
TD 1 – Fonctions récursives primitives

Exercice 1. Montrer que pour tout $n \in \mathbb{N}$ l'ensemble $\{n\}$ est récursif primitif. En déduire que tout sous-ensemble de \mathbb{N} qui est fini ou qui est le complémentaire d'un sous-ensemble fini de \mathbb{N} est récursif primitif.

Solution de l'exercice 1. On va montrer que les singletons sont récursifs primitifs car leur fonction caractéristique est récursive primitive. On sait que l'égalité f définie par $f(x, y) = 1$ si $x = y$ et 0 sinon est récursive primitive. Si on note c_n la fonction constante égale à n à une variable, on a $\chi_{\{n\}} = f(\pi_1^1, c_n)$ ce qui montre que $\{n\}$ est bien un ensemble récursif primitif.

L'ensemble des ensembles récursifs primitifs étant clos par opérations booléennes, un sous ensemble fini de \mathbb{N} est la réunion de singletons de \mathbb{N} qui sont récursifs primitifs et donc est récursif primitif. De même les ensembles cofinis de \mathbb{N} sont récursifs primitifs par passage au complémentaire.

Exercice 2. Montrer que l'ensemble des nombres pairs est récursif primitif.

Solution de l'exercice 2. La fonction caractéristique f des nombres pairs se définit de la manière suivante par récurrence : $f(0) = 1$ et $f(n+1) = 1 - f(n)$; elle est donc bien récursive primitive.

Exercice 3. Soit p un entier non nul. Montrer que la fonction $\text{sup}_p : \mathbb{N}^p \rightarrow \mathbb{N}$ qui à (x_1, \dots, x_p) associe le maximum de x_1, \dots, x_p est récursive primitive.

Solution de l'exercice 3. Soit p un entier non nul, on veut montrer que la fonction sup_p qui au p -uplet (x_1, \dots, x_p) associe le plus grand des x_i pour i appartenant à $\{1, \dots, p\}$. On fait une démonstration par récurrence sur p .

$$\text{sup}_1(x_1) = x_1, \quad \text{sup}_1 = \pi_1^1 \quad (\text{c'est la projection à un argument})$$

On suppose que les sup_i pour i appartenant à $\{1, \dots, p-1\}$, sont récursives primitives on regarde pour sup_{p+1} :

$$\text{sup}_{p+1}(x_1, \dots, x_{p+1}) = \text{sup}_2(\text{sup}_p(x_1, \dots, x_p), x_{p+1})$$

où $\text{sup}_2(x, y) = x$ si $x \geq y$ et y sinon. Ainsi, sup_{p+1} est récursive primitive.

Donc la fonction sup_p est récursive primitive pour tout $p \in \mathbb{N}^*$.

Exercice 4. Montrer que les fonctions *quotient* et *reste* sont récursives primitives (par définition, on pose $x/y = 0$ si $y = 0$). En déduire que $\{(x, y) \mid y \text{ divise } x\}$ est récursif primitif.

Solution de l'exercice 4. Notons q la fonction quotient. On a $q(x, y) = \mu k \leq x \cdot (k+1)y > x$, donc q est récursive primitive. La fonction reste est égale à $r(x, y) = x - yq(x, y)$ l'est également.

Exercice 5. Montrer que l'ensemble des nombre premiers est récursif primitif. En déduire que la fonction π qui à n fait correspondre le $(n+1)$ -ième nombre premier est récursive primitive.

Solution de l'exercice 5. Un entier n est premier ssi $n > 1$ et pour tout $a \in \{2, n-1\}$, a ne divise pas n (qui peut s'écrire : pour tout $a < n$, $a < 2$ ou a ne divise pas n). Ceci montre que l'ensemble des nombre premiers est récursif primitif.

La fonction f qui à n fait correspondre le $(n+1)$ -ième nombre premier est définie par récurrence : $f(0) = 2$ et $f(n+1) = \mu a \leq (f(n)! + 1) \cdot (a > f(n) \text{ et } a \text{ premier})$.

Exercice 6. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ une fonction récursive primitive. Montrer que la fonction g définie par $g(x) = \underbrace{f \circ f \circ \dots \circ f}_{x+1 \text{ fois}}(0)$ est récursive primitive.

Solution de l'exercice 6. On définit g par schéma de récurrence :
$$\begin{cases} g(0) = f(0) \\ g(x+1) = f(g(x)) \end{cases} .$$

Exercice 7. Montrer que la fonction $x \mapsto x^{x^{\dots^x}}$, où le nombre de x dans la tour d'exponentielles est $x+1$, est récursive primitive.

Solution de l'exercice 7. On définit d'abord par schéma de récurrence une fonction récursive primitive $F(x, n) = x^{x^{\cdot^{\cdot^{\cdot^x}}}}$, où le nombre de x dans la tour d'exponentielles est n :
$$\begin{cases} F(x, 0) = 1 \\ F(x, n+1) = x^{F(x, n)} \end{cases}$$
. La fonction cherchée est $F(x, x)$, qui est donc récursive primitive.

Exercice 8. Soit E l'ensemble des entiers naturels qui sont la somme de deux carrés. Montrer que E est récursif primitif.

Solution de l'exercice 8. $x \in E$ s'exprime par $\exists s, t \leq x (s^2 + t^2 = x)$.

Exercice 9. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ définie par $f(a, b) = \alpha_2(c, d)$ où c/d est l'écriture simplifiée de la fraction a/b si $b \neq 0$, et $c = d = 0$ sinon. Montrer que la fonction f est récursive primitive.

Solution de l'exercice 9. Il existe plusieurs solutions. On peut par exemple définir $f(a, b)$ comme le plus petit entier $k \leq \alpha_2(a, b)$ tel que $\beta_2^1(k) \cdot b = \beta_2^2(k) \cdot a$ et $k \neq 0$.

