Formules qui représentent (ou non) la suite des nombres premiers

Réalisé par :
BALBAL Chadi

Encadré par :
RIVOAL Tanguy

Table des matières

Notations et définitions

On trouvera ci-dessous des notations et des définitions utilisées dans le texte.

1. On note par $a \mid b$ que a divise b, et par $a \nmid b$ que a ne divise pas b.
2. La définition d'une **machine de Turing** est donnée en section 6.2.
3. La définition d'un ensemble **récursif** ou **décidable** est donnée en section 6.5.
4. La définition d'un ensemble **récursivement énumérable** ou **semi-décidable** est donnée en section 6.3.
5. **Une équation diophantienne** est une équation de la forme $D(x_1, \ldots, x_m) = 0$, où D est un polynôme à coefficients entiers relatifs.
6. Un ensemble A des n-uplets est **diophantien** si et seulement si :

$$(a_1, \ldots, a_n) \in A \Leftrightarrow \exists x_1, \ldots x_m \; [D(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0], \tag{1}$$

   où D est un polynôme à coefficients entiers relatifs en les variables $a_1, \ldots, a_n, x_1, \ldots, x_m$, qui se répartissent en *paramètres* $a_1, \ldots, a_n$ et *inconnues* $x_1, \ldots, x_m$. L'équivalence (1) est appelée une **représentation diophantienne** de l'ensemble A.
7. Une **fonction algébrique** d'indéterminées $x_1, \ldots, x_m$ est une fonction F qui satisfait l'équation non triviale $P(F, x_1, \ldots, x_m) = 0$, où P est un polynôme irréductible non identiquement nul à $n + 1$ variables sur un corps commutatif K.
8. Une **fonction transcendante** est une fonction qui n'est pas algébrique.
9. Pour tous entiers non nuls a et b, l'entier $a \wedge b$ est le plus grand entier naturel qui les divise tous les deux.

1 Introduction et objectif

La suite des nombres premiers $(p_n)_{n\geq 1}$ joue un rôle central en mathématiques, et pourtant, il n'existe pas de "formules simples" permettant d'engendrer ces nombres. Bien que certaines formules, comme celles construites à partir du crible d'Ératosthène ou du théorème de Wilson, permettent de calculer p_n pour toute valeur de n, elles ne fournissent pas d'information facilement utilisable pour déterminer leur comportement asymptotique, qui est décrit par le théorème des nombres premiers à savoir que $p_n \sim n\ln(n)$. De plus, elles ne sont pas utilisables en pratique pour de grandes valeurs de n.

En 1976, Jones, Sato, Wada et Wiens ont démontré le résultat remarquable suivant.

Théorème 1.1. *L'ensemble des valeurs strictement positives que prend le polynôme ci-dessous, lorsque ses variables varient sur les entiers naturels, correspond exactement à l'ensemble de tous les nombres premiers :*

$$
\begin{aligned}
&(k+2)\Big\{1-[wz+h+j-q]^2-[(x+cu)^2-((a+u^2(u^2-a))^2-1)(n+4dy)^2-1]^2\\
&-[z-(gk+2g+k+1)(h+j)-h]^2-[m^2-(a^2-1)\ell^2-1]^2\\
&-[16(k+1)^3(k+2)(n+1)^2+1-f^2]^2-[ai+k+1-1-\ell-i]^2\\
&-[e-p-q-z-2n]^2-[n+\ell+v-y]^2-[e^3(e+2)(a+1)^2+1-o^2]^2\\
&-[m-p-\ell(a-n-1)-b(2a(n+1)-(n+1)^2-1)]^2-[x^2-(a^2-1)y^2-1]^2\\
&-[x-q-y(a-p-1)-s(2a(p+1)-(p+1)^2-1)]^2\\
&-[u^2-16(a^2-1)r^2y^4-1]^2-[pm-z-p\ell(a-p)-t(2ap-p^2-1)]^2\Big\}.
\end{aligned}
$$

(On précise que l'expression [...] ne représente pas la partie entière, mais uniquement des parenthèses ordinaires, et de même pour {...} qui ne représente pas la partie fractionnaire).

On précise tout d'abord qu'une valeur strictement positive prise par ce polynôme est exactement le facteur $(k+2)$ lorsque le deuxième facteur du polynôme est égal à 1. Notons le fait paradoxal que personne n'a jamais observé un nombre premier produit par ce polynôme. En effet, nous verrons dans ce mémoire que pour obtenir un nombre premier, quelques variables du polynôme doivent prendre des valeurs très grandes.

De plus, ce théorème est une des conséquences des travaux qui ont établi la réponse négative au dixième problème de Hilbert. Ce problème, énoncé en 1900, porte sur les équations diophantiennes et peut être formulé ainsi :

"Existe-t-il un algorithme qui, recevant en entrée une équation diophantienne à coefficients entiers naturels, détermine en un nombre fini d'opérations si cette équation admet une solution en nombres entiers ?"

La notion rigoureuse d'algorithme n'a été développée qu'au cours des années 1930, notamment grâce aux travaux de Gödel, Church et Turing dans le domaine de la calculabilité. Ces avancées ont permis de démontrer l'inexistence d'algorithmes pour résoudre certains problèmes bien définis. En particulier, Alan Turing a montré l'existence d'ensembles récursivement énumérables non décidables.

À partir des années 1950, Martin Davis, Hilary Putnam et Julia Robinson ont montré que tout ensemble récursivement énumérable pouvait être exprimé à l'aide d'équations diophantiennes incluant des exponentielles, établissant ainsi que ces ensembles sont exponentiellement diophantiens. Julia Robinson, dans ses recherches visant à démontrer que la fonction exponentielle est diophantienne, s'est intéressée aux solutions de l'équation de Pell. Elle a alors formulé une hypothèse centrale : il existe une relation diophantienne à croissance exponentielle. Elle a par ailleurs démontré que si une telle relation existe, cela impliquerait que l'exponentiation est effectivement diophantienne. Il ne restait alors qu'à exhiber une relation diophantienne à croissance exponentielle pour montrer que tout ensemble récursivement énumérable est diophantien, ce qui permettait de conclure à l'équivalence des deux classes et de répondre négativement au dixième problème de Hilbert. En 1970, Youri Matiiassevitch a apporté la pièce manquante en prouvant que la suite de Fibonacci constitue une telle relation. Cette découverte a permis de clore le dixième problème de Hilbert en montrant qu'il n'existe pas d'algorithme permettant de déterminer si une équation diophantienne à coefficients entiers donnée admet ou non des solutions entières.

L'objectif de ce mémoire est de présenter en détail les résultats et démonstrations exposés dans [3], en vue d'établir le théorème 1.1. Pour ce faire, le mémoire sera structuré comme suit. La section 2 est consacrée à quelques préliminaires, dont certains seront utilisés par la suite. On y présentera quelques formules exprimant explicitement les nombres premiers, notamment une formule récursive issue du théorème de Wilson et une formule donnée par Legendre et inspirée du crible d'Ératosthène. On y montrera également qu'il est impossible qu'un polynôme non constant prenne uniquement des valeurs premières pour les entiers positifs, et cela même pour une fonction algébrique. Dans la section 3, on s'intéressera aux solutions des équations de Pell, qui forment une suite à croissance exponentielle. Cette section a pour but d'établir le caractère diophantien de ces suites, ce qui jouera un rôle central par la suite. La section 4 portera sur une représentation de la fonction factorielle en termes d'exponentiation. Cela a pour conséquence que si l'exponentiation admet une représentation diophantienne, alors il en va de même pour la factorielle. Dans la section 5, on combinera les résultats des sections 3 et 4, et l'on utilisera le théorème de Wilson (qui caractérise les nombres premiers par la factorielle) pour construire une représentation diophantienne des nombres premiers, telle que formulée dans le théorème 1.1. Enfin, la section 6 sera consacrée au dixième problème de Hilbert. On introduira d'abord la notion de codage et des machines de Turing, permettant de formaliser rigoureusement la notion d'algorithme. Cela nous mènera aux définitions des ensembles décidables et semi-décidables. On montrera ensuite que tout ensemble diophantien est semi-décidable, en traduisant l'idée intuitive selon laquelle, pour savoir si un polynôme s'annule sur les entiers naturels, il suffit de tester successivement tous les entiers, sans garantie d'arrêt. On présentera enfin la démonstration (éventuellement avec certains résultats admis pour des raisons de longueur) du fait que tout ensemble récursivement énumérable est diophantien, ce qui constitue la clé de la réponse négative au dixième problème de Hilbert. C'est à ce stade que l'on fera intervenir le caractère diophantien de l'exponentiation.

2 Représentations des nombres premiers

Le théorème fondamental pour la construction de notre polynôme est le théorème de Wilson suivant.

Théorème 2.1 (Théorème de Wilson)**.** *Quel que soit* $k \in \mathbb{N}$*, on a que*

$$k+1 \text{ est premier } \Leftrightarrow \ (k+1) \mid (k!+1).$$

Démonstration. Supposons que $(k+1) \mid (k!+1)$, on a

$$\forall\ 1 < n \leq k \quad k!+1 \equiv 1 \mod (n),$$

donc

$$\forall\ 1 < n \leq k \quad n \nmid k!+1.$$

Il en découle que

$$\forall\ 1 < n \leq k \quad n \nmid k+1,$$

et donc que $k+1$ est premier.

Montrons la réciproque et supposons maintenant que $k+1$ est premier. Comme $\mathbb{Z}/(k+1)\mathbb{Z}$ est un corps, on a que

$$\forall\ a \in \{\overline{2}, \ldots, \overline{k}\},\ \exists\ b \in \{\overline{2}, \ldots, \overline{k}\} \quad \text{tel que} \quad \overline{a}\overline{b} = \overline{1} \quad \text{dans } \mathbb{Z}/(k+1)\mathbb{Z}.$$

Or, on a que si $\overline{a}^2 = 1$, alors $(\overline{a}-1)(\overline{a}+1) = \overline{0}$, donc $\overline{a} = 1$ ou $\overline{a} = -\overline{1} = \overline{k}$. Alors

$$\overline{k!} = \overline{k}\ \overline{(k-1)!} = -\overline{1},$$

donc $\overline{k!+1} = \overline{0}$. Cela signifie que $k+1$ divise $k!+1$. □

Ce théorème nous permet de construire une formule pour le n-ième nombre premier suivante, que l'on trouve dans [6].

On définit la suite des nombres premiers $(p_n)_{n\geq 1}$ avec $p_1 = 2,\ p_2 = 3 \ldots$.

Proposition 2.2. *Le n-ième nombre premier est donné par la formule suivante :*

$$p_n = \sum_{i=0}^{n^2} \left[1 \dot{-} \left(\left(\sum_{j=0}^{i} r\left((j \dot{-} 1)!^2, j \right) \right) \dot{-} n \right) \right],$$

où $y \dot{-} x = y - x$ *si* $y \geq x$ *et 0 si* $y < x$*, et* $r(a,b)$ *est le reste de la division euclidienne de* a *par* b*.*

Démonstration. D'après le théorème de Wilson, si j est premier alors

$$(j-1)! \equiv -1 \mod (j).$$

De plus, si j n'est pas premier, alors, soit $j = 4$, et $(j-1)! \equiv 2 \mod (j)$, soit $j > 4$, et il existe $p > 2, q \geq 2$ tel que $j = pq$.

Si $p \neq q$, $j = pq \mid (j-1)!$ et si $p = q$, on a $2p \in \{1, \ldots j-1\}$ et $2p \neq p$, alors $j \mid 2p.p \mid (j-1)!$, alors si $j \geq 5$ non premier, on a $(j-1)! \equiv 0 \mod (j)$.

On conclut alors que

$$r((j-1)!^2, j) = \begin{cases} 1 & \text{si } j \text{ est premier}\,; \\ 0 & \text{sinon.} \end{cases}$$

Le nombre de nombres premiers inférieurs ou égaux à i avec $i > 0$ est donné par

$$\pi(i) = \sum_{j=1}^{i} r((j-1)!^2, j) \quad et\ \pi(0) = 0.$$

On pose $c(a,n) = 1 \dot{-} ((1+a) \dot{-} n)$. Cette fonction caractérise la relation $<$ sur les entiers, puisque

$$c(a,n) = \begin{cases} 1 & \text{si } a < n; \\ 0 & \text{sinon.} \end{cases}$$

On a $\pi(i) < n \Leftrightarrow i < p_n$, donc $c(\pi(i), n) = \begin{cases} 1 & \text{si } i < p_n; \\ 0 & \text{sinon.} \end{cases}$

Donc

$$p_n = \sum_{i=0}^{k} c(\pi(i), n) \quad \text{avec } k \geq p_n - 1.$$

D'après le résultat de J. Barkley Rosser et L. Schoenfeld dans [8] selon lequel $p_n < n(\log(n) + \log(\log(n))$ pour $n > 5$, on peut prendre $k = n^2$. On obtient ainsi

$$p_n = \sum_{i=0}^{n^2} 1 \dot{-} \left(\left(1 + \sum_{j=1}^{i} r\big((j-1)!^2, j\big) \right) \dot{-} n \right).$$

Comme $r((0)!^2, 0) = 1$, alors

$$p_n = \sum_{i=0}^{n^2} \left[1 \dot{-} \left(\left(\sum_{j=0}^{i} r\big((j \dot{-} 1)!^2, j\big) \right) \dot{-} n \right) \right].$$

□

On peut construire une autre formule traduisant le crible d'Ératosthène, qui donne le nombre de nombres premiers inférieurs ou égaux à un réel x noté $\pi(x)$, et qui a été donnée par Legendre en 1808.

Description du crible d'Ératosthène

Le crible d'Ératosthène est une méthode permettant de trouver tous les nombres premiers inférieurs à un entier N en éliminant les multiples des entiers successifs. On supprime

d'abord les multiples de 2 sauf 2, puis ceux de 3 sauf 3 qui est donc premier. Le nombre 4 est déjà éliminé, donc le nombre 5 est premier et on élimine alors ses multiples sauf lui-même. On s'arrête lorsque le carré du plus petit nombre restant dépasse N, car tout nombre composé $n \leq N$ possède au moins un diviseur premier p tel que $p \leq \sqrt{N}$. En effet, si n était composé uniquement de facteurs premiers supérieurs à $\sqrt{N}$, alors il serait strictement supérieur à N, ce qui est impossible. Ainsi, tous les nombres non premiers auront déjà été éliminés à ce stade. À la fin, les nombres $\leq N$ qui n'ont pas été supprimés sont exactement les nombres premiers inférieurs ou égaux à N. La formule du crible d'Ératosthène-Legendre traduit cette méthode.

Proposition 2.3. *Le nombre de nombres premiers inférieurs ou égaux à x est donné par*

$$\pi(x) - \pi(\sqrt{x}) = -1 + \sum_{d} \mu(d) \left\lfloor \frac{x}{d} \right\rfloor,$$

où la somme est prise sur tous les diviseurs d du produit $p_1 p_2 \dots p_n$, où $p_1, \dots, p_n$ sont les nombres premiers inférieurs ou égaux à $\sqrt{x}$. La fonction de Möbius $\mu(k)$ est définie de la manière suivante : elle vaut 0 si k est divisible par le carré d'un entier, et $(-1)^r$ si k peut s'écrire comme le produit de r nombres premiers distincts. $\lfloor \cdot \rfloor$ désigne la partie entière.

Pour démontrer cette formule, on utilisera le principe d'inclusion-exclusion suivant.

Lemme 2.4. *Soient $A_1, A_2, \dots, A_n$ n ensembles finis. On a*

$$\mathrm{Card}\Big(\bigcup_{i=1}^{n} A_i\Big) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} \mathrm{Card}\big(A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_k}\big).$$

Démonstration. On montre le résultat par récurrence. Pour $n = 2$, on a $\mathrm{Card}(A_1 \cup A_2) = \mathrm{Card}(A_1) + \mathrm{Card}(A_2) - \mathrm{Card}(A_1 \cap A_2)$. Supposons maintenant que le résultat est vrai pour un certain n, on a

$$\mathrm{Card}\Big(\bigcup_{i=1}^{n+1} A_i\Big) = \mathrm{Card}\Big(\bigcup_{i=1}^{n} A_i\Big) + \mathrm{Card}\Big(A_{n+1}\Big) - \mathrm{Card}\Big(\big(\bigcup_{i=1}^{n} A_i\big) \cap A_{n+1}\Big) \tag{2}$$

$$= \mathrm{Card}\Big(\bigcup_{i=1}^{n} A_i\Big) + \mathrm{Card}\Big(A_{n+1}\Big) - \mathrm{Card}\Big(\bigcup_{i=1}^{n} \big(A_i \cap A_{n+1}\big)\Big). \tag{3}$$

On obtient ainsi le résultat par hypothèse de récurrence. □

On peut maintenant démontrer la proposition 2.3.

