Tristan Stérin
FDI
DM #2
21/12/2015

Exercice 1

Question 1 Montrer que les fonctions suivantes sont primitves récursives.

(a) Les fonctions constantes $c_a(n) = a$.

(b) La fonction *somme* définie par $somme(n, m) = n + m$.

(c) La fonction *prédécesseur* p définie par $p(n) = max(0, n - 1)$.

(d) La fonction *prod* définie par $prod(n, m) = nm$.

(e) La fonction *eq0* définie par $eq0(m) = 1$ si $m = 0$ et $eq0(m) = 0$ sinon.

(f) La fonction *eq* définie par $eq(n, m) = 1$ si $m = n$ et $eq(n, m) = 0$ sinon.

(g) Toute fonction $f : N \to \mathbb{N}$ à support fini :

$$\exists E \subseteq \mathbb{N}, |E| < \infty \quad x \notin E \implies f(x) = 0$$

.

(a) Soit $a \in \mathbb{N}$.
On montre que :

$$\boxed{\forall n \in \mathbb{N}, c_a(n) = s^a(\odot(n))}$$

Avec s^a s composé a fois.
Une récurrence sur a montre que $s^a(0) = a$.
Écrit sous cette forme on constate que c_a est récursive primitive par $a + 1$ applications du schéma de composition.

■

(b) On définit $somme'$:

$$\boxed{\begin{aligned} \forall (n,m) \in \mathbb{N}^2, \quad & somme'(n, 0) = n \\ & somme'(n, m+1) = (s \circ p_3^3)(n, m, somme'(n, m)) \end{aligned}}$$

$somme'$ est PR par application une fois du schéma de récurrence primitive : $s \circ p_3^3$ est PR par composition.

Soit $n \in \mathbb{N}$, on montre par récurrence sur $m \in \mathbb{N}$:

$$\mathbf{P}(m) \ : \ "somme'(n,m) = somme(n,m) = n+m"$$

— $\mathbf{P}(0)$: $somme'(n,0) = n = n+0$
— **Hérédité** : Soit $m \in \mathbb{N}$ tel que $\mathbf{P}(m)$, on a :

$$\begin{aligned} somme'(n,m+1) &= (s \circ p_3^3)(n,m,somme'(n,m)) \\ &= s(somme'(n,m)) \\ &= s(n+m) \\ &= (n+m)+1 \\ &= n+(m+1) \end{aligned}$$

D'où $\mathbf{P}(m+1)$.

Donc $somme = somme'$ et $somme$ est PR.

■

(c) L'énoncé est un peu ambigü comme soulevé dans la FAQ.
On considère malgré les définitions que les $\mathbb{N}^0 \to \mathbb{N}$ sont PR (les constantes sont PR, ça justifie).
Ainsi on peut définir des fonctions à une seule variable en schéma de récursion primitive.
On montre que p vérifie :

$$\boxed{\begin{array}{c} p(0) = 0 \\ \forall n \in \mathbb{N}, \quad p(n+1) = p_1^2(n, p(n)) \end{array}}$$

En effet, $p(0) = max(0,-1) = 0$ et pour $n \in \mathbb{N}$:

$$p(n+1) = max(0,n) = n = p_1^2(n,p(n))$$

Sous cette forme on constate que p est PR.

■

(d) On définit *prod'* :

$$\boxed{\begin{array}{rl} \forall (n,m) \in \mathbb{N}^2, & prod'(n,0) = 0 \\ & prod'(n,m+1) = somme_{13}(n,m,prod'(n,m)) \\ & \text{avec : } somme_{13}(x_1,x_2,x_3) = somme(p_1^3(x_1,x_2,x_3), p_3^3(x_1,x_2,x_3)) \end{array}}$$

$somme_{13}$ est PR par schéma de composition et 1b) donc $prod'$ est PR par schéma de récurrence.

Soit $n \in \mathbb{N}$, on montre par récurrence sur $m \in \mathbb{N}$:

$$\mathbf{P}(m)\ :\ "prod'(n,m) = prod(n,m) = n * m"$$

- $\mathbf{P}(0)$: $prod'(n,0) = 0 = n * 0$
- **Hérédité** : Soit $m \in \mathbb{N}$ tel que $\mathbf{P}(m)$, on a :

$$\begin{aligned} prod'(n,m+1) &= somme_{13}(n,m,prod'(n,m)) \\ &= n + prod'(n,m) \\ &= n + n * m \\ &= n * (m+1) \end{aligned}$$

  D'où $\mathbf{P}(m+1)$.

Donc $prod = prod'$ et $prod$ est PR. ∎

(e) On montre que $eq0$ vérifie :

$$\boxed{\begin{aligned} & eq0(0) = 1 \\ \forall n \in \mathbb{N}, \quad & eq0(n+1) = (\odot \circ p_1^2)(n, eq0(n)) \end{aligned}}$$

En effet, $eq(0) = 1$ et pour $n \in \mathbb{N},\ n+1 \neq 0$:

$$eq0(n+1) = 0 = \odot(n)$$

Sous cette forme on constate que $eq0$ est PR. ∎

(f) On montre que eq vérifie :

$$\boxed{\forall (n,m) \in \mathbb{N}^2, \quad eq(n,m) = (eq0 \circ somme)(sub(n,m), sub(m,n))}$$

Avec :

$$\boxed{\begin{aligned} & sub(n,0) = n \\ \forall (n,m) \in \mathbb{N}^2, \quad & sub(n,m+1) = (p \circ p_3^3)(n,m,sub(n,m)) \end{aligned}}$$

On constate que sub est PR.
Une récurrence montre que $sub(n,m) = max(0, n-m)$.
Soient $(n,m) \in \mathbb{N}^2$:

- Si $n = m,\ sub(n,m) = sub(m,n) = 0$ et donc :
  $eq(n,m) = 1 = eq0(sub(n,m) + sub(m,n))$
- Si $n \neq m$, mettons $n > m$.
  Alors $sub(n,m) = n - m \neq 0$ et $sub(m,n) = 0$ aisni $eq0(sub(n,m) + sub(m,n)) = 0$
  Ainsi $eq(n,m) = 0 = eq0(sub(n,m) + sub(m,n))$

Mis sous cette forme, on constate que eq est PR.

