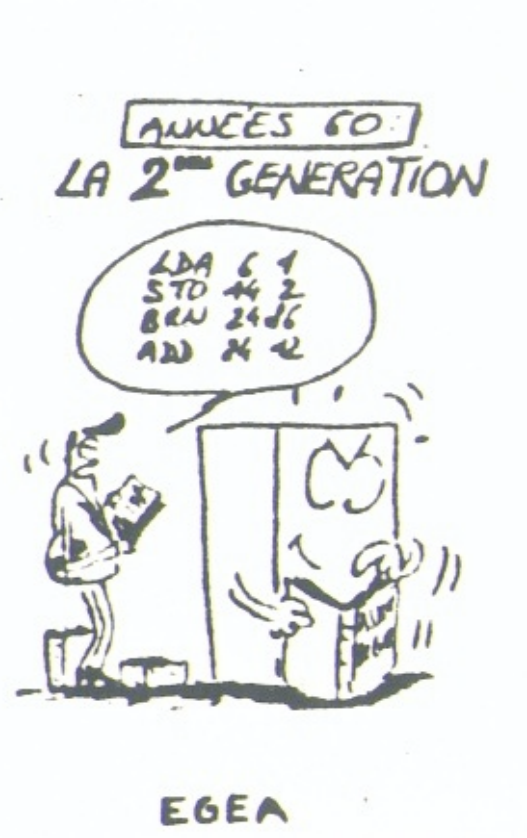
BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
COMPUTATION - INFORMATION

Volumes 2014



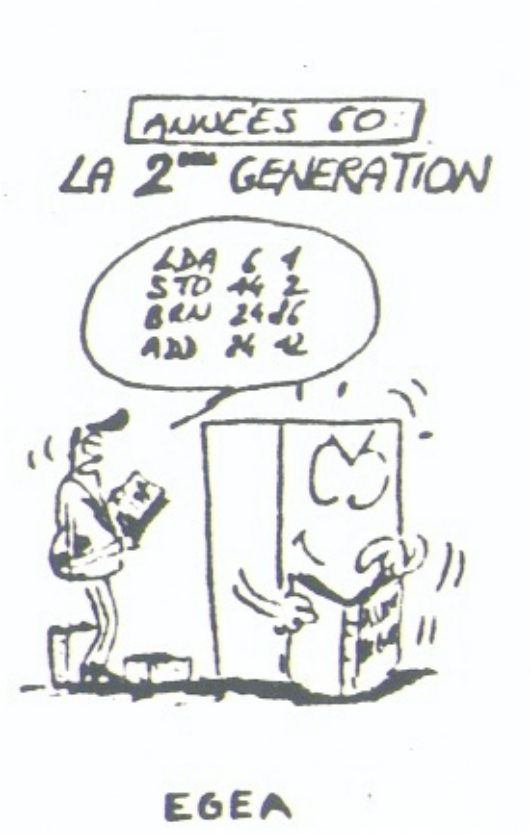
Publication trimestrielle, gratuite, de l'Université d'Aix-Marseille
http://sites.univ-provence.fr/biaa

Dépôt légal : janvier 2014 ISSN 0291 - 5413

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
COMPUTATION - INFORMATION

Volume 97 – mars 2014

Aix*Marseille université



Publication trimestrielle, gratuite, de l'Université d'Aix-Marseille
http://sites.univ-provence.fr/biaa

Impression : mars 2014

ISSN 0291 - 5413

Couverture : dessin de Michel Avezard (Zevar) interprété par la MMIAGe (1985-1987)

B.I.A.A.

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
Revue fondée par Edmond Bianco
Publication trimestrielle de l'Université d'Aix-Marseille
ISSN 0291-5413

Le bulletin d'informatique approfondie et applications est une revue pluridisciplinaire destinée à éclairer les connaissances fondamentales informatiques. Les fondements sont un domaine vaste allant de la structure intérieure de l'ordinateur, où se matérialise la machine universelle, à l'algorithme qui devient programme, pour aboutir à la notion de système. Nous contribuons ainsi à ce que les autres disciplines plus anciennes (sciences humaines et de la société, sciences de la matière et de l'énergie, sciences mathématiques, sciences de la nature, sciences de la terre, sciences de l'univers, sciences de la vie, etc.) n'aient pas tendance à considérer l'informatique comme un simple outil définitivement figé. Il importe de continuer à maîtriser les développements fondamentaux de l'informatique pour que nos disciplines puissent en tirer un meilleur parti.

Notre publication est ouverte à l'ensemble de la communauté scientifique. Le périodique est diffusé vers les bibliothèques universitaires de France et vers quelques bibliothèques des cinq continents.

DIRECTEUR DE LA PUBLICATION
Jean - Michel Knippel

RESPONSABLE DE L'ÉDITION
Eric Olivier

SERVEUR DE PUBLICATION
Christian Blanvillain

SECRÉTARIAT
Kalassoumi Adjilani
Université d'Aix-Marseille,
Site St Charles, Case 33,
3 place Victor Hugo
F -13331 Marseille Cedex 3
Téléphone : +33 (0) 413 550 252
Télécopie : +33 (0) 491 509 110

DÉPOSITAIRE
Université d'Aix-Marseille,
Bibliothèque Universitaire
Site St Charles,
3 place Victor Hugo
F -13331 Marseille Cedex 3
Téléphone : +33 (0) 413 550 579
Télécopie : +33 (0) 491 957 557

IMPRIMEUR
Université d'Aix-Marseille,
Service Reprographie
Site St Charles,
3 place Victor Hugo
F -13331 Marseille Cedex 3
Téléphone : +33 (0) 413 550 626
Télécopie : +33 (0) 413 550 650

TABLE DES MATIÈRES

(2014) B.I.A.A. no **97** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

ÉDITORIAL : Développement en fractions continues et algorithmes

Jean - Michel KNIPPEL

Résumé. – Je rappelle dans cette introduction à l'article de mon collègue Christian Faivre, le principe du développement en fractions continues d'un nombre positif X par des exemples simples. Je ne ferai pas l'économie de vous citer une application connue du domaine de Christian Huygens, un automate planétaire.

Commençons par étudier le principe du développement en fractions continues d'un nombre positif X sur deux exemples simples. Je les ai tirés de documents de Jean-Paul Davalan [1]. Soit le nombre positif X dont on cherche le développement en fractions continues : supposons qu'il s'écrive sous la forme d'une fraction A/B, avec A, B entiers ou d'un nombre décimal D (qui est une fraction particulière). Notons pour l'induction $A = A0$ et $B = B0$: si la fraction $A0/B0$ ne se réduit pas à un entier, on l'écrit $A0/B0 = C0 + 1/(A1/B1)$ où $C0$ est la partie entière de $A0/B0$ et $A1/B1$ est l'inverse (bien défini) de $A0/B0 - C0$. Si la fraction $A1/B1$ ne se réduit pas à un entier, on peut lui appliquer *l'algorithme précédent* qui donne l'entier $C1$ (à la place de $C0$) et la fraction $A2/B2$ (à la place de $A1/B1$) etc... L'algorithme *s'arrête* (nécessairement pour une fraction *non continue*) au bout de k étapes lorsque $Ak/Bk = Ck$ est un entier. Finalement,

$$X = A/B = A0/B0 = C0 + 1/(C1 + 1/(C2 + 1/(C3 + 1/(\cdots + 1/Ck)\cdots)))$$

que l'on écrit $X = [C0\,; C1, C2, C3, \ldots, Ck]$ (le ";" sépare la partie entière du nombre – soit C0 – de sa partie fractionnaire). Le même algorithme s'applique pour les nombres irrationnels (non réductibles à une fraction d'entier), mais le temps d'arrêt étant infini, cela donne... *une fraction continue.*

Exemple 1 : $X = 3.1415926$ est écrit sous la forme $A0/B0 = 31415926/10000000$. La partie entière de $A0/B0 = 3.1415926$ est 3 et on calcule : $A0/B0 - 3 = 1415926/10000000$. L'inverse de $1415926/10000000$ est $A1/B1 = 10000000/1415926$. On reprend le procédé en mettant $A1/B1$ à la place de $A0/B0$.

Exemple 2 : Pour $745/237 = [3, 6, 1, 33]$, les réduites successives sont :

$$\begin{aligned} 3 &= 3/1 \\ 19/6 &= 3 + 1/6 \\ 22/7 &= 3 + 1/(6 + 1/1) \\ 745/237 &= 3 + 1/(6 + 1/(1 + 1/33)) \end{aligned}$$

Les usages des fractions continues sont nombreux. Christian Huygens a construit un automate planétaire pour déterminer les positions relatives des corps célestes du système solaire. Christian Huygens souhaite construire, à l'aide d'un mécanisme de type horlogerie un automate représentant le mouvement des planètes autour du soleil. La difficulté

1. http ://jean-paul.davalan.perso.sfr.fr/

à laquelle il est confronté est liée au rapport de la durée d'une année terrestre et de celle de Saturne. En un an, la Terre tourne de $359^\circ\ 45'\ 40''\ 30'''$ et Saturne de $12^\circ\ 13'\ 34''\ 18'''$. Le rapport est égal à 77 708 431/2 640 858. Combien faut-il de dents sur les deux engrenages supportant respectivement la Terre et Saturne ? Huygens sait que les fractions continues offrent le meilleur compromis, ce qu'il exprime ainsi : « *Or, lorsqu'on néglige à partir d'une fraction quelconque les derniers termes de la série et celles qui la suivent, et qu'on réduit les autres plus le nombre entier à un commun dénominateur, le rapport de ce dernier au numérateur sera voisin de celui du plus petit nombre donné au plus grand ; et la différence sera si faible qu'il serait impossible d'obtenir un meilleur accord avec des nombres plus petits.* » [2]

En mathématiques, et plus précisément en analyse, le théorème de Lochs, démontré par Gustav Lochs en 1964, est un résultat concernant la vitesse de convergence du développement en fractions continues d'un nombre réel typique. Je vous laisse lire l'article de Christian Faivre qui donne des résultats de simulation sur ordinateur basés sur les travaux de Lochs.

2. http ://fr.wikipedia.org/wiki/Fraction_continue

(2014) B.I.A.A. no **97** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

Approximations décimales et développement en fraction continue

Christian FAIVRE [1]

Résumé. – Nous abordons dans cet article (sans chercher à être exhaustif) un problème pratique, celui de déterminer les k premiers quotients partiels (où $k \geq 1$ est un entier fixé) du développement fraction continue d'un nombre x défini "formellement" comme par exemple $x = \sqrt[3]{2}$ ou bien $x = \pi$.