Démonstration. Soit k un entier naturel et soit x un nombre réel positif. Le nombre $\lfloor \frac{x}{k} \rfloor$ donne le nombre de multiples de k entre 1 et x. Soient maintenant $p_1, \dots, p_n$ des nombres premiers et x un entier naturel. On note pour tout $1 \leq i \leq n$, A_i l'ensemble des multiples de p_i entre 1 et x. On a alors, le nombre d'entiers inférieurs ou égaux à x et qui sont divisibles

par au moins un des p_i est donné par $\mathrm{Card}(\bigcup_{i=1}^n A_i)$. Ainsi, en utilisant le lemme 2.4, on a que le nombre d'entiers inférieurs ou égaux à x qui ne sont divisibles par aucun p_i est

$$x - \mathrm{Card}\Big(\bigcup_{i=1}^{n} A_i\Big) = x - \sum_{k=1}^{n}(-1)^{k-1} \sum_{1\le i_1<i_2<\ldots<i_k\le n} \mathrm{Card}\big(A_{i_1}\cap A_{i_2}\cap\ldots\cap A_{i_k}\big) \tag{4}$$

$$= x - \sum_i \Big\lfloor \frac{x}{p_i} \Big\rfloor + \sum_{i<j} \Big\lfloor \frac{x}{p_i p_j} \Big\rfloor - \sum_{i<j<k} \Big\lfloor \frac{x}{p_i p_j p_k} \Big\rfloor + \ldots, \tag{5}$$

où $p_1, \ldots, p_n$ sont les nombres inférieurs ou égaux à $\sqrt{x}$. Le nombre $x - \mathrm{Card}\big(\bigcup_{i=1}^n A_i\big)$ donne le nombre des nombres premiers $\sqrt{x} < p \le x$ en plus du nombre 1 puisqu'il n'est divisible par aucun des p_i. Alors $\pi(x) - \pi(\sqrt{x}) = -1 + \sum_d \mu(d)\big\lfloor \frac{x}{d} \big\rfloor$ où d prend les valeurs des diviseurs du nombre $p_1 p_2 \ldots p_n$ puisque l'expression $\mu(d)\big\lfloor \frac{x}{d} \big\rfloor$ est nulle sauf si d est de la forme $p_{i_1} p_{i_2} \ldots p_{i_k}$ avec les p_{i_j} sont deux à deux distincts. □

Remarque 2.5. *Bien que le crible d'Ératosthène soit une méthode ancienne pour identifier les nombres premiers, il reste un outil utile en mathématiques. Il y a même des améliorations très récentes, comme celles décrites dans l'article [9], qui renforcent encore son efficacité.*

La relation $p_n = m$ est récursive, puisqu'on peut construire un algorithme calculant p_n. En effet, la formule donnée par la proposition 2.2 n'utilise que des opérations réalisables par une machine de Turing, comme nous le verrons à la fin. Or, tout ensemble récursif est récursivement énumérable et, d'après le résultat négatif du dixième problème de Hilbert, la classe des ensembles diophantiens coïncide avec celle des ensembles récursivement énumérables. Il en résulte que l'ensemble des nombres premiers est diophantien. Alors, il existe $Q \in \mathbb{Z}[X_1 \ldots X_m]$ tel que $p_n = m \Leftrightarrow \exists x_1, \ldots, x_l \in \mathbb{N}$ tel que $Q(n, m, x_1, \ldots, x_l) = 0$. On pose alors $P = x_{l+1}(1 - Q^2(n, x_{l+1}, x_1, \ldots, x_l))$ et ce serait bien le polynôme qu'on cherche.

On remarque que P peut prendre des valeurs négatives non premières, et cela est dû aux résultats suivants qui montrent l'impossibilité d'avoir une fonction algébrique de plusieurs variables qui représente seulement les nombres premiers. Cela montre la nécessité d'utiliser une fonction transcendante si on veut y parvenir. Dans [7], Julia Robinson a remarqué que si on prend le polynôme M défini par : $k + 2$ est premier si et seulement s'il existe $x_1, \ldots, x_l \in \mathbb{N}$ tel que $M(k, x_1, \ldots, x_l) = 0$, alors l'ensemble des nombres premiers est exactement l'image de la fonction $2 + k \times 0^{M(k, x_1, \ldots, x_l)}$ avec la convention $0^0 = 1$.

Proposition 2.6. *Si un polynôme $P \in \mathbb{C}[X_1, \ldots, X_k]$ est tel que, pour tout $(n_1, \ldots, n_k) \in \mathbb{N}^k$, $P(n_1, \ldots, n_k)$ est premier, alors P est constant.*

Démonstration. Soit $P \in \mathbb{C}[X_1, \ldots, X_k]$ vérifiant l'hypothèse de la proposition. Montrons tout d'abord que $P \in \mathbb{Q}[X_1, \ldots, X_k]$ par récurrence sur k. Pour $k = 1$, on pose $P = \sum_{j=0}^t a_j X^j$. On a alors quelque soit $i \in \{1, \ldots, t+1\}$, $P(i) \in \mathbb{Q}$.

Alors

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ 1 & 2 & \ldots & 2^t \\ \vdots & & & \\ 1 & t+1 & \ldots & (t+1)^t \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_t \end{bmatrix} = \begin{bmatrix} P(1) \\ \vdots \\ P(t+1) \end{bmatrix}.$$

La matrice à gauche est de Vandermonde, donc elle est inversible, puisque les valeurs $1, 2, \ldots, t+1$ sont toutes distinctes. Donc pour tout $i \in \{0, \ldots, t\}$, $a_i \in \mathbb{Q}$.

Supposons que le résultat est vérifié pour un certain $k \in \mathbb{N}^*$. On écrit alors

$$P(X_1, \ldots, X_k) = \sum_{j=0}^{t} Q_j(X_1, \ldots, X_{k-1}) X_k^j.$$

On fixe $n_1, \ldots, n_{k-1} \in \mathbb{N}$, et on a, pour tout $n \in \mathbb{N}$,

$$P(n_1, \ldots, n_{k-1}, n) = \sum_{j=0}^{t} Q_j(n_1, \ldots, n_{k-1}) n^j \in \mathbb{N}.$$

On a alors d'après le cas $k = 1$, pour tous $n_1, \ldots, n_{k-1} \in \mathbb{N}$, $Q_j(n_1, \ldots, n_{k-1}) \in \mathbb{Q}$, et par hypothèse de récurrence on obtient le résultat.

On pose $P(1, \ldots, 1) = p$ premier par hypothèse. Si on écrit

$$P(X_1, \ldots, X_k) = \sum_{\alpha \in \mathbb{N}^k} \frac{p_\alpha}{q_\alpha} X_1^{\alpha_1} \ldots X_k^{\alpha_k}$$

avec $(p_\alpha, q_\alpha) \in \mathbb{N} \times \mathbb{N}^*$ et p_α nuls sauf pour un nombre fini de α, on a $P(1, \ldots, 1) = \sum_{\alpha \in \mathbb{N}^k} \frac{p_\alpha}{q_\alpha}$.

Soit $\ell \in \mathbb{N}$ un multiple commun des q_α, on a

$$\forall n_1, \ldots, n_k \in \mathbb{N}, \quad P(1 + n_1 \ell p, \ldots, 1 + n_k \ell p) = \sum_{\alpha \in \mathbb{N}^k} \frac{p_\alpha}{q_\alpha} (1 + n_1 \ell p)^{\alpha_1} \ldots (1 + n_k \ell p)^{\alpha_k},$$

donc

$$\forall n_1, \ldots, n_k \in \mathbb{N}, \quad P(1 + n_1 \ell p, \ldots, 1 + n_k \ell p) \equiv P(1, \ldots, 1) = p \mod (p).$$

On sait que $P(1 + n_1 \ell p, \ldots, 1 + n_k \ell p)$ est aussi premier, donc

$$\forall n_1, \ldots, n_k \in \mathbb{N}, \quad P(1 + n_1 \ell p, \ldots, 1 + n_k \ell p) = p.$$

Donc P est constant. □

Proposition 2.7. *Si une fonction algébrique $F(z_1, z_2, \ldots, z_k)$ est telle que, pour tout k-uplet $(n_1, \ldots, n_k) \in \mathbb{N}^k$, $F(n_1, \ldots, n_k)$ est premier, alors F est constante.*

Démonstration. On montre tout d'abord que si pour tout $z \in \mathbb{N}$, $F(z) \in \mathbb{Z}$, alors F est un polynôme. On restreint z aux valeurs réelles. La fonction F admet en $z = +\infty$ un développement en séries de Puiseux $F(z) = \sum_{l=0}^{\infty} a_l z^{\alpha - l\delta}$ où $\alpha \in \mathbb{Q}$ et $\delta \in \mathbb{Q}^{+*}$ [11, p.98]. On pose $\Delta F(z) = F(z+1) - F(z)$ et $\Delta^{k+1} F(z) = \Delta(\Delta^k F(z))$ pour tout entier $k \geq 1$. Les fonctions $\Delta^k F$ sont aussi algébriques telles que pour tout $z \in \mathbb{N}$, $\Delta^k F(z) \in \mathbb{Z}$. On a par récurrence $\Delta^k F(n) = \int_0^1 \int_0^1 \cdots \int_0^1 F^{(k)}(n + x_1 + \ldots + x_k) dx_1 \ldots dx_k$ où $F^{(k)}$ est la dérivée

$k-$ième de F. Puisque $\delta \in \mathbb{Q}^{+*}$, après avoir dérivé k fois pour un certain k, les termes dans la série de Puiseux auront tous un exposant strictement négatif. Alors pour un k assez grand, on a $F^{(k)}(z) \to 0$ quand $z \to \infty$. Ainsi $\Delta^k F(n) \to 0$ quand $n \to \infty$. Puisque $\Delta^k F(n) \in \mathbb{Z}$, alors à partir d'un certain n, on a pour tout $m \geq n$ $\Delta^k F(m) = 0$.

Montrons qu'une fonction algébrique non nulle ne peut pas avoir une infinité de zéros. Soit F une fonction algébrique définie par $P(z, F(z)) = 0$. Soit $P(X, Y) = YQ(X, Y) + R(X)$ la division euclidienne de $P(X, Y)$ par Y. Supposons que F admet une infinité de zéros qu'on note $(z_n)_{n\in\mathbb{N}}$. Alors, pour tout $n \in \mathbb{N}$, on a $P(z_n, F(z_n)) = P(z_n, 0) = 0$. Donc, pour tout $n \in \mathbb{N}$, $F(z_n)Q(z_n, F(z_n)) + R(z_n) = R(z_n) = 0$. Ainsi, $R = 0$. Puisque P est irréductible, alors Q est le polynôme constant non nul. Alors, $F = 0$. On conclut qu'une fonction algébrique non nulle ne peut pas avoir une infinité de zéros. On a alors, pour un k assez grand, $\Delta^k F = 0$.

Montrons que si $\Delta F = 0$, alors F est constante. On a pour tout réel z, $P(z + 1, F(z + 1)) - P(z, F(z)) = 0$, et $F(z + 1) = F(z)$, donc $P(z + 1, F(z)) - P(z, F(z)) = 0$. En écrivant $P(X, Y) = \sum_{i,j=0}^{n} a_{i,j} X^i Y^j$, on obtient $\sum_{i,j=0}^{n} a_{i,j}((z + 1)^i - z^i)F(z)^j = 0$, ce qui nous donne $Q(z, F(z)) = 0$ avec Q est de degré en X inférieur strictement à celui de P. On conclut alors qu'il existe $W \in \mathbb{C}[X]$ non nul tel que pour tout $z \in \mathbb{R}$ $W(F(z)) = 0$. Sachant que W n'a qu'un nombre fini de zéros et F est continue dans le connexe $\mathbb{R}$, alors F est constante.

Montrons maintenant que si $\Delta F = Q$, avec Q un polynôme de $\mathbb{R}[X]$ de degré n, alors F est un polynôme de degré $n + 1$. On a pour tout Q polynôme de degré n, il existe (à une constante prés) un polynôme de degré $n + 1$ tel que $\Delta P = Q$. Ainsi, $\Delta(F - P) = 0$, et d'après le point précédent, $F = P + cst$.

On a maintenant puisque $\Delta^k F = 0$, alors $\Delta(\Delta^{k-1} F) = 0$, et d'après le premier point $\Delta^{k-1} F = cst$, et en utilisant le deuxième point, on aura par récurrence que F est un polynôme. Pour passer à une variable complexe, il suffit d'utiliser le théorème des zéros isolés, et ainsi, si F coïncide avec un polynôme pour les réels, elle coïncide encore avec ce polynôme prolongé dans le plan complexe.

Montrons maintenant le résultat généralisé suivant : si une fonction algébrique est telle que pour tous $n_1, \ldots, n_k \in \mathbb{N}$, $F(n_1, \ldots, n_k) \in \mathbb{Z}$, alors F est un polynôme. D'après le résultat sur une seule variable, on en déduit que pour tous $n_1, \ldots, n_{i-1}, n_{i+1}, \ldots, n_k \in \mathbb{N}$ $F(n_1, \ldots, n_{i-1}, z_i, n_{i+1}, \ldots, n_k)$ est un polynôme en z_i. Étant donné que F est une fonction algébrique, il existe un entier naturel d_i tel que pour tous $n_1, \ldots, n_{i-1}, n_{i+1}, \ldots, n_k \in \mathbb{N}$, le degré du polynôme $F(n_1, \ldots, n_{i-1}, z_i, n_{i+1}, \ldots, n_k)$ est majoré par d_i. D'où, pour tous $n_1, \ldots, n_{i-1}, n_{i+1}, \ldots, n_k \in \mathbb{N}$,

$$\frac{\partial^{d_i} F(n_1, \ldots, n_{i-1}, z_i, n_{i+1}, \ldots, n_k)}{\partial z_i^{d_i}} = 0.$$

Puisque la dérivée d'une fonction algébrique est algébrique, on a pour tous $z_1, \ldots, z_k \in \mathbb{C}$

$$\frac{\partial^{d_i} F(z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_k)}{\partial z_i^{d_i}} = 0. \tag{6}$$

Les points de branchement d'une fonction algébrique sont finis, car il s'agit des points où la dérivée partielle du polynôme définissant la fonction F s'annule. Ces points sont donc les zéros de la dérivée du polynôme et ils forment ainsi un ensemble fini. En conséquence, on peut effectuer un développement de Taylor autour d'une infinité de points. D'après l'égalité $\frac{\partial^{d_i} F}{\partial z_i^{d_i}} = 0$, ce développement est un polynôme. On a que la fonction F coïncide avec ce polynôme sur un nombre infini de points, alors F est un polynôme.

Pour déduire notre proposition, il suffit de constater qu'une fonction algébrique ne prenant que des valeurs premières est un polynôme, qui, d'après la proposition 2.6, doit être constant. □

3 Relation diophantienne à croissance exponentielle

Dans cette section, notre but est de donner une représentation diophantienne d'une suite à croissance exponentielle. Nous nous intéressons alors aux solutions de l'équation de Pell :

$$x^2 - (a^2-1)y^2 = 1 \quad \text{avec} \quad a \geq 2 \quad \text{et} \quad a, x, y \in \mathbb{N}. \tag{7}$$

Nous nous appuierons sur les résultats obtenus dans [5] pour caractériser ces solutions. On montre tout d'abord que les solutions de l'équation (7) sont exactement données par les suites de Lucas :

$$U_a(0) = 1, \quad U_a(1) = a, \quad U_a(n+2) = 2aU_a(n+1) - U_a(n) \quad \textit{pour} \quad n \geq 0, \tag{8}$$

$$V_a(0) = 0, \quad V_a(1) = 1, \quad V_a(n+2) = 2aV_a(n+1) - V_a(n) \quad \textit{pour} \quad n \geq 0. \tag{9}$$

On notera $d = a^2 - 1$. On définit les deux suites $\chi_a(n)$ et $\psi_a(n)$ dans $\mathbb{N}$ par :

$$\chi_a(n) + \sqrt{d}\psi_a(n) = (a + \sqrt{d})^n.$$

Nous montrons aussi que $\chi_a(n)$ et $\psi_a(n)$ sont les seules solutions de (8) et (9) respectivement. Le but de cette section est de démontrer la Proposition suivante.

Proposition 3.1 (Représentation diophantienne de la relation $y = \psi_a(n)$).

Pour tous $a, n, y \in \mathbb{N}$ tels que $n \geq 1, a \geq 2$, on a :

$$y = \psi_a(n) \Leftrightarrow \exists\, b, c, d, r, s, t, u, v, x \quad \textit{tels que}$$

(i)	$x^2 = (a^2-1)y^2 + 1,$	(v)	$b = a + u^2(u^2 - a),$
(ii)	$u^2 = (a^2-1)v^2 + 1,$	(vi)	$s = x + cu,$
(iii)	$s^2 = (b^2-1)t^2 + 1,$	(vii)	$t = n + 4dy,$
(iv)	$v = 4ry^2,$	$(viii)$	$n \leq y.$

La preuve de cette proposition nécessite plusieurs lemmes préparatoires.

Lemme 3.2. *Si (x, y) est une solution de (7), alors il n'est pas possible d'avoir l'inégalité*

$$1 < x + y\sqrt{d} < a + \sqrt{d}.$$

Démonstration. Soit (x, y) une solution de (7), alors on a $(x + y\sqrt{d})(x - y\sqrt{d}) = 1$. On a $x - y\sqrt{d} \neq 0$ puisque $d \notin \mathbb{Q}$, alors l'inégalité devient

$$1 < \frac{1}{x - y\sqrt{d}} < a + \sqrt{d}.$$

On remarque que $(a, 1)$ est une solution de (7), donc $x - y\sqrt{d} > a - \sqrt{d}$ et $x - y\sqrt{d} < 1$, alors $-1 < -x + y\sqrt{d} < -a + \sqrt{d}$. En ajoutant cette dernière inégalité à notre inégalité, on obtient $0 < 2y\sqrt{d} < 2\sqrt{d}$. Cela donne $0 < y < 1$ car $d > 0$, ce qui est absurde. □

Lemme 3.3. *Si (x, y) et (x', y') sont des solutions de (7) alors, en posant*

$$x'' + \sqrt{d}y'' = (x + \sqrt{d}y)(x' + \sqrt{d}y'),$$

on a que (x'', y'') est aussi une solution de (7).