Exercice 10. Soit u la fonction qui à $n \in \mathbb{N}$ associe le $(n+1)$ -ième entier naturel (dans l'ordre croissant) qui est une somme de deux carrés. Montrer que u est récursive primitive.

Solution de l'exercice 10. On définit la fonction u par un schéma de récurrence :

$$\begin{cases} u(0) = 0 \\ u(n+1) = \mu k \leq (u(n)+1)^2 \cdot (k > u(n) \text{ et } \exists x, y \leq k (x^2 + y^2 = k)) \end{cases}$$

Exercice 11. Soit f la fonction qui à n associe le nombre d'entiers premiers inférieurs ou égaux à n . Montrer que f est récursive primitive.

Solution de l'exercice 11. Deux solutions :

1. On utilise le test de primalité vu dans un exercice précédent : $P(k) = 1$ si k est premier et 0 sinon. Ce test est récursif primitif. On a alors $f(n) = \sum_{k=0}^n P(k)$ qui est récursive primitive comme somme bornée de fonctions récursives primitives.

2. Par schéma de récurrence on pose :
$$\begin{cases} f(0) & = 0 \\ f(n+1) & = h(n, f(n)) \end{cases}$$
, avec $h(n, y) = \begin{cases} y+1 & \text{si } P(n+1) = 1 \\ y & \text{sinon} \end{cases}$.

On a donc f RP car définie par schéma de récurrence à partir de fonctions elles mêmes RP (h est définie par cas en fonction de P , donc RP).

Exercice 12. Montrer que les fonctions calculant le plus petit commun multiple et le plus grand diviseur commun de deux nombres sont récursives primitives.

Solution de l'exercice 12. Soit $m(x, y) = \mu k \leq xy (\text{div}(x, k) \text{ et } \text{div}(y, k) \text{ et } (\forall s \leq xy (\text{div}(x, s) \text{ et } \text{div}(y, s) \Rightarrow \text{div}(k, s))))$, où $\text{div}(u, v)$ est la fonction testant si u divise v . Alors m calcule le plus petit commun multiple de x et y , car elle cherche le plus petit entier k qui soit divisible par x et y et tel que si un autre entier s est divisible par x et y alors k divise s . L'expression symbolique donnée ci-dessus traduit cette phrase, en ajoutant des bornes pour le schéma μ et la quantification.

De même, le plus grand diviseur commun pourrait s'obtenir par une expression similaire, en traduisant ses propriétés, mais il peut aussi se calculer à partir du produit de x et y et de leur p.p.c.m..

Exercice 13. Soit n entier, on lui associe la suite $(a_i)_{i \in \mathbb{N}}$ de l'écriture décimale de \sqrt{n} : $\sqrt{n} = a_0, a_1 a_2 a_3 \dots$, où a_0 est un entier et chaque a_i pour $i \geq 1$ est un chiffre de 0 à 9 (si \sqrt{n} est un entier, a_i vaut 0 pour $i \geq 1$). Montrer que la fonction f qui à (n, i) associe a_i (dans la suite associée à \sqrt{n}) est récursive primitive.

Solution de l'exercice 13. Montrons que la fonction qui à n associe $\lfloor \sqrt{n} \rfloor$ est récursive primitive.

On cherche le plus petit entier k inférieur ou égal à x tel que $(k+1)^2$ soit supérieur strictement à x . Cette fonction est récursive primitive par composition de fonctions récursives primitives et par minimisation bornée. Deux solutions pour la suite :

1. On a vu que $n \mapsto \lfloor \sqrt{n} \rfloor$ est récursive primitive. Donc $\phi : (n, i) \mapsto \lfloor 10^i \sqrt{n} \rfloor = \lfloor \sqrt{10^{2i} n} \rfloor$ est récursive primitive par composition. On pose alors $f(n, i) = \begin{cases} \lfloor \sqrt{n} \rfloor & \text{si } i = 0 \\ \phi(n, i) \text{ modulo } 10 & \text{si } i \geq 1 \end{cases}$, qui est récursive primitive car définie par cas à partir de fonctions récursives primitives.

$$2. f(n, 0) = \begin{cases} \lfloor \sqrt{n} \rfloor & \text{si } i = 0 \\ \mu k \leq 9 \quad (10 \mid \lfloor 10^i \sqrt{n} \rfloor - k) & \text{si } i \geq 1 \end{cases}$$

Exercice 14. Soit A l'ensemble des nombres entiers dont l'écriture en base 10 est un palindrome (par exemple 13266231 ou 754434457). Montrer que A est récursif primitif.

Solution de l'exercice 14. On a vu comment extraire un chiffre d'un nombre dans l'exercice concernant le développement décimal de \sqrt{n} . Soit donc c la fonction qui sur la donnée d'un entier n renvoie la valeur du i ème chiffre de n (en partant de la droite, donc le chiffre correspondant à la puissance i de 10) dans le développement de n en base 10. $c(n, i)$ est le reste modulo 10 de la division entière de n par 10^i et est donc bien RP.

De même, il est facile de définir une fonction RP l qui renvoie la plus grande puissance de 10 apparaissant dans n : $l(n) = \mu k \leq n \ (10^{k+1} > n)$.

Un entier n est alors un palindrome ssi $\forall i \leq \lfloor \frac{n}{2} \rfloor, c(n, i) = c(n, l(n) - i)$, ce qui nous donne bien un test RP.

Exercice 15. Soit g une fonction récursive primitive. Soit f la fonction définie par : $f(0, x) = g(x)$ et $f(n+1, x) = f(n, f(n, x))$. Montrer que f est récursive primitive.