$\blacksquare$

(g) Soit $f : \mathbb{N} \to \mathbb{N}$ à support fini.
Il existe $k \in \mathbb{N}$ et $E \subseteq \mathbb{N} = \{x_1, \dots, x_k\}$, $|E| = k$, tels que :

$$\forall x \in \mathbb{N} \quad x \notin E \implies f(x) = 0$$

Si $k = 0$, $f = \odot$ et le résultat est connu.
On suppose donc $k \geq 1$.
On note pour $1 \leq i \leq k \quad y_i = f(x_i)$.
On définit f' :

$$\boxed{\forall x \in \mathbb{N}, \quad f'(x) = somme_k(d_1(eq0(eq(x, x_1))), \dots, d_p(eq0(eq(x, x_p))))}$$

Avec, pour $1 \leq i \leq k$:

$$\boxed{\begin{array}{c} d_i(0) = y_i \\ \forall n \in \mathbb{N}, \quad d_i(n+1) = (\odot \circ p_1^2)(n, d_i(n)) \end{array}}$$

Et :

$$\boxed{\begin{array}{l} somme_1 = p_1^1 \\ \forall n \in \mathbb{N}^* \quad \forall (x_1, \dots, x_{n+1}) \in \mathbb{N}^{n+1}, \\ somme_{n+1} : \mathbb{N}^{n+1} \to \mathbb{N} \\ somme_{n+1}(x_1, \dots, x_{n+1}) = somme(p_1^{n+1}(x_1, \dots, x_{n+1}), Somme_n(x_1, \dots, x_{n+1})) \end{array}}$$

$$\boxed{\forall n \in \mathbb{N}^* \quad Somme_n(x_1, \dots, x_{n+1}) = somme_n(p_2^{n+1}(x_1, \dots, x_{n+1}), \dots, p_{n+1}^{n+1}(x_1, \dots, x_{n+1}))}$$

Par récurrence, on obtient que :

- $\forall k \in \mathbb{N}^*, \ somme_k$ est PR
- $\forall k \in \mathbb{N} - \{0, 1\} \ \forall (x_1, \dots, x_k) \in \mathbb{N}^k, \ somme_k(x_1, \dots, x_k) = x_1 + \cdots + x_k$

De plus les d_i sont PR donc f' est PR.

On constate que pour $1 \leq i \leq k$:

$$\forall x \in \mathbb{N} \quad x = x_i \Leftrightarrow d_i(eq0(eq(x, x_i))) = y_i$$

$$\forall x \in \mathbb{N} \quad x \neq x_i \Rightarrow d_i(eq0(eq(x, x_i))) = 0$$

(eq0 agit comme un not logique ici).

Donc :
$$\forall x \in \mathbb{N} \quad x \notin E \implies f'(x) = 0$$
(tous les termes de la somme sont nuls)
$$\forall 1 \leq i \leq k \quad f'(x_i) = y_i$$
(un seul terme non nul, les x_i sont nécessairement distincts puisque $|E| = k$).
On en conclut $f = f'$ et f est PR. ■

Question 2 On définit la fonction d'Ackermann via la récurrence double suivante :
$$\begin{aligned} Ack(0, x) &= x + 2 \\ Ack(1, 0) &= 0 \\ Ack(n+2, 0) &= 1 \\ Ack(n+1, x+1) &= Ack(n, Ack(n+1, x)) \end{aligned}$$
Montrer que pour tout $n \in \mathbb{N}$, la fonction $Ack_n : x \mapsto Ack(n, x)$ est PR.

- **Cas** $n = 0$: $Ack_0(x) = x + 2 = s(s(x)) = s^2(x)$, PR par schéma de composition.
- **Cas** $n = 1$:
$$\boxed{\begin{aligned} & Ack_1(0) = 0 \\ \forall x \in \mathbb{N}, \quad & Ack_1(x+1) = Ack(1, x+1) = Ack(0, Ack(1, x)) = (Ack_0 \circ p_2^2)(x, Ack_1(x)) \end{aligned}}$$
PR par schémas de composition et de récurrence.
- **Cas** $n \geq 2$:
$$\boxed{\begin{aligned} & Ack_n(0) = 1 \\ \forall x \in \mathbb{N}, \quad & Ack_n(x+1) = Ack(n, x+1) = (Ack_{n-1} \circ p_2^2)(x, Ack_n(x)) \end{aligned}}$$
Par récurrence sur $n \geq 2$ on obtient que les $Ack_{n \geq 2}$ sont PR.
Ainsi, pour tout $n \in \mathbb{N}$, la fonction $Ack_n : x \mapsto Ack(n, x)$ est PR.

■

Question 3
On veut montrer dans cette question que la fonction d'Ackermann n'est pas primitive récursive.