1. CALCULER LES QUOTIENTS PARTIELS

Nous abordons dans cet article (sans chercher à être exhaustif) un problème pratique, celui de déterminer les k premiers quotients partiels (où $k \geq 1$ est un entier fixé) du développement fraction continue d'un nombre x défini "formellement" comme par exemple $x = \sqrt[3]{2}$ ou bien $x = \pi$. En général, le seul moyen est de passer par le développement décimal de x. Le schéma est alors le suivant : à partir des n premières décimales de x (avec n à choisir convenablement) on en déduit une approximation rationnelle r_n de x. On développe alors r_n en fraction continue. Notons que le développement en fraction continue de r_n peut se déterminer exactement car il s'agit d'un nombre rationnel. En effet pour un nombre rationnel p/q l'algorithme de développement en fraction continue montre que l'on a besoin de faire uniquement des opérations sur des nombres entiers. Si r_n est suffisamment proche de x, alors les k premiers quotients partiels de r_n coïncideront avec ceux de x. Le problème est donc le choix de n par rapport à k, en admettant bien entendu que l'on puisse déterminer concrètement les n premières décimales de x, ce qui n'est pas toujours évident selon le nombre x surtout si n est grand.

2. UNE EXCEPTION : LES NOMBRES ALGÉBRIQUES

Dans le cas très particulier d'un nombre algébrique non rationnel x, Lagrange a donné le premier une méthode directe pour déterminer le développement en fraction continue de x sans utiliser le développement décimal. C'est cette méthode que nous allons rapidement exposer ici. Soit $S \in \mathbf{Q}[X]$ de degré ≥ 2 tel que $S(x) = 0$. On ne suppose pas dans ce qui suit que S est le polynôme minimal de x. On peut toujours supposer (quitte à remplacer S par le polynôme $S/PGCD(S, S')$) que S n'a pas de racine multiple. Notons $a_0 = [x]$, de sorte que $a_0 < x < a_0 + 1$. On supposera ici pour simplifier l'exposition de la méthode que x est la seule racine de S dans l'intervalle $[a_0, a_0 + 1]$. On écrit alors :

$$x = a_0 + \frac{1}{\alpha_1},$$

avec $\alpha_1 > 1$. En substituant alors cette valeur dans l'équation à la place de x, on aura (après avoir tout multiplié par α_1^d, où $d = d^\circ S$) que α_1 est racine d'un polynôme S_1 de même degré que S et que α_1 est la seule racine de S_1 dans l'intervalle $(1, \infty)$ car dans

1. Université d'Aix-Marseille

le cas contraire S aurait alors plusieurs racines dans (a_0, a_0+1). On calcule ensuite les valeurs :

$$S_1(1),\ S_1(2),\ S_1(3), \ldots,$$

jusqu'à trouver deux entiers consécutifs a_1 et a_1+1 tels que les quantités $S_1(a_1)$ et $S_1(a_1+1)$ soient de signes différents. Le polynôme S_1 admet donc une racine dans l'intervalle (a_1, a_1+1) d'où $a_1 = [\alpha_1]$ puisque α_1 est la seule racine de S_1 dans $(1, \infty)$. On écrit alors :

$$\alpha_1 = a_1 + \frac{1}{\alpha_2},$$

avec $\alpha_2 > 1$ et on recommence. On remplace α_1 par $a_1 + \frac{1}{\alpha_2}$ dans l'équation $S_1(\alpha_1) = 0$ et on en déduit un polynôme S_2 tel que $S_2(\alpha_2) = 0$ qui n'a qu'une seule racine dans l'intervalle $(1, \infty)$. On calcule les valeurs $S_2(1), S_2(2), \ldots$, etc... On en déduit ainsi de proche en proche une suite $\alpha_1, \alpha_2, \ldots$, de réels > 1 et une suite $a_1, a_2, \ldots$, d'entiers ≥ 1 tels que

$$x = a_0 + \frac{1}{\alpha_1}, \qquad \alpha_k = a_k + \frac{1}{\alpha_{k+1}} \quad (k \geq 1).$$

On a alors par construction

$$x = a_0 + \frac{1}{\alpha_1} = a_0 + \frac{1}{a_1 + \dfrac{1}{\alpha_2}} = \cdots,$$

ce qui donne le développement en fraction continue de x :

$$x = [a_0; a_1, a_2, \ldots].$$

On constate que α_k, l'unique racine > 1 de S_k coincïde avec le quotient complet d'ordre k de x. On a donc l'algorithme suivant pour calculer de proche en proche les polynômes S_k et les entiers a_k :

$$\begin{cases} S_{k+1}(x) = x^d S_k(a_k + \frac{1}{x}) \\ S_{k+1}(a_{k+1}) \cdot S_{k+1}(a_{k+1}+1) < 0 \end{cases}$$

pour tout $k \geq 0$, avec $S_0 = S$ et $a_0 = [x]$. Pour k suffisamment grand, on peut montrer (voir par exemple [BvdP]) que les polynômes S_k deviennent "réduits" i.e. à part la racine $\alpha_k > 1$, les autres racines β de S_k vérifient toutes les inégalités $-1 < Re(\beta) < 0$ et $|\beta| < 1$. Si S_{k+1} est réduit, on peut alors en déduire un procédé rapide pour calculer $a_{k+1} = [\alpha_{k+1}]$. En écrivant

$$S_{k+1}(x) = s_{k+1,d}x^d + \cdots + s_{k+1,1}x + s_{k+1,0},$$

on a $\alpha_{k+1} + \sigma = -s_{k+1,d-1}/s_{k+1,d}$, où σ désigne la somme de toutes les racines de S_{k+1} autres que α_{k+1}. On a donc $u < \alpha_{k+1} < u + (d-1)$, avec $u = -s_{k+1,d-1}/s_{k+1,d}$, ce qui permet de limiter la recherche de a_{k+1} aux d valeurs $[u], [u]+1, \ldots, [u]+(d-1)$.
Dans le cas particulier de $x = \sqrt{N}$, Lagrange a décrit un autre algorithme pour déterminer le développement en fraction continue. On pose $P_0 = 0$ et $Q_0 = 1$, puis on détermine α_0 et a_0 par

$$\alpha_0 = \frac{P_0 + \sqrt{N}}{Q_0}, \quad a_0 = [\alpha_0].$$

Ensuite P_k, Q_k, α_k, a_k étant déterminés, on calcule $P_{k+1}, Q_{k+1}, \alpha_{k+1}, a_{k+1}$ par les formules :

$$P_{k+1} = a_k Q_k - P_k, \qquad Q_{k+1} = \frac{N - P_{k+1}^2}{Q_k}$$

$$\alpha_{k+1} = \frac{P_{k+1} + \sqrt{N}}{Q_{k+1}}, \qquad a_{k+1} = [\alpha_{k+1}].$$

Dès que l'on a trouvé un entier $h \geq 1$ tel que $\alpha_{h+1} = \alpha_1$, on a alors le développement en fraction continue :

$$\sqrt{N} = [a_0; \overline{a_1, \ldots, a_h}].$$

Dans cet algorithme, les α_k coincident avec les quotients complets de $\sqrt{N}$.

3. La méthode de Lochs

Reprenons le problème abordé au début, celui de déterminer les k premiers quotients partiels du développement en fraction continue d'un nombre irrationnel x à partir de son développement décimal. Pour tout $n \geq 1$, soient x_n, y_n, les approximations décimales d'ordre n respectivement par défaut et par excès de x i.e.

$$x_n = \frac{[10^n x]}{10^n}, \qquad y_n = x_n + \frac{1}{10^n}.$$

Suivant Lochs [Loc64], supposons que l'on ait trouvé un entier $n \geq 1$ pour lequel les développements en fraction continue de x_n et y_n (ces deux développements sont finis puisqu'il s'agit de nombres rationnels) coïncident jusqu'à l'ordre k i.e.

$$x_n = [u_0; u_1, \ldots, u_k, \ldots], \quad y_n = [u_0; u_1, \ldots, u_k, \ldots]. \tag{1}$$

Alors si $x = [a_0; a_1, a_2, \ldots]$, on aura

$$a_0 = u_0, \ldots, a_k = u_k,$$

en d'autres termes $u_0, \ldots, u_k$ seront les k premiers quotients partiels exacts de x. En effet les nombres qui admettent un développement en fraction continue qui commence par $u_0, \ldots, u_k$ est un intervalle (intervalle fondamental). Désignons par $k_n(x)$ pour tout entier $n \geq 1$, le nombre maximal de quotients partiels de x donnés par x_n et y_n suivant le procédé ci-dessus c'est-à-dire que $k_n(x)$ est le plus grand entier $p \geq 0$ tel que l'on puisse écrire

$$x_n = [u_0; u_1, \ldots, u_p, \ldots], \qquad y_n = [u_0; u_1, \ldots, u_p, \ldots], \tag{2}$$

pour certains entiers $u_0, \ldots, u_p$. Notons qu'un tel entier p existe toujours ; en effet en posant $u_0 = [x]$, on a $[x_n] = u_0$ et $[y_n] = u_0$ ou bien $y_n = u_0 + 1$. Dans ce dernier cas, on peut écrire $y_n = [u_0; 1]$. La quantité $k_n(x)$ peut donc s'interpréter comme le nombre maximal de quotients partiels de x donnés par les n premières décimales de x. Remarquons que le développement en fraction continue d'un nombre rationnel n'est pas unique. En fait il existe exactement deux développements. Habituellement celui où le dernier quotient partiel est ≥ 2 est appelé le développement régulier et l'autre le développement irrégulier. Il faut souligner que nous ne choisisssons pas systématiquement un de ces deux développements au profit de l'autre, mais on essaie d'obtenir le maximum de quotients partiels en jouant sur les deux développements. Par exemple si le développement décimal de x commence par $x = 0.5\ldots$, alors

$$x_1 = 0.5 = [0; 2], \qquad y_1 = 0.6 = [0; 1, 1, 2].$$

Mais $[0; 2] = [0; 1, 1]$, donc ici on a $k_1(x) = 2$ (et non $k_1(x) = 0$) et on en déduit $x = [0; 1, 1, \ldots]$. Il faut cependant souligner que cette situation est plutôt exceptionnelle car

dans la majorité des cas, les développements réguliers de x_n et y_n se présenteront suivant le schéma :

$$\begin{cases} x_n = [u_0; u_1, \ldots, u_l, u_{l+1}, u_{l+2}, \ldots] \\ y_n = [u_0; u_1, \ldots, u_l, u'_{l+1}, u'_{l+2}, \ldots] \end{cases}$$

avec $u_{l+1} \neq u'_{l+1}$ et donc on aura $k_n(x) = l$. A titre illustratif, considérons maintenant des exemples de calcul de quantités $k_n(x)$. Prenons $x = \pi = 3.14159265\cdots$. Nous avons les développements en fraction continue :

$$\begin{cases} 3.1 = [3; 10] \\ 3.2 = [3; 5] \end{cases} \qquad \begin{cases} 3.14 = [3; 7, 7] \\ 3.15 = [3; 6, 1, 2] \end{cases}$$