Solution de l'exercice 15. Montrer par récurrence sur n que $f(n, x) = g^{2^n}(x)$.

Exercice 16. Montrer qu'une fonction f est primitive récursive ssi le graphe de f est primitif récursif et f est majorée par une fonction primitive récursive.

Solution de l'exercice 16. Si f est récursive primitive elle est majorée par elle-même et la fonction caractéristique de son graphe est $\chi_{G_f}(x, y) = \chi_{=(x, f(x))}$. Réciproquement, si on sait que f est majorée par g , alors on peut écrire $f(x) = \mu y \leq g(x). (x, y) \in G_f$.

Exercice 17. Soit \mathcal{S}^* l'ensemble des suites finies d'entiers non vides et $\alpha : \mathcal{S}^* \Rightarrow \mathbb{N}$ la fonction qui à une suite non vide (x_1, \dots, x_p) associe $\alpha(x_1, \dots, x_p) = \alpha_2(p, \alpha_p(x_1, \dots, x_p))$.

1. Montrer que α est une injection dont l'image est un ensemble récursif primitif.
2. Montrer que la fonction $\phi : \mathbb{N}^3 \Rightarrow \mathbb{N}$ définie par :

$$\phi(i, p, x) = \begin{cases} \beta_p^i(x) & \text{si } 1 \leq i \leq p \\ 0 & \text{sinon} \end{cases}$$

est récursive primitive.

3. En déduire qu'il existe une fonction récursive primitive $\gamma \in \mathcal{F}_2$ qui sur la donnée de deux entiers z et i renvoie le i -ième élément de la suite finie d'entiers codée par z via α .

Solution de l'exercice 17. Remarque : α n'est pas récursive primitive car ce n'est pas une fonction de \mathbb{N}^k dans \mathbb{N} pour un entier k fixé.

1. Montrons que α est injective. Soit (x_1, \dots, x_m) et (y_1, \dots, y_n) appartenant à \mathcal{S}^* . Si $\alpha(x_1, \dots, x_m) = \alpha(y_1, \dots, y_n)$, alors $\alpha_2(m, \alpha_m(x_1, \dots, x_m)) = \alpha_2(n, \alpha_n(y_1, \dots, y_n))$. Ceci implique par injectivité de α_2 que $m = n$ et $\alpha_m(x_1, \dots, x_m) = \alpha_m(y_1, \dots, y_m)$, donc $m = n$ et $x_i = y_i, \forall i \in \{1, \dots, m\}$. α est bien injective.

Montrons maintenant que l'image de α est un ensemble récursif primitif. Comme α_2 est une bijection de \mathbb{N}^2 dans \mathbb{N} , on doit déterminer l'ensemble A des couples de la forme $(m, \alpha_m(x_1, \dots, x_m))$ pour $(x_1, \dots, x_m) \in \mathcal{S}^*$. Tout couple de la forme $(0, n)$ n'appartient pas à A .

Soit un couple $(m, n) \in \mathbb{N}^* \times \mathbb{N}$. On sait que α_m est une bijection de \mathbb{N}^m dans \mathbb{N} , donc il existe (x_1, \dots, x_m) tels que $n = \alpha_m(x_1, \dots, x_m)$. Donc $\text{Im}(\alpha) = \{n \mid \beta_2^1(n) \neq 0\}$ Comme le test " $\beta_2^1(n) \neq 0$ " est récursif primitif, $\text{Im}(\alpha)$ est un ensemble récursif primitif.

2. Soit la fonction ϕ de \mathbb{N}^3 dans \mathbb{N} définie par :

$$\phi(i, p, x) = \begin{cases} \beta_p^i(x) & \text{si } 1 \leq i \leq p, \\ 0 & \text{sinon.} \end{cases}$$

On utilise le schéma de récurrence sur p pour donner une définition récursive primitive de ϕ .

$$\begin{cases} \phi(i, 1, x) &= \begin{cases} x & \text{si } i = 1, \\ 0 & \text{sinon.} \end{cases} \\ \phi(i, p+1, x) &= \begin{cases} 0 & \text{si } i = 0 \text{ ou si } i > p+1, \\ \phi(i, p, x) & \text{si } i \leq p-1, \\ \beta_2^1(\phi(p, p, x)) & \text{si } i = p, \\ \beta_2^2(\phi(p, p, x)) & \text{si } i = p+1. \end{cases} \end{cases}$$

Mais la définition ci-dessus ne correspond pas au schéma de récurrence vu en cours, car on a seulement le droit d'utiliser la valeur "précédente" de la fonction qu'on est en train de calculer, à savoir ici $\phi(i, p, x)$. Cela ne pose pas de problèmes pour le cas $i = p$ de la troisième ligne, car alors $\phi(i, p, x) = \phi(p, p, x)$, mais il reste la dernière ligne et l'utilisation de $\phi(p, p, x)$ qui n'est pas la même chose que $\phi(i, p, x)$, car on est dans le cas $i = p + 1$. On définit donc une fonction auxiliaire ψ pour calculer $\phi(p, p, x) = \beta_p^p(x)$. On montre que cette fonction est récursive primitive par application d'un schéma de récurrence :

$$\begin{cases} \psi(1, x) &= x \\ \psi(p+1, x) &= \beta_{p+1}^{p+1}(x) = \beta_2^2(\psi(p, x)) \end{cases}$$

Ceci nous permet de compléter la définition par schéma de récurrence de la la fonction ϕ et de montrer ainsi qu'elle est bien récursive primitive.

3. On définit maintenant facilement la fonction γ qui est bien récursive primitive : $\gamma(z, i) = \phi(i, \beta_2^1(z), \beta_2^2(z))$.