Démonstration. On a

$$x'' + \sqrt{d}y'' = xx' + dyy' + \sqrt{d}(xy' + x'y),$$

donc $x'' = xx' + dyy'$ et $y'' = xy' + x'y$, alors $x'' - \sqrt{d}y'' = (x - \sqrt{d}y)(x' - \sqrt{d}y')$. On obtient

$$(x'' + \sqrt{d}y'')(x'' - \sqrt{d}y'') = (x - \sqrt{d}y)(x + \sqrt{d}y)(x' - \sqrt{d}y')(x' + \sqrt{d}y') = 1.$$

Alors, (x'', y'') est aussi une solution de (7). □

Lemme 3.4. *Le couple $(\chi_a(n), \psi_a(n))$ satisfait l'équation (7).*

Démonstration. On a $\chi_a(1) = a$ et $\psi_a(1) = 1$, qui satisfont l'équation (7). Alors, par récurrence et en utilisant le Lemme 3.3, on a $\chi_a(n)$ et $\psi_a(n)$ satisfont l'équation (7) pour tout $n \in \mathbb{N}$. □

Proposition 3.5. *Si x et y sont des solutions dans $\mathbb{N}$ de l'équation (7), alors*

$$\exists n \in \mathbb{N} \quad \textit{tel que} \quad x = \chi_a(n) \quad \textit{et} \quad y = \psi_a(n).$$

Démonstration. Soit (x, y) une solution de l'équation (7). On a $(a + \sqrt{d})^n \to +\infty$ et pour tout $n \in \mathbb{N}$, $(a + \sqrt{d})^{n+1} > (a + \sqrt{d})^n$, donc il existe $n \in \mathbb{N}$ tel que $(a + \sqrt{d})^n \leq x + \sqrt{d}y < (a + \sqrt{d})^{n+1}$. Alors, $1 \leq (a - \sqrt{d})^n(x + \sqrt{d}y) < (a + \sqrt{d})$. D'après le Lemme 3.4 et puisque si (x, y) est une solution de (7), alors $(x, -y)$ l'est aussi, on déduit que $(a - \sqrt{d})^n(x + \sqrt{d}y)$ est une solution de (7). Alors, d'après le Lemme 3.2, on a $x + \sqrt{d}y = (a + \sqrt{d})^n = \chi_a(n) + \sqrt{d}\psi_a(n)$. □

Lemme 3.6. *On a les égalités suivantes :*

$$\chi_a(m \pm n) = \chi_a(m)\chi_a(n) \pm d\ \psi_a(m)\psi_a(n),$$

$$\psi_a(m \pm n) = \psi_a(m)\chi_a(n) \pm \chi_a(m)\psi_a(n).$$

Si le symbole $\pm$ est $+$, alors il l'est pour les deux côtés, et de même pour $-$.

Démonstration. On a

$$(a + \sqrt{d}))^{m+n} = (\chi_a(m) + \sqrt{d}\psi_a(m))(\chi_a(n) + \sqrt{d}\psi_a(n)).$$

Si $m \geq n$, on a

$$(a + \sqrt{d}))^{m-n} = (\chi_a(m) + \sqrt{d}\psi_a(m))(\chi_a(n) - \sqrt{d}\psi_a(n)).$$

On obtient notre résultat par identification. □

Corollaire 3.7. *On a les égalités suivantes :*

$$\chi_a(n \pm 1) = a\chi_a(n) \pm d\ \psi_a(n),$$

$$\psi_a(n \pm 1) = a\psi_a(n) \pm \chi_a(n).$$

Démonstration. On utilise le Lemme 3.6 en prenant n et 1 au lieu de m et n respectivement. □

Corollaire 3.8. *On a les égalités suivantes :*

$$\chi_a(n+2) = 2a\chi_a(n+1) - \chi_a(n),$$

$$\psi_a(n+2) = 2a\psi_a(n+1) - \psi_a(n).$$

Démonstration. D'après le Corollaire 3.7, on a $\chi_a(n+2) = a\chi_a(n+1) + d\ \psi_a(n+1)$, et $\chi_a(n) = a\chi_a(n+1) - d\ \psi_a(n+1)$. En additionnant les deux égalités, on obtient notre résultat et de même pour ψ. □

Nous avons désormais établi que les solutions de l'équation (7) sont données par les suites de Lucas (8) et (9). Nous cherchons à présent à étudier la variation de ces suites en fonction de l'indice, dans le but de localiser leurs termes de manière diophantienne, c'est-à-dire de caractériser, à l'aide d'une représentation diophantienne, le fait qu'un entier donné soit le n-ième terme de la suite.

Lemme 3.9. *Pour tout $n \in \mathbb{N}$, on a $\chi_a(n) \wedge \psi_a(n) = 1$.*

Démonstration. Soit d un diviseur commun de $\chi_a(n)$ et de $\psi_a(n)$. On a que

$$d \mid (\chi_a(n)^2 - (a^2-1)\psi_a(n)^2).$$

Puisque le membre de droite vaut 1, on en déduit que $d = 1$. □

Lemme 3.10. *Pour tous $n, k \in \mathbb{N}$, on a $\psi_a(n) \mid \psi_a(nk)$.*

Démonstration. Soit $n \in \mathbb{N}$, on montre le résultat par récurrence sur k.

Pour $k = 0$, on a $\psi_a(n) \mid \psi_a(0) = 0$. Supposons que le résultat soit vrai pour un certain $k \in \mathbb{N}$. On a d'après le Lemme 3.6,

$$\psi_a(n(k+1)) = \psi_a(nk+n) = \psi_a(nk)\chi_a(n) + \chi_a(nk)\psi_a(n).$$

Alors, par hypothèse de récurrence, on a $\psi_a(n) \mid \psi_a(n(k+1))$. □

Lemme 3.11. *Pour tout $n \in \mathbb{N}$, on a*

$$\psi_a(n+1) > \psi_a(n) \geq n \quad \textit{et} \quad \chi_a(n+1) > \chi_a(n) \geq n.$$

Démonstration. Montrons la première inégalité par récurrence double.

On a $\psi_a(0) = 0,\ \psi_a(1) = 1,\ \psi_a(2) = 2a,\ \chi_a(0) = 1,\ \chi_a(1) = a,$ et $\chi_a(2) = 2a^2 - 1$. On a $2a^2 - 1 - a = a(2a-1) - 1$ et $a > 1$, alors $\chi_a(2) > \chi_a(1)$ et les autres inégalités sont évidentes. Supposons que la première inégalité est vérifiée pour un certain n dans $\mathbb{N}$ et $n+1$. On a par le Corollaire 3.8

$$\psi_a(n+2) = 2a\psi_a(n+1) - \psi_a(n) > (2a-1)\psi_a(n+1) > \psi_a(n+1),$$

et de même pour $\chi_a(n+2)$. Alors on a, pour tout $n \in \mathbb{N}$,

$$\forall n \in \mathbb{N} \quad \psi_a(n+1) > \psi_a(n) \quad \text{et} \quad \chi_a(n+1) > \chi_a(n).$$

Finalement puisqu'on a $\psi_a(1) = 1$ et $\chi_a(1) \geq 1$, alors pour tout $n \in \mathbb{N}$,

$$\psi_a(n) \geq n \quad \text{et} \quad \chi_a(n) \geq n.$$

□

Lemme 3.12. *Pour tous $n, k \in \mathbb{N}$, on a l'équivalence : $\psi_a(n) \mid \psi_a(k) \Leftrightarrow n \mid k$.*

Démonstration. La deuxième implication est donnée par le Lemme 3.10.

Montrons la réciproque. Soit $n, k \in \mathbb{N}$ tel que $\psi_a(n) \mid \psi_a(k)$ et soit $k = nq + r$ la division euclidienne de k par n. On a par le Lemme 3.6, $\psi_a(nq+r) = \psi_a(nq)\chi_a(r) + \chi_a(nq)\psi_a(r)$, et on a $\psi_a(n) \mid \psi_a(nq)$, alors $\psi_a(n) \mid \chi_a(nq)\psi_a(r)$. Par le Lemme 3.9, on a $\chi_a(nq) \wedge \psi_a(nq) = 1$, donc $\chi_a(nq) \wedge \psi_a(n) = 1$, alors $\psi_a(n) \mid \psi_a(r)$. On a $\psi_a(n) > \psi_a(r)$, donc $\psi_a(r) = 0$. On obtient alors $r = 0$, donc $n \mid k$. □

Lemme 3.13. *Pour tous $n, k \in \mathbb{N}$, on a $\psi_a(nk) \equiv k\chi_a(n)^{k-1}\psi_a(n) \mod (\psi_a(n)^3)$.*

Démonstration. On a

$$\chi_a(nk) + \sqrt{d}\psi_a(nk) = (a + \sqrt{d})^{nk} = (\chi_a(n) + \sqrt{d}\psi_a(n))^k = \sum_{j=1}^{k} \binom{k}{j} \chi_a(n)^{k-j} d^{\frac{j}{2}} \psi_a(n)^j.$$

Donc

$$\psi_a(nk) = \sum_{\substack{j=1 \\ j \text{ impair}}}^{k} \binom{k}{j} \chi_a(n)^{k-j} d^{\frac{j-1}{2}} \psi_a(n)^j \equiv k\chi_a(n)^{k-1}\psi_a(n) \mod (\psi_a(n)^3).$$

□

Lemme 3.14. *Pour tous $n, t \in \mathbb{N}$, on a que si $\psi_a(n)^2 \mid \psi_a(t)$, alors $\psi_a(n) \mid t$.*

Démonstration. Si $n = 0$, alors $t = 0$ et le résultat est vrai. On suppose maintenant que $n \neq 0$. On a $\psi_a(n) \mid \psi_a(t)$, alors par le Lemme 3.12, on a $n \mid t$. On pose $t = nk$ et on a par le Lemme 3.13, $\psi_a(t) \equiv k\chi_a(n)^{k-1}\psi_a(n) \mod (\psi_a(n)^3)$. Alors, $\psi_a(n)^2 \mid k\chi_a(n)^{k-1}\psi_a(n)$, et $\psi_a(n) \neq 0$, donc $\psi_a(n) \mid k\chi_a(n)^{k-1}$, et puisque $\chi_a(n) \wedge \psi_a(n) = 1$, alors $\psi_a(n) \mid k$, d'où $\psi_a(n) \mid t$. □

Lemme 3.15. *Pour tout $n \in \mathbb{N}$, on a*

$$\psi_a(n) \equiv n \mod (a-1).$$

Démonstration. Pour $n = 0$ et $n = 1$, le résultat est évident. Supposons que le résultat soit vrai pour un certain $n \in \mathbb{N}$ et $n+1$. On a par le Corollaire 3.8

$$\psi_a(n+2) = 2a\psi_a(n+1) - \psi_a(n) \equiv 2(n+1) - n \equiv n+2 \mod (a-1).$$

□

Lemme 3.16. *Soit $a \geq 2, b \geq 2$ et c dans $\mathbb{N}$ tels que $a \equiv b \mod (c)$, alors*

$$\forall n \in \mathbb{N} \quad \chi_a(n) \equiv \chi_b(n) \mod (c),$$

et de même pour ψ.

Démonstration. En utilisant le Corollaire 3.8 et par récurrence double, on obtient le résultat. □

Lemme 3.17. *Pour tous $n, j \in \mathbb{N}$ tels que $2n \pm j \in \mathbb{N}$, on a*

$$\chi_a(2n \pm j) \equiv -\chi_a(j) \mod (\chi_a(n)).$$

Démonstration. On a par le Lemme 3.6, $\chi_a(2n \pm j) = \chi_a(2n)\chi_a(j) \pm d\ \psi_a(2n)\psi_a(j)$, et $\psi_a(2n) = 2\psi_a(n)\chi_a(n)$, donc $\chi_a(2n \pm j) \equiv \chi_a(2n)\chi_a(j) \mod (\chi_a(n))$, et $\chi_a(2n) = \chi_a(n)^2 + d\psi_a(n)^2 \equiv -1 \mod (\chi_a(n))$, alors $\chi_a(2n \pm j) \equiv -\chi_a(j) \mod (\chi_a(n))$. □

Lemme 3.18. *Pour tous $n, j \in \mathbb{N}$ tels que $4n \pm j \in \mathbb{N}$, on a*

$$\chi_a(4n \pm j) \equiv \chi_a(j) \mod (\chi_a(n)).$$

Démonstration. D'après le Lemme 3.17, on a

$$\chi_a(4n \pm j) \equiv -\chi_a(2n \pm j) \equiv \chi_a(j) \mod (\chi_a(n)).$$

□

Lemme 3.19. *Si $\chi_a(i) \equiv \chi_a(j) \mod (\chi_a(n))$ avec $i \leq j \leq 2n$ et $n > 0$ alors $i = j$, sauf pour le cas particulier : $a = 2, n = 1, i = 0$ et $j = 2$.*

Démonstration. Si $\chi_a(n)$ est impair, on pose $q = \frac{\chi_a(n)-1}{2}$. On a $-q, -q+1, \ldots, -1, 0, 1, \ldots, q$ sont deux à deux non-congrus modulo $\chi_a(n)$, et on a par le Corollaire 3.7 $\chi_a(n) = a\chi_a(n-1) + d\ \psi_a(n-1)$, donc $\chi_a(n-1) \leq \frac{\chi_a(n)}{a} \leq \frac{\chi_a(n)}{2}$, alors $\chi_a(n-1) \leq q$. Comme $1 = \chi_a(0) < \chi_a(1) < \ldots < \chi_a(n-1)$, on a

$$\{\chi_a(0), \chi_a(1), \ldots, \chi_a(n-1)\} \subset \{1, \ldots, q\},$$

et d'après le Lemme 3.17, on a

$$\begin{cases} \chi_a(n+1) \equiv -\chi_a(n-1) \mod (\chi_a(n)); \\ \chi_a(n+2) \equiv -\chi_a(n-2) \mod (\chi_a(n)); \\ \vdots \\ \chi_a(2n) \equiv -\chi_a(0) \mod (\chi_a(n)). \end{cases}$$

Alors, $\chi_a(0), \chi_a(1), \ldots, \chi_a(2n)$ sont deux à deux non-congrus modulo $\chi_a(n)$.

Si $\chi_a(n)$ est pair, on pose $q = \frac{\chi_a(n)}{2}$ et cette fois on a seulement $-q+1, \ldots, 0, \ldots, q$ qui sont deux à deux non-congrus modulo $\chi_a(n)$ puisque $-q \equiv q \mod (2q)$ et on a $\chi_a(n+1) \leq q$.
Si $\chi_a(n+1) < q$ le résultat est vrai par le même raisonnement du cas précédent.
Sinon, on a $\chi_a(n+1) \equiv -q \equiv q \equiv \chi_a(n-1) \mod (\chi_a(n))$ qui contredit le résultat du lemme puisque $n+1 \neq n-1$. Ce cas se produit quand $\chi_a(n-1) = \frac{\chi_a(n)}{2}$. Par le Corollaire 3.7, on a

$$0 \leq (a-2)\chi_a(n-1) = -d\ \psi_a(n-1) \leq 0,$$

alors $a = 2$ et $\psi_a(n-1) = 0$, donc $n = 1$, et on obtient $i = 0$ et $j = 2$. □

Lemme 3.20. *Si $\chi_a(i) \equiv \chi_a(j) \mod (\chi_a(n))$, $n > 0$, $0 < i \leq n$ et $0 \leq j \leq 4n$, alors, soit $i = j$, soit $j = 4n - i$.*

Démonstration. Si $j \leq 2n$, alors $i = j$ ou bien le cas particulier du Lemme 3.19 se produit. Or on a $i > 0$, donc le cas particulier se produit pour $j = 0$, $i = 2$, $n = 1$ et $a = 2$, ce qui est impossible puisque $i \leq n$.

Si $j > 2n$, on pose $j' = 4n - j$. On a d'après le Lemme 3.18,

$$\chi_a(j') \equiv \chi_a(j) \mod (\chi_a(n)).$$

Donc

$$\chi_a(j') \equiv \chi_a(i) \mod (\chi_a(n))$$

et $j' < 2n$, et d'après le Lemme 3.19 $j' = i$, donc $j = 4n - i$ et le cas particulier du Lemme 3.19 ne peut pas se produire car $j < 4n$ ce qui donne $j' > 0$. □

Lemme 3.21. *Si $0 < i \leq n$ et $\chi_a(j) \equiv \chi_a(i) \mod (\chi_a(n))$ alors $j \equiv \pm i \mod (4n)$.*

Démonstration. Soit $j = 4nq + r$ la division euclidienne de j par $4n$. On a par le Lemme 3.18, $\chi_a(j) \equiv \chi_a(r) \mod (\chi_a(n))$, et $0 \leq r < 4n$, alors par le Lemme 3.20 on a $i = r$ ou $i = 4n - r$. Alors $j \equiv \pm i \mod (4n)$. □

Nous pouvons maintenant démontrer la Proposition 3.1.

Démonstration. Soit $a, n, y \in \mathbb{N}$ tel que $n \geq 1, a \geq 2$.