Ainsi $k_1(\pi) = k_2(\pi) = 0$. Ensuite

$$\begin{cases} 3.141 = [3; 7, 10, 1, 5, 2] \\ 3.142 = [3; 7, 23, 1, 2] \end{cases}$$

et donc $k_3(\pi) = 1$. Même avec six décimales nous n'obtenons qu'un seul quotient partiel puisque

$$\begin{cases} 3.141592 = [3; 7, 15, 1, 84, 6, 2] \\ 3.141593 = [3; 7, 16, 983, 4, 2] \end{cases}$$

d'où $k_6(\pi) = 1$. L'exemple de π pourrait donc faire penser qu'en général $k_n(x)$ est beaucoup plus petit que n. De manière étonnante cette affirmation est fausse en général et c'est précisément le résultat opposé qui est vrai. En 1964, G. Lochs [Loc64] a démontré le résultat suivant :

Théorème 3.1. *Pour presque tout nombre irrationnel x (au sens de la mesure de Lebesgue), on a*

$$\lim_{n\to\infty} \frac{k_n(x)}{n} = \frac{6 \log 2 \log 10}{\pi^2} \simeq 0.9702.$$

La constante $6 \log 2 \log 10/\pi^2$ étant très proche de 1, on peut presque dire que les n premières décimales déterminent les n premiers quotients partiels de x. Ainsi les conclusions que l'on peut tirer de l'examen des 6 premières décimales de π sont tout à fait erronnées. L'explication de ce phénomène réside dans le développement en fraction continue de π :

$$\pi = [3; 7, 15, 1, 292, \ldots].$$

On constate en effet la présence du grand quotient partiel 292, un nombre inhabituellement grand pour un quatrième quotient partiel. Il est donc intéressant de regarder un plus grand nombre de décimales. Il a été prouvé par Lochs lui-même [Loc63] que les 1000 premières décimales de π déterminent 968 quotients partiels exacts i.e. $k_{1000}(\pi) = 968$. Avec 10000 décimales, on obtient 9757 quotients partiels. Il semble donc que l'on ait

$$\lim_{n\to\infty} \frac{k_n(\pi)}{n} = \frac{6 \log 2 \log 10}{\pi^2},$$

c'est-à-dire que π n'est pas dans l'ensemble exceptionnel du théorème de Lochs. Pour certains nombres irrationnels x, $k_n(x)$ peut être plus grand que n (c'est le cas pour le Nombre d'Or G, voir l'exemple après le corollaire 4.2), en revanche $k_n(x)$ est toujours un $O(n)$. On a en effet le résultat suivant [Fai01] :

Théorème 3.2. *Pour tout nombre irrationnel x et $n \geq 1$, on a l'inégalité*

$$k_n(x) \leq Cn + \frac{1}{2},$$

avec $C = \log 10/(2 \log G) \simeq 2.3924$. La constante Cest optimale.

Le théorème de Lochs est un résultat relatif à deux développements distincts d'un même nombre réel (en l'occurrence le développement décimal et le développement en fraction continue). La constante $(6 \log 2 \log 10)/\pi^2$ est égale au rapport des entropies

$$\frac{h_m(S)}{h_\mu(T)}$$

où S désigne la transformation associée au développement décimal (i.e. $S(x) = 10x - [10x]$) et T la transformation de Gauss associée au développement en fraction continue. En utilisant le théorème de Shannon–McMillan–Breiman, Bosma, Dajany et Kraaikamp [BDK06] ont pu généraliser le théorème de Lochs à d'autres développements.

4. Améliorations du théorème de Lochs

Je me suis intéressé au théorème de Lochs et j'ai pu améliorer ce théorème dans deux directions différentes. Dans la première, je montre que la suite $k_n(x)/n$ admet toujours une limite si x a une constante de Lévy et si les quotients partiels de x ne sont pas trop grands [Fai01] :

Théorème 4.1. *Soit $x = [a_0; a_1, a_2, \ldots,]$ un nombre irrationnel ayant une constante de Lévy et tel que $a_n = o(\alpha^n)$ pour tout $\alpha > 1$. Alors*

$$\lim_{n\to\infty} \frac{k_n(x)}{n} = \frac{\log 10}{2\beta(x)}.$$

Le théorème de Lochs découle de ce résultat puisque presque tout x admet une constante de Lévy égale à $\pi^2/(12 \log 2)$ (théorème de Lévy) et que toujours pour presque tout x, on a par exemple $a_n(x) = O(n^2)$. Ce dernier résultat découlant du théorème classique de Bernstein sur les fractions continues puisque la série $\sum n^{-2}$ est convergente. Une application importante du théorème 4.1 concerne le cas des nombres quadratiques. Le développement en fraction continue d'un nombre quadratique étant périodique et tout nombre quadratique admettant une constante de Lévy, on peut alors énoncer :

Corollaire 4.2. *Pour tout nombre quadratique x, on a*

$$\lim_{n\to\infty} \frac{k_n(x)}{n} = \frac{\log 10}{2\beta(x)}.$$

Par exemple si l'on prend $x = G$, le Nombre d'Or, on obtient

$$\lim_{n\to\infty} \frac{k_n(x)}{n} = \frac{\log 10}{2 \log G} \simeq 2.3924,$$

puisque $\beta(G) = \log G$. Ce qui montre bien que l'on ne peut améliorer la constante C dans le théorème 3.2. Dans [Fai01], il est aussi prouvé que si pour un irrationnel x, on a $\lim_{n\to\infty} \frac{1}{n} \log q_n(x) = \infty$, alors

$$\lim_{n\to\infty} \frac{k_n(x)}{n} = 0.$$

C'est le cas par exemple de e. Le théorème 4.1 a été amélioré récemment par Wu [Wu06] qui a montré en substance que la condition sur les quotients partiels $a_n = o(\alpha^n)$ pouvait être supprimée. Précisément, le résultat de Wu est que l'on a toujours pour tout nombre irrationnel x :

$$\liminf_{n\to\infty} \frac{k_n(x)}{n} = \frac{\log 10}{2\beta^*(x)}, \quad \limsup_{n\to\infty} \frac{k_n(x)}{n} = \frac{\log 10}{2\beta_*(x)},$$

où $\beta_*(x)$ et $\beta^*(x)$ sont définis par :

$$\beta_*(x) = \liminf_{n\to\infty} \frac{\log q_n(x)}{n}, \quad \beta^*(x) = \limsup_{n\to\infty} \frac{\log q_n(x)}{n}.$$

Le deuxième résultat mentionné sur k_n est un théorème de large déviations [Fai97] :

Théorème 4.3. *Pour tout $\epsilon > 0$, il existe des constantes C, r avec $C > 0$ et $0 < r < 1$ telles que*

$$m\left\{x \in [0,1];\ \left|\frac{k_n(x)}{n} - a\right| \geq \epsilon\right\} \leq Cr^n \qquad (n \geq 1),$$

où $a = 6\log 2 \log 10/\pi^2$.

Ainsi pour tout $\epsilon > 0$,

$$\sum_{n\geq 1} m\left\{x \in [0,1];\ \left|\frac{k_n(x)}{n} - a\right| \geq \epsilon\right\} < \infty,$$

et le résultat de Lochs s'en déduit par une application du lemme de Borel–Cantelli. J'ai obtenu plus précisément les deux inégalités suivantes :

$$\begin{aligned}
\limsup_{n\to\infty} \frac{1}{n} \log\left[m\left\{x \in [0,1];\ \frac{k_n(x)}{n} \leq a - \epsilon\right\}\right] &\leq \theta_1(\epsilon), \\
\limsup_{n\to\infty} \frac{1}{n} \log\left[m\left\{x \in [0,1];\ \frac{k_n(x)}{n} \geq a + \epsilon\right\}\right] &\leq \theta_2(\epsilon),
\end{aligned}$$

avec

$$\theta_1(\epsilon) = \inf_{0<t<\frac{1}{2}} \frac{1}{t+1}\Big(-t\log 10 + (a-\epsilon)\log\lambda(2-2t)\Big)$$

et

$$\theta_2(\epsilon) = \inf_{t>0}\Big(t\log 10 + (a+\epsilon)\log\lambda(2+2t)\Big).$$

Dans ces deux formules, $\lambda(2-2t)$ et $\lambda(2+2t)$ sont les valeurs propres dominantes des opérateurs de transfert L_{2-2t} et L_{2+2t} définis dans le chapitre 3. On peut montrer que $\theta_1(\epsilon) < 0$ et $\theta_2(\epsilon) < 0$. Par exemple pour prouver que $\theta_1(\epsilon) < 0$, on considère la fonction

$$h(t) = -t\log 10 + (a-\epsilon)\log\lambda(2-2t),$$

définie pour $t < \frac{1}{2}$. On a

$$h'(t) = -\log(10) - 2(a-\epsilon)\frac{\lambda'(2-2t)}{\lambda(2-2t)}.$$

Comme on l'a vu au chapitre 3,

$$\lambda(2) = 1, \qquad \lambda'(2) = -\frac{\pi^2}{12\log 2},$$

donc $h(0) = 0$ et $h'(0) < 0$. Ainsi $h(t) < 0$ pour $t > 0$ suffisamment petit ce qui implique que $\theta_1(\epsilon) < 0$.

5. Un théorème central limite. Application

Dans [Fai98], un théorème central limite est énoncé pour la suite $(k_n)_{n\geq 1}$.

Théorème 5.1. *Il existe une constante $\sigma > 0$ telle que*

$$\lim_{n\to\infty} m\left\{x \in [0,1];\ \frac{k_n(x) - na}{\sigma\sqrt{n}} \leq z\right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt,$$

pour tout $z \in \mathbf{R}$.

Récemment Wu [Wu] a montré que $(k_n)_{n\geq 1}$ vérifiait aussi une loi du logarithme itérée :

$$\limsup_{n\to\infty} \frac{k_n(x) - na}{\sigma\sqrt{2n\log\log n}} = 1, \qquad \liminf_{n\to\infty} \frac{k_n(x) - na}{\sigma\sqrt{2n\log\log n}} = -1,$$

pour presque tout irrationnel x. La constante σ est reliée à une autre constante importante dans la théorie métrique des fractions continues. Comme on l'a vu au chapitre 4, la suite $(\log q_n)_{n\geq 1}$ vérifie un théorème central limite :

$$\lim_{n\to\infty} m\left\{x \in [0,1],\ \frac{\log q_n(x) - nb}{\sigma_1\sqrt{n}} \leq z\right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt,$$

pour une certaine constante $\sigma_1 > 0$ et avec $b = \pi^2/(12\log 2)$. On a alors que les constantes σ and σ_1 sont reliées par la formule ([Fai98] p. 462) :

$$\sigma^2 = 864\, \frac{(\log 2)^3 \log 10}{\pi^6}\, \sigma_1^2. \tag{3}$$

B. Vallée [Val97] a montré une belle expression entre σ_1 et la valeur propre dominante de l'opérateur de transfert L_2 (cf. chapitre 3) :

$$\sigma_1^2 = \lambda''(2) - (\lambda'(2))^2.$$

On en déduit alors une relation simple entre σ^2 et la constante de Hensley α_H qui intervient dans l'analyse en moyenne de l'algorithme d'Euclide. En effet, en utilisant (3) ainsi que l'expression suivante pour la constante de Hensley [FV98]