(*a*) Montrer que $\forall n \in \mathbb{N}, x \in \mathbb{N}^*, Ack_n(x) > x$. En déduire que pour tout entier n, Ack_n est strictement croissante et que Ack est croissante en son premier argument : $\forall x \geq 2\ \forall n \in \mathbb{N},\ Ack(n, x) \leq Ack(n+1, x)$

(a) On va procéder par induction bien fondée sur $\mathbb{N} \times \mathbb{N}^*$ muni de l'ordre lexicographique naturel (total et bien fondé). Il admet pour minimum : $(0, 1)$.
On veut montrer :

$$\forall n \in \mathbb{N}\ \forall x \in \mathbb{N}^*\ \mathbf{P}(n, x) : \text{"}Ack_n(x) > x\text{"}$$

— $\mathbf{P}(0, 1)$: $Ack_0(1) = 3 > 1$

— **Induction** : Soit $(n, x) \in \mathbb{N} \times \mathbb{N}^*$ (et donc $x > 0$), on suppose $\mathbf{P}$ vraie pour tous les couples lexicographiquement strictement plus petits. Si $n = 0$ on conclut car $x + 2 > x$, sinon :

$$Ack_n(x) = Ack_n((x-1)+1) = Ack_{n-1}(Ack_n(x-1)) > Ack_n(x-1) > x - 1$$

Ceci par hypothèse d'induction car $\forall a \in \mathbb{N}^* (n, x) > (n-1, a)$ et $(n, x) > (n, x-1)$.
Par les deux inégalités strictes on en conclut :

$$Ack_n(x) > x$$

Soit $n \in \mathbb{N}$ on souhaite montrer :

$$\forall x \in \mathbb{N}\ Ack_n(x+1) > Ack_n(x)$$

Si $n = 0$ on constate que c'est vrai. On suppose $n \geq 1$:
Soit $x \in \mathbb{N}$:

$$Ack_n(x+1) = Ack_{n-1}(Ack_n(x)) > Ack_n(x) > x$$

Ceci d'après la preuve précédente.
D'où le résultat.
On veut maintenant montrer que :

$$\forall x \geq 2\ \forall n \in \mathbb{N},\ Ack_n(x) \leq Ack_{n+1}(x)$$

Soit $x \geq 2$, on peut appliquer le premier résultat à $x - 1 \geq 1$, soit $n \in \mathbb{N}$ on a :
$Ack_{n+1}(x-1) > x - 1$ donc $Ack_{n+1}(x-1) \geq x$.
Par croissance de Ack_n (précédent résultat) :

$$Ack_{n+1}(x) = Ack_n(Ack_{n+1}(x-1)) \geq Ack_n(x)$$

D'où le résultat.

∎

(*b*) On pose pour k entier, $Ack_n^k = Ack_n \circ \ldots \circ Ack_n$, où la composition est prise k fois. Montrer que $\forall n, k, x \in \mathbb{N}\ Ack_n^k(x) \leq Ack_{n+1}(x+k)$.

(b) Soient $n, x \in \mathbb{N}$ on montre par récurrence sur $k \in \mathbb{N}$:

$$\mathbf{P}(k) : \text{"}Ack_n^k(x) \leq Ack_{n+1}(x+k)\text{"}$$

— $\mathbf{P}(0)$: $Ack_n^0(x) = x$ on doit donc montrer :

$$x \leq Ack_{n+1}(x)$$

Pour $x \neq 0$ on sait déjà $x < Ack_{n+1}(x)$ (par 3a) d'où le résultat.
Si $x = 0$:

- — $Ack_0(0) = 2 > 0$
- — $Ack_1(0) = 0 \geq 0$
- — $Ack_{n>1}(0) = 1 > 0$

D'où le résultat dans tous les cas.

— **Hérédité** : Soit $k \in \mathbb{N}$ on suppose $\mathbf{P}(k)$. On a :

$$Ack_n^{k+1}(x) = Ack_n(Ack_n^k(x)) \leq Ack_n(Ack_{n+1}(x+k))$$

Par croissance et hypothèse de récurrence.
Or $Ack_n(Ack_{n+1}(x+k)) = Ack_{n+1}(x+k+1)$.
D'où $\mathbf{P}(k+1)$
D'où le résultat.

■

(*c*) Montrer que Ack_n^k est dominée par Ack_{n+1}.

(c) On procède en quatre étapes.

Lemme 0

(*a*) $\forall n \geq 0 \quad Ack_n(1) \geq 2$

(*b*) $\forall n \geq 0 \quad Ack_n(2) > 3$

Preuve (a) :
Par récurrence sur n : $Ack_0(1) = 3 \geq 2$
$Ack_{n+1}(1) = Ack_n(Ack_{n+1}(0)) = Ack_n(1) \geq 2$ (par HR).
Preuve (b) : c'est analogue. ■

Lemme 1 Soient $k, n \in \mathbb{N}$:
Soit $x_0 > 0$ tel que $Ack_n^k(x_0) \leq Ack_{n+1}(x_0)$
Alors : $\forall x \geq x_0 \ Ack_n^k(x) \leq Ack_{n+1}(x)$

Preuve
On se donne un tel x_0.
Il suffit de montrer : $Ack_n^k(x_0+1) \leq Ack_{n+1}(x_0+1)$
Si $n = 0$ et $k = 0$ c'est connu par 3a.
On a :

$$Ack_n^k(x_0+1) = Ack_n^{k-1}(Ack_{n-1}(Ack_n(x_0))) \leq Ack_n^{k-1}(Ack_n(Ack_n(x_0)))$$

En effet, $Ack_n(x_0) \geq Ack_n(1) \geq 2$ par le **Lemme 0 (a)** et on conclut par croissance de Ack en le premier argument (3a) et par croissance de Ack_n^{k-1} (récurrence sur k). Donc :

$$Ack_n^k(x_0+1) \leq Ack_n(Ack_n^k(x_0)) \leq Ack_n(Ack_{n+1}(x_0)) = Ack_{n+1}(x_0+1)$$

D'où le résultat.