Exercice 18. On rappelle que π est la fonction qui à n associe le $(n+1)$ -ième nombre premier. Soit $\Omega' : \mathcal{S}^* \Rightarrow \mathbb{N}$ la fonction qui à une suite non vide d'entiers (x_0, \dots, x_n) associe $\Omega'(x_0, \dots, x_n) = \prod_{i=0}^n \pi(i)^{1+x_i}$.

1. Montrer que Ω' est une injection dont l'image est un ensemble récursif primitif.
2. Montrer que l'on peut définir une fonction récursive primitive $\delta' : \mathbb{N}^2 \Rightarrow \mathbb{N}$ telle que, si z est de la forme $\Omega'(x_0, \dots, x_n)$ et si $0 \leq i \leq n$, alors $\delta'(z, i)$ est égal à x_i .

Solution de l'exercice 18. Soit $\Omega' : \mathcal{S}^* \rightarrow \mathbb{N}$ la fonction qui à une suite non vide d'entiers (x_0, \dots, x_{n-1}) associe $\prod_{i=0}^{n-1} p(i)^{1+x_i}$.

1. Montrons que Ω' est injective : si $\Omega'(x_0, \dots, x_{m-1}) = \Omega'(y_0, \dots, y_{n-1})$, alors $\prod_{i=0}^{m-1} p(i)^{1+x_i} = \prod_{i=0}^{n-1} p(i)^{1+y_i}$. D'après le caractère unique de la décomposition d'un entier en facteurs premiers, $m = n$ et $x_i = y_i$ pour $0 \leq i \leq n$.

Étudions l'image de $\Omega' : n$ appartient à l'image de Ω' si et seulement si $n \geq 2$ et il ne manque pas un élément dans la suite croissante des facteurs premiers de la décomposition de n . Nous allons essayer d'exprimer ceci de différentes manières, ce qui se traduira par différentes expressions primitives récursives. Dans la suite, p désigne le plus grand nombre premier divisant n , ce qui peut être calculé de manière récursive primitive : $p = n \ominus (\mu k \leq n (P(n \ominus k) \wedge (n \ominus k)|n))$.

1. Pour tout q premier inférieur ou égal à p , q divise n . Ceci s'exprime par $\forall q \leq p, (P(q) \Rightarrow q|n)$, donc par combinaisons booléennes et quantification bornée sur des fonctions récursives primitives.
2. p est égal au plus grand facteur premier de n "avant un trou" (i.e. tel que le nombre premier suivant p n'apparaisse pas dans n). On fait ce test en posant $i = \mu k \leq n(p(k+1) \text{ ne divise pas } n)$ et en testant l'égalité de p et $p(i)$.
3. Pour tout nombre premier q divisant n , alors tout nombre premier r inférieur ou égal à q divise n . On exprime ceci de manière récursive primitive : $\forall q \leq n, (P(q) \wedge q|n) \Rightarrow (\forall r \leq q, P(r) \Rightarrow r|n)$.
4. Si un nombre premier divise n , alors le nombre premier précédent (s'il existe) divise aussi n . On a ici une expression récursive primitive très simple : $\forall i \leq n, p(i)|n \Rightarrow p(i \ominus 1)|n$.

Notons que dans tous les cas ci-dessus (et en général), il faut faire attention aux cas "limites". Ici pour n valant 0 ou 1 il peut y avoir des problèmes, qu'on résoudra via une définition par cas.

2. Soit la fonction δ' de \mathbb{N}^2 dans $\mathbb{N} : \delta'(z, i) = \mu k \leq z, p(i)^{k+2}$ ne divise pas z (on peut décider que δ' vaut 0 si z n'appartient pas à l'image de Ω'). δ' est bien récursive primitive.

Exercice 19.

1. Rappeler la définition d'un codage des suites finies vu en cours (α, Ω ou Ω').

- Donner une définition récursive primitive de la fonction qui à un entier n associe le code de la suite de diviseurs premiers de n dans l'ordre croissant.

Solution de l'exercice 19.

- Codage α ou codage Ω ou Ω' . Nous choisirons Ω' .
- On définit d'abord la fonction R qui à deux entiers n et k associe le $k + 1^{\text{e}}$ diviseur premier de n : $R(n, k) = \mu t \leq n$. $\sum_{i=0}^t \chi_A(n, t) = k + 1$, où A est l'ensemble des couples (n, t) tels que t est un diviseur premier de n , dont on sait qu'il est récursif primitif (cours, TD). On renvoie ensuite $\prod_{k=0}^n \pi(k)^{1+R(n,k)}$.

Exercice 20. On rappelle que le codage Ω' est défini par $\Omega'(x_0, \dots, x_n) = \prod_{i=0}^n \pi(i)^{1+x_i}$. Montrer que le tri dans l'ordre croissant d'une suite, l'entrée et la sortie étant codées par la fonction Ω' , est récursif primitif.