$$\alpha_H = -\frac{\lambda''(2) - (\lambda'(2))^2}{(\lambda'(2))^3},$$

on obtient

$$\sigma^2 = \frac{\log 10}{2}\alpha_H.$$

Des simulations sur ordinateur à partir de la loi de k_n avaient permis de donner dans [Fai01] l'approximation

$$\sigma \approx 0.769. \tag{4}$$

On peut faire maintenant beaucoup mieux. En effet Lohte [Lho04] a obtenu récemment des estimations précises de la constante de Hensley à partir de l'algorithme DFV introduit la première fois par Daudé, Flajolet et Vallée [DFV97]. On a par exemple

$$\alpha_H = 0.5160624089\cdots, \tag{5}$$

avec 10 décimales exactes, améliorant de ce fait une estimation antérieure de Flajolet et Vallée [FV00]. En utilisant (5), on obtient alors l'estimation suivante pour σ qui précise (4) :

$$\sigma = 0.7708039990\cdots$$

k	10	20	30	40	50	60	70	80	90	100
$N_{k,\alpha}$	16	28	40	51	62	74	85	96	107	118

TABLE 1. Nombres de décimales

La distribution exacte de k_n (lorsque x est choisi au hasard dans $[0,1]$ suivant une répartition uniforme) a été déterminée pour $1 \leq n \leq 7$. Même pour les petites valeurs de n et en fait pour $n \geq 4$, on peut considérer la distribution de k_n comme approximativement normale si on procède à une correction de continuité comme c'est habituellement le cas quand on essaie d'approcher une distribution discrète par une distribution continue. Des calculs sur ordinateur (voir annexe en fin de la thèse) ont justifié que pour tout entier i, on obtient une bonne approximation de $P(k_n \leq i)$ en prenant $P(N_n \leq i+1)$, où N_n suit une distribution normale de moyenne na et d'écart type $\sigma\sqrt{n}$. En utilisant cette approximation, on peut donner une règle pratique pour savoir combien il faut prendre de décimales si l'on veut obtenir (au moins) k quotients partiels exacts dans le développement en fraction continue d'un nombre irrationnel. La règle est la suivante :

Règle. *Pour un niveau de confiance de $\alpha\%$, (par exemple 95%), si l'on veut k quotients partiels (au moins), il faut prendre un nombre de décimales égal à $N_{k,\alpha}$ avec*

$$N_{k,\alpha} = \frac{\left(-q\sigma + \sqrt{q^2\sigma^2 + 4ak}\right)^2}{4a^2},$$

où q désigne le $(1-\alpha)$-quantile pour la distribution normale $N(0,1)$.

Essayons de justifier cette règle. Si l'on veut au moins k quotients partiels, avec un niveau de confiance de α, on doit prendre n tel que

$$P(k_n \geq k) \geq \alpha.$$

Mais comme k_n ne prend que des valeurs entières,

$$P(k_n \geq k) = 1 - P(k_n \leq k-1) \simeq 1 - P(N_n \leq k),$$

grâce à l'approximation vue plus haut. Ainsi il faut prendre n tel que

$$P(N_n \leq k) \leq 1 - \alpha.$$

Mais

$$P(N_n \leq k) = \Psi\left(\frac{k - na}{\sigma\sqrt{n}}\right),$$

où Ψ est la fonction de répartition de la loi normale $N(0,1)$. Donc n doit être choisi de telle sorte que

$$\frac{k - na}{\sigma\sqrt{n}} \leq q,$$

ce qui donne

$$n \geq \frac{\left(-q\sigma + \sqrt{q^2\sigma^2 + 4ak}\right)^2}{4a^2}.$$

Remarque. Il n'existe pas de borne *absolue* N_k pour le nombre de décimales de telle sorte que pour tout irrationnel (ou même pour presque tout irrationnel) en prenant N_k

k	200	300	400	500	1000	2000	3000	4000	5000	10000
$N_{k,\alpha}$	226	334	440	546	1074	2122	3166	4208	5248	10440

TABLE 2. Nombres de décimales

décimales on soit sûr d'obtenir au moins k quotients partiels. Ainsi on ne peut améliorer le résultat précédent sans spécifier de niveau de confiance.

Les tableaux 1 et 2 donnent le nombre de décimales pour plusieurs valeurs de k et pour un seuil de confiance de 95% ($\alpha = 0.95$). Par exemple si l'on veut 1000 quotients partiels, on prendra 1074 décimales.

RÉFÉRENCES

[BDK06] W. Bosma, K. Dajani, and C. Kraaikamp. Entropy quotients and correct digits in number-theoretic expansions. *IMS Lecture Notes–Monograph Series Dynamics and Stochastics*, 48 :176–188, 2006.

[BvdP] E. Bombieri and A. van der Poorten. Continued fractions of algebraic numbers. Computational Algebra and Number Theory, Sydney 1992, W. Bosma and A. van der Poorten eds.,. (Kluwer 1995), 137–152.

[DFV97] H. Daudé, P. Flajolet, and B. Vallée. An average-case analysis of the Gaussian algorithm for lattice reduction. *Comb. Probab. Comput.*, 6(4) :397–433, 1997.

[Fai97] C. Faivre. On decimal and continued fraction expansions of a real number. *Acta Arith.*, 82(2) :119–128, 1997.

[Fai98] C. Faivre. A central limit theorem related to decimal and continued fraction expansions. *Arch. Math. (Basel)*, 70 :455–463, 1998.

[Fai01] C. Faivre. On calculating a continued fraction expansion from a decimal expansion. *Acta Sci. Math. (Szeged)*, 67 :505–519, 2001.

[FV98] P. Flajolet and B. Vallée. Continued fraction algorithms, functional operators, and structure constants. *Theor. Comput. Sci.*, 194(1–2) :1–34, 1998.

[FV00] P. Flajolet and B. Vallée. Continued fractions, comparison algorithms, and fine structure constants. Théra, Michel (ed.), Constructive, experimental, and nonlinear analysis. Selected papers of a workshop, Limoges, France, September 22-23, 1999. Providence, RI : American Mathematical Society (AMS), publ. for the Canadian Mathematical Society. CMS Conf. Proc. 27, 53-82 (2000)., 2000.

[Lho04] L. Lhote. Computation of a Class of Continued Fraction Constants. In *Proceedings of Alenex–ANALCO04*, pages 199–210, 2004.

[Loc63] G. Lochs. Die ersten 968 Kettenbruchnenner von π. *Monatsh. Math.*, 67 :311–316, 1963.

[Loc64] G. Lochs. Vergleich der Genauigkeit von Dezimalbruch und Kettenbruch. *Abh. Math. Sem. Univ. Hamburg*, 27 :142–144, 1964.

[Val97] B. Vallée. Opérateurs de Ruelle–Mayer généralisés et analyse en moyenne des algorithmes d'Euclide et de Gauss. *Acta Arith.*, 81 :101–144, 1997.

[Wu] J. Wu. An Iterated Logarithm Law Related to Decimal and Continued Fraction Expansions. To appear in Monatsh. Math.

[Wu06] J. Wu. Continued fraction and decimal expansions of an irrational number. *Adv. in Math.*, 206 :684–694, 2006.

6. ANNEXE : NORMALITÉ DE k_n

Comme on l'a vu, lorsque x est choisi au hasard dans $[0,1]$ suivant une répartition uniforme la quantité

$$\frac{k_n - na}{\sigma\sqrt{n}}$$

avec

$$a = \frac{6\log 2\log 10}{\pi^2} \simeq 0.9702, \qquad \sigma \simeq 0.7708039990,$$

tend vers une loi normale $N(0,1)$ quand $n \to \infty$. Nous avons affirmé que la loi k_n pouvait être considérée comme approximativement normale dès que $n \geq 4$ à condition de faire une correction de continuité i.e. pour tout entier i, on obtient une bonne approximation de $P(k_n \leq i)$ en prenant $P(N_n \leq i+1)$, où N_n suit une distribution normale de moyenne na et d'écart type $\sigma\sqrt{n}$. Le but de cette annexe est de fournir une justification numérique de cette affirmation.

Dans les divers tableaux ci-après, on a fait figurer à titre comparatif l'approximation de $P(k_n \leq i)$ sans correction de continuité i.e. $P(N_n \leq i)$ et l'approximation avec correction de continuité $P(N_n \leq i+1)$. Les diverses probabilités ont été arrondies avec deux chiffres après la virgule. Pour $1 \leq n \leq 7$, les probabilités $P(k_n \leq i)$ ont été calculées a partir de la loi exacte de k_n et par simulations pour $n \geq 8$. On peut voir sur les divers tableaux que l'approximation est très correcte dés que $n \geq 4$. En revanche, l'approximation par $P(N_n \leq i)$ n'est pas très bonne (surtout dans la partie centrale) même pour $n = 100$.

$n=1$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.40	0.10	0.52
1	0.90	0.52	0.91
2	1.00	0.91	1.00

$n=2$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.15	0.04	0.19
1	0.49	0.19	0.52
2	0.82	0.52	0.83
3	0.98	0.83	0.97
4	1.00	0.97	1.00

$n = 3$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.06	0.01	0.08
1	0.23	0.08	0.25
2	0.53	0.25	0.53
3	0.80	0.53	0.79
4	0.94	0.79	0.94
5	0.99	0.94	0.99
6	1.00	0.99	1.00
7	1.00	1.00	1.00

$n = 4$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.02	0.01	0.03
1	0.10	0.03	0.11
2	0.29	0.11	0.28
3	0.54	0.28	0.53
4	0.77	0.53	0.77
5	0.92	0.77	0.92
6	0.98	0.92	0.98
7	1.00	0.98	1.00
8	1.00	1.00	1.00
9	1.00	1.00	1.00

$n = 5$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.01	0.00	0.01
1	0.04	0.01	0.05
2	0.14	0.05	0.14
3	0.31	0.14	0.31
4	0.54	0.31	0.53
5	0.76	0.53	0.75
6	0.90	0.75	0.89
7	0.97	0.89	0.97
8	0.99	0.97	0.99
9	1.00	0.99	1.00
10	1.00	1.00	1.00
11	1.00	1.00	1.00

$n = 6$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.00	0.00	0.01
1	0.01	0.01	0.02
2	0.06	0.02	0.07
3	0.16	0.07	0.17
4	0.34	0.17	0.33
5	0.55	0.33	0.54
6	0.74	0.54	0.73
7	0.88	0.73	0.88
8	0.96	0.88	0.95
9	0.99	0.95	0.99
10	1.00	0.99	1.00
11	1.00	1.00	1.00
12	1.00	1.00	1.00
13	1.00	1.00	1.00
14	1.00	1.00	1.00

$n = 7$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.00	0.00	0.00
1	0.01	0.00	0.01
2	0.03	0.01	0.03
3	0.08	0.03	0.09
4	0.19	0.09	0.19
5	0.35	0.19	0.35
6	0.55	0.35	0.54
7	0.73	0.54	0.72
8	0.87	0.72	0.86
9	0.95	0.86	0.94
10	0.98	0.94	0.98
11	1.00	0.98	0.99
12	1.00	0.99	1.00
13	1.00	1.00	1.00
14	1.00	1.00	1.00