■

Lemme 2 Soit $n \in \mathbb{N}^*$:

$$\forall k \in \mathbb{N} \ \exists x_{n,k} \in \mathbb{N} \quad Ack_n(x_{n,k}) > x_{n,k} + k + 1$$

Par récurrence sur k :

- $k = 0$: par le **Lemme 0 (b)** $x_{n,0} = 2$ convient.
- **Hérédité** ($n \neq 0$) : $Ack_n(x_{n,k}+1) = Ack_{n-1}(Ack_n(x_{n,k})) > Ack_{n-1}(x_{n,k}+k+1)$ par HR. On a alors :

$$\begin{aligned} Ack_n(x_{n,k}+1) > Ack_{n-1}(x_{n,k}+k+1) &> x_{n,k}+k+1 \\ Ack_n(x_{n,k}+1) &> x_{n,k}+k+2 \end{aligned}$$

$x_{n,k+1} = x_{n,k} + 1$ convient donc.

D'où le résultat.

■

On peut maintenant répondre à la question en démontrant :

$$\boxed{\forall n \in \mathbb{N}\ \forall k \in \mathbb{N}\ \exists C_{n,k} \in \mathbb{N} \quad \forall x \in \mathbb{N}\ Ack_n^k(x) \leq Ack_{n+1}(max(x, C_{n,k}))}$$

Soient $n, k \in \mathbb{N}$.

— Si $n \neq 0$, on se donne un $x_{n,k}$ du **Lemme 2**, d'après **3b** et le **Lemme 2** :

$$Ack_n^k(x_{n,k} + k + 1) < Ack_n^k(Ack_n(x_{n,k})) \leq Ack_{n+1}(x_{n,k} + k + 1)$$

On pose $C_{n,k} = x_{n,k} + k + 1 > 0$ En application du **Lemme 1** à $C_{n,k}$ on a :

$$\forall x \geq C_{n,k} \quad Ack_n^k(x) \leq Ack_{n+1}(x)$$

De plus, par croissance de Ack_n^k :

$$Ack_{n+1}(C_{n,k}) \geq Ack_n^k(C_{n,k}) \implies \forall x \leq C_{n,k}\ Ack_{n+1}(C_{n,k}) \geq Ack_n^k(x)$$

C'est ce qu'on voulait.

— Si $n = 0$ on ne peut pas se servir du **Lemme 2** mais on a par récurrence :

$$Ack_0^k(x) = x + 2k$$

Alors par récurrence sur k, on montre l'existence d'un $x_{0,k} \neq 0$ tel que :

$$x_{0,k} + 2k \leq Ack_1(x_{0,k})$$

Pour $k = 0$ tout $x \neq 0$ convient par (3a).
L'argument d'hérédité est le même que pour le lemme 2, $x_{0,k+1} = x_{0,k} + 1$ convient.
Muni de cela on peut conclure comme dans le cas $n \neq 0$ en prenant $C_{0,k} = x_{0,k}$.

D'où le résultat. (ouf ! !) ∎

(d) Montrer que si f PR est définie avec n utilisations du schéma de récurrence , alors Ack_{n+1} domine f.

(d) On procède par récurrence sur n. En montrant la propriété :

$\mathbf{P}(n)$: "$\forall f$ PR f construite avec n utilisations du SR $\implies \exists k \in \mathbb{N}\ Ack_n^k$ domine f"

— $n = 0$: on vérifie que $\odot$, s et les p_i^k sont dominées par Ack_1^1 (on utilise 3a).
Ensuite, par distinction de multiples cas on constate que f définie par schéma de composition avec h et $g_1 \ldots g_p$ des fonctions de base est dominée par Ack_1.

— **Hérédité** : On suppose la propriété vraie pour les fonctions définies à l'aide de $k \leq n$ utilisations du schéma de récurrence. Soit f définie à l'aide de $n+1$ utilisations, on peut supposer sans perte de généralité que :

$$f(a_1, \ldots, a_p, 0) = g(a_1, \ldots, a_p)$$
$$f(a_1, \ldots, a_p, x+1) = h(a_1, \ldots, a_p, x, f(a_1, \ldots, a_p, x))$$

Avec h et g PR définie avec au plus n utilisations du schémas de récurrence. En effet car si f est en étape finale contruite par schéma de composition on ramène l'étude aux fonctions qui interviennent.

Par HR on a l'existence de k_1, C_1 et k_2, C_2 tels que :

$$\forall a_1, \ldots, a_p \quad g(a_1, \ldots, a_p) \leq Ack_n^{k_1}(sup(a_1, \ldots, a_p, C_1))$$
$$\forall a_1, \ldots, a_p{+}1, a_{p+2} \quad h(a_1, \ldots, a_{p+1}, a_{p+2}) \leq Ack_n^{k_2}(sup(a_1, \ldots, a_{p+1}, a_{p+2}, C_2))$$

En effet si g ou h sont construits avec moins de n applications du SR, la croissance de Ack en le premier argument permet quand même d'établir l'inégalité.
Par récurrence sur x on montre qu'on a alors :

$$f(a_1, \ldots, a_p, x) \leq Ack_n^{k_1+xk_2}(sup(a_1, \ldots, a_p, x, C_1, C_2))$$

— x = 0 : $C_2 \leq C_1$, c'est connu. $C_2 > C_1$ par croissance.

— **Hérédité** :

$$\begin{aligned} f(a_1, \ldots, a_p, x+1) &\leq Ack_n^{k_2}(sup(a_1, \ldots, a_p, x, f(a_1, \ldots, a_p, x))) \\ &\leq Ack_n^{k_2}(sup(a_1, \ldots, a_p, x, Ack_n^{k_1+xk_2}(sup(a_1, \ldots, a_p, x, C_1, C_2)))) \end{aligned}$$

Or :

$$Ack_n^{k_1+xk_2}(sup(a_1, \ldots, a_p, x, C_1, C_2)) \geq Ack_n^{k_1+xk_2}(a_1) \geq a_1$$
$$\vdots$$
$$Ack_n^{k_1+xk_2}(sup(a_1, \ldots, a_p, x, C_1, C_2)) \geq Ack_n^{k_1+xk_2}(x) \geq x$$

Donc :

$$\begin{aligned} f(a_1, \ldots, a_p, x+1) &\leq Ack_n^{k_2}(Ack_n^{k_1+xk_2}(sup(a_1, \ldots, a_p, x, C_1, C_2))) \\ &\leq Ack_n^{k_1+(x+1)k_2}(sup(a_1, \ldots, a_p, x, C_1, C_2)) \end{aligned}$$

D'où le résultat.