Solution de l'exercice 20. Il existe plusieurs façons d'exprimer cette fonction de tri. Nous allons décrire une solution possible. Nous commençons par remarquer qu'il existe une fonction récursive primitive λ qui, sur la donnée d'un entier $z = \Omega'(x_0, \dots, x_n)$, renvoie l'entier n (et une valeur quelconque si z n'est pas dans l'image de Ω') : $\lambda(z) = \mu k \leq z$. ($\pi(k + 1)$ ne divise pas z). Nous allons ensuite coder une permutation σ de $\{0, \dots, n\}$ par l'entier $\Omega'(\sigma(0), \dots, \sigma(n))$. Il est facile d'écrire une fonction P qui teste si un entier p code bien une permutation comme ci-dessus : $P(p)$ vaut 1 ssi $\forall i \leq \lambda(p), \exists j \leq \lambda(p), \delta'(p, j) = i$. Nous considérons maintenant la fonction $E(y, z, p)$ qui teste si la permutation codée par p transforme la suite codée par y en la suite codée par z (et renvoie 0 si ce ne sont pas des codes ou si les longueurs ne sont pas compatibles) : E renvoie 1 ssi $\forall i \leq \lambda(y), \delta'(y, i) = \delta'(z, \delta'(p, i))$. Il nous faudra aussi la fonction $T(y)$ qui vérifie si une liste est triée : $\forall i \leq \lambda(y), \delta'(y, i - 1) \leq \delta'(y, i)$. Enfin, sur l'entrée z , notre fonction de tri va renvoyer le plus petit entier $y \leq z^{(z+1)^2}$ tel que y soit une liste triée et qu'il existe un code de permutation $p \leq \Omega'(0, \dots, \lambda(z))$ de sorte que $E(y, z, p)$.

Exercice 21. Soit f la fonction de \mathbb{N} dans \mathbb{N} définie par :

$$f(0) = 1, f(1) = 1, f(n + 2) = f(n + 1) + f(n).$$

Montrer que f est récursive primitive.

Solution de l'exercice 21. Cette fonction est définie (mathématiquement) par récurrence, mais il ne s'agit pas d'une application du schéma de récurrence, car on utilise les deux valeurs précédentes de la fonction. La définition donnée ne nous permet donc pas de conclure que la fonction est récursive primitive. Pour cela, on définit une fonction intermédiaire F par $F(n) = \alpha_2(f(n), f(n + 1))$. Montrons par le schéma de récurrence que F est récursive primitive :

$$\begin{cases} F(0) &= \alpha_2(f(0), f(1)) = \alpha_2(1, 1), \\ F(n + 1) &= \alpha_2(f(n + 1), f(n + 2)) = \alpha_2(f(n + 1), f(n + 1) + f(n)) \\ &= \alpha_2(\beta_2^2(F(n)), \beta_2^2(F(n)) + \beta_2^1(F(n))) \end{cases}$$

F est bien récursive primitive. Comme $f = \beta_2^1 \circ F$, f est récursive primitive par composition.

Exercice 22. Montrer que l'ensemble des fonctions récursives primitives est clos par le schéma de récurrence sur la suite des valeurs, c'est-à-dire que le calcul d'une fonction en n peut faire intervenir toutes les valeurs prises par la fonction sur les valeurs plus petites que n .

Soit $g : \mathbb{N}^p \rightarrow \mathbb{N}$ et $h : \mathbb{N}^{p+2} \rightarrow \mathbb{N}$ des fonctions récursives primitives, montrer que la fonction $f : \mathbb{N}^{p+1} \rightarrow \mathbb{N}$ définie ci-dessous est récursive primitive :

$$\begin{aligned} f(a_1, \dots, a_p, 0) &= g(a_1, \dots, a_p), \\ f(a_1, \dots, a_p, n + 1) &= h\left(a_1, \dots, a_p, n, \Omega'(f(a_1, \dots, a_p, 0), \dots, f(a_1, \dots, a_p, n))\right). \end{aligned}$$

Solution de l'exercice 22.

On note \bar{a} pour a_1, \dots, a_p afin d'alléger les notations. On considère la fonction $F(\bar{a}, n) = \Omega'(f(\bar{a}, 0), \dots, f(\bar{a}, n))$. Montrons par un schéma de récurrence classique que F est récursive primitive sous les hypothèses de l'énoncé. Commençons par donner le cas de base :

$$F(\bar{a}, 0) = \Omega'(f(\bar{a}, 0)) = 2^{1+g(\bar{a})}.$$

Donnons maintenant l'étape de récurrence :

$$\begin{aligned} F(\bar{a}, n+1) &= \Omega'(f(\bar{a}, 0), \dots, f(\bar{a}, n), f(\bar{a}, n+1)) \\ &= \Omega'(f(\bar{a}, 0), \dots, f(\bar{a}, n)) \cdot p(n+1)^{f(\bar{a}, n+1)} \\ &= F(\bar{a}, n) \cdot p(n+1)^{h(\bar{a}, n, F(\bar{a}, n))}. \end{aligned}$$

La dernière expression est bien récursive primitive et ne fait intervenir que $F(\bar{a}, n)$, c'est-à-dire la valeur de la fonction F pour le paramètre n . La fonction F est alors récursive primitive, ainsi que la fonction f obtenue par $f(\bar{a}, n) = \delta'(F(\bar{a}, n), n)$.

Exercice 23. Soit g_1, g_2, h des fonctions récursives primitives. Montrer que la fonction f définie par

$$f(0, y) = g_1(y), \quad f(x+1, 0) = g_2(x), \quad f(x+1, y+1) = h(x, y, f(x, y+1), f(x+1, y))$$

est récursive primitive.

Solution de l'exercice 23. On va calculer la fonction par diagonales (i.e., on calcule d'un coup toutes les valeurs $f(x, y)$ pour $x+y$ constant). Posons donc $F(n) = \Omega'(f(n, 0), \dots, f(0, n))$. Montrons que la fonction F est récursive primitive par un schéma de récurrence :

$$\left\{ \begin{array}{l} F(0) = \Omega'(f(0, 0)) = 2^{1+g_1(0)} \\ F(n+1) = \Omega'(f(n+1, 0), \dots, f(i, n+1 \ominus i), \dots, f(0, n+1)) \\ \quad = \Omega'(g_2(n), \dots, f(i, n+1 \ominus i), \dots, g_1(n+1)) \\ \quad = 2^{1+g_2(n)} \cdot \left(\prod_{i=1}^n p(i)^{h(n \ominus i, i \ominus 1, f(n \ominus i, i), f(n+1 \ominus i, i \ominus 1))} \right) \cdot p(n+1)^{1+g_1(n+1)} \\ \quad = 2^{1+g_2(n)} \cdot \left(\prod_{i=1}^n p(i)^{h(n \ominus i, i \ominus 1, \delta'(F(n), i), \delta'(F(n), i \ominus 1))} \right) \cdot p(n+1)^{1+g_1(n+1)} \end{array} \right.$$

La fonction f est donc elle aussi récursive primitive car $f(x, y) = \delta'(F(x+y), y)$.