$n = 8$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.01	0.00	0.01
3	0.04	0.01	0.04
4	0.10	0.04	0.10
5	0.21	0.10	0.21
6	0.37	0.21	0.36
7	0.55	0.36	0.54
8	0.72	0.54	0.71
9	0.85	0.71	0.85
10	0.93	0.85	0.93
11	0.98	0.93	0.97
12	0.99	0.97	0.99
13	1.00	0.99	1.00
14	1.00	1.00	1.00
15	1.00	1.00	1.00

$n = 9$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
1	0.00	0.00	0.00
2	0.00	0.00	0.01
3	0.02	0.01	0.02
4	0.05	0.02	0.05
5	0.12	0.05	0.12
6	0.23	0.12	0.23
7	0.38	0.23	0.38
8	0.55	0.38	0.55
9	0.72	0.55	0.71
10	0.84	0.71	0.84
11	0.92	0.84	0.92
12	0.97	0.92	0.97
13	0.99	0.97	0.99
14	1.00	0.99	1.00
15	1.00	1.00	1.00

$n = 10$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
2	0.00	0.00	0.00
3	0.01	0.00	0.01
4	0.02	0.01	0.03
5	0.06	0.03	0.06
6	0.13	0.06	0.13
7	0.24	0.13	0.24
8	0.39	0.24	0.39
9	0.56	0.39	0.55
10	0.71	0.55	0.70
11	0.83	0.70	0.83
12	0.91	0.83	0.91
13	0.96	0.91	0.96
14	0.99	0.96	0.99
15	1.00	0.99	1.00
16	1.00	1.00	1.00

$n = 50$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
32	0.00	0.00	0.00
33	0.00	0.00	0.00
34	0.01	0.00	0.01
35	0.01	0.01	0.01
36	0.02	0.01	0.02
37	0.03	0.02	0.03
38	0.04	0.03	0.04
39	0.06	0.04	0.06
40	0.08	0.06	0.08
41	0.12	0.08	0.12
42	0.16	0.12	0.16
43	0.21	0.16	0.20
44	0.26	0.20	0.26
45	0.32	0.26	0.32
46	0.39	0.32	0.39
47	0.47	0.39	0.46
48	0.54	0.46	0.54
49	0.61	0.54	0.61
50	0.68	0.61	0.68
51	0.74	0.68	0.74
52	0.80	0.74	0.79
53	0.84	0.79	0.84
54	0.88	0.84	0.88
55	0.92	0.88	0.92
56	0.94	0.92	0.94
57	0.96	0.94	0.96
58	0.97	0.96	0.97
59	0.98	0.97	0.98
60	0.99	0.98	0.99
61	0.99	0.99	0.99
62	1.00	0.99	1.00
63	1.00	1.00	1.00
64	1.00	1.00	1.00

$n = 100$			
i	$P(k_n \leq i)$	$P(N_n \leq i)$	$P(N_n \leq i+1)$
75	0.00	0.00	0.00
76	0.00	0.00	0.00
77	0.01	0.00	0.01
78	0.01	0.01	0.01
79	0.01	0.01	0.01
80	0.02	0.01	0.02
81	0.02	0.02	0.03
82	0.03	0.03	0.03
83	0.04	0.03	0.05
84	0.06	0.05	0.06
85	0.08	0.06	0.08
86	0.10	0.08	0.10
87	0.12	0.10	0.12
88	0.15	0.12	0.15
89	0.18	0.15	0.18
90	0.22	0.18	0.22
91	0.26	0.22	0.26
92	0.30	0.26	0.30
93	0.35	0.30	0.35
94	0.40	0.35	0.40
95	0.45	0.40	0.45
96	0.50	0.45	0.50
97	0.55	0.50	0.55
98	0.60	0.55	0.60
99	0.65	0.60	0.65
100	0.70	0.65	0.70
101	0.74	0.70	0.74
102	0.78	0.74	0.78
103	0.82	0.78	0.82
104	0.85	0.82	0.85
105	0.88	0.85	0.88
106	0.90	0.88	0.90
107	0.92	0.90	0.92
108	0.94	0.92	0.94
109	0.95	0.94	0.95
110	0.96	0.95	0.97
111	0.97	0.97	0.97
112	0.98	0.97	0.98
113	0.99	0.98	0.99
114	0.99	0.99	0.99
115	0.99	0.99	0.99
116	1.00	0.99	1.00
117	1.00	1.00	1.00
118	1.00	1.00	1.00

(2014) B.I.A.A. no **97** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

Qu'est-ce qu'une machine ? (I/III)

Eric OLIVIER [1 2]

Résumé. – La théorie des machines de Turing reformule et clarifie un certain nombre de questions portant sur les fondements (logiques) des mathématiques. Ainsi la question *"Qu'est-ce qu'une machine ?"* est-elle équivalente à la question *"Qu'est-ce qu'un calcul ?"*. Richard Feynman résume cela en affirmant que *n'importe quelle procédure de calcul à laquelle on pourrait penser, est équivalente au calcul d'une machine de Turing – les fonctions récursives générales sont Turing-calculables et vice-versa – et on peut donc prendre "Turing-calculable" pour un synonyme effectif de "calculable"*. Notons enfin que le calcul *automatique* (i.e. le calcul effectué par une machine de Turing) distingue la notion de *proposition démontrable* de celles de *proposition vraie, décidable, indécidable* : cela éclaire les travaux révolutionnaires de Gödel sur la *complétude* et la *consistance* des théories mathématiques.

1. ALPHABETS ET LANGAGES

Un alphabet $\mathcal{A}$ est un ensemble fini ou infini dénombrable dont les éléments sont appelés lettres, chiffres, symboles ou encore digits ; on note $\mathcal{A}^{(0)} := \{\phi\}$ (ϕ est le mot vide) et pour tout entier $n \geq 1$ l'ensemble $\mathcal{A}^{(n)}$ est constitué des mots finis composés de (la concaténation de) n lettres prises dans $\mathcal{A}$. (Pour tout $n \geq 1$, il est facile de mettre en bijection $\mathcal{A}^{(n)}$ avec le produit cartésien $\mathcal{A}^n = \mathcal{A} \times \cdots \times \mathcal{A}$ ayant n facteurs égaux à $\mathcal{A}$.) Le langage associé à $\mathcal{A}$ est

$$\mathcal{A}^* = \bigcup_{n=0}^{\infty} \mathcal{A}^{(n)}.$$

(Ici on utilise un cas particulier de la notation de Kleene). L'application $(w, m) \mapsto wm$, définie sur $\mathcal{A}^* \times \mathcal{A}^*$ et à valeur dans $\mathcal{A}^*$, est appelée la concaténation : elle munit $\mathcal{A}^*$ d'une structure de monoïde (non commutatif) dont l'élément neutre est le mot vide. Soit w et m deux mots de $\mathcal{A}^*$; on dit que m est un préfixe (resp. suffixe) de w s'il existe $w' \in \mathcal{A}^*$ tel que $w = mw'$ (resp. $w = w'm$) ; s'il existe deux mots w' et w'' dans $\mathcal{A}^*$ tels que $w = w'mw''$, alors on dit que m factorise w. La longueur d'un mot $w \in \mathcal{A}^*$, notée $|w|$, est le nombre de lettres qui le composent (avec la convention que $|\phi| = 0$) ; il est alors facile de voir que $w \mapsto |w|$ est un morphisme de monoïde de $\mathcal{A}^*$ sur le monoïde additif $(\mathbb{N}, +)$ des nombres entiers positifs ou nuls. Dans la suite nous supposerons toujours les alphabets considérés totalement ordonnés ; cela permet de définir sur $\mathcal{A}^*$ un ordre total naturel appelé *ordre canonique*. Pour w et m deux mots distincts de $\mathcal{A}^*$ on note $w < m$ lorsque $|w| < |m|$ ou lorsque w précède m dans l'ordre lexicographique, dans le cas où $|w| = |m|$ (notons que l'ordre canonique diffère de l'ordre lexicographique). Un sous-ensemble $\mathcal{L}$ de $\mathcal{A}^*$ est appelé un *langage* sur $\mathcal{A}$.

1. GDAC-I2M UMR 7373 CNRS Université d'Aix-Marseille
2. eric.olivier@univ-amu.fr

2. MACHINES DE TURING

Le concept de *machine* introduit par Turing dans son article fondateur de 1936 [Tur36] peut se décrire comme un *mécanisme abstrait* pouvant se trouver dans un nombre fini d'*états internes*, dont on note l'ensemble $\mathcal{Q} = \{q_1 = \mathtt{I}, \ldots, q_r = \mathtt{F}\}$. La machine est pourvue d'une tête de lecture/écriture permettant de lire/écrire sur un *ruban* des lettres prises dans un alphabet $\mathcal{A}$ – *fini* – appelé *alphabet d'exécution* ; le ruban lui-même est formé d'une infinité (linéaire bilatérale) de cases. Les cases du ruban sont vides, à l'exception d'un nombre *fini* d'entre elles où figurent des lettres de l'alphabet d'exécution ; par convention, l'alphabet d'exécution contient le symbole $\star$ permettant d'indiquer qu'une case est vide. Après chaque lecture/écriture, la tête peut se déplacer d'une case à droite (action `R`) où à gauche (action `L`) ou encore rester immobile (action `S`). Ici les états internes $q_1 = \mathtt{I}$ et $q_r = \mathtt{F}$ sont respectivement l'état initial et l'état final (`I` et `F` seront toujours supposés distincts). Initialement, la machine est positionnée dans son état initial `I`, la tête étant prête à lire la lettre de la case en position de lecture/écriture. Plus précisément, c'est la *table de transition* qui définit les actions de la machine en fonction des lettres figurant sur le ruban : on la représente comme une application

$$T : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q} \times \mathcal{A} \times \{\mathtt{L}, \mathtt{S}, \mathtt{R}\}.$$

Par exemple (en supposant que `0` et `1` appartiennent à $\mathcal{A}$) la transition $T(\mathtt{I}, \mathtt{1}) = (\mathtt{F}, \mathtt{0}, \mathtt{L})$, signifie que T étant dans l'état interne `I` et lisant la lettre `1` sur la case courante du ruban, écrit la lettre `0` sur cette même case, puis se positionne dans l'état interne `F` tout en déplaçant la tête de lecture/écriture d'une case à gauche. Un couple $(\mathtt{Q}, \mathtt{a}) \in \mathcal{Q} \times \mathcal{A}$ est appelé une *configuration* de T ; par convention on impose que

$$T(\mathtt{Q}, \mathtt{a}) = (\mathtt{Q}, \mathtt{a}, \mathtt{S}) \iff \mathtt{Q} = \mathtt{F} \tag{1}$$

de sorte que les seules *configurations d'arrêt* de la machine sont de la forme $(\mathtt{F}, \mathtt{a})$. Dans la suite $\mathfrak{T}(\mathcal{A})$ désigne l'ensemble (dénombrable) des machines de Turing dont l'alphabet d'exécution est $\mathcal{A}$.