D'où :

$$\begin{aligned}
f(a_1, \dots, a_p, x) &\leq Ack_n^{k_1+xk_2}(sup(a_1, \dots, a_p, x, C_1, C_2)) \\
&\leq Ack_{n+1}(sup(a_1, \dots, a_p, x, C_1, C_2) + k_1 + xk_2) \\
&\leq Ack_{n+1}(Ack_2^h(sup(a_1, \dots, a_p, x, C_3)) \\
&\leq Ack_{n+1}^{h+1}(sup(a_1, \dots, a_p, x, C_3))
\end{aligned}$$

Car on montre que *somme*, *sup* et $x \mapsto kx$ sont toutes dominées par Ack_2^h pour un certain h.
Donc f est dominée par Ack_{n+2} par ce qui précède.
D'où le résultat.

■

(e) Conclure.

(e) Supposons par l'absurde que Ack est PR. Alors : $f : n \mapsto Ack(n, n+1)$ est PR par composition. Supposons f construite avec m utilisations du SR.
On a montré précédemment qu'alors il existe C tel que $\forall n > C$:

$$Ack(n, n+1) \leq Ack_{m+1}(n)$$

Maintenant, si $n > m+1$ par 3a on a :

$$Ack_{m+1}(n) < Ack_{m+1}(n+1) \leq Ack_n(n+1) = Ack(n, n+1)$$

Absurde en prenant $n > sup(n, C)$.
D'où le résultat. ■

■

Question 4
On définit l'ensemble R des fonctions récursives comme le plus petit ensemble de fonctions, éventuellement partielle, qui contient PR et qui est clos par le *schéma de minimisation non-borné* : si $f : A \subseteq \mathbb{N}^{p+1} \to \mathbb{N}$ est dans R, alors la fonction $g : B \subseteq \mathbb{N}^p \to \mathbb{N}$ est dans R, où g est définie par :

$$g(a_1, \ldots, a_p) = min\,\{z : f(a_1, \ldots, a_p, z) = 0\} \text{ si non vide}$$

Quelle est la cardinalité de R ? Exhiber une fonction qui n'est pas dans R.

Si on pose :

$$\begin{aligned}
&R_0 = \{\odot,\, p_i^k,\, s\}\\
&R_{n+1} =\\
&R_n \cup (\cup_p \cup_m \{SC(h, g_1, \ldots, g_p) \mid h \in N_p \cap R_n \; g_{i\leq p} \in N_m \cap R_n\})\\
&\cup (\cup_p \{SR(g,h) \mid g \in N_p \cap R_n \; h \in N_{p+2} \cap R_n\})\\
&\cup (\cup_p \{SMNB(f) \mid f \in N_{p+1} \cap R_n\})
\end{aligned}$$

Avec N_p les fonctions (pouvant être partielles) de $\mathbb{N}^p \to \mathbb{N}$. Et SC, SR, $SMNB$ les différents schémas introduits. Alors d'après la théorie des ensembles inductifs (PROG) on sait que :

$$R = \cup_n R_n$$

Par récurrence on a que les R_n sont dénombrables par réunion dénombrable d'ensembles dénombrables ($N_p \cap R_n$ dénombrable).
Ainsi R est réunion dénombrable d'ensembles dénombrables : R est dénombrable.

Par argument de cardinalité on a donc l'existence de fonctions non R. On peut en exhiber une par argument diagonal.
On pose $R = \{f_n\}$ et :

$$g(n) = f_n(n) + 1$$

Supposons par l'absurde que g est R. Il existe m tel que $g = f_m$ et alors :

$$g(m) = f_m(m) + 1 \neq f_m(m)$$

Absurde.

■

Question 5 Montrer que si f est R, alors f est calculable par une machine de Turing.

On raisonne par induction structurelle (on prend des entrées en binaire, machine à deux rubans I/O) :

— **Cas** $\odot$: une machine qui écrit 0 quelque soit l'entrée.

— **Cas** p_i^k : on sépare les différents paramètre d'un espace sur le ruban, il suffit de maintenir un compteur pour atteindre i et recopier l'entrée sur la sortie.

— **Cas** s : on sait ajouter $+1$ avec une MT en binaire (cf TD).

— **Cas SC** : soit $h \in N_p \cap R$ et $g_1, \ldots, g_p \in N_n \cap R$. On considère $f = SC(h, g_1, \ldots, g_p)$. Par hypothèse d'induction, on dispose de M_0 qui calcule h et $M_1, \ldots, M_p$ qui simulent $g_1, \ldots, g_p$. Il suffit de composer ces machines en faisant écrire les résultats de $M_1, \ldots, M_p$ (sur $x_1, \ldots x_n$) séparés d'un espace sur le ruban d'entrée de M_0 et appliquer M_0.

— **Cas SR** : Soit $f = SR(g, h)$, $M_0 = M(f)$ et $M_1 = M(h)$. Pour calculer f il faut simuler la stack comme sur un ordi, on empile sur le ruban les arguments successifs jusqu'à $x = 0$ auquel cas on calcul avec M_0 et après on remonte tant que le ruban n'est pas vide en appliquant M_1 et en utilisant le résultat comme paramètre manquant de l'appel d'après.