On montre l'implication $\Leftarrow$.
Supposons qu'il existe $b, c, d, r, s, t, u, v, x \in \mathbb{N}$ tels que les équations $(i) - (viii)$ soient vérifiées. On a par la Proposition 3.5,

$$(i) \Rightarrow \exists k \in \mathbb{N} \quad \text{tel que} \quad x = \chi_a(k) \quad \text{et} \quad y = \psi_a(k),$$
$$(ii) \Rightarrow \exists j \in \mathbb{N} \quad \text{tel que} \quad u = \chi_a(j) \quad \text{et} \quad v = \psi_a(j),$$
$$(iii) \Rightarrow \exists i \in \mathbb{N} \quad \text{tel que} \quad s = \chi_b(i) \quad \text{et} \quad t = \psi_b(i).$$

On veut montrer que $k = n$.

On a $(iv) \Rightarrow y^2 \mid v$. Donc par le Lemme 3.14, on a $y \mid j$.

On a $(v) \Rightarrow b \equiv a \mod (u)$, alors, par le Lemme 3.16, on obtient $\chi_b(i) \equiv \chi_a(i) \mod (\chi_a(j))$.

On a $(vi) \Rightarrow s \equiv x \mod (u)$, donc $\chi_b(i) \equiv \chi_a(k) \mod (\chi_a(j))$. Alors $\chi_a(k) \equiv \chi_a(i) \mod (\chi_a(j))$ et on a par (iv) et $(viii)$ $0 < y \leq v$, de sorte que, par le Lemme 3.11, on a $0 < k \leq j$. En utilisant le Lemme 3.21, on obtient alors que $i \equiv \pm k \mod (4j)$, et en utilisant le fait que $y \mid j$, on obtient $i \equiv \pm k \mod (4y)$.

On a par le Lemme 3.15, $t = \psi_b(i) \equiv i \mod (b-1)$, par (v) $b = a(1-u^2) + u^4$ et par (ii) $u^2 \equiv 1 \mod (v^2)$. Donc $(iv) \Rightarrow u^2 \equiv 1 \mod (4y)$, alors $b \equiv 1 \mod (4y)$. On obtient alors $\psi_b(i) \equiv i \mod (4y)$.

On a $(vii) \Rightarrow \psi_b(i) \equiv n \mod (4y)$, alors $n \equiv \pm k \mod (4y)$. On a par le Lemme 3.11 $y = \psi_a(k) \geq k$, et $(viii) \Rightarrow n \leq y$, alors $n = k$.

Montrons maintenant l'implication $\Rightarrow$.

Supposons que $y = \psi_a(n)$, alors par le Lemme 3.11, $(viii)$ est vérifiée. Posons $x = \chi_a(n)$ et (i) est bien vérifiée. Pour que (ii) et (iv) soient réalisées, on pose tout d'abord $v' = \psi_a(ny)$. On a par le Lemme 3.13 que $y^2 \mid v'$. On cherche alors $v = \psi_a(m)$ tel que $4v' \mid v$, ce qui revient à chercher u et k tels que $u^2 - (a^2-1)(4kv')^2 = 1$. Ils existent bien car $(a^2-1)(4v')^2$ n'est pas un carré parfait [2, Section 14]. On a trouvé alors $v = \psi_a(m)$ tel que $4y^2 \mid v$. Donc, il existe r vérifiant (iv), et en posant $u = \chi_a(m)$ (ii) est vérifiée. On pose $b = a + u^2(u^2 - a)$ et (v) est vérifiée et donne $b \equiv a \mod (u)$. Donc en posant $s = \chi_b(n)$ on obtient par le Lemme 3.16 $s \equiv x \mod (u)$ ce qui donne l'existence de c qui vérifie (vi). On pose $t = \psi_b(n)$. Par le Lemme 3.15, $t \equiv n \mod (b-1)$, et en utilisant (ii),(iv) et (v) on obtient $b \equiv 1 \mod (4y)$ alors $t \equiv n \mod (4y)$ ce qui donne d et (vii) est ainsi vérifiée. On a bien trouvé alors $b, c, d, r, s, t, u, v, x \in \mathbb{N}$ vérifiant les équations $(i) - (viii)$. □

On peut donner une version compacte de le Proposition 3.1, qui est celle que nous utiliserons après.

Corollaire 3.22.

Pour tous $a, n, y \in \mathbb{N}$ *tels que* $n \geq 1, a \geq 2$, *on a :*

$$y = \psi_a(n) \Leftrightarrow \exists\, c, d, r, u, x \quad \textit{tels que}$$

(i) $x^2 = (a^2-1)y^2 + 1$, (iii) $(x + cu)^2 = ((a + u^2(u^2-a))^2 - 1)(n + 4dy)^2 + 1$,
(ii) $u^2 = 16(a^2-1)r^2y^4 + 1$, (iv) $n \leq y$.

Nous énonçons à présent quatre propriétés que nous utiliserons par la suite et qui montrent le caractère exponentiel des solutions de l'équation (7).

Proposition 3.23. *Pour tout $n \in \mathbb{N}$, on a*

$$(2a-1)^n \leq \psi_a(n+1) \leq (2a)^n.$$

Démonstration. La proposition est vérifiée pour $n = 0$ et $n = 1$. Supposons que la proposition soit vraie pour un certain $n \in \mathbb{N}$ et $n+1$. On a par le Corollaire 3.8 que

$$\psi_a(n+2) = 2a\psi_a(n+1) - \psi_a(n),$$

et par le Lemme 3.11 que $n \leq \psi_a(n) < \psi_a(n+1)$, d'où $\psi_a(n+2) \leq 2a(2a)^n$. On déduit que

$$\psi_a(n+2) \geq (2a-1)\psi_a(n+1) \geq (2a-1)^{n+1}.$$

□

On désigne par l'égalité $x = \square$ le fait que l'entier x est le carré d'un entier.

Proposition 3.24. *Soit $e \geq 2$. On a*

$$e^3(e+2)(n+1)^2 + 1 = \square \Rightarrow e - 1 + e^{e-2} \leq n. \tag{10}$$

Réciproquement, pour tous $e, t \in \mathbb{N}$, on peut satisfaire (10) avec n tel que $t \mid n+1$.

Démonstration. On pose $a = e+1$. Alors, (10) devient $(a-1)^2(n+1)^2(a^2-1)+1 = \square$. Alors, par la Proposition 3.5,

il existe $j \in \mathbb{N}^*$ tel que $(a-1)(n+1) = \psi_a(j)$ (on a exclu le cas $j = 0$ puisque $a \geq 3$).

On a par le Lemme 3.15, $\psi_a(j) \equiv j \mod (a-1)$, donc $a-1 \mid j$, et $j \neq 0$, alors $a-1 \leq j$. En utilisant la Proposition 3.23, on a

$$(2a-1)^{a-2} \leq (2a-1)^{j-1} \leq \psi_a(j) = (n+1)(a-1),$$

et

$$(2a-1)^{a-2} > (2a-2)^{a-2} = 2^{a-2}(a-1)^{a-2} \geq 2(a-1)^{a-2} \quad (\text{car } a \geq 3).$$

Comme

$$(a-1)^{a-3} \geq a-2 \quad (\text{si } a = 3 \text{ on obtient une égalité et si } a > 3, \text{ c'est vérifié}),$$

on a

$$(2a-1)^{a-2} > (a-1)^{a-2} + (a-1)(a-2),$$

de sorte que

$$(a-1)^{a-2} + (a-1)(a-2) < (n+1)(a-1).$$

Donc $(a-1)^{a-3} + (a-2) < n+1$. On obtient donc que $e^{e-2} + e - 1 \leq n$.

Réciproquement, soit $e, t \in \mathbb{N}$. Soit l'équation $e^3(e+2)(kt)^2 + 1 = \square$ qu'on peut écrire $e^3(e+2)t^2k^2+1 = \square$. Puisque $e^3(e+2)t^2$ n'est pas un carré parfait (en écrivant $a = e+1$), on peut trouver k vérifiant l'équation. En posant $n+1 = kt$, on obtient le résultat souhaité. □

Proposition 3.25. *Pour tous $n, p, a \in \mathbb{N}$ tel que $a \geq 2$, on a :*

$$\chi_a(n) \equiv p^n + \psi_a(n)(a-p) \mod (2ap - p^2 - 1).$$

Démonstration. Le résultat est vérifié pour $n = 0$ et $n = 1$. Supposons que le résultat soit vérifié pour un certain $n \in \mathbb{N}$ et $n+1$. On a par le Corollaire 3.8 :

$$\begin{aligned}\chi_a(n+2) &= 2a\chi_a(n+1) - \chi_a(n) \\ &\equiv 2a(p^{n+1} + \psi_a(n+1)(a-p)) - p^n - \psi_a(n)(a-p) \mod (2ap - p^2 - 1).\end{aligned}$$

Donc

$$\chi_a(n+2) \equiv 2ap^{n+1} - p^n + (a-p)\psi_a(n+2) \mod (2ap - p^2 - 1).$$

Comme $2ap - 1 \equiv p^2 \mod (2ap - p^2 - 1)$, on en déduit que

$$\chi_a(n+2) \equiv p^{n+2} + \psi_a(n+2)(a-p) \mod (2ap - p^2 - 1).$$

□

Proposition 3.26. *Pour tous $n, p, a \in \mathbb{N}$ tel que $0 < p^n < a$, on a :*

$$p^n + \psi_a(n)(a-p) \leq \chi_a(n).$$

Démonstration. On remarque que l'inégalité est vérifiée pour $n = 0$. On suppose désormais que $n \geq 1$, ce qui donne $p < a$. On a par l'équation (7) $\sqrt{a^2-1}\psi_a(n) < \chi_a(n)$, donc

$$\psi_a(n) < \frac{1}{\sqrt{a^2-1}}\chi_a(n) \quad (a > 1).$$

On a $\sqrt{a^2-1} > a-1$, alors $\psi_a(n) < \frac{1}{a-1}\chi_a(n)$. Donc

$$(a-p)\psi_a(n) < \frac{a-p}{a-1}\chi_a(n) \quad (a-p > 0),$$

et

$$p^n + (a-p)\psi_a(n) < a + \frac{a-p}{a-1}\chi_a(n).$$

Il suffit donc de montrer que $a + \frac{a-p}{a-1}\chi_a(n) \leq \chi_a(n)$ pour obtenir notre résultat. On pose $A = \chi_a(n) - a - \frac{a-p}{a-1}\chi_a(n)$. On a

$$A = \frac{1}{a-1}((a-1)\chi_a(n) - a(a-1) - (a-p)\chi_a(n)) = \frac{1}{a-1}((p-1)\chi_a(n) - a(a-1)).$$

Si $p = 1$, on a $(a-1)\psi_a(n) < \chi_a(n)$, alors $1 + (a-1)\psi_a(n) \leq \chi_a(n)$ et l'inégalité est ainsi vérifiée.

On suppose maintenant que $p > 1$, donc $A > \frac{1}{a-1}(\chi_a(n) - a(a-1))$. On a $\chi_a(2) = 2a^2 - 1$, donc

$$\chi_a(2) - (a^2 - a) = a^2 + a - 1 \geq 0 \ , \text{ puisque } \ a \geq 1 \geq \frac{-1+\sqrt{5}}{2}.$$

Et puisque $\forall n \geq 2 \ \chi_a(n) \geq \chi_a(2)$, alors $A > 0$ quand $n \geq 2$.

Il nous reste de montrer le cas où $n = 1$, $\psi_a(1) = 1$, qui est bien vérifié puisque $p^1 + \psi_a(1)(a-p) = a = \chi_a(1)$. □

4 Représentation de la fonction factorielle à l'aide de fonctions exponentielles

Le but de cette section est de présenter et démontrer la représentation de la factorielle suivante qui est donnée en [3].

Proposition 4.1. *Pour tous* $k, f \in \mathbb{N}^*$, $\quad f = k! \Leftrightarrow \exists\ j, h, n, p, q, w, z \in \mathbb{N}$ *tels que :*

$$\begin{array}{lll} (i) & q = wz + h + j, & (iv) \quad p = (n+1)^k, \\ (ii) & z = f(h+j) + h, & (v) \quad q = (p+1)^n, \\ (iii) & (2k)^3(2k+2)(n+1)^2 + 1 = \square, & (vi) \quad z = p^{k+1}. \end{array}$$

On rappelle que $\square$ *représente le carré d'un entier.*

Pour ce faire, on aura besoin des résultats suivants :

Lemme 4.2. *Soit* $q \in \mathbb{N}^*$.

$$Si \quad 0 \leq \alpha < \frac{1}{q} \quad alors \quad (1 - q\alpha) \leq (1-\alpha)^q.$$

Démonstration. Montrons l'inégalité par récurrence sur q. L'inégalité est évidente pour $q = 1$. Supposons que le résultat soit vrai pour un certain $q \in \mathbb{N}^*$, et soit $0 \leq \alpha < \frac{1}{q+1} < \frac{1}{q}$. On a

$$(1-\alpha)^{q+1} = (1-\alpha)^q(1-\alpha) \geq (1-\alpha)(1-q\alpha) = (1 - (q+1)\alpha + q\alpha^2) \geq (1-(q+1)\alpha).$$

□

Lemme 4.3. *Si* $\quad 0 \leq \alpha \leq \frac{1}{2}, \quad alors \quad (1-\alpha)^{-1} \leq 1 + 2\alpha.$

Démonstration. Soit $0 \leq \alpha \leq \frac{1}{2}$. On a $\alpha - 2\alpha^2 = \alpha(1-2\alpha) \geq 0$, alors $(1-\alpha)(1+2\alpha) = 1 + \alpha - 2\alpha^2 \geq 1$, et on a $(1-\alpha) > 0$, donc $(1-\alpha)^{-1} \leq 1 + 2\alpha$. □

Le lemme suivant énonce la propriété essentielle permettant d'exprimer la fonction factorielle au moyen d'une représentation diophantienne exponentielles.

Lemme 4.4. *Pour tout* $k \in \mathbb{N}^*$, *si* $(2k)^k \leq n$ *et* $n^k < p$, *alors*

$$k! < \frac{(n+1)^k p^k}{r((p+1)^n, p^{k+1})} < k! + 1,$$

où $r(a,b)$ *désigne le reste de la division euclidienne de* a *par* b.

Démonstration. Montrons tout d'abord que $r((p+1)^n, p^{k+1}) = \sum_{i=0}^{k} \binom{n}{i} p^i$. On a

$$(p+1)^n = \sum_{i=0}^{k} \binom{n}{i} p^i + p^{k+1} \sum_{i=k+1}^{n} \binom{n}{i} p^{i-k-1}.$$

Il suffit de montrer que $\sum_{i=0}^{k} \binom{n}{i} p^i < p^{k+1}$. On a

$$\sum_{i=0}^{k} \binom{n}{i} p^i \leq \sum_{i=0}^{k} n^i p^i = \frac{(np)^{k+1} - 1}{np - 1},$$

alors il suffit encore de montrer que $(np)^{k+1} - 1 < (np-1)p^{k+1}$. On a

$$\begin{aligned}(np)^{k+1} - 1 = n^k n (p)^{k+1} - 1 &\leq (p-1)np^{k+1} - 1 \\ &= np^{k+2} - np^{k+1} - 1 < np^{k+2} - p^{k+1} = (np-1)p^{k+1}.\end{aligned}$$

Donc $r((p+1)^n, p^{k+1}) = \sum_{i=0}^{k} \binom{n}{i} p^i$.

Montrons maintenant que $k!(\sum_{i=0}^{k} \binom{n}{i} p^i) < (n+1)^k p^k$. On a

$$\sum_{i=0}^{k} \binom{n}{i} p^i = \sum_{i=0}^{k-1} \binom{n}{i} p^i + \binom{n}{k} p^k \leq (\sum_{i=0}^{k-1} \binom{n}{i}) p^{k-1} + \binom{n}{k} p^k.$$

Comme $2k \leq (2k)^k \leq n$, on a $k \leq \frac{n}{2}$, d'où

$$\forall i \in \{0, \ldots, k-1\}, \ \binom{n}{i} \leq \binom{n}{k-1},$$

si bien que

$$\sum_{i=0}^{k} \binom{n}{i} p^i \leq k \binom{n}{k-1} p^{k-1} + \binom{n}{k} p^k \leq k \frac{n^{k-1}}{(k-1)!} p^{k-1} + \frac{n^k}{k!} p^k = \frac{k^2 n^{k-1} p^{k-1} + n^k p^k}{k!}.$$

D'où

$$k! \sum_{i=0}^{k} \binom{n}{i} p^i \leq k^2 n^{k-1} p^{k-1} + n^k p^k < k n^k p^{k-1} + n^k p^k < p^k (k + n^k) \leq p^k (n+1)^k.$$

Montrons maintenant que $\frac{(n+1)^k p^k}{\sum_{i=0}^{k} \binom{n}{i} p^i} < k! + 1$. On a

$$\frac{(n+1)^k p^k}{\sum_{i=0}^{k} \binom{n}{i} p^i} = \frac{(n+1)^k}{\sum_{i=0}^{k} \binom{n}{i} p^{i-k}} < \frac{(n+1)^k}{\binom{n}{k}}.$$

Il suffit alors de montrer que $\frac{(n+1)^k}{\binom{n}{k}} < k! + 1$. Cette inégalité est bien vérifiée quand $k = 1$ puisque $n \geq 2$ ce qui donne $n+1 < 2n$. On suppose alors que $k \geq 2$. On a $\frac{(n+1)^k}{\binom{n}{k}} = \frac{(n+1)^k k!}{\frac{n!}{(n-k)!}}$, et $\frac{n!}{(n-k)!} > (n+1-k)^k$. Alors,

$$\frac{(n+1)^k}{\binom{n}{k}} < \frac{k!}{\frac{(n+1-k)^k}{(n+1)^k}} = \frac{k!}{(1-\frac{k}{n+1})^k} < k!\left((1-\frac{k}{n})^k\right)^{-1}.$$

Et on a $0 < k^2 < n$ et $k \geq 2$, donc $k^2 < (2k)^k \leq n$. En prenant $\alpha = \frac{k}{n}$ et $q = k$, on a $0 < \frac{k}{n} \leq \frac{1}{k}$. Par le Lemme 4.2, on obtient $0 < 1 - \frac{k^2}{n} \leq (1-\frac{k}{n})^k$, donc $(1-\frac{k^2}{n})^{-1} \geq (1-\frac{k}{n})^{-k}$, alors $\frac{(n+1)^k}{\binom{n}{k}} < k!(1-\frac{k^2}{n})^{-1}$. On a $0 < \frac{k^2}{n} \leq \frac{1}{2}$ (en utilisant $(2k)^k \leq n$), alors, par le Lemme 4.3, $(1-\frac{k^2}{n})^{-1} \leq 1 + 2\frac{k^2}{n}$, donc

$$\frac{(n+1)^k}{\binom{n}{k}} < k!\left(1 + \frac{2k^2}{n}\right) \leq k!\left(1 + \frac{2k^2}{(2k)^k}\right).$$

On a $(2k)^k = 2 \times k \times 2k \times (2k)^{k-2}$, et $2 \times (2k)^{k-2} \geq k!$, alors $(2k)^k \geq 2k^2 k!$. Alors

$$\frac{(n+1)^k}{\binom{n}{k}} < k!\left(1 + \frac{1}{k!}\right) = k! + 1.$$

□

Nous pouvons maintenant démontrer la proposition 4.1.