Il y a beaucoup de définitions possibles (et équivalentes) de la machine de Turing : pour un approfondissement, citons l'introduction heuristique de Feynman [Fey99] ainsi que les présentations plus systématiques dans les livres de Minsky [Min67], Bianco [Bia79] ou encore chez Yablonski [Yab79]. On trouvera aussi un survol complet des apports scientifiques et mathématiques de Turing dans [Mar13] ; il y a aussi la belle biographie de Lassègue [Las98].

3. ASPECT FONCTIONNEL DES MACHINES DE TURING

La composition (fonctionelle) des machines de Turing est définie de la façon suivante : pour $i = 0, 1$ soit T_i une machine de $\mathfrak{T}(\mathcal{A})$ définie avec un alphabet d'états internes $\mathcal{Q}_i$ et d'état initial et final $\mathtt{I}_i$ et $\mathtt{F}_i$. En supposant de plus que $\mathcal{Q}_0 \cap \mathcal{Q}_1 = \emptyset$ (il est toujours possible de se ramener à cette situation), on définit la composée $T_1 \circ T_0$ comme la machine de $\mathfrak{T}(\mathcal{A})$ dont les états internes sont dans $\mathcal{Q} := \mathcal{Q}_0 \sqcup \mathcal{Q}_1$, d'état initial et final $\mathtt{I}_0$ et $\mathtt{F}_1$, la

table de transition étant définie pour tout $(\mathtt{Q},\mathtt{a}) \in \mathcal{Q}\times\mathcal{A}$ en posant :

$$T_1\circ T_0(\mathtt{Q},\mathtt{a}) = \begin{cases} T_0(\mathtt{Q},\mathtt{a}) & \text{si } \mathtt{Q}\in\mathcal{Q}_0\backslash\{\mathtt{F}_0\}\ ;\\ (\mathtt{I}_1,\mathtt{a},\mathtt{S}) & \text{si } \mathtt{Q}=\mathtt{F}_0\ ;\\ T_1(\mathtt{Q},\mathtt{a}) & \text{si } \mathtt{Q}\in\mathcal{Q}_1. \end{cases}$$

Pour $T\in\mathfrak{T}(\mathcal{A})$ une machine de Turing (d'états internes $\mathcal{Q}$), l'itération (fonctionnelle) de T (composition successive avec elle-même) donne

$$T^0(\mathtt{Q},\mathtt{a}) = (\mathtt{Q},\mathtt{a}),\quad T^1(\mathtt{Q},\mathtt{a}) = T(\mathtt{Q},\mathtt{a}),\quad T^2(\mathtt{Q},\mathtt{a}) = T\circ T(\mathtt{Q},\mathtt{a}),\dots$$

Afin d'éviter toute ambiguïté d'écriture on supposera toujours que $\mathcal{Q}\cup\{[,)\}$ est disjoint de $\mathcal{A}$. L'ensemble $\mathrm{St}(T)$ des *états courants* de T est constitué des mots de la forme $w[\mathtt{Q}]m$, où w et m sont dans $\mathcal{A}^*$ et $\mathtt{Q}\in\mathcal{Q}$. Ainsi, $\cdots\star\star\star\, wm\,\star\star\star\cdots$ est la suite des symboles écrit sur le ruban, la première lettre de m étant le symbole de la case lue par la tête de lecture/éciture (si m est le mot vide le symbole lu est $\star$) ; enfin, $\mathtt{Q}$ est l'état interne de la machine. Par abus de notation

$$T : \mathrm{St}(T)\to\mathrm{St}(T)$$

est l'application telle que $T(w[\mathtt{Q}]m)$ soit l'état courant obtenu par application d'un cycle de T initialement dans l'état courant $w[\mathtt{Q}]m$. L'ensemble des entrées (resp. sorties) de T est le sous-ensemble de $\mathrm{St}(T)$ formé des états courants de la forme $w[\mathtt{I}]m$ (resp. $w[\mathtt{F}]m$). On note $\mathrm{Out}(T)$ le sous ensemble de $\mathrm{St}(T)$ qui sont des *sorties* (ou encore des *états terminaux*) de T : plus précisément,

$$w[\mathtt{Q}]m\in\mathrm{Out}(T)\iff T(w[\mathtt{Q}]m)=w[\mathtt{Q}]m\iff \mathtt{Q}=\mathtt{F}.$$

Par définition, le temps d'arrêt d'un état courant $w[\mathtt{Q}]m\in\mathrm{St}(T)$ est

$$\mathrm{Stop}(T,w[\mathtt{Q}]m) = \min\Big\{n\in\mathbb{N}\ ;\ T^n(w[\mathtt{Q}]m)\in\mathrm{Out}(T)\Big\}.$$

avec $\mathrm{Stop}(T,w[\mathtt{Q}]m)=+\infty$ si $T^n(w[\mathtt{Q}]m)\notin\mathrm{Out}(T)$ pour tout $n\in\mathbb{N}$. On définit aussi l'application $T^*:\mathrm{St}(T)\to\mathrm{Out}(T)\cup\{\omega\}$ telle que

$$T^*(w[\mathtt{Q}]m) = \begin{cases} T^{\mathrm{Stop}(T,w[\mathtt{Q}]m)}(w[\mathtt{Q}]m) & \text{si } \mathrm{Stop}(T,w[\mathtt{Q}]m)<+\infty\,;\\ \omega & \text{si } \mathrm{Stop}(T,w[\mathtt{Q}]m)=+\infty. \end{cases} \tag{2}$$

Remarque 3.1. *Le fait que* $\mathrm{Stop}(\cdot,\cdot)$ *soit définie par une minimisation déterminée par l'arrêt de la machine est à mettre en rapport avec le schéma de minimisation (introduit par Kleene dans* [Kle36]*) permettant de définir les fonctions récursives (voir l'identification des fonctions Turing-calculables et des fonctions récursives dans la démonstration du Théorème 5.4).*

La fonction $\mathrm{Stop}(\cdot,\cdot)$ permet aussi de retrouver une interprétation fonctionnelle de la composition des machines de Turing introduite en début de paragraphe ; en effet, étant données S et T deux machines dans $\mathfrak{T}(\mathcal{A})$, on a :

$$S\circ T^*(w[\mathtt{I}]m) = \begin{cases} S^*\Big(T^{\mathrm{Stop}(T,w[\mathtt{Q}]m)}(w[\mathtt{Q}]m)\Big) & \text{si } \mathrm{Stop}(T,w[\mathtt{Q}]m)<+\infty\,;\\ \omega & \text{si } \mathrm{Stop}(T,w[\mathtt{I}]m)=+\infty. \end{cases} \tag{3}$$

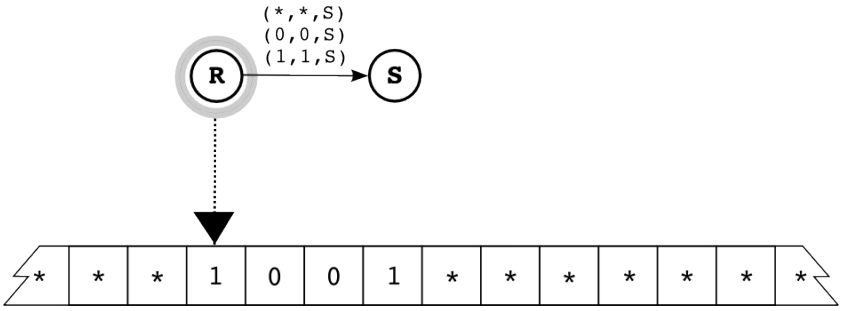


FIGURE 1. Représentation de Minsky d'une machine identité I (dans $\mathfrak{T}\{\star, 0, 1\}$) : ici on a $I^*([\mathtt{I}])1001) = [\mathtt{F}])1001$.

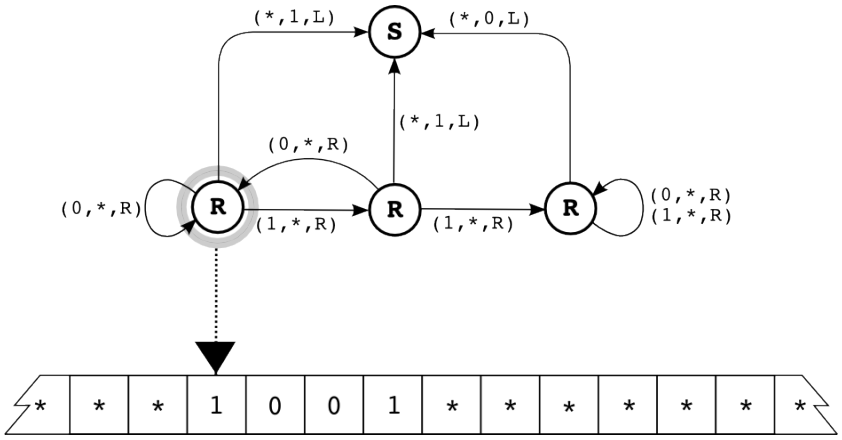


FIGURE 2. Représentation de Minsky d'une machine de Turing F reconnaissant les mots (de Fibonacci) qui ne sont pas factorisables par 11 : ici $F^*([\mathtt{I}])1001) = [\mathtt{F}])1$.

4. REPRÉSENTATION DES MACHINES DE TURING : GRAPHE DE MINSKY

Afin de donner une représentation graphique simple d'une machine de Turing de table de transition $T : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q} \times \mathcal{A} \times \{\mathtt{L}, \mathtt{S}, \mathtt{R}\}$, nous utiliserons *le graphe de Minsky* (voir [Min67]) : suivant la configuration de la machine on choisit (le plus judicieusement) une application $\Delta : \mathcal{Q} \ni \mathtt{Q} \mapsto (\mathtt{Q}, \mathtt{D_Q}) \in \mathcal{Q} \times \{\mathtt{L}, \mathtt{S}, \mathtt{R}\}$ qui fixe une direction privilégiée pour chaque état interne ; le graphe de Minsky est alors le graphe orienté dont l'ensemble des noeuds (resp. labels) est $\Delta(\mathcal{Q})$ (resp. $\mathcal{A} \times \mathcal{A} \times \{\mathtt{L}, \mathtt{S}, \mathtt{R}\}$) et défini par l'application

$$G : \Delta(\mathcal{Q}) \times \Delta(\mathcal{Q}) \times \Big(\mathcal{A} \times \mathcal{A} \times \{\mathtt{L}, \mathtt{S}, \mathtt{R}\}\Big) \to \{0, 1\}$$

telle que pour la transition $T((\mathtt{Q}, \mathtt{a})) = (\mathtt{P}, \mathtt{b}, \mathtt{D})$,

$$G\Big((\mathtt{Q}, \mathtt{D_Q}), (\mathtt{P}, \mathtt{D_P}), (\mathtt{a}, \mathtt{b}, \mathtt{D})\Big) = 1 \iff (\mathtt{P}, \mathtt{b}, \mathtt{D}) \neq (\mathtt{Q}, \mathtt{b}, \mathtt{D_Q}).$$

Autrement dit, le graphe de Minsky ne représente pas une transitions qui ne changent ni l'état interne Q, ni le symbole de la case courante et pour laquelle la tête de lecture/écriture se déplace dans la direction privilégiée $\mathtt{D_Q}$. En pratique, un noeud d'un graphe de Minsky ne portent que le symbole de sa direction privilégiée.