— **Cas SMNB** : Soit $g = SMNB(f)$ et $M_0 = M(f)$. On interprète la non définition d'une fonction en un point comme la non terminaison sur cette entrée de la machine qui calcule f. Muni de cette définition on construit une machine qui teste tous les z dans l'ordre croissant en utilisant le calcul de M_0. Si un tel z existe on l'écrit sur le ruban sinon le calcul ne termine pas : g n'est pas définie.

■

Question 6 On s'intéresse maintenant à la réciproque : soit f calculée par une machine de Turing M, qu'on suppose déterministe, avec un seul état d'acceptation q^+,qui ne bloque jamais, avec un seul ruban bi-infini.

(a) On suppose que les états de M sont les entiers $\{0, \ldots, n-1\}$ et que les symboles de bande sont les entiers $\{0, \ldots, m-1\}$, le symbole blanc étant l'entier 0. Proposer un codage d'un bout de ruban semi-infini (à support fini), ie à $u = u_0u_1u_2...$ ou $u = ...u_2u_1u_0$ on associe $< u > \in \mathbb{N}$ tel que $< . >$ soit injective.

a) Comme les u_i sont bornés par m on va prendre le codage en base m (on ne différencie pas semi-inf gauche de droite quitte à retourner). Si $u = u_0 \ldots u_p$ on pose :

$$< u > = \sum_{k=0}^{p} u_k m^k$$

C'est injectif.

■

(b) On considère : $F: \ < u > \mapsto u_0$. Montrer que F est PR.

b) On a :

$$\boxed{F(< u >) = mod(div(< u >, 2), m)}$$

$$\boxed{\begin{aligned} div(0, m) &= 0 \\ div(n+1, m) &= div(n, m) + eq(m * (1 + div(n, m)), n+1) \end{aligned}}$$

$$\boxed{mod(n, m) = n - m * div(n, m)}$$

Par récurrence on montre que *div* et *mod* correspondent bien à la division entière et au modulo. Elles sont PR car on peut les réécrire sous une forme explicitement PR à l'aide de *prod*, *somme*, *sub* et des schémas PR.
Alors F est bien PR (on peut la réécrire explicitement PR en utilisant c_2 et c_m.

■

(c) Soit $(q, a, q', b, d) \in \delta(M)$. Montrer que $\delta State(q, a) = q'$, $\delta Write(q, a) = b$ et $\delta Move(q, a) = d$ sont PR.

c) Pour être PR on a besoin de fonctions totales. On prend la convention que les transitions invalides entraînent la nullité de la fonction considérée.
Alors $\delta State$, $\delta Write$ et $\delta Move$ sont à support fini et donc PR d'après **1g)**.

■

(*d*) Montrer que les fonctions suivantes peuvent être définies par récurrence mutuelle : $State(< w >, t)$,$Left(< w >, t)$,$Right(< w >, t)$. Qui donnent respectivement l'état de la machine, le contenu à gauche (exclu) de la tête et à droite (inclus) après t étapes de calcul.

d) On suppose que le mot <w> est donné initialement en forme semi-ruban droit.

$$State(< w >, 0) = 0$$
$$Left(< w >, 0) = 0$$
$$Right(< w >, 0) =< w >$$

$q_t = State(< w >, t) \quad u_t = F(Right(< w >, t))$

$$State(< w >, t+1) = \delta State(q_t, u_t)$$

Si $\delta Move(q_t, u_t) = R$:

$$Left(< w >, t+1) = 2 * (m\frac{Left(< w >, t)}{2} + \delta Write(q_t, u_t))$$
$$Right(< w >, t+1) = 2 * \frac{\frac{Right(<w>,t)}{2} - u_t}{m}$$

Si $\delta Move(q_t, u_t) = L$:

$$Left(< w >, t+1) = 2 * \frac{\frac{Left(<w>,t)}{2} - u_t}{m}$$
$$Right(< w >, t+1) = 2 * (m\frac{Right(< w >, t)}{2} + \delta Write(q_t, u_t))$$

Ce qu'on a fait ici c'est d'attribuer les codages correspondant à chaque mouvement.

■

(e) Conclure $f \in R$.

e) On suppose qu'avant d'arriver en q^+ la machine a remis la tête de lecture au début du mot, quitte à le faire nous même en q^+ et à le connecter à un nouvel état q^{++}.
Alors on a :

$$f(w) = Right(< w >, min\{t : eq0(eq(State(< w >, t), q^+)) = 0\})$$

On assimile ici le codage d'entrée de f, pour passer de $n \in \mathbb{N}$ à un mot en entrée de M, à cet l'entier n. De même $Right(...)$ est le codage d'un mot α par $<>$, on l'assimile à la valeur que représente ce α selon le codage d'entrée. Ainsi l'expression a du sens.

$State$, $Left$ et $Right$ sont PR : on peut rendre le schéma de récurrence mutuelle en schéma de récurrence simple en regroupant ces 3 fonctions dans une seule avec un paramètre en plus qui raccorde à la fonction voulue.
C'est aussi comme ça qu'on peut gérer la condition L/R avec un paramètre en plus et en utilisant $eq0$ pour sélectionner l'expression qu'on désire. Après toutes les fonctions utilisées par composition sont PR ($prod$, div, s, F, $\delta State$, $\delta Write$ et $\delta Move$).

Enfin on recourt à une utilisation couplée du schéma de composition et de minimisation non bornée. Tout cela est donc R.
Donc $f \in R$.

■

Question 7
(e) En déduire que toute fonction de R peut s'exprimer à l'aide d'au plus un schéma de minimisation non-bornée.

Si f est R, avec **5** on construit une machine de Turing M qui calcule f. Et avec 6 on ré-exprime f avec une seule occurence du SMNB (voir la construction).
D'où le résultat.