Exercice 24. Soit p un entier non nul. Montrer que l'ensemble ci-dessous est récursif primitif :

$$E_p = \{(a_0, \dots, a_p) \in \mathbb{N}^{p+1} \mid \text{le polynôme } a_0 + a_1X + \dots + a_pX^p \text{ a un zéro dans } \mathbb{Z}\}.$$

Solution de l'exercice 24. Soit p un entier non nul et soit E_p l'ensemble des uplets $(a_0, \dots, a_p) \in \mathbb{N}^{p+1}$ tels que le polynôme $a_0 + a_1X + \dots + a_pX^p$ a une racine dans \mathbb{Z} . Montrons que E_p est récursif primitif. On note Q le polynôme $a_0 + a_1X + \dots + a_pX^p$. Comme Q a des coefficients positifs ses racines ne peuvent être que négatives ou nulles.

Si $a_0 = 0$, 0 est une racine entière de Q et l'uplet (a_0, \dots, a_p) appartient à E_p .

Comme nous pouvons faire le test sur a_0 de manière récursive primitive, nous supposons maintenant que $a_0 \neq 0$. Dans ce cas, toute racine n est strictement négative et les parties positives et négatives dans l'évaluation du polynôme doivent se compenser : $a_0 + a_2n^2 + \dots + a_pn^p = a_1(-n) + a_3(-n)^3 + \dots + a_{p-1}(-n)^{p-1}$. (On a supposé ici que p est pair. Une petite modification sur les indices suffit pour écrire le cas où p est impair.)

Nous avons donc une caractérisation récursive primitive des racines, il nous reste à avoir une borne sur une racine entière, pour pouvoir utiliser une quantification bornée. Or $Q(n) = 0$ implique que $a_0 = -n(a_1 + a_2n + \dots + a_pn^{p-1})$, donc que n divise a_0 .

La fonction caractéristique de E_p (p pair) teste donc s'il existe un m inférieur ou égal à a_0 tel que $a_0 + a_2m^2 + \dots + a_pm^p = a_1m + a_3m^3 + \dots + a_{p-1}m^{p-1}$ ou si $a_0 = 0$. Notre ensemble E_p est bien récursif primitif.

Exercice 25.

1. Montrer que les fonctions polynômes à plusieurs variables à coefficients dans \mathbb{N} sont primitives récursives.
2. En s'inspirant du codage des suites finies dans \mathbb{N} utilisant la décomposition en facteur premiers, proposer un codage Θ de l'ensemble $\mathbb{N}[X]$ des polynômes à une variable à coefficients dans \mathbb{N} qui soit *bijectif*. Plus précisément, on veut :
 - (i) $\Theta : \mathbb{N}[X] \Rightarrow \mathbb{N}$ bijective ;

- (ii) il existe une fonction primitive récursive $c : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, telle que $c(i, n)$ est le coefficient de degré i dans le polynôme $P_n = \Theta^{-1}(n)$;
- (iii) il existe une fonction primitive récursive $d : \mathbb{N} \rightarrow \mathbb{N}$ telle que $d(n)$ est le degré du polynôme $P_n = \Theta^{-1}(n)$ (on considère que le degré du polynôme nul est 0).
On note dorénavant P_n pour $\Theta^{-1}(n)$ (ou pour la fonction polynôme associée à P_n).
- 3. Montrer que l'on peut coder la fonction qui applique une fonction polynôme à un entier, plus précisément, montrer que la fonction $a : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ vérifiant $a(n, x) = P_n(x)$ est primitive récursive.
- 4. Soit les deux fonctions $p : \mathbb{N} \rightarrow \mathbb{N}$ et $i : \mathbb{N} \rightarrow \mathbb{N}$ telles que $P_{p(n)}$, respectivement $P_{i(n)}$ soit le polynôme dont les monômes sont exactement les monômes de puissance paire, respectivement impaire, de P_n . Montrer que p et i sont primitives récursives.
- 5. En déduire une fonction partielle récursive $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que si $P_n = \Theta^{-1}(n)$ possède un zéro dans \mathbb{Z} , alors $f(n)$ est la valeur absolue d'un zéro de P_n , sinon f n'est pas définie.
- 6. En fait on peut trouver une fonction primitive récursive qui calcule un zéro de P_n . Montrer qu'il existe une fonction primitive récursive $g : \mathbb{N} \rightarrow \mathbb{N}$, telle que si P_n possède un zéro dans \mathbb{Z} , alors $g(n) > 0$ et $g(n) - 1$ est la valeur absolue d'un zéro de P_n , sinon $g(n) = 0$.
- 7. En déduire que l'ensemble des n tels que P_n possède un zéro dans \mathbb{Z} est primitif récursif.

Solution de l'exercice 25. On utilise la fonction $\pi : \mathbb{N} \rightarrow \mathbb{N}$ telle que $\pi(n)$ soit le $n + 1$ nombre premier, dont on sait qu'elle est primitive récursive.