Donnons comme exemple une machine F permettant de reconnaître les mots de *Fibonacci* c'est-à-dire les mots binaires qui ne sont pas factorisables par 11. L'ensemble des

états internes de F est $\mathcal{Q} = \{\mathtt{I}, \mathtt{U}, \mathtt{V}, \mathtt{F}\}$ et l'alphabet d'exécution $\mathcal{A} = \{\star, \mathtt{0}, \mathtt{1}\}$. Le fonctionnement complet de la machine est donné par la table de transition suivante :

	$\star$	0	1
I	(F, 1, L)	(I, $\star$, R)	(U, $\star$, R)
U	(F, 1, L)	(I, $\star$, R)	(V, $\star$, R)
V	(F, 0, L)	(V, $\star$, R)	(V, $\star$, R)
F	(F, $\star$, S)	(F, 0, S)	(F, 1, S)

Ainsi on aura par exemple $F^*([\mathtt{I})\mathtt{1001}) = [\mathtt{F})\mathtt{1}$ (ce qui signifie que 1001 est un mot de Fibonacci) alors que $F^*([\mathtt{I})\mathtt{1011}) = [\mathtt{F})\mathtt{0}$ (ce qui signifie que 1011 n'est pas un mot de Fibonacci). Une représentation du graphe de Minsky de F (appliquée au mot 1001) est donnée dans la Figure 2.

5. Langage récursivement énumérables et langages récursifs

Supposons $\{[,), |, \#\}$ disjoint de l'alphabet $\mathcal{A}$; la machine $G \in \mathfrak{T}(\mathcal{B})$, avec $\mathcal{B} \supset \mathcal{A} \cup \{|, \#\}$, est un *générateur* du langage $\mathcal{L} \subset \mathcal{A}^*$, lorsqu'il existe une suite strictement croissante d'entiers $n_0, n_1, \ldots$ (cette suite étant finie lorsque $\mathcal{L}$ est fini) et telle que

$$G^{n_k}([\mathtt{I})m_0|w_0\#) = [\mathtt{I})m_k|w_0\#w_1\#\cdots\#w_k\#$$

et où $w_0, w_1, \ldots$ forment une liste complète des mots dans $\mathcal{L}$. Le symbole | (dont on s'assure qu'il ne possède qu'une seule occurrence au cours des calculs de G) délimite la zone d'affichage de la liste des mots de $\mathcal{L}$ (à droite) d'avec la zone de calcul (à gauche) où sont affichés les mots $m_0, m_1, \ldots$ (ce sont des mot liés au fonctionnement de G).

Définition 5.1. *Le langage $\mathcal{L} \subset \mathcal{A}^*$ est dit récursivement énumérable s'il existe un générateur G qui engendre tous les mots de $\mathcal{L}$ (et uniquement ceux-là) ; il est dit récursif s'il existe un générateur G qui engendre tous les mots de $\mathcal{L}$ dans l'ordre canonique sur $\mathcal{A}^*$.*

Soit $\mathcal{L}$ un sous-langage de $\mathcal{A}^*$ et $w \in \mathcal{A}^*$; un problème classique consiste à déterminer si oui ou non w appartient à $\mathcal{L}$. Si le langage $\mathcal{L}$ est récursif, on pourra répondre à cette question dans tous les cas en un temps fini (en général on ne connait pas le temps d'attente). Par contre, si $\mathcal{L}$ n'est que récursivement énumérable, et que w n'est pas dans $\mathcal{L}$, alors on a pas accès à cette information en temps fini.

Le *langage reconnu* (on dit aussi *accepté*) par une machine $T \in \mathfrak{T}(\mathcal{A} \sqcup \{\star\})$, noté $\mathrm{Ackn}(T)$, est le sous-ensemble de $\mathcal{A}^*$ constitué des mots w pour lesquels le temps d'arrêt $\mathrm{Stop}(T, [\mathtt{I})w)$ est fini ($\mathrm{Ackn}(\cdot)$ vient du terme anglais *acknowlegement*).

Théorème 5.2. *Soit $\mathcal{L} \subset \mathcal{A}^*$ un langage ; alors (i) : $\mathcal{L}$ est récursivement énumérable si et seulement si il est reconnu par une machine de Turing $T \in \mathfrak{T}(\mathcal{A} \cup \{\star\})$ et (ii) : $\mathcal{L}$ est récursif si et seulement si $\mathcal{L}$ et $\mathcal{A}^* \backslash \mathcal{L}$ sont récursivement énumérables.*

Preuve. Exercice.

□

Dans la suite $\mathcal{N}_2 = \{\mathtt{0}\} \cup \mathtt{1}\{\mathtt{0}, \mathtt{1}\}^*$ désigne le langage des écritures binaires des nombres entiers naturels, de sorte qu'en utilisant (grossièrement) l'ordre naturel sur les

entiers $\mathcal{N}_2 = \{\mathtt{0}, \mathtt{1}, \mathtt{10}, \mathtt{11}, \mathtt{100}, \ldots\}$. L'ensemble $\mathfrak{T}(\mathcal{A} \sqcup \{\star\})$ des machines de Turing sur $\mathcal{A}$ étant dénombrable on peut l'écrire sous la forme d'une liste injective, soit encore :

$$T_{\mathtt{0}}, T_{\mathtt{1}}, T_{\mathtt{10}}, T_{\mathtt{11}}, T_{\mathtt{100}}, \ldots$$

Proposition 5.3. *Le langage*

$$\mathcal{D} := \Big\{ w \in \mathcal{N}_2 \ ; \ w \notin \mathrm{Ackn}(T_w) \Big\} = \Big\{ w \in \mathcal{N}_2 \ ; \ \mathrm{Stop}(T_w, [\mathtt{I}]w) = +\infty \Big\}. \tag{4}$$

n'est pas récursivement énumérable.

Preuve. Supposons par l'absurde $\mathcal{D}$ récursivement énumérable : comme $\mathcal{D} \subset \{\mathtt{0}, \mathtt{1}\}^*$ et que $T_{\mathtt{0}}, T_{\mathtt{1}}, T_{\mathtt{10}} \ldots$ est la liste complète des machines de $\mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$, d'après le Théorème 5.2 il existe $w_0 \in \mathcal{N}_2$ tel que $\mathcal{D} = \mathrm{Ackn}(T_{w_0})$. La contradiction vient du fait que $w_0 \in \mathcal{D}$ entraîne $w_0 \notin \mathrm{Ackn}(T_{w_0}) = \mathcal{D}$ alors que $w_0 \in \mathcal{N}_2 \backslash \mathcal{D}$ entraîne $w_0 \in \mathrm{Ackn}(T_{w_0}) = \mathcal{D}$.

□

Preuve (bis). Par l'absurde, supposons que $\mathcal{D} = \mathrm{Ackn}(T_w)$ avec $w \in \mathcal{N}_2$; la contradiction vient du fait qu'on a :

$$w_A \in \mathcal{D} \Rightarrow \begin{cases} \mathrm{Stop}(T_w, [\mathtt{I}]w) = \infty & \text{par définition de } \mathcal{D} \\ \mathrm{Stop}(T_w, [\mathtt{I}]w) < \infty & \text{par définition de } T_w \end{cases}$$

et

$$w \in \mathcal{N}_2 \backslash \mathcal{D} \Rightarrow \begin{cases} \mathrm{Stop}(T_w, [\mathtt{I}]w_A) < \infty & \text{par définition de } \mathcal{D} \\ \mathrm{Stop}(T_w, [\mathtt{I}]w_A) = \infty & \text{par définition de } T_w \end{cases}$$

□

6. Machine de Turing universelle

Une machine de Turing est dite *universelle* si le ruban de lecture/écriture peut être *programmée* afin de *simuler* les calculs de n'importe quelle machine de Turing. En suivant la présentation guidée de Feynman [3] dans [Fey99], la construction (abstraite) effective d'une machine universelle est un exercice abordable et même amusant ! Le théorème suivant, résume les propriétés de la machine universelle que nous utiliserons.

Théorème 6.1. *Il existe une machine de Turing $U \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ – dite universelle – associée à une application $(T, w[\mathtt{Q}]m) \mapsto \mathrm{Pr}(T, w[\mathtt{Q}]m) \in \{\star, \mathtt{0}, \mathtt{1}\}^*$ définie pour tout $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ et tout $w[\mathtt{Q}]m \in \mathrm{St}(T)$, et telle que pour tout couple $(T, w[\mathtt{Q}]m)$, il existe un rang $n \geq 1$ pour lequel*

$$U^n\Big([\mathtt{I}]\mathrm{Pr}(T, w[\mathtt{Q}]m)\Big) = [\mathtt{I}]\mathrm{Pr}(T, T(w[\mathtt{Q}]m)) \ ; \tag{5}$$

de plus, en posant $\mathrm{Pr}(T, \omega) = \omega$, on a dans tous les cas

$$U^*\Big([\mathtt{I}]\mathrm{Pr}(T, w[\mathtt{Q}]m)\Big) = [\mathtt{I}]\mathrm{Pr}(T, T^*(w[\mathtt{Q}]m)) \ ; \tag{6}$$

d'autre part, il existe une machine $\Delta \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ (une décodeur) telle que

$$(\Delta \circ U)^*\Big([\mathtt{I}]\mathrm{Pr}(T, w[\mathtt{Q}]m)\Big) = T^*(w[\mathtt{Q}]m). \tag{7}$$

3. Cette présentation est issue des discussions entre Minsky et Feynman à propos des machines et qui ont participé à jeter les bases de ce qu'on appelle maintenant l'*ordinateur quantique*.