■

Exercice 3

On s'intéresse aux formules de l'arithmétique de Presburger, ie les formules qui quantifient sur des entiers et qui utilisent comme prédicat l'addition et l'égalité. Étant donnée une formule de l'arithmétique de Presburger, on s'intéresse à la vérité d'une telle formule. On va montrer que cette propriété est décidable

Question 1 Montrer que pour toute formule ϕ, il existe une formule ψ qui est vraie si et seulement si ϕ est vraie et qui est sous forme prénexe, ie $\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \hat{\psi}$ avec Q_i des quantificateurs et $\hat{\psi}$ une formule sans quantificateur.

On le démontre par récurrence sur le nombre n de variables liées dans la formule (i.e : variable assujetie à un quantificateur). On s'autorise des variables libres dans la formule.

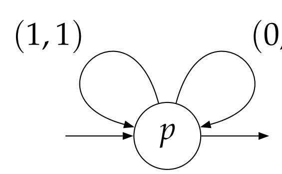
- $n = 0$: l'absence de variable liée implique l'absence de quantificateur et donc ϕ est sous forme prénexe.
- **Hérédité** : $\phi = \phi' c(Q_1 x_1 \phi'')$ avec ϕ' ne possédant pas de variable liée, c un connecteur binaire (ϕ' peut être vide et c inexistant) et ϕ'' en possédant n. Comme aucune variable liée n'apparaît dans ϕ' (en particulier x_1), ϕ est équivalente à $Q_1 x_1(\phi' c \phi'')$. $(\phi' c \phi'')$ possède toujours n variables liées, par HR elle est équivalente à ϕ_1 en forme prénexe.
  Finalement ϕ est équivalente à $Q_1 x_1 \phi_1$ qui est sous forme prénexe.

■

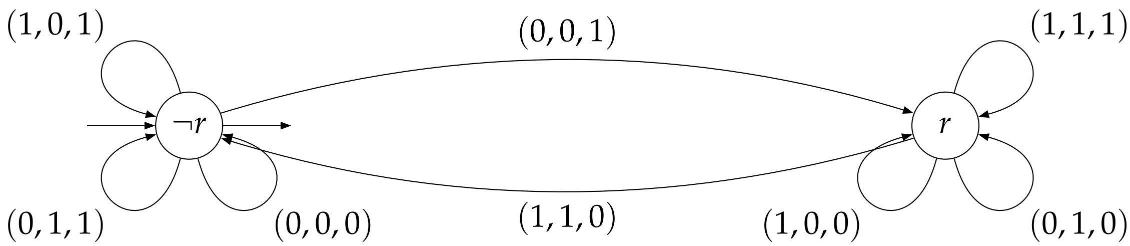
Question 2 On considère un codage usuel des entiers en binaire avec le bit de poids fort à gauche, avec éventuellement des zéros en tête pour ajuster les longueurs. On se placera donc sur un alphabet $\sum = 0, 1^p$, pour p que l'on fixera en fonction de la formule à décider. Montrer que les langages suivant sont rationnels :

$$L_+ = \{(n_1, n_2, n_3) \mid n_1 + n_2 = n_3\}$$
$$L_= = \{(n_1, n_2) \mid n_1 = n_2\}$$

En prenant comme alphabet $\{0,1\}^2$ et en représentant 2 nombres comme les couples successifs de leurs bits on a pour $L_=$ l'automate :



Et pour L_+ en utilisant des triplets :



Cet automate vérifie l'addition "à l'envers". L'état c correspondant à la retenue et l'autre à l'absence de retenue.

■

Question 3 En déduire que pour tout ϕ formule sans quantificateurs avec p variables libres l'ensemble suivant est rationnel :

$$L_\phi = \{(n_1, n_2, \dots, n_p) \mid \phi(n_1, n_2, \dots, n_p) \text{ vraie}\}$$

On a vu que les composants atomiques de ϕ donnent des langages rationnels. On va récupérer L_ϕ par intersection, union ou complémentation des sous langages correspondant à ces formes atomiques (on peut les étendre à p variables en ignorant les $p-2$ ou $p-3$ composantes qui ne servent à rien). Ça reste rationnel.

■

Question 4 Étant donnée une formule ψ sous forme prénexe $\psi = Q_1x_1Q_2x_2 \dots Q_nx_n\hat{\psi}$, on définit $\psi_k = Q_{k+1}x_{k+1} \dots Q_nx_n\hat{\psi}$. Montrer par récurrence que les langages suivant sont rationnels :

$$X_k = \{(x_1, x_2, \dots, x_k) \mid \psi_k(x_1, x_2, \dots, x_k) \text{ vraie}\}$$

On suppose ici que ψ ne possède pas de variables libres (elle est close).
Par définition ψ_k possède k variables libres. On procède par récurrence descendante. Par convention : $\psi_0 = \psi$ et $\psi_n = \hat{\psi}$.

— $k = n$: $X_n = \{(x_1, x_2, \dots, x_n) \mid \hat{\psi}(x_1, x_2, \dots, x_n) \text{ vraie}\}$ est rationnel par **Question 3** puisque $\hat{\psi}$ est sans quantificateur.

— **Hérédité** : Soit $0 \leq k < n$ on suppose X_{k+1} rationnel.
On a donc un AFD $\mathcal{A}_{k+1}$ sur l'alphabet $\{0,1\}^{k+1}$ qui reconnait X_{k+1}.
On va construire $\mathcal{A}_k$ sur $\{0,1\}^k$ qui reconnait X_k.
$\psi_k = Q_{k+1}x_{k+1}\psi_{k+1}$. Deux cas :

— $Q_{k+1} = \exists$:
La formule devient : $\psi_k = \exists x_{k+1} \psi_{k+1}$. Et alors si pour $\mathcal{A}_k$ on prend $\mathcal{A}_{k+1}$ où l'on supprime la dernière composante de chaque transition, on trouve un automate non déterministe où l'acceptance d'un mot $(x_1, \ldots, x_k)$ revient à l'existence d'un x_{k+1} (formé par un certain choix de la composante qu'on a enlevé sur chaque transition) tel que $\mathcal{A}_{k+1}$ accepte $(x_1, \ldots, x_{k+1})$. La réciproque est vraie, si $\mathcal{A}_{k+1}$ accepte $(x_1, \ldots, x_{k+1})$ en prenant le même chemin, $\mathcal{A}_k$ accepte $(x_1, \ldots, x_k)$. $\mathcal{A}_k$ convient donc et X_k est rationnel.