Démonstration. Montrons l'implication $\Leftarrow$.

Supposons qu'il existe $j, h, n, p, q, w, z \in \mathbb{N}$ tel que $(i) - (vi)$ soient vérifiées. On a $(ii) \Rightarrow\ h + j < z$, car $f \in \mathbb{N}^*$, alors $(1) \Rightarrow\ r(q,z) = h + j$.
Par (iii) et la Proposition **??**, on a $2k - 1 + (2k)^{2k-2} \leq n$. Donc $(2k)^k \leq n$ (si $k = 1$ c'est vérifié et si $k \geq 2$ $2k - 2 \geq k$). On a d'après (iv), $n^k < p$, de sorte que d'après le Lemme 4.4 on a

$$k! < \frac{(n+1)^k p^k}{r((p+1)^n, p^{k+1})} < k! + 1.$$

Par (iv) et (v) et (vi), on a

$$k! < \frac{z}{h+j} < k! + 1,$$

alors, par (ii) on a

$$k! < f + \frac{h}{h+j} < k! + 1.$$

Donc, $f = k!$.

Supposons maintenant que $f = k!$.

D'après la Proposition 3.24, on peut trouver n vérifiant (3) tel que $(2k)^k \leq n$, p et q et z sont alors donnés et (4)-(5)-(6) sont vérifiées. On pose $w = \frac{q - r(q,z)}{z}$ qui appartient bien à $\mathbb{N}$. On pose $h = z - fr(q,z)$ et $j = r(q,z) - h = (f+1)r(q,z) - z$. Par le Lemme 4.4, on a $f < \frac{z}{r(q,z)} < f + 1$, alors $h, j\ \in \mathbb{N}$. □

5 Représentation diophantienne de l'ensemble des nombres premiers

On présente maintenant le théorème central de ce mémoire. Il est dû à J. P. Jones, D. Sato, H. Wada et D. Wiens dans [3] :

Théorème 5.1. *L'ensemble des valeurs strictement positives que prend le polynôme ci-dessous, lorsque ses variables varient sur les entiers naturels, correspond exactement à l'ensemble de tous les nombres premiers :*

$$\begin{aligned}(k+2)\Big\{&1-[wz+h+j-q]^2-[(x+cu)^2-((a+u^2(u^2-a))^2-1)(n+4dy)^2-1]^2\\&-[z-(gk+2g+k+1)(h+j)-h]^2-[m^2-(a^2-1)\ell^2-1]^2\\&-[16(k+1)^3(k+2)(n+1)^2+1-f^2]^2-[ai+k+1-1-\ell-i]^2\\&-[e-p-q-z-2n]^2-[n+\ell+v-y]^2-[e^3(e+2)(a+1)^2+1-o^2]^2\\&-[m-p-\ell(a-n-1)-b(2a(n+1)-(n+1)^2-1)]^2-[x^2-(a^2-1)y^2-1]^2\\&-[x-q-y(a-p-1)-s(2a(p+1)-(p+1)^2-1)]^2\\&-[u^2-16(a^2-1)r^2y^4-1]^2-[pm-z-p\ell(a-p)-t(2ap-p^2-1)]^2\Big\}.\end{aligned}$$

(On précise que l'expression [...] ne représente pas la partie entière, mais uniquement des parenthèses ordinaires, et de même pour {...} qui ne représente pas la partie fractionnaire).

Pour montrer ce théorème, on montre tout d'abord le théorème suivant.

Théorème 5.2. *Pour tout entier naturel $k \geq 1$, pour que $k+1$ soit premier, il est nécessaire et suffisant qu'il existe des entiers naturels $a, b, c, d, e, f, g, h, i, j, k, \ell, m, n, o, p, q, r, s, t, u, v, w, x, y, z$ tels que :*

$$\begin{array}{ll}
(i) & q = wz + h + j \\
(ii) & z = (gk + g + k)(h + j) + h \\
(iii) & (2k)^3(2k+2)(n+1)^2 + 1 = f^2 \\
(iv) & e = p + q + z + 2n \\
(v) & e^3(e+2)(a+1)^2 + 1 = o^2 \\
(vi) & x^2 = (a^2-1)y^2 + 1 \\
(vii) & u^2 = 16(a^2-1)r^2y^4 + 1 \\
(viii) & (x+cu)^2 = ((a+u^2(u^2-a))^2-1)(n+4dy)^2 + 1 \\
(ix) & m^2 = (a^2-1)\ell^2 + 1 \\
(x) & \ell = k + i(a-1) \\
(xi) & n + \ell + v = y \\
(xii) & m = p + \ell(a-n-1) + b(2a(n+1) - (n+1)^2 - 1) \\
(xiii) & x = q + y(a-p-1) + s(2a(p+1) - (p+1)^2 - 1) \\
(xiv) & pm = z + p\ell(a-p) + t(2ap - p^2 - 1).
\end{array}$$

Démonstration. **Montrons la condition de suffisance.** Pour ce faire, on utilise le théorème de Wilson. Soit $k \in \mathbb{N}^*$, et supposons qu'il existe 26 entiers naturels notés $a, b, \ldots, z$ vérifiant les équations $(i) - (xiv)$. On remarque que si on ajoute les conditions $p = (n+1)^k$, $q = (p+1)^n$, $z = p^{k+1}$ aux équations $(i) - (iii)$, on aura, par la Proposition 4.1, que $gk + g + k = k!$, ce qui donne $(g+1)(k+1) = k! + 1$. Cela implique, par le théorème de Wilson, que $k+1$ est premier. Ainsi, pour montrer notre résultat, il suffit

de montrer les conditions $p = (n+1)^k$, $q = (p+1)^n$ et $z = p^{k+1}$ en utilisant les résultats obtenus sur les équations de Pell.

L'équation (iii) nous donne, par la Proposition 3.24, que $2k - 1 + 2k^{2k-2} \leq n$. Alors, on a (i') $k < n$ et (ii') $n \geq 2$.

Les équations (iv) et (v) nous donnent par la Proposition 3.24 :

$$(iii')\ e - 1 + e^{e-2} = p + q + z + 2n - 1 + (p + q + z + 2n)^{p+q+z+2n-2} \leq a \quad \text{et} \quad (iv')\ n < a.$$

D'après (i'), (ii') et (iii'), on a

$$(v')\ p < a, \quad (vi')\ (n+1)^k < a \quad \text{et} \quad (vii')\ 2a(n+1) - (n+1)^2 - 1 = (n+1)(2a-n-1) - 1 > a.$$

Les équations $(vi) - (viii)$ et (xi) nous donnent, par le Corollaire 3.22,

$$y = \psi_a(n) \quad \text{et} \quad x = \chi_a(n).$$

L'équation (ix) montre qu'il existe $k' \in \mathbb{N}$ tel que

$$\ell = \psi_a(k') \quad \text{et} \quad m = \chi_a(k').$$

L'équation (xi) et l'inégalité (ii') impliquent que $\ell < y$, alors $k' < n$, et en utilisant (iv'), on a $k' < a - 1$. On a de même $k < a - 1$, alors en utilisant l'équation (x) et le Lemme 3.15, on a $k \equiv \ell \equiv k' \mod (a-1)$, ainsi $k = k'$ et $\ell = \psi_a(k)$ et $m = \chi_a(k)$.

On a d'après la Proposition 3.25, appliquée à $n+1$, puisque $a > 2$,

$$m \equiv (n+1)^k + \ell(a - n - 1) \mod (2a(n+1) - (n+1)^2 - 1).$$

D'après l'équation (xii),

$$m \equiv p + \ell(a - n - 1) \mod (2a(n+1) - (n+1)^2 - 1).$$

Alors,

$$p \equiv (n+1)^k \mod (2a(n+1) - (n+1)^2 - 1).$$

En utilisant (v'),(vi') et (vii'), on obtient ainsi $p = (n+1)^k$. En utilisant (ii') et (iii'), on a

$$(viii')\ q < a, \quad (ix')\ (p+1)^n < a \quad \text{et} \quad (x')\ 2a(p+1) - (p+1)^2 - 1 = (p+1)(2a-p-1) - 1 > a.$$

En utilisant l'équation $(xiii)$, et par le même raisonnement qu'avant, on trouve $q = (p+1)^n$. D'après (i'), (ii'), (iii') et le fait que $p > 0$, on a

$$z < a, \quad p^{k+1} < a \quad \text{et} \quad a < 2ap - p^2 - 1.$$

En utilisant cette fois l'équation (xiv) et la Proposition 3.25, appliquée à p, on a

$$z \equiv pm - p\ell(a-p) \equiv p^{k+1} + p\ell(a-p) - p\ell(a-p) \equiv p^{k+1} \mod (2ap - p^2 - 1).$$

Alors, $z = p^{k+1}$. On a montré ainsi la suffisance.

Montrons la nécessité. Supposons que $k+1$ soit premier. D'après le théorème de Wilson $(k+1) \mid (k!+1)$. Alors il existe $g \in \mathbb{N}$ tel que $k!+1 = (g+1)(k+1)$. Donc, d'après la Proposition 4.1, il existe $j, h, n, p, q, w, z, f \in \mathbb{N}$ tels que :

$$\begin{array}{llll} (i) & q = wz + h + j, & (iv'') & p = (n+1)^k, \\ (ii) & z = (gk + g + k)(h + j) + h, & (v'') & q = (p+1)^n, \\ (iii) & (2k)^3(2k+2)(n+1)^2 + 1 = f, & (vi'') & z = p^{k+1}. \end{array}$$

On pose $e = p + q + z + 2n$ et (iv) est satisfaite. D'après la Proposition 3.24, (v) admet une solution, ce qui donne a et o. On pose $y = \psi_a(n)$ et d'après le Corollaire 3.22, on peut trouver c, d, r, u, x tels que les équations $(vi) - (viii)$ soient satisfaites. On pose $m = \chi_a(k)$ et $\ell = \psi_a(k)$ et on aura que (ix) est satisfaite. D'après le Lemme 3.15, on a $\ell \equiv k \mod (a-1)$, ce qui donne l'existence de i et (x) est satisfaite. Pour montrer l'existence de v satisfaisant (xi), on doit montrer que $y \geq n + \ell = n + \psi_a(k)$, et on a déjà montré que (iii) implique que $k < n$. Il suffit alors de montrer que $\psi_a(n) \geq n + \psi_a(n-1)$. En utilisant $2 \leq n < a$ et le Corollaire 3.8, on trouve notre résultat par récurrence. On a donc montré que v existe dans $\mathbb{N}$ et que (xi) est vérifiée.

On a par la Proposition 3.25 que

$$m = \chi_a(k) \equiv (n+1)^k + \ell(a-n-1) \mod (2a(n+1) - (n+1)^2 - 1),$$

et en utilisant le fait que $p = (n+1)^k$, on trouve b vérifiant (12). Par le même raisonnement sur x et $q = (p+1)^n$, ainsi que m et $z = p^{k+1}$, on trouve s vérifiant $(xiii)$ et t vérifiant (xiv). Ceci termine la preuve du théorème 5.2. □

On obtient ainsi la preuve du théorème 5.1 comme conséquence.

Démonstration. On prend $k+2$ au lieu de $k+1$ dans le Théorème 5.2 et on somme les carrés des équations $(i) - (xiv)$ du théorème 5.2. La somme obtenue est nulle si $k+2$ est premier et strictement positive, sinon. □

Remarque 5.3. *Le polynôme construit n'a pas d'utilité pour le calcul effectif des nombres premiers. D'après la démonstration du théorème 5.2, si on veut montrer que $k+1 = 2$ est premier on a $k = 1$ et $n \geq 2$, donc :*

$$p \geq (2+1)^1 = 3, \quad q \geq (3+1)^2 = 16, \quad z \geq 3^{1+1} = 9.$$

Ainsi,

$$e \geq 3 + 16 + 9 + 2 \times 2 = 32.$$

En utilisant l'inégalité $e - 1 + e^{e-2} \leq a$, on obtient :

$$a \geq 31 + 32^{30} > 10^{45}.$$

En utilisant $x = \chi_a(n)$, $\psi_a(n) \geq (2a-1)^{n-1}$ et $\chi_a(n) > \psi_a(n)\sqrt{a^2-1}$, alors

$$x > \psi_a(2)(a-1) \geq (2a-1)(a-1) > 10^{90}.$$

Et si, de plus, on veut montrer qu'un nombre supérieur à 3 est premier, on aura $k \geq 2$. *On obtient ainsi,*

$$n \geq 3, \quad p \geq (3+1)^2 = 16, \quad q \geq (16+1)^3 = 4913, \quad z \geq 16^{2+1} = 4096.$$

Alors nécessairement $a > 9028^{9028}, \quad x > 9028^{18056}$.

Cela montre que pour obtenir 2 comme sortie de notre polynôme, il faut choisir des variables $a > 10^{45}$ *et* $x > 10^{90}$. *De même, pour observer un nombre premier supérieur à 3, il est nécessaire de prendre* $a > 9028^{9028}$ *et* $x > 9028^{18056}$, *ce qui explique pourquoi aucun nombre premier n'a été trouvé avec ce polynôme.*

6 Indécidabilité du dixième problème de Hilbert

L'objectif de cette section est d'exposer une réponse négative au dixième problème de Hilbert. Pour ce faire, nous introduirons d'abord les outils nécessaires afin de donner une formulation rigoureuse au problème. Plus précisément, nous introduirons les machines de Turing qui représentent adéquatement les "algorithmes".

Nous rappelons d'abord quelques définitions essentielles pour la suite.

Définition 6.1. *Une équation diophantienne est une équation de la forme* $D(x_1, \ldots, x_m) = 0$, *où* D *est un polynôme à coefficients entiers relatifs.*

Définition 6.2. *Un ensemble* A *des* n*-uplets d'entiers naturels est* ***diophantien*** *si et seulement s'il existe un polynôme* D *à coefficients entiers relatifs en les variables* $a_1, \ldots, a_n$ *(paramètres) et* $x_1, \ldots, x_m$ *(inconnues) tel que :*

$$(a_1, \ldots, a_n) \in A \Leftrightarrow \exists x_1, \ldots x_m \ \in \mathbb{Z} \quad \textit{tels que} \quad D(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0. \tag{11}$$

L'équivalence (1) *est appelée une* ***représentation diophantienne*** *de l'ensemble* A.

Définition 6.3. *Une relation* R *sur* n *entiers naturels est dite diophantienne si l'ensemble des* $n-$*uplets pour lesquels la relation est satisfaite est diophantien. Ce qui est équivalent au fait qu'il existe un polynôme* D *à coefficients entiers relatifs tel que :*

$$R(a_1, \ldots, a_n) \Leftrightarrow \exists x_1, \ldots x_m \ \in \mathbb{Z} \quad \textit{tels que} \quad D(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0. \tag{12}$$

L'équivalence (12) *est appelée une* ***représentation diophantienne*** *de la relation* R.

Définition 6.4. *Une fonction* $f : \mathbb{N}^n \to \mathbb{N}^m$ *est diophantienne si son graphe est diophantien. Autrement dit, il existe un polynôme* D *à coefficients entiers relatifs tel que*

$$a = f(b_1, \ldots, b_n) \Leftrightarrow \exists x_1, \ldots x_m \ \in \mathbb{Z} \quad \textit{tels que} \quad D(a, b_1, \ldots, b_n, x_1, \ldots, x_m) = 0. \tag{13}$$

L'équivalence (13) *est appelée une* ***représentation diophantienne*** *de la fonction* f.

6.1 Codage diophantien

6.1.1 Numérotation de Cantor

Nous présentons maintenant l'énumération des couples d'entiers naturels suivante :

$$(0,0), (0,1), (1,0), (0,2), (1,1), (2,0), \ldots$$

Proposition 6.5. *La fonction* Cantor: $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$, *définie par* $(a, b) \mapsto \frac{(a+b)^2+3a+b}{2}$ *est une bijection. On appelle* Cantor(a, b) *le nombre de Cantor de* (a, b).