- 1. Les fonctions polynômes s'obtiennent par composition à partir de l'addition, de la multiplication et des fonctions de projections, et sont donc primitives récursives. (Faire éventuellement une démonstration par récurrence sur le degré.)
- 2. On utilise la définition d'un polynôme comme une suite d'entiers à support fini, i.e. la suite est nulle sauf pour un nombre fini d'indices. Soit $P = (c_i)_{i \in \mathbb{N}}$ un polynôme et d le plus petit indice tel que $\forall j > d, c_j = 0$, alors posons $\Theta(P) = \prod_{i=0}^d \pi(i)^{c_i} - 1$. L'existence de la décomposition en facteur premiers d'un entier non nul nous assure que cette application est surjective, l'unicité à des exposants nuls près nous assure de l'injectivité.
 - (i) Θ est bijective d'après ce qui précède.
 - (ii) Soit $c : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, telle que

$$c(i, n) = \mu p \leq n (\pi(i)^{p+1} \nmid (n + 1)).$$

On a bien défini ci-dessus $c(i, n)$ le coefficient de degré i dans le polynôme $P_n = \Theta^{-1}(n)$: le coefficient est forcément plus petit que le code du polynôme car pour tous i et p , on a $p < \pi(i)^p$, et comme $c(i, n) = p$ si et seulement si $\pi(i)^p \mid (n + 1)$, alors $p < n + 1$. La fonction c est primitive récursive par minimisation bornée sur un prédicat primitif récursif (stabilité par combinaison booléenne ; l'exponentielle, π et le prédicat \mid sont primitifs récursifs).

- (iii) Soit $d : \mathbb{N} \rightarrow \mathbb{N}$ telle que

$$d(n) = n \ominus (\mu p \leq n ((\pi(n \ominus p) \mid (n + 1)))).$$

On a bien défini $d(n)$ comme le degré du polynôme $P_n = \Theta^{-1}(n)$, en effet ce degré est forcément inférieur ou égal au code du polynôme car pour tout entier $q, q < \pi(q), d(n)$ est donc le plus grand nombre premier qui divise $n + 1$, pour $n \neq 0$, et $d(0) = 0$. On a bien que d est primitive récursive (composition, minimisation bornée, l'exponentielle, π et le prédicat \mid sont primitifs récursifs).

- 3. soit $a : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ vérifiant

$$a(n, x) = \sum_{i=0}^{d(n)} c(i, n).x^i.$$

On a bien $a(n, x) = P_n(x)$, et a_n est primitive récursive, d'après les questions précédentes et par somme bornée (remarque : on n'a pas vraiment besoin de d , car on a aussi $a(n, x) = \sum_{i=1}^n c(i, n).x^n$).

- 4. On sait que «être pair» est un prédicat primitif récursif, soit χ sa fonction caractéristique. On définit $p : \mathbb{N} \rightarrow \mathbb{N}$ et $i : \mathbb{N} \rightarrow \mathbb{N}$ par :

$$p(n) = \prod_{i=0}^{d(n)} \pi(i)^{\chi(i).c(i, n)} \ominus 1 \quad i(n) = \prod_{i=0}^{d(n)} \pi(i)^{(1 \ominus \chi(i)).c(i, n)} \ominus 1.$$

On a bien que $P_{p(n)}$, respectivement $P_{i(n)}$ est le polynôme dont les monômes sont exactement les monômes de puissance paire, respectivement impaire, de P_n . On a aussi p et i sont primitives récursives : produit borné et question 2.

5. Un zéro d'un polynôme à coefficients dans \mathbb{N} est forcément négatif. Si $f(n)$ est la valeur absolue d'un zéro de P_n , alors $-f(n)$ est un zéro de P_n , or, pour $x \in \mathbb{N}$, $P_n(-x) = 0$ s'écrit aussi $P_{p(n)}(x) = P_{i(n)}(x)$. Si l'on définit $f : \mathbb{N} \rightarrow \mathbb{N}$ par

$$f(n) = \mu x (P_{p(n)}(x) = P_{i(n)}(x)) = \mu x (a(p(n), x) = a(i(n), x)).$$

on a donc bien que $-f(n)$ est un zéro de P_n . Par ailleurs f est partielle récursive par minimisation sur un prédicat qui est récursif d'après les questions 3 et 4.

6. On peut facilement borner la minimisation ci-dessus en remarquant qu'un zéro entier d'un polynôme à coefficients entiers est un diviseur de son coefficient de degré nul, et donc, si dernier est non nul, est toujours inférieur ou égal à ce coefficient. Si le coefficient de degré nul est 0, on choisit 0 comme zéro du polynôme. Pour pouvoir distinguer les polynômes qui n'ont pas de zéro de ceux qui s'annulent en zéro, on peut translater de 1 le zéro cherché, on définit $g : \mathbb{N} \rightarrow \mathbb{N}$,

$$g(n) = \mu x \leq c(0, n) [x \geq 1 \text{ et } a(p(n), x \ominus 1) = a(i(n), x \ominus 1)].$$

On a bien que si P_n possède un zéro dans \mathbb{Z} , alors $g(n) > 0$ et $g(n) - 1$ est la valeur absolue d'un zéro de P_n , sinon $g(n) = 0$. On a aussi que g est primitive récursive par minimisation bornée sur un prédicat primitif récursif d'après les questions 3 et 4.

7. L'ensemble des n tels que P_n possède un zéro dans \mathbb{Z} a pour fonction caractéristique $sgn \circ g$, où sgn est la fonction qui vaut 1 ssi son argument est non nul. Cet ensemble est donc primitif récursif.