— $Q_{k+1} = \forall$:
On se ramène au cas précédent par équivalence de ψ_k avec $\neg \exists x_{k+1} \neg \psi_{k+1}$. On conclut par stabilité des langages rationnels par passage au complémentaire.

D'où le résultat. ∎

Question 5 Conclure en considérant X_0.

On a $\psi_0 = \psi$.

— Si $\psi = \forall x_1 \psi_1(x_1)$ on doit vérifier que $X_1 = \mathcal{L}(\mathcal{A}_1) = \{0,1\}^*$

— Si $\psi = \exists x_1 \psi_1(x_1)$ on doit vérifier que $X_1 = \mathcal{L}(\mathcal{A}_1) \neq \emptyset$

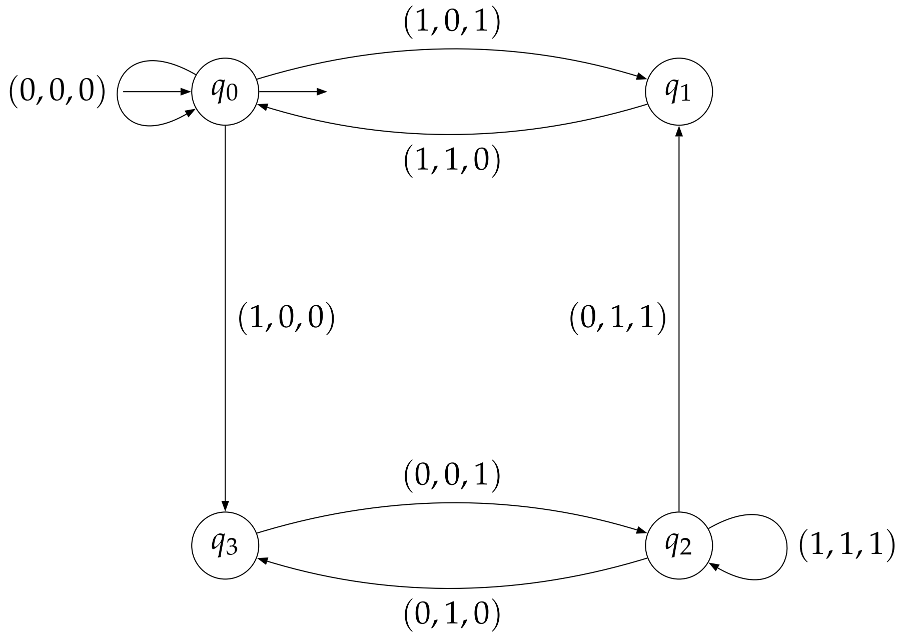
Ces deux problèmes sont décidables.

On peut aussi conclure avec $\mathcal{A}_0$ associé à X_0 : $\mathcal{A}_0$ est un graphe orienté.
Par la construction des $\mathcal{A}_k$ il suffit désormais de tester l'accessibilité d'un état final à partir d'un état initial (on conclut en fonction de Q_1) ce qui est décidable.
D'où le résultat. ∎

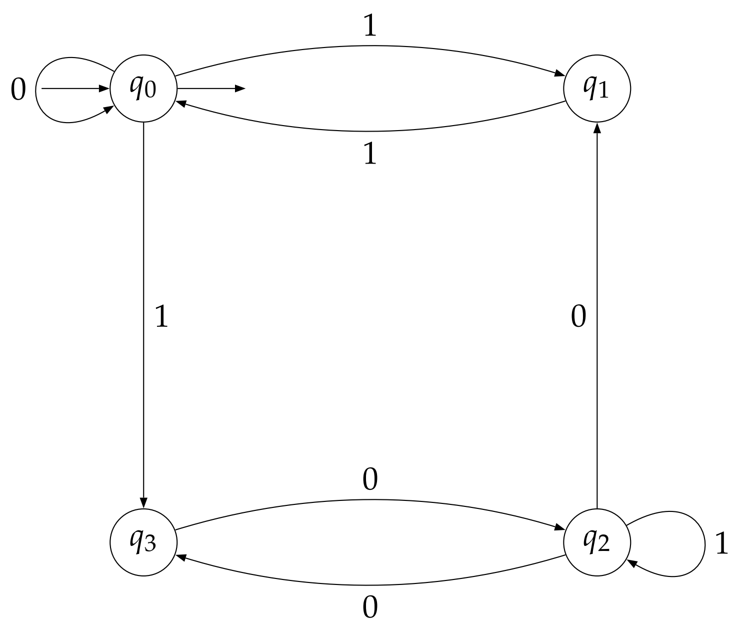
Question 6 Faites tourner l'algorithme sur les formules :

$$\psi_0(x) \equiv \exists y, \exists z, x = y + z \wedge z = y + y$$
$$\psi_1 \equiv \forall x, \psi_0(x)$$

La première formule est équivalente à l'équation $x \equiv 0 \ mod\ 3$, on veut donc montrer que ψ_1 est fausse.
Pour $x = y + z \wedge z = y + y$ on propose l'automate suivant :



On utilise l'agorithme pour trouver l'automate associé à X_1 (défini plus haut) :



On se rend compte que la formule est fausse puisque 1 n'est pas dans le langage de cet automate alors que ψ_0 a le quantificateur $\forall$ (voir première conclusion **Question 5**). D'où le résultat. ■