Démonstration. On prend un couple $(a, b) \in \mathbb{N} \times \mathbb{N}$. On considère la diagonale dans $\mathbb{N}^2$ définie par $x, y \in \mathbb{N}$ tels que $x + y = c$ avec c une constante. Si (a, b) et $(0, y)$ appartiennent à la même diagonale, alors $y = a + b$. Si on énumère $\mathbb{N}^2$ par diagonalisation en commençant par 0, le nombre de Cantor de $(0, a + b)$ est $\frac{(a+b)(a+b+1))}{2}$. Donc, le nombre de Cantor de (a, b) est $\frac{(a+b)(a+b+1))}{2} + a = \frac{(a+b)^2+3a+b}{2}$. □

On généralise maintenant notre fonction Cantor à un uplet $(a_1, \ldots, a_m)$ d'entiers naturels de longueur m quelconque de la manière suivante :

$$\text{Cantor}_1(a_1) = a_1,$$

$$\text{Cantor}_{n+1}(a_1, \ldots, a_{n+1}) = \text{Cantor}_n(a_1, \ldots, a_{n-1}, \text{Cantor}(a_n, a_{n+1})).$$

On appelle $\text{Cantor}_n(a_1, \ldots, a_n)$ le nombre de Cantor du uplet $(a_1, \ldots, a_n)$. La fonction Cantor_m est une bijection de $\mathbb{N}^m$ vers $\mathbb{N}$.

On peut alors définir les fonctions $\text{Elem}_{n,m}(c)$, qui renvoient le m-ième élément du n-uplet associé au nombre de Cantor c. Cependant, ces fonctions présentent un inconvénient : nous travaillerons avec des n-uplets de longueur non fixée, et il n'est pas évident de montrer que $\text{Elem}_{n,m}(c)$ est une fonction diophantienne en les trois variables c, m et n. Pour surmonter cette difficulté, nous introduisons le codage positionnel, qui nous servira après.

6.1.2 Code positionnel

Soit $(a_1, \ldots, a_n)$ un $n-$uplet de longueur n. Le code positionnel de cet uplet est donné par le triplet (a, b, c) tel que $c = n$, b doit satisfaire les inégalités $b > a_i$, $i = 1, \ldots n$, et a est donné par

$$a = a_n b^{n-1} + a_{n-1} b^{n-2} + \ldots + a_1 b^0.$$

Ainsi, $a_n, \ldots, a_1$ sont les chiffres de la représentation en base b de a. Contrairement au nombre de Cantor d'un uplet, le code d'un uplet n'est pas unique et il existe même une infinité de codes présentant le même uplet et cela est dû au choix arbitraire de la base b. En plus, tout triplet n'est pas nécessairement un code d'un uplet. On introduit alors la relation diophantienne *être un code positionnel* qui est vraie si le triplet (a, b, c) est le code d'un uplet et fausse sinon. On a :

$$\text{Code}(a, b, c) \Leftrightarrow b \geq 2 \text{ et } a < b^c.$$

On définit la fonction $\text{Elem}(a, b, d)$ qui donne le $d-$ième élément de l'uplet qui a comme code (a, b, c) avec $c \geq d$. Cette fonction est diophantienne puisque :

$$e = \text{Elem}(a, b, d) \Leftrightarrow \exists x, y, z \in \mathbb{N} \quad \text{tels que} \quad d = z+1 \text{ et } a = xb^d + eb^z + y \text{ et } e < b \text{ et } y < b^z. \tag{14}$$

Remarque 6.6. *Le fait que la relation* $\text{Code}(a, b, c)$ *et la fonction* $\text{Elem}(a, b, d)$ *sont diophantiennes découle du fait que l'exponentiation est elle-même diophantienne. En réalité,*

Julia Robinson a montré que l'existence d'une relation diophantienne à croissance exponentielle impliquerait que l'exponentiation est diophantienne. Par la suite, avec Davis et Putnam, elle a démontré que tout ensemble récursivement énumérable est exponentiellement diophantien. Comme il était déjà connu que tout ensemble diophantien est récursivement énumérable, il ne restait alors plus qu'à exhiber une relation à croissance exponentielle (aussi connue sous le nom de relation de Julia Robinson) qui soit diophantienne, afin d'établir l'équivalence entre les ensembles diophantiens et les ensembles récursivement énumérables, ce qui impliquerait l'indécidabilité du dixième problème de Hilbert. Elle a tenté de démontrer que les solutions de l'équation de Pell forment un ensemble diophantien, comme nous l'avons vu en section 3. Inspiré par cette idée, Matiiassevitch a finalement réussi, en utilisant la suite de Fibonacci, à exhiber une relation diophantienne à croissance exponentielle, achevant ainsi la réponse négative au dixième problème de Hilbert. Voir [4, Commentaire, Section 2].

On introduit maintenant un nouvel outil qui nous servira après dans la simulation diophantienne des machines de Turing. On fixe une base $b \geq 3$. Les éléments des uplets appartiendront à un ensemble $M_b = \{0, 1, \ldots, b-1\}$. Soit F une fonction définie sur M_b et à valeurs dans M_b. On construit de cette fonction une fonction $F[b]$, définie sur $\mathbb{N} \times \mathbb{N}$ de la façon suivante : si (a, b, c) est le code positionnel d'un uplet $(a_1, \ldots, a_c)$, alors $(F[b](a, c), b, c)$ est le code positionnel de l'uplet $(F(a_1), \ldots, F(a_c))$, et si (a, b, c) n'est pas un code positionnel, alors $F[b](a, c)$ n'est pas définie. On a le résultat suivant, dont la preuve se trouve dans [4, p. 52].

Proposition 6.7. *La fonction $F[b]$ est diophantienne.*

On peut généraliser cet outil à une fonction de m variables. Soit F une fonction définie sur M_b^m à valeurs dans M_b. On définit $F[b] : \mathbb{N}^m \times \mathbb{N} \to \mathbb{N}$ ainsi : si $(a_1, b, c), \ldots (a_m, b, c))$ sont les codes positionnels des uplets $(a_{1,1}, \ldots, a_{1,c}), \ldots, (a_{m,1}, \ldots, a_{m,c})$ respectivement, alors $(F[b](a_1, \ldots, a_m, c), b, c)$ est le code positionnel de l'uplet

$$(F(a_{1,1}, \ldots, a_{m,1}), \ldots, F(a_{1,c}, \ldots, a_{m,c})).$$

6.2 Machines de Turing

Dans cette section, nous présenterons une description d'une machine de Turing, un outil de calcul abstrait [4, p. 77]. Il existe une formulation ensembliste purement mathématique de cette machine, mais, comme indiqué dans [4, p. 77], nous en donnerons une description en la considérant comme une machine physique. La machine est composée d'une mémoire sous la forme d'un *ruban* divisé en cellules. On suppose que ce ruban a une seule extrémité gauche et est infini à droite afin d'éviter, comme pour un vrai ordinateur, des problèmes de type "Mémoire insuffisante". Par contre, à chaque instant, seul un nombre fini de cellules va être rempli. Chaque cellule peut soit être vide soit contenir un seul *symbole* pris dans un ensemble fini $A = \{\alpha_1, \ldots, \alpha_w\}$ qu'on nomme un *alphabet*. On utilise le symbole Λ pour désigner une cellule vide et on utilise le symbole "$\star$" pour marquer exclusivement la cellule la plus à gauche du ruban (cellule "initiale"). La machine contient aussi une *tête* qui va servir pour lire et écrire les symboles sur les cellules du ruban. Elle se positionne sur une seule cellule et à chaque instant du temps, qu'on suppose discret, elle peut se déplacer vers la droite et vers la gauche. À chaque instant, la machine prend une valeur q_i parmi un ensemble fini d'*états* $q_1, \ldots, q_v$. On notera q_1 l'état *initial*, c'est l'état de la machine au premier instant. Les états *finaux* sont les états pour lesquels la machine s'arrête.

À une étape donnée, la machine fonctionne de la manière suivante : la tête examine la cellule et y écrit un symbole (qui peut être identique à celui déjà présent). Ensuite, elle se déplace soit vers la gauche, soit vers la droite, ou bien reste immobile. Finalement, elle passe dans un autre état (qui peut être le même état qu'avant). Ce fonctionnement est décrit par des *instructions* de la forme

$$q_i \alpha_j \Rightarrow \alpha_{A(i,j)} D(i,j) q_{Q(i,j)} \tag{15}$$

où

(a) q_i est l'état en cours, qui ne doit pas être final ;

(b) α_j est le symbole de l'alphabet examiné par la tête. On pose $\alpha_0 = \Lambda$;

(c) $\alpha_{A(i,j)}$ est le symbole de l'alphabet à écrire ;

(d) $D(i,j)$ est le déplacement de la tête. Il y a trois choix :
- L, la tête se déplace à gauche,
- R, la tête se déplace à droite,
- S, la tête ne se déplace pas ;

(e) $q_{Q(i,j)}$ est le nouvel état.

Pour chaque couple consistant d'un état non final et d'un symbole de $A \cup \{\Lambda\}$, il doit y avoir exactement une instruction dont ce couple est le membre à gauche. Par convention, le ruban à tout instant doit être occupé par des symboles de A et sans trous, c'est-à-dire qu'il n'existe pas de cellule vide située à gauche d'une cellule non vide. Le reste du ruban infini est vide. La machine reçoit comme entrée le contenu initial du ruban et la position de la tête et commence à travailler selon les instructions. Si elle atteint un état final, la sortie est donnée par l'état de la machine, le contenu du ruban et la position de la tête. Il se peut que la machine n'atteigne jamais un état final.

6.2.1 Composition de machines

L'objectif de cette sous-section est de construire des machines de Turing ayant des propriétés spécifiques.

Toutes les machines auront le même alphabet $\{\star, 0, 1, 2, 3, \lambda\}$ avec λ désignant que la cellule est vide. On introduit λ puisque la tête doit nécessairement écrire un symbole sur la cellule examinée. On l'interprète exactement comme une cellule vide. Ainsi, à droite d'une cellule contenant λ, on ne peut avoir que des cellules contenant λ ou Λ (c'est-à-dire une cellule vide). Les machines auront deux états finaux q_2 et q_3 qui seront interprétés comme les réponses "OUI" et "NON" respectivement. Afin de ne pas fournir le système complet des instructions, qui peut être long à écrire et peu pratique, on introduit deux méthodes de construction de machines à partir d'autres machines.

a) La machine $M_1; M_2$

Soit M_1 et M_2 deux machines de Turing, on construit une nouvelle machine M qu'on note $M_1; M_2$ de la manière suivante :

(a) Dans toutes les instructions de M_1, on remplace l'état q_2 par q_{v+1} où v est le nombre des états de la machine M_1.

(b) Dans toutes les instructions de M_2, on remplace les états non-finals q_i par q_{v+i}.

(c) L'ensemble des instruction de la nouvelle machine M est constitué des instructions des deux machines M_1 et M_2 modifiés comme ci-dessus.

La machine $M_1; M_2$ fonctionne de la manière suivante pour notre ensemble d'états q_1, q_2, q_3 : l'état initial de la machine est celui de la machine M_1. Si M_1 s'arrête dans q_2, les actions de M_2 se lancent, sinon la machine $M_1; M_2$ s'arrête dans q_3. $M_1; M_2$ permet alors d'exécuter M_1 et M_2 sauf si M_1 s'arrête dans q_3.

Remarque 6.8. *On remarque que l'opération ; est associative. Le terme* $M1; M2; M3$ *a donc un sens.*

b) La machine *while* M_1 *do* M_2 *od :*

Soit M_1 et M_2 deux machines de Turing, on construit une nouvelle machine M' qu'on note *while* M_1 *do* M_2 *od* de la manière suivante :

(a) Dans toutes les instructions de M_1, on remplace l'état q_2 par q_{v+1} où v est le nombre des états de la machine M_1, et on remplace l'état q_3 par q_2.

(b) Dans toutes les instructions de M_2, on remplace les états non-finals q_i par q_{v+i}, et on remplace l'état final q_2 par q_1.

(c) L'ensemble des instruction de la nouvelle machine M' est constitué des instructions des deux machines M_1 et M_2 modifiés comme ci-dessus.

La machine *while* M_1 *do* M_2 *od* fonctionne de la manière suivante pour notre ensemble d'états q_1, q_2, q_3 : l'état initial de la machine est celui de la machine M_1. Si M_1 s'arrête dans q_2, M_2 entre dans q_1, sinon la machine *while* M_1 *do* M_2 *od* s'arrête dans q_2. Si M_2 s'arrête dans q_2, M_1 entre dans q_1 sinon notre machine s'arrête dans q_3. *while* M_1 *do* M_2 *od* permet alors d'exécuter cycliquement les deux machines jusqu'à ce que l'une d'elles s'arrête dans q_3.

6.2.2 Machines de base

On introduit maintenant une liste de machines de Turing dont nous aurons besoin. Les machines de Turing que nous allons considérer manipulent principalement des entiers naturels. Nous utilisons la représentation des nombres suivante.

Un entier strictement positif m est représenté par une suite de $m+1$ cellules consécutives, où la première contient le symbole "0" et les m suivantes contiennent le symbole "1". Le nombre 0 est simplement représenté par le symbole "0".

Le symbole "0" au début d'un nombre joue le rôle d'une virgule dans un n-uplet. La représentation d'un n-uplet $(a_1, a_2, \ldots, a_n)$ est obtenue en juxtaposant les représentations des nombres $a_1, a_2, \ldots, a_n$ sans espace entre elles, en commençant à la deuxième cellule du ruban, la première contenant toujours le symbole $\star$. Les cellules situées après le dernier symbole "1" de la représentation de a_n contiendront soit le symbole λ, soit elles resteront vides. La représentation qui ne contient pas λ sera appelé *la représentation canonique*.

Par exemple, deux représentations possibles de l'uplet $(1, 2, 0, 3)$ sont :

$$\star|0|1|0|1|1|0|0|1|1|1|\lambda| \ \ |.... \quad \text{et} \quad \star|0|1|0|1|1|0|0|1|1|1| \ \ | \ \ |....$$

TABLE 1: Machines de Turing

Machine	**Description**	**Construction**
LEFT	La machine déplace la tête vers la gauche sauf si la tête examine la cellule $\star$.	• $q_1\star \Rightarrow \star S q_2$ • $q_1 0 \Rightarrow 0 L q_2$ • $q_1 1 \Rightarrow 1 L q_2$ • $q_1 2 \Rightarrow 2 L q_2$ • $q_1 3 \Rightarrow 3 L q_2$ • $q_1\lambda \Rightarrow \lambda L q_2$ • $q_1\Lambda \Rightarrow \lambda L q_2$
RIGHT	La machine déplace la tête vers la droite sauf si la tête examine la cellule λ ou Λ.	• $q_1\star \Rightarrow \star R q_2$ • $q_1 0 \Rightarrow 0 R q_2$ • $q_1 1 \Rightarrow 1 R q_2$ • $q_1 2 \Rightarrow 2 R q_2$ • $q_1 3 \Rightarrow 3 R q_2$ • $q_1\lambda \Rightarrow \lambda L q_2$ • $q_1\Lambda \Rightarrow \lambda L q_2$

WRITE(n)	La machine écrit n qui doit appartenir à $\{0, 1, 2, 3, \lambda\}$ sauf si la tête examine la cellule $\star$.	• $q_1\star \Rightarrow \star S q_2$ • $q_1 0 \Rightarrow 0 S q_2$ • $q_1 1 \Rightarrow 0 S q_2$ • $q_1 2 \Rightarrow 0 S q_2$ • $q_1 3 \Rightarrow 0 S q_2$ • $q_1 \lambda \Rightarrow 0 S q_2$ • $q_1 \Lambda \Rightarrow 0 S q_2$
READ(n)	La machine s'arrête dans l'état q_2 si la tête examine n et s'arrête en état q_3 sinon. Pour la machine READ(λ), elle s'arrête en état q_2 si la tête examine λ ou Λ.	• $q_1\star \Rightarrow \star S q_3$ • $q_1 0 \Rightarrow 0 S q_2$ • $q_1 1 \Rightarrow 1 S q_3$ • $q_1 2 \Rightarrow 2 S q_3$ • $q_1 3 \Rightarrow 3 S q_3$ • $q_1 \lambda \Rightarrow \lambda S q_3$ • $q_1 \Lambda \Rightarrow \lambda S q_3$
STOP	La machine va directement dans l'état q_3, sans déplacer la tête.	• $q_1\star \Rightarrow \star S q_3$ • $q_1 0 \Rightarrow 0 S q_3$ • $q_1 1 \Rightarrow 1 S q_3$ • $q_1 2 \Rightarrow 2 S q_3$ • $q_1 3 \Rightarrow 3 S q_3$ • $q_1 \lambda \Rightarrow \lambda S q_3$ • $q_1 \Lambda \Rightarrow \lambda S q_3$
NEVERSTOP	La machine reste toujours dans l'état q_1, donc ne s'arrête jamais.	• $q_1\star \Rightarrow \star S q_1$ • $q_1 0 \Rightarrow 0 S q_1$ • $q_1 1 \Rightarrow 1 S q_1$ • $q_1 2 \Rightarrow 2 S q_1$ • $q_1 3 \Rightarrow 3 S q_1$ • $q_1 \lambda \Rightarrow \lambda S q_1$ • $q_1 \Lambda \Rightarrow \lambda S q_1$