Exercice 26. On définit les fonctions élémentaires \mathcal{F}^E comme le plus petit sous-ensemble de \mathcal{F}^* contenant :

- les fonctions nulles 0_p ;
- la fonction successeur ;
- la relation d'égalité $R_=($ (définie par $R_=(x, y) = 1$ si $x = y$ et $R_=(x, y) = 0$ sinon) ;
- les projections ;

et clos par les opérations suivantes :

- composition ;
- somme bornée et produit borné. Cela signifie que si cet ensemble contient la fonction

$$(y, x_1, \dots, x_n) \mapsto f(y, x_1, \dots, x_n),$$

alors il contient aussi les fonctions

$$(z, x_1, \dots, x_n) \mapsto \sum_{y < z} f(y, x_1, \dots, x_n) \text{ et } (z, x_1, \dots, x_n) \mapsto \prod_{y < z} f(y, x_1, \dots, x_n).$$

1. Montrer que \mathcal{F}^E est un sous-ensemble des fonctions récursives primitives.
2. a) Montrer que la fonction $(x, y) \mapsto xy$ est élémentaire.
b) Montrer que la relation R_{\leq} est élémentaire (la relation R_{\leq} est définie par $R_{\leq}(x, y) = 1$ si $x \leq y$ et $R_{\leq}(x, y) = 0$ sinon).
c) Montrer que la soustraction propre \ominus est élémentaire (on rappelle que $x \ominus y = x - y$ si $x \geq y$ et $x \ominus y = 0$ sinon).
d) Montrer que $(x, y) \mapsto x + y$ est élémentaire. *Indication* : exprimer la somme à l'aide des produits $(x + 1)(y + 1)$ et xy , et de la soustraction propre.

On définit les fonctions $e_r : \mathbb{N} \rightarrow \mathbb{N}$ par $e_0(x) = x$ et $e_{r+1}(x) = 2^{e_r(x)}$.

3. Montrer que pour tout $x \in \mathbb{N}$, $x^2 < 2^{2^x} = e_2(x)$.

Dans cet exercice, on dit qu'une fonction $f(x_1, \dots, x_p)$ est *dominée* par e_r si pour tout $x_1, \dots, x_p \in \mathbb{N}$, $f(x_1, \dots, x_p) \leq e_r(\max(x_1, \dots, x_p))$.

4. Montrer que pour toute fonction élémentaire $(x_1, \dots, x_p) \mapsto f(x_1, \dots, x_p)$, il existe un entier r tel que e_r domine f .
5. Montrer que la fonction $n \mapsto e_n(n)$ est récursive primitive.
6. En déduire que \mathcal{F}^E est un sous-ensemble *strict* des fonctions récursives primitives.

Solution de l'exercice 26.

1. On sait les fonctions listées sont récursives primitives et l'ensemble des fonctions récursifs primitifs est clos par composition et somme et produit bornés, donc \mathcal{F}^E est un sous-ensemble des fonctions récursives primitives.
2. a) $xy = \sum_{z < y} p_2^1(x, y)$.
 b) $R_{\leq}(x, y) = R_{=}(x, y) + \sum_{z < y} R_{=}(z, x)$.
 c) $x \ominus y = \sum_{z < y} R_{\leq}(x, z)$.
 d) $x + y = (x + 1)(y + 1) \ominus xy \ominus 1$.
3. $x < 2^x$, donc $x^2 < 2^{2x} \leq 2^{2^x}$.
4. Nous allons montrer cette propriété par induction. Elle est vraie pour les fonctions élémentaires "de base". Vérifions alors qu'elle est stable par composition. Soit $f : \mathbb{N}^n \Rightarrow \mathbb{N}$ et $g_1, \dots, g_n : \mathbb{N}^p \Rightarrow \mathbb{N}$ des fonctions élémentaires. Alors par hypothèse d'induction il existe r, r_1, \dots, r_n tels que e_r domine f et e_{r_i} domine g_i . Soit $x_1, \dots, x_p \in \mathbb{N}$, alors $g_i(x_1, \dots, x_p) \leq e_{r_i}(\max(x_1, \dots, x_p))$. On a alors $f(g_1, \dots, g_n)(x_1, \dots, x_p) \leq e_r(\max(g_1(\bar{x}), \dots, g_n(\bar{x}))) \leq e_r(e_s(\max(x_1, \dots, x_p)))$, où s est le maximum des r_i . On voit donc que $f(g_1, \dots, g_n)$ est dominée par e_{r+s} .
 Vérifions qu'elle est stable par somme bornée. Soit f une fonction élémentaire, donc dominée par un certain e_r . Alors $\sum_{y < z} f(y, \bar{x}) \leq \sum_{y < z} e_r(\max(y, x_1, \dots, x_n)) \leq z \cdot e_r(\max(z, x_1, \dots, x_n)) \leq (e_r(\max(z, x_1, \dots, x_n)))^2 \leq e_{r+2}(\max(z, x_1, \dots, x_n))$.
 Vérifions qu'elle est stable par produit borné. Soit f une fonction élémentaire, donc dominée par un certain e_r . Alors $\prod_{y < z} f(y, \bar{x}) \leq \prod_{y < z} e_r(\max(y, x_1, \dots, x_n)) \leq (e_r(\max(z, x_1, \dots, x_n)))^z \leq 2^{z \cdot (e_{r-1}(\max(z, x_1, \dots, x_n)))} \leq 2^{(e_{r-1}(\max(z, x_1, \dots, x_n)))^2} \leq e_{r+2}(\max(z, x_1, \dots, x_n))$.
5. Il est facile de voir que $e : (n, x) \mapsto e_n(x)$ est récursive primitive, par schéma de récurrence sur n , puis on en déduit que $n \mapsto e_n(n)$ l'est aussi.
6. $n \mapsto e_n(n)$ ne peut pas être dominée par une fonction e_r , car si elle l'était on aurait $e_{r+1}(r+1) \leq e_r(r+1)$. Donc $n \mapsto e_n(n)$, qui est récursive primitive, n'est pas élémentaire.