n^k(\sup(\overline{x}, K)), K_0))$$

or $K_0 \leq K \leq \operatorname{Ack}_n^k(K)$ donc $f(\sup \overline{x}, K)) \leq \operatorname{Ack}_n^{k+k_0} \sup(\overline{x}, K))$, d'où le résultat pour la composition.

Récurrence primitive : on suppose que $f \in \mathscr{C}_{n+1}$, $f = \operatorname{Rec}(g, h)$avec $g \in \mathscr{C}_n$ et $h \in \mathscr{C}_n$. Par hypothèse d'induction on a des entiers k_0 et k_1 tels que $\operatorname{Ack}_n^{k_0}$ domine g, $\operatorname{Ack}_n^{k_1}$ domine h. On a donc des entiers K_0 et K_1 tels que :

$$g(\sup(\overline{x}, K_0)) \leq \operatorname{Ack}_n^{k_0}((\sup(\overline{x}, K_0))\ ; \quad h(\sup(\overline{x}, y, z, K_1)) \leq \operatorname{Ack}_n^{k_1}((\sup(\overline{x}, y, z, K_1))\ ;$$

On pose $K = \sup(K_0, K_1)$. Par croissance de Ack_n il vient que

$$g(\sup(\overline{x}, K_0)) \leq \operatorname{Ack}_n^{k_0}((\sup(\overline{x}, K))\ ; \quad h(\sup(\overline{x}, y, z, K_1)) \leq \operatorname{Ack}_n^{k_1}((\sup(\overline{x}, y, z, K))\ ;$$

et en reprenant la définition par récurrence primitive on en déduit, ainsi que de la croissance de Ack_n, par récurrence sur y que :

$$f(\sup(;x, y, K) \leq A_n^{k_0 + yk_1}(\sup(\overline{x}, y, K)$$

donc (question **8**)

$$f(\sup(\overline{x}, y, K) \leq A_{n+1}(\sup(\overline{x}, y, K) + k_0 + yk_1)\ .$$

La fonction $h : (y, z) \mapsto z + k_1 y + k_0$ est dominée par $\operatorname{Ack}_1^k : x \mapsto 2^k x$ pour un k suffisamment grand, donc par $\operatorname{Ack}_{n+1}^k$ (question **6**). Donc f est dominée par $\operatorname{Ack}_{n+1}^{k+1}$.

10. Montrer que Ack_n^k est dominée par Ack_{n+1} (on pourra montrer que pour $y > 0$, $\operatorname{Ack}_{n+1}(y) \geq 2y$, puis que pour $x > 2k$, $\operatorname{Ack}_{n+1}(x - k) \geq x$).

Solution - $\operatorname{Ack}_1(y) = 2y$, donc d'après la question **6**, pour $y > 0$, $\operatorname{Ack}_{n+1}(y) \geq 2y$. On en déduit que pour $x > 2k$, $\operatorname{Ack}_{n+1}(x - k) \geq 2x - k \geq x$. Donc, par croissance de Ack_{n+1}, pour $x > 2k$ et d'après la question **8** :

$$\operatorname{Ack}_n^k(x) \leq \operatorname{Ack}_n^k(\operatorname{Ack}_{n+1}(x - k)) = \operatorname{Ack}_{n+1}(x)\ .$$

11. En déduire que la fonction d'Ackermann n'est pas récursive primitive.

On peut également montrer que la fonction diagonale $n \mapsto \operatorname{Ack}(n, n)$ domine toutes les fonctions récursives primitives.

Solution - D'après les deux précédentes questions, si $f \in \mathscr{C}_n$, alors Ack_{n+1} domine f.
Donc si la fonction d'Ackermann était primitive récursive, alors $s \circ \operatorname{Ack}$ serait dominée par une fonction Ack_N pour un certain $N \geq 1$. On aurait alors un K tel que pour tout entier x :

$$\operatorname{Ack}(n, x) + 1 \leq \operatorname{Ack}_N(\sup(n, x, K))\ .$$

En particulier pour $x > K$ et $x > N + 1$, $\operatorname{Ack}(N + 1, x) + 1 \leq \operatorname{Ack}(N, x)$, ce qui contredit le résultat de la question **6**.
On peut montrer aussi que la fonction diagonale $x \mapsto \operatorname{Ack}(x, x)$ domine toutes les fonctions récursives primitives, ce qui aboutirait à une contradiction si cette fonction était récursive primitive (même argument).
La relation « domine » est transitive (si pour tout $\overline{x}$, $g(\overline{x}) \leq f(\sup(\overline{x}, K))$ et pour tout y, $f(y) \leq h(\sup(y, K'))$, alors $g(\overline{x}) \leq h(\sup(\overline{x}, K, K')))$.
Il suffit donc de montrer que pour tout entier n, Ack_n est dominée par $x \mapsto \operatorname{Ack}(x, x)$: or pour $x \geq n$, $\operatorname{Ack}(n, x) \leq \operatorname{Ack}(x, x)$ d'après la question **6**, pour tout x, $\operatorname{Ack}_n(x) \leq \operatorname{Ack}(\sup(x, n), \sup(x, n))$.