READNOT(n)	La machine fait l'opposé de la machine READ(n).	*while*READ(n) *do* STOP *od*
STAR	La machine déplace la tête vers la cellule $\star$.	*while*READNOT($\star$) *do* LEFT *od*
VACANT	La machine déplace la tête vers la cellule la plus à gauche contenant le symbole λ si elle existe ; sinon, elle déplace la tête vers la cellule vide la plus à gauche.	STAR ;*while*READNOT(λ) *do* RIGHT *od*
JUMP	La machine déplace la tête à droite jusqu'à ce qu'elle examine le symbole "0". Si aucune cellule contenant le symbole "0" ne se trouve à droite de la tête, la machine ne s'arrêtera jamais.	*while*READNOT(0) *do* RIGHT *od*
FIND(k)	La machine déplace la tête à droite jusqu'à ce qu'elle examine le $k-$ième symbole "0". En lui donnant en entrée la représentation de l'uplet $(a_1, a_2, \ldots, a_n)$ avec $k \leq n$, la tête se déplace vers le symbole "0" qui débute la représentation de 'élément a_k.	FIND(1)=STAR ;JUMP FIND(k+1)=FIND(k) ;RIGHT ;JUMP
LAST	En lui donnant en entrée la représentation de l'uplet $(a_1, a_2, \ldots, a_n)$, la machine se comporte comme FIND(n). Elle déplace la tête vers le symbole "0" qui débute la représentation de 'élément a_n.	VACANT ;*while*READNOT(0) *do* LEFT *od*
NEW	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n, 0)$.	VACANT ;WRITE(0)
INC	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n + 1)$.	VACANT ;WRITE(1)

DEC	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n - 1)$ si $a_n \neq 0$. Si $a_n = 0$, la machine ne change pas le ruban, et s'arrête dans l'état q_3.	VACANT ;LEFT ;[READ(1) ;WRITE(λ)]
DELETE	La machine tronque l'uplet $(a_1, a_2, \ldots, a_n)$ en le transformant en $(a_1, a_2, \ldots, a_{n-1})$.	VACANT ; *while*READNOT(0) *do* WRITE(λ) ;LEFT *od* ; WRITE(λ)
MARK(n) où $n = 2$ ou 3	La machine remplace les occurrences consécutives du symbole "1" juste à droite de la position initiale de la tête par des symboles "n".	*while*RIGHT ;READ(1) *do* WRITE(n) *od*
THEREIS(n) où $n = 2$ ou 3	La machine détermine si le symbole "2" existe dans le ruban en commençant par la cellule $\star$. Si le symbole existe elle s'arrête dans l'état q_2, sinon elle s'arrête dans l'état q_3.	STAR ; *while*READNOT(n) *do* if READNOT(λ) then RIGHT *od*
THEREWAS(n) où $n = 2$ ou 3	La machine détermine si le symbole "n" existe dans le ruban. En cas de sa présence elle le remplace par "1" et s'arrête. Alors, elle restaure le premier symbole "n" à gauche seulement.	if THEREIS(n) then WRITE(1)
RESTORE	La machine change tous les symboles "2" et "3" dans le ruban par "1".	*while*THEREIS(2) *do* THEREWAS(2) *od* ; *while*THEREIS(3) *do* THEREWAS(3) *od* ;
APPEND(k)	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n + a_k)$.	FIND(k) ;MARK(2) ; *while*THEREWAS(2) *do* INC *od*
COPY(k)	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n, a_k)$.	NEW ;APPEND(k)
ADD(k, ℓ)	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n, a_k + a_\ell)$.	COPY(k) ;APPEND(ℓ)

MULT(k, ℓ)	La machine transforme l'uplet $(a_1, a_2, \ldots, a_n)$ en $(a_1, a_2, \ldots, a_n, a_k a_\ell)$.	Si $k \neq \ell$: MULT(k, ℓ)= NEW ;FIND(k) ;MARK(3) ; *while*THEREWAS(3) *do* APPEND(ℓ) *od* Si $k = l$: MULT(k, k)= COPY(k) ;LAST ;MARK(3) ; *while*THEREWAS(3) *do* *while*THEREWAS(3) *do* APPEND(k) *od* *od*

NOTGREATER (k, ℓ)	La machine s'arrête dans l'état q_2 si $a_k \leq a_\ell$ et dans l'état q_3 si $a_k > a_\ell$ quand elle reçoit en entrée la représentation de l'uplet $(a_1, a_2, \ldots, a_n)$.	FIND(k) ;MARK(2) ;FIND(ℓ) ;MARK(3) ;
		*while*THEREIS(2) ;THEREIS(3) *do*
		THEREWAS(2) ;THEREWAS(3)
		od ;
		*while*THEREIS(2) *do*
		RESTORE ;STOP
		od ;
		RESTORE

EQUAL(k, ℓ)	La machine s'arrête dans l'état q_2 si $a_k = a_\ell$ et dans l'état q_3 si $a_k \neq a_\ell$ quand elle reçoit en entrée la représentation de l'uplet $(a_1, a_2, \ldots, a_n)$.	NOTGREATER(k, ℓ) ;NOTGREATER(ℓ, k)
NOTEQUAL(k, ℓ)	Donne l'inverse de la machine EQUAL(k, ℓ).	*while*EQUAL(k, ℓ) *do* STOP *od*
NEXT	La machine transforme l'uplet $(a_1, \ldots, a_n)$ en $(a_1, \ldots, a_{n-2}, b, c)$, où (b,c) est le couple qui suit le couple (a_{n-1}, a_n) dans l'énumération de Cantor donnée en 6.1.1.	LAST ;WRITE(1) ;RIGHT ; *while*READ(λ) *do* WRITE(1) ;LAST ;RIGHT *od* ; WRITE(0)
DECODE	La machine transforme l'uplet $(a_1, \ldots, a_n)$ en $(a_1, \ldots, a_n, b, c)$, où (b, c) est le couple de nombre de Cantor a_n.	LAST ;MARK(2) ;NEW ;NEW ; *while*THEREWAS(2) *do* NEXT *od*

6.3 La semi-décidabilité des ensembles diophantiens

Après avoir défini les machines de Turing, on est maintenant capable d'introduire la notion d'un ensemble *récursivement énumérable* ou *semi-décidable*.

Définition 6.9. *Un ensemble A de $n-$uplets d'entiers naturels est récursivement énumérable s'il existe une machine de Turing M qui, démarrant dans l'état q_1 avec un ruban qui contient la représentation canonique de l'uplet $(a_1, \ldots, a_n)$ et dont la tête examine initialement la cellule "$\star$", s'arrête si et seulement si $(a_1, \ldots, a_n) \in A$. On dit que M semi-décide l'ensemble A. En particulier, il n'est pas garanti que la machine s'arrête, d'où le nom de semi-décidable. En effet, comme nous le verrons dans la section 6.5, un ensemble est dit décidable si la machine s'arrête dans tous les cas.*

Le but de cette section est de démontrer le résultat suivant :

Proposition 6.10. *Les ensembles diophantiens sont récursivement énumérables.*

Démonstration. Pour démontrer ce résultat, on construit, pour une équation diophantienne paramétrée

$$D(a_1, \ldots, a_n, x_1, \ldots, x_{m+1}) = 0, \tag{16}$$

une machine de Turing qui commence par la représentation canonique de l'uplet $(a_1, \ldots, a_n)$ et s'arrête si et seulement si l'équation (16) admet une solution en les inconnues $x_1, \ldots, x_{m+1}$.

L'idée est de calculer les valeurs $D(a_1, \ldots, a_n, x_1, \ldots, x_{m+1})$ pour tous les $(m+1)$-uplets de $\mathbb{N}^{m+1}$ jusqu'à ce qu'on trouve une valeur nulle. Si l'équation (16) n'admet aucune solution, alors la machine ne s'arrêtera jamais.

On commence par construire une machine M_1 qui, en recevant la représentation d'un uplet $(a_1, \ldots, a_n, y_0)$, vérifie si y_0 est le nombre de Cantor d'un $(m+1)-$uplet $(x_1, \ldots, x_{m+1})$ qui n'est pas solution de l'équation (16).

En appliquant la machine DECODE sur l'uplet $(a_1, \ldots, a_n, y_0)$, on obtient en sortie l'uplet $(a_1, \ldots, a_n, y_0, x_1, y_1)$ avec $y_0 = \text{Cantor}(x_1, y_1)$. En appliquant encore une fois la machine DECODE sur notre uplet $(a_1, \ldots, a_n, y_0, x_1, y_1)$, on obtient en sortie l'uplet $(a_1, \ldots, a_n, y_0, x_1, y_1, x_2, y_2)$, avec $y_1 = \text{Cantor}(x_2, y_2)$. Donc

$$y_0 = \text{Cantor}(x_1, \text{Cantor}(x_2, y_2)) = \text{Cantor}_3(x_1, x_2, y_2).$$

On conclut que la machine M_1'=DECODE; ...; DECODE, qui contient m copies de la machine DECODE, transforme l'uplet $(a_1, \ldots, a_n, y_0)$ en l'uplet

$$(a_1, \ldots, a_n, y_0, x_1, y_1, \ldots, x_m, y_m),$$

avec $y_0 = \text{Cantor}_{m+1}(x_1, x_2, \ldots, x_m, y_m)$. L'entier y_m est le $(m+1)-$ème élément du $(m+1)-$uplet dont le nombre de Cantor est y_0. Donc $y_m = x_{m+1}$.

Le polynôme D est à coefficients entiers relatifs. Si l'on souhaite travailler avec des entiers relatifs, il est possible de coder le signe, mais les machines qu'on a construites

sont uniquement adaptées aux nombres entiers. On réécrit l'équation (16) sous la forme suivante :

$$C_L(a_1, \ldots, a_n, x_1, \ldots, x_m, x_{m+1}) = C_R(a_1, \ldots, a_n, x_1, \ldots, x_m, x_{m+1}), \tag{17}$$

avec les polynômes C_L et C_R à coefficients entiers naturels et obtenus en faisant passer tous les coefficients négatifs dans l'équation (16) à droite de (17).

On veut maintenant calculer les deux valeurs

$$C_L(a_1, \ldots, a_n, x_1, \ldots, x_m, y_m) \quad \text{et} \quad C_R(a_1, \ldots, a_n, x_1, \ldots, x_m, y_m)$$

et les comparer. On remarque que le calcul de ces deux valeurs nécessite seulement un enchaînement des deux opérations + et × appliquées sur les quantités $1, a_1, \ldots, a_n, x_1, \ldots, x_m$, y_m. La quantité 1 est utilisée pour le calcul des constantes. Par exemple, pour calculer $x^2 + xy + 2$, on calcule $z_1 = x \times x$, puis $z_2 = x \times y$, puis $z_3 = 1 + 1$, puis $z_4 = z_1 + z_2$ et finalement $z_5 = z_4 + z_3$. Alors, on peut représenter le calcul de C_L et C_R par la suite des opérations

$$z_1 = \alpha_1 \ R_1 \ \beta_1,$$
$$\vdots$$
$$z_k = \alpha_k \ R_k \ \beta_k,$$

où $R_i \in \{+, \times\}$ et $\alpha_i, \beta_i \in \{1, a_1, \ldots, a_n, x_1, \ldots, x_m, y_m, z_1, \ldots, z_{k-1}\}$. La valeur de C_R est égale à z_k et celle de C_L est égale à z_ℓ pour un $1 \leq \ell \leq k$ fixé.

On peut alors construire une machine de Turing M_1'', en combinant les machines de Turing NEW, INC, ADD et MULT et en utilisant la méthode " ;", qui transforme l'uplet $(a_1, \ldots, a_n, y_0, x_1, y_1, \ldots, x_m, y_m)$ en l'uplet $(a_1, \ldots, a_n, y_0, x_1, y_1, \ldots, x_m, y_m, 1, z_1, \ldots, z_k)$. Finalement, on considère la machine NOTEQUAL$(n+1+2m+1+l, n+1+2m+1+l+k)$ qui compare les valeurs de $z_l = C_L$ et $z_k = C_R$. Ainsi, on a construit notre machine $M_1 = M_1'; M_1''; \text{NOTEQUAL}(n + 1 + 2m + 1 + l, n + 1 + 2m + 1 + l + k)$.

Soit M_2 la machine DELETE; ... ; DELETE qui contient $2m+k+1$ machines DELETE. Cette machine transforme l'uplet $(a_1, \ldots, a_n, y_0, x_1, y_1, \ldots, x_m, y_m, 1, z_1, \ldots, z_k)$ en l'uplet $(a_1, \ldots, a_n, y_0)$.

On définit alors la machine M par

$$M = \text{NEW}; \textit{while } M_1 \textit{ do } M_2; \text{INC } \textit{od}.$$

Si on démarre cette machine avec une représentation de l'uplet $(a_1, \ldots, a_n)$, elle commence par le transformer en $(a_1, \ldots, a_n, 0)$ en utilisant NEW. Après elle commence la boucle $\textit{while} M_1 \textit{ do } M_2; \text{INC } \textit{od}$ qui vérifie si 0 est le nombre de Cantor d'un $(m+1)-$uplet qui n'est pas solution de l'équation (16). Si cet uplet est une solution, la machine s'arrête, sinon elle reprend l'uplet $(a_1, \ldots, a_n, 0)$ et lui ajoute 1 par INC pour le transformer en $(a_1, \ldots, a_n, 1)$ et elle refait la même procédure. Sachant que Cantor_{m+1} est une bijection entre $\mathbb{N}^{m+1}$ et $\mathbb{N}$, cette machine vérifie tous les $(m+1)-$uplets possibles. Ce qui achève notre preuve. □

6.4 Caractère diophantien des ensembles semi-décidables

Dans la section précédente, on a montré que tout ensemble diophantien est semi-décidable (récursivement énumérable). Le but de cette section est de montrer la réciproque.

Proposition 6.11. *Tout ensemble récursivement énumérable est diophantien.*

Soit A un ensemble récursivement énumérable et soit M une machine de Turing qui le *semi-décide*. Pour démontrer notre résultat, on doit trouver une représentation diophantienne de l'ensemble A. Ainsi, on cherche un polynôme D à coefficients entiers relatifs tel que

$$(a_1, \ldots, a_n) \in A \Leftrightarrow \exists x_1, \ldots x_m \in \mathbb{Z} \quad \text{tels que} \quad D(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0. \tag{18}$$

Le résultat est alors équivalent au fait que l'équation $D(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0$ admet une solution en les variables $x_1, \ldots, x_m$ si et seulement si la machine M démarrant dans l'état q_1 avec un ruban qui contient la représentation canonique de l'uplet $(a_1, \ldots, a_n)$ et dont la tête examine initialement la cellule "$\star$" s'arrête.

Soit $\{\alpha_1, \ldots, \alpha_w\}$ l'alphabet de la machine M et soit $\{q_1, \ldots, q_v\}$ l'ensemble de ses états. On sait qu'à chaque instant le ruban ne contient qu'un nombre fini ℓ de cellules non vides. Donc, on peut représenter le ruban à cet instant par l'uplet

$$(s_1, s_2, \ldots, s_\ell), \tag{19}$$

où les $s_i \in \{1, 2, \ldots, w\}$ sont les indices des symboles de l'alphabet. On a par convention s_1 est l'indice du symbole "$\star$". L'état q_i de la machine et la position de la tête à chaque instant seront représentés par un uplet de même longueur que l'uplet (19)

$$(0, \ldots, 0, i, 0, \ldots, 0), \tag{20}$$

où le seul élément non nul est l'indice de l'état et la position de cet élément correspond à la position de la tête.

On appelle le triplet (contenu du ruban, état de la machine, position de la tête) à un instant fixé une *configuration*. Les deux uplets (19) et (20) déterminent de façon unique la *configuration*.

On fixe une base $\beta > \max(4, w, v)$, où w et v sont les nombres de symboles et d'états de la machine respectivement. On va utiliser le codage positionnel relativement à cette base β pour représenter les deux uplets (19) et (20) pour tout instant. On appelle *code de configuration* le couple (p, t) où p est le numéro de l'uplet (20) et t est le numéro de l'uplet (19) en base β (on rappelle que le numéro d'un uplet $(a_1, \ldots, a_n)$ est le nombre a si (a, b, c) est le code positionnel de cet uplet).

Nous ne fixons pas de longueur spécifique dans notre code de configuration, car nous considérons les éléments nuls de l'uplet (19) comme représentant des cellules vides, ce qui est cohérent avec notre convention d'un ruban infini à droite. De plus, l'uplet (20) ne contient qu'un seul élément non nul, ce qui explique que sa longueur n'a pas d'importance.

On donne tout d'abord le code de la configuration initiale (c'est la configuration donnée dans la définition 6.9 d'un ensemble récursivement énumérable), et on montre que la relation être un code de configuration initiale est une relation diophantienne afin de simplifier la construction du polynôme (18). Pour ce faire, on aura besoin de la fonction suivante :

Lemme et définition 6.12. *Soit la fonction* Repeat : $\mathbb{N}^3 \to \mathbb{N}$ *définie par :*

$$\text{Repeat}(p,q,r) = p\frac{q^r-1}{q-1}.$$

Si $p < q$, *alors* $(\text{Repeat}(p,q,r),q,r)$ *est le code positionnel du* $r-$*uplet* $(p,\ldots,p)$.

Démonstration. On a

$$p\frac{q^r-1}{q-1} = p(1+q+\ldots+q^{r-1}) = p.q^0 + p.q^1 + \ldots + p.q^{r-1}.$$

Ce qui donne le résultat par définition du code positionnel. □

On donne maintenant le code de la configuration initiale.