7. L'AUTORÉFÉRENCE ET LE PROBLÈME DE L'ARRÊT

Soit $U \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ une machine universelle donnée par le Théorème 6.1. Toute machine $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ est associé au code $\mathrm{Pr}(T, w[\mathtt{Q}]m) \in \{\star, \mathtt{0}, \mathtt{1}\}^*$ de simulation par U du calcul de T sur l'entrée $w[\mathtt{Q}]m$: si $T^k(w[\mathtt{Q}]m) = w_k[\mathtt{Q}_k]m_k$ alors il existe une suite d'entiers $0 = n_0, n_1, \ldots$ strictement croissante telle que

$$U^{n_k}\Big([\mathtt{I}]\mathrm{Pr}(T, w[\mathtt{Q}]m)\Big) = [\mathtt{I}]\mathrm{Pr}(T, w_k[\mathtt{Q}_k]m_k).$$

Rappelons que $\mathcal{N}_2 := \{\mathtt{0}\} \sqcup \mathtt{1}\{\mathtt{0}, \mathtt{1}\}^*$ est l'ensemble des entiers binaires : ce langage est ordonné canoniquement en prenant $\mathtt{0} < \mathtt{1}$. Chacune machine $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ est affecté un entier binaire unique noté w_T qui est le *codage binaire (standard) du rang de* $\mathrm{Pr}(T, [\mathtt{I}])$ *dans l'ordre canonique sur* $\{\mathtt{0}, \mathtt{1}\}^*$ (le code $\mathrm{Pr}(T, [\mathtt{I}])$ décrit complètement, et de manière univoque, la table de transition de T) ; on notera aussi $T = T_{w_T}$, de sorte que $T_{\mathtt{0}}, T_{\mathtt{1}}, T_{\mathtt{10}}, \ldots$ est la liste complète des machines dans $\mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$. Cette numérotation des machines précise celle (plus vague) qui nous a permis de définir le langage $\mathcal{D}$ introduit en (4). Sachant (Proposition 5.3) que $\mathcal{D}$ n'est pas récursivement énumérable, le théorème suivant établit l'existence d'un langage strictement récursivement énumérable.

Proposition 7.1. *Le langage* $\mathcal{N}_2\backslash\mathcal{D} = \big\{w \in \mathcal{N}_2 \ ; \ \mathrm{Stop}(T_w, [\mathtt{I}]w) < +\infty\big\}$ *est strictement récursivement énumérable (i.e. récursivement énumérable et non récursif).*

Preuve du Théorème 7.1. D'après la Proposition 5.3, nous savons que $\mathcal{D}$ n'est pas récursivement énumérable : il reste donc à démontrer que $\mathcal{N}_2\backslash\mathcal{D}$ est récursivement énumérable. Par construction de la machine universelle U, le langage

$$\mathcal{E}_U := \Big\{\mathrm{Pr}(T, w[\mathtt{Q}]m) \ ; \ T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\} \text{ et } w[\mathtt{Q}]m \in \mathrm{St}(T)\Big\}.$$

est récursif : le langage $\mathcal{N}_2$ étant lui aussi récursif, il en est de même de

$$\Big\{\mathrm{Pr}(T_w, [\mathtt{I}]w) \ ; \ w \in \mathcal{N}_2\Big\}.$$

Ainsi, il existe une machine S telle que $\mathrm{Stop}(S, [\mathtt{I}]w) = \infty$ lorsque $w \in \{\mathtt{0}, \mathtt{1}\}^*\backslash\mathcal{N}_2$ et $S^*([\mathtt{I}]w) = [\mathtt{F}]\mathrm{Pr}(T_w, [\mathtt{I}]w)$, pour tout $w \in \mathcal{N}_2$. On obtient $\mathcal{N}_2\backslash\mathcal{D} = \mathrm{Ackn}(U \circ S)$ du fait que $\mathrm{Stop}(U \circ S, [\mathtt{I}]w) = \infty$, pour tout $w \in \{\mathtt{0}, \mathtt{1}\}^*\backslash\mathcal{N}_2$ et que $\mathrm{Stop}(T_w, [\mathtt{I}]w) < \infty$ dès que $w \in \mathcal{N}_2$ (i.e. $w \in \mathcal{N}_2\backslash\mathcal{D}$) si et seulement si $\mathrm{Stop}(U \circ S, [\mathtt{I}]w) < \infty$.

□

Insistons sur le fait que l'existence d'un langage strictement récursivement énumérable est un sous-produit du *problème de l'arrêt* des machines de Turing. Nous donnons maintenant un deuxième argument pour le Théorème 7.1 : celui-ci est essentiellement basé sur le fait que la machine universelle U permet de conclure que

$$\mathcal{F}_U := \Big\{\mathrm{Pr}(T, [\mathtt{I}]w) \ ; \ T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\} \text{ et } w \in \{\mathtt{0}, \mathtt{1}\}^* \text{ t.q. } \mathrm{Stop}(T, [\mathtt{I}]w) < \infty\Big\} \tag{8}$$

est un langage récursivement énumérable. Le problème de l'arrêt porte sur la récursivité de $\mathcal{F}_U$, c'est-à-dire qu'il pose la question de l'existence d'une machine – appelée un *Oracle* – prédisant (pour tout $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ et tout $w \in \{\mathtt{0}, \mathtt{1}\}$) si $\mathrm{Stop}(T, [\mathtt{I}]w) < \infty$.

Théorème 7.2. *Le langage* $\mathcal{F}_U$ *est strictement récursivement énumérable.*

Preuve. Supposons par l'absurde qu'il existe un oracle $O \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ décidant le problème de l'arrêt des machines dans $\mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$, c'est-à-dire tel que pour tout $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ et pour tout $w \in \{\mathtt{0}, \mathtt{1}\}^*$,

$$O^*\Big([\mathtt{I}]\mathrm{Pr}(T, [\mathtt{I}]w)\Big) = \begin{cases} [\mathtt{F}]\mathtt{1} & \text{si } \mathrm{Stop}(T, [\mathtt{I}]w) < \infty \\ [\mathtt{F}]\mathtt{0} & \text{si } \mathrm{Stop}(T, [\mathtt{I}]w) = \infty \end{cases}$$

Les langages $\mathcal{N}_2$ et $\mathcal{E}_U$ étant récursifs, le langage $\{\mathrm{Pr}(T_w, [\mathtt{I}]w) \ ; \ w \in \mathcal{N}_2\}$ est lui aussi récursif. Il existe donc une machine S pour laquelle $\mathrm{Stop}(S, [\mathtt{I}]w) = \infty$ lorsque $w \in \{\mathtt{0}, \mathtt{1}\}^* \backslash \mathcal{N}_2$ et telle que $S^*([\mathtt{I}]w) = [\mathtt{F}]\mathrm{Pr}(T_w, [\mathtt{I}]w)$, pour tout entier binaire w. Par suite, pour tout $w \in \{\mathtt{0}, \mathtt{1}\}^*$,

$$(O \circ S)^*([\mathtt{I}]w) = \begin{cases} [\mathtt{F}]\mathtt{1} & \text{si } w \in \mathcal{N}_2 \text{ et } \mathrm{Stop}(T_w, [\mathtt{I}]w) < \infty \\ [\mathtt{F}]\mathtt{0} & \text{si } w \in \mathcal{N}_2 \text{ et } \mathrm{Stop}(T_w, [\mathtt{I}]w) = \infty \\ [\mathtt{F}]\mathtt{0} & \text{si } w \in \{\mathtt{0}, \mathtt{1}\}^* \backslash \mathcal{N}_2 \end{cases}$$

ce qui contredit le fait (Théorème 7.1) que $\mathcal{N}_2 \backslash \mathcal{D}$ n'est pas récursif.

□

La théorie des langages récursivement énumérables est un point clef du dixième problème de Hilbert [Hil02] résolu par Yuri Matiyasevich (voir [Rob52, Dav53, DPR61, Mat70, DH73, Mat99]). Suivant une définition de Julia Robinson, une partie A de l'ensemble $\mathbb{N}$ des entiers naturels positifs ou nuls, est dite *diophantienne* s'il existe un polynôme $P \in \mathbb{Z}[X_0, \ldots, X_N]$ tel que $n \in A$ si et seulement si l'équation $P(n, x_1, \ldots, x_N) = 0$ possède une solution dans $\mathbb{Z}^{N+1}$. En d'autres termes, A est la projection sur $\mathbb{N}$ d'une courbe algébrique de $\mathbb{Z}^{N+1}$. Par exemple, l'ensemble $A = \{0, 1, 4, 9, \ldots\}$ des carrés d'entiers est diophantiens puisque $n \in A$ si et seulement si $P(n, n^2) = 0$ avec

$$P(X, Y) = X^2 - Y.$$

Pour $A \subset \mathbb{N}$ soit $\mathcal{L}_A$ le sous-ensemble de $\{\mathtt{0}, \mathtt{1}\}^*$ des écritures binaires des éléments de A. L'ensemble A est alors dit récursivement énumérable (resp. récursif) si le langage $\mathcal{L}_A$ l'est. Si A est diophantien alors A est récursivement énumérable. La réciproque répond à la question de Hilbert portant sur le calcul des solutions des équations diophantiennes [4].

Théorème 7.3. *[Robinson-Matijasevic] Une partie de $\mathbb{N}$ est diophantienne si et seulement si elle est récursivement énumérable.*

En particulier, il existe des parties diophantiennes de $\mathbb{N}$ qui sont strictement récursivement énumérable : cela rend le calcul systématique (mécanique) des solutions d'équations diophantiennes impossible (voir [Oli14, § 5]).

8. Complexité de Kolmogorov

Dans ce paragraphe, tout nombre entier naturel n est identifié à son développement binaire, c'est-à-dire à un mot du language $\mathcal{N}_2 = \{\mathtt{0}\} \cup \mathtt{1}\{\mathtt{0}, \mathtt{1}\}^*$; dans la suite, $\ell(w)$ désigne

4. Est-il possible de trouver une procédure *mécanique* permettant de trouver les solutions des équations diophantiennes ? D'après le Théorème de Robinson-Matijasevic, la réponse est non.

la longueur d'un mot de w relativement à son alphabet de référence [5] : en particulier,

$$\log_2 n \leq \ell(n) \leq 1 + \log_2(\max\{1, n\}).$$

Admettons provisoirement l'existence d'une famille $\{K^{(n)} : \mathbb{N}^n \to \mathbb{N}\ ;\ n \geq 1\}$ t.q.

(K0) : il existe une infinité d'entiers n t.q. $K^{(1)}(n) \geq \ell(n)$ (n est dit K-aléatoire) ;

(K1) : si $f : \mathbb{N}^p \to \mathbb{N}^q$ est calculable alors il existe $C_f > 0$ t.q.

$$K^{(q)}\big(f(x_1, \cdots, x_p)\big) \leq 4\big(\ell(x_1) + \cdots + \ell(x_n)\big) + C_f\ ;$$

Théorème 8.1 (Euclide)**.** *Il existe une infinité de nombres premiers.*

Preuve (Chaitin-Kolmogorov [Kol65][Cha66, Cha75]**).** D'après **(K0)**, il existe une suite strictement croissante $x_1, x_2, \ldots$ d'entiers K-aléatoires, de sorte que pour tout rang k,

$$K^{(1)}(x_k) \geq \ell(x_k). \tag{9}$$

Supposons qu'il existe un nombre fini de nombre premiers, soient $p_1, \ldots, p_N$ et écrivons

$$x_k = p_1^{\alpha_1(x_k)} \cdots p_N^{\alpha_N(x_k)} =: f(\alpha_1(x_k), \cdots, \alpha_N(x_k))$$

la décomposition de x_k en facteurs premiers (ici les exposants $\alpha_i(x_k) \in \mathbb{N}$ sont des entiers positifs où nuls). En particulier, du fait que $x_k \geq 2^{\alpha_1(x_k)+\cdots+\alpha_N(x_k)}$, on tire l'inégalité

$$\log_2 x_k \geq \alpha_