Lemme 6.13. *Le code de configuration d'une machine de Turing en état* q_1 *avec un ruban qui contient la représentation canonique de l'uplet* $(a_1,\ldots,a_n)$ *et dont la tête examine la cellule "*$\star$*", est donné par* $p = 1$ *et de plus* t *est donné par*

$$(t,\beta,a) = (\kappa,\beta,1) + (\mu,\beta,1) + (\text{Repeat}(\nu,\beta,a_1),\beta,a_1) + \ldots + (\mu,\beta,1) + (\text{Repeat}(\nu,\beta,a_n),\beta,a_n),$$

où $a = a_1 + \ldots + a_n + n + 1$ *et* κ,μ,ν *sont les indices des symboles "*$\star$*", "0" et "1" respectivement de l'alphabet de la machine M.*

Démonstration. On a $p = 1\cdot\beta^0 + 0\cdot\beta^1 + \ldots$, donc p est le numéro de l'uplet $(1,0,0,\ldots)$ en base β qui correspond bien à l'état q_1 et la tête qui examine la cellule initiale "$\star$".

D'après le Lemme 6.12, t est le numéro d'un uplet de longueur $a_1+\ldots+a_n+n+1$ qui est de la forme $(\kappa,\mu,\nu,\ldots\nu,\ldots,\mu,\nu,\ldots\nu)$ où chaque $i-$ème séquence de $\mu,\nu,\ldots\nu$ est de longueur a_i+1. Puisque κ,μ,ν sont les indices des symboles "$\star$", "0" et "1" respectivement de l'alphabet de la machine M, alors t est le code en base β de l'uplet qui donne la représentation canonique de l'uplet $(a_1,\ldots,a_n)$. □

Puisque la fonction Repeat est diophantienne, il suffit, pour construire l'équation (18), de construire une autre équation diophantienne

$$D_1(p,t,x_1,\ldots,x_m) = 0 \tag{21}$$

telle que si (p,t) est le code d'une configuration, alors on a l'équivalence : l'équation (21) admet une solution en $x_1,\ldots,x_m$ si et seulement si, la machine M démarrant dans cette configuration s'arrête.

On commence maintenant par simuler le passage d'une configuration à la suivante. On a alors besoin d'introduire les deux fonctions suivantes.

Définition 6.14. *Soient* NextP *et* NextT *deux fonctions sur* $\mathbb{N}^2$ *telles que si* (p,t) *est le code de configuration de la machine* M *à une étape, alors* $(\mathrm{NextP}(p,t), \mathrm{NextT}(p,t))$ *est le code de la machine à l'étape suivante. Dans le cas où* (p,t) *est le code de configuration d'une étape dont l'état est final, on pose* $(\mathrm{NextP}(p,t), \mathrm{NextT}(p,t)) = (0,t)$. *On peut comprendre cela de la façon suivante : après l'état final, le contenu du ruban ne change plus, la tête n'est plus positionnée dans une cellule et la machine n'est dans aucun état. Dans le cas où* (p,t) *n'est pas un code de configuration, les fonctions ne sont pas définies.*

On veut montrer que les deux fonctions NextT et NextP sont diophantiennes. On remarque d'abord que, d'après les instructions de la machine données par (15), les deux fonctions NextT et NextP ne dépendent que des fonctions A, D et Q. Pour ajouter le cas où l'on est dans un état final de manière conforme avec la définition des deux fonctions NextT et NextP, on pose $A(i,j) = j$, $Q(i,j) = 0$ et $D(i,j) = S$ si q_i est un état final.
On définit maintenant la fonction A' qui est un prolongement de A pour qu'on puisse utiliser l'outil qu'on a introduit dans la section 6.1.2 qui permet de passer du code d'un uplet au code de l'uplet obtenu en appliquant la fonction suivante à tous ses éléments :

$$A'(i,j) : \begin{array}{ccc} \{0,1,\dots,\beta-1\}^2 & \to & \{0,1,\dots,\beta-1\} \\ (i,j) & \mapsto & \begin{cases} A(i,j) & \text{si } 0 < i \leq v, 0 \leq j \leq w, \\ j & \text{sinon.} \end{cases} \end{array}$$

On remarque que c'est ici où l'on a besoin de la condition indiquée au début que $\beta > \max(4,w,v)$. On peut maintenant énoncer le résultat suivant.

Lemme 6.15. *Les fonctions* NextT *et* NextP *sont diophantiennes.*

Démonstration. (i) Supposons que le code de configuration de la machine (p,t) est tel que p est le numéro de l'uplet $(0,\dots,0,i,0,\dots,0)$ où i est placé dans la $m-$ième cellule et t est le numéro de l'uplet $(s_1,\dots,s_m,\dots,s_\ell)$. Cela se traduit par le fait que la machine est dans l'état q_i, la tête se trouve dans la cellule m et cette cellule contient le symbole α_{s_m}. Donc, la machine applique l'instruction $q_i\alpha_{s_m} \Rightarrow \alpha_{A(i,s_m)}D(i,s_m)q_{Q(i,s_m)}$. Si l'on est dans une position k différente de m, le symbole dans l'uplet de numéro t reste le même donc d'indice $k = A'(0,k)$. Et si l'on est dans la position m, le nouveau symbole dans l'uplet de numéro t est d'indice $A(i,s_m) = A'(i,s_m)$. Le contenu du ruban dans la nouvelle configuration est alors donné par $(A'(0,s_1),\dots,A'(0,s_m-1),A'(i,s_m),A'(0,s_m+1),\dots,A'(0,s_\ell))$ qui est de code positionnel $A'[\beta](p,t,\ell')$ d'après la section 6.1.2 avec l' un entier donné. On a alors que

$$t' = \mathrm{NextT}(p,t) \Leftrightarrow \exists \ell' \in \mathbb{N} \quad \text{tel que} \quad t' = A'[\beta](p,t,\ell').$$

D'après la proposition 6.7, on a que $A'[\beta]$ est diophantienne, de sorte que NextT l'est aussi.

(ii) Contrairement à la fonction NextT, l'élément en position n dans l'uplet de numéro $\mathrm{NextP}(p,t)$ dépend des éléments des uplets de numéros p et t en positions $n-1$, n et $n+1$ puisque le nouvel état de la machine est donné après le déplacement de la tête. On ne peut pas utiliser directement l'outil donné en section 6.1.2. On introduit alors les nouveaux

numéros d'uplets $p^R = p\beta$, $p^L = p$ div β, $t^R = t\beta$, $t^L = t$ div β, avec a div b est la partie entière de $\frac{a}{b}$. On remarque que t^R est le numéro de l'uplet $(0, s_1, \ldots, s_\ell)$, t_L est le numéro de l'uplet $(s_2, \ldots, s_\ell, 0)$, p^R est le numéro de l'uplet $(0, \ldots, 0, 0, i, \ldots, 0)$ tel que la position de i est décalée à droite et p_L est le numéro de l'uplet $(0, \ldots, i, 0, 0, \ldots, 0)$ où i est décalé à gauche. Cela est dû au fait que multiplier par β est équivalent à ajouter un 0 à droite dans la représentation en base β et diviser par β puis passer à la partie entière permet de supprimer le premier chiffre, ce sont ainsi des décalages à droite et à gauche de l'uplet d'un seul pas. L'élément en position n dans l'uplet de numéro NextP(p,t) dépend maintenant des éléments des uplets de numéros p, t, p^R, p^L, t^R et t^L de même position n. On peut alors définir la fonction qui permet de passer à l'uplet de numéro NextP(p,t). Soit la fonction $DQ : \{0,1,\ldots,\beta-1\}^6 \to \{0,1,\ldots,\beta-1\}$ définie par

$$DQ(i^L, i, i^R, j^L, j, j^R) = \begin{cases} Q(i^L, j^L) & \text{si } i^L > 0\ ,\ i = i^R = 0 \text{ et } D(i^L, j^L) = L, \\ Q(i,j) & \text{si } i > 0\ ,\ i^L = i^R = 0 \text{ et } D(i,j) = S, \\ Q(i^R, j^R) & \text{si } i^R > 0\ ,\ i = i^L = 0 \text{ et } D(i^L, j^L) = R, \\ 0 & \text{sinon.} \end{cases}$$

Cette fonction permet de donner la représentation de NextP suivante :

$$p' = \text{NextP(p,t)} \Leftrightarrow \exists \ell' \text{tel que } p' = DQ[\beta](p\beta, p, p\text{div}\beta, t\beta, t, t\text{div}\beta, \ell').$$

La fonction div est diophantienne puisque

$$c = a \text{ div } b \Leftrightarrow \exists r \in \mathbb{N} \text{ tel que } r < b,\ b \text{ divise}(a-r) \text{ et } a = cb + r.$$

Comme $DQ[\beta]$ est diophantienne par la proposition 6.7, alors NextP est diophantienne. □

Les deux fonctions NextT et NextP donnent la configuration de la machine après un pas. On introduit maintenant les deux fonctions AfterT et AfterP définies comme suit :

$$\begin{aligned} \text{AfterT}(0,p,t) &= t, \\ \text{AfterP}(0,p,t) &= p, \\ \text{AfterT}(k+1,p,t) &= \text{NextT}(\text{AfterP}(k,p,t), \text{AfterT}(k,p,t)), \\ \text{AfterP}(k+1,p,t) &= \text{NextP}(\text{AfterP}(k,p,t), \text{AfterT}(k,p,t)). \end{aligned}$$

On remarque que le code de configuration de la machine commençant par la configuration (p,t) est donné par $(\text{AfterP}(k,p,t), \text{AfterT}(k,p,t))$ après k pas. On admet le résultat suivant, dont la preuve se trouve dans [4, p. 95].

Lemme 6.16. *Les fonctions* AfterP *et* AfterT *sont diophantiennes.*

On est maintenant capable de représenter le fait que la machine commençant par la configuration (p,t) s'arrête. Soient $w_1, \ldots, w_z$ les indices des états finaux de la machine M. On a la machine M, démarrant avec la configuration (p,t), s'arrête si et seulement si :

$\exists k, r \in \mathbb{N}$ tel que $\text{Elem}(\text{AfterP}(k,p,t), \beta, r) = w_1$ ou $\cdots$ ou $\text{Elem}(\text{AfterP}(k,p,t), \beta, r) = w_z$.

Cette condition exprime le fait qu'après un nombre k d'étapes, un état final w_i sera atteint. Or si deux relations R_1 et R_2 sont diophantiennes, alors la relation R définie par

$$R(a_1, \ldots, a_n) \Leftrightarrow R_1(a_1, \ldots, a_n) \text{ ou } R_2(a_1, \ldots, a_n)$$

est aussi diophantienne. En effet, si

$$R_i(a_1, \ldots, a_n) \Leftrightarrow \exists x_1, \ldots x_m \text{ tel que } D_i(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0$$

est une représentation diophantienne des relations R_i $(i = 1, 2)$, alors

$$R(a_1, \ldots, a_n) \Leftrightarrow \exists x_1, \ldots x_m \text{ tel que } D_1(a_1, \ldots, a_n, x_1, \ldots, x_m) \cdot D_2(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0$$

est une représentation diophantienne de la relation R ($D_1 D_2$ s'annule si et seulement si l'un des deux s'annule).

Sachant que les deux fonctions AfterP et Elem sont diophantiennes, on est capable de construire notre polynôme de l'équation (21). Ce qui achève notre preuve que tout ensemble récursivement énumérable est diophantien.

6.5 Indécidabilité du dixième problème de Hilbert

On introduit à présent la notion de décidabilité, qu'on utilisera pour formuler précisément le dixième problème de Hilbert.

Définition 6.17. *Un ensemble A de $n-$uplets d'entiers naturels est récursive ou décidable s'il existe une machine de Turing M qui, démarrant dans l'état q_1 avec un ruban qui contient la représentation canonique de l'uplet $(a_1, \ldots, a_n)$ et dont la tête examine initialement la cellule "$\star$", s'arrête dans l'état q_2 si $(a_1, \ldots, a_n) \in A$ et s'arrête dans l'état q_3 sinon. On dit que M décide A.*

Remarque 6.18. *Une fois l'équivalence établie entre les notions d'ensemble diophantien et d'ensemble semi-décidable, il est devenu possible de démontrer directement l'indécidabilité du dixième problème de Hilbert en se basant sur l'indécidabilité du problème de l'arrêt de Turing [10, p.183], qui consiste à déterminer si une machine de Turing s'arrête ou non. En effet, puisque l'arrêt d'une machine de Turing est équivalent à la résolubilité d'une équation diophantienne en entiers, on en déduit immédiatement l'indécidabilité du dixième problème de Hilbert. Toutefois, il est également possible d'aborder ce problème directement à travers les équations diophantiennes. L'un des résultats majeurs des travaux sur le dixième problème de Hilbert est l'existence d'une équation diophantienne universelle, dont la définition sera présentée ci-dessous. Sa construction figure dans [4, Section 4].*

Lemme et définition 6.19. *Il existe un polynôme $U_0(k, y_1, \ldots, y_m)$ à coefficients entiers relatifs, tel que pour toute équation diophantienne sans paramètres*

$$D(x_1, \ldots, x_w) = 0, \tag{22}$$

il existe un entier naturel k_D tel que l'équation (22) possède des solutions entières en $x_1, \ldots, x_w$ si et seulement si l'équation $U_0(k_D, y_1, \ldots, y_m) = 0$ admet des solutions entières en $y_1, \ldots, y_m$.

On dit que $U_0(k, y_1, \ldots, y_m) = 0$ est une équation diophantienne universelle et que k_D est le code de l'équation diophantienne (22).

On note A_0 l'ensemble diophantien des entiers naturels défini par cette équation diophantienne universelle, et on remarque que ses éléments sont exactement les codes des équations diophantiennes sans paramètres qui admettent des solutions entières.

On peut maintenant énoncer le dixième problème de Hilbert comme suit : **L'ensemble A_0 est-il récursif ?** La réponse est la suivante.

Théorème 6.20. *L'ensemble A_0 n'est pas récursif. Donc, il n'existe aucune machine de Turing pouvant décider si une équation diophantienne admet ou non une solution en nombres entiers.*

Afin de présenter la preuve du théorème 6.20, on donne tout d'abord une propriété de l'ensemble A_0 qui a été prouvée en [4, p. 69].

Lemme 6.21. *L'ensemble A_0 est de complémentaire non diophantien.*

On donne ensuite des résultats qui montrent la relation entre la décidabilité et la semi-décidabilité.

Lemme 6.22. *Un ensemble récursif est semi-décidable.*

Démonstration. Soit M une machine de Turing qui décide un ensemble récursif A. On pose $M' = \textit{while } M \textit{ do}$ STOP $\textit{od}$; NEVERSTOP. La machine M' s'arrête si et seulement si la machine M s'arrête dans l'état q_2 donc si et seulement si $(a_1, \ldots, a_n) \in A$. Donc, A est semi-décidable. □

Lemme 6.23. *Si un ensemble est récursif, il en est de même pour son complémentaire.*

Démonstration. Soit M une machine de Turing qui décide un ensemble récursif A. On pose $M' = \textit{while } M \textit{ do}$ STOP $\textit{od}$. La machine M' s'arrête dans l'état q_3 si la machine M s'arrête dans l'état q_2 et dans l'état q_2 si la machine M s'arrête dans l'état q_3, d'où le résultat. □

Le résultat suivant sera admis. La première implication découle des deux lemmes mentionnés ci-dessus, tandis que la deuxième implication est démontrée dans [4, p. 99].

Lemme 6.24. *Un ensemble A est récursif si et seulement si A et son complémentaire sont semi-décidables.*

On est maintenant en mesure de donner une preuve du théorème 6.20.

Démonstration. L'ensemble A_0 est diophantien de complémentaire non diophantien. Donc, il est semi-décidable et son complémentaire n'est pas semi-décidable puisque la classe des ensembles diophantiens coïncide avec la classe des ensembles semi-décidables. Ainsi, on conclut par le lemme 6.24 que A_0 n'est pas récursif. □

Références

[1] W. J. Ellison, M. Mendès-France, *Les nombres premiers*, Éditions Hermann, 1975.

[2] G. H. Hardy, E. M. Wright, *An introduction to the theory of numbers*, Oxford University Press, 1938.

[3] J. P. Jones, D. Sato, H. Wada, D. Wiens, Diophantine representation of the set of prime numbers, *Am. Math. Mon.* **86** (1976), 449–464.

[4] Y. Matiiassevitch, *Le dixième problème de Hilbert : son indécidabilité*, Éditions Masson, 1995.

[5] M. Davis, Hilbert's tenth problem is unsolvable, *Am. Math. Mon.* **80** (1973), 233–269.

[6] J. P. Jones, Formula for the Nth prime number, *Canad. Math. Bull.* **18** (3), (1975), 433–434.

[7] H. Putnam and J. Robinson, The decision problem for exponential Diophantine equations, *Ann. of Math.* **74** (1961), 425–436.

[8] J. B. Rosser and L. Schoenfeld, Approximate formulas for some functions of prime numbers, *Illinois Jour. of Math.* **6** (1962), 64–94.

[9] H. A. Helfgott, An improved sieve of Eratosthenes, *arXiv* :1712.09130v5, (2019). `https://arxiv.org/abs/1712.09130v5`.

[10] J. Patarin, *Théorie des ensembles et logique mathématique : des infinis mathématiques aux théorèmes de Gödel*, Références Sciences, Ellipses, Paris, (2020), viii+292 pp.

[11] C. L. Siegel, *Topics in Complex Function Theory*, Vol. **1**, Interscience Tracts in Pure and Applied Mathematics, n° 25, Wiley, New York, (1969), ix+186 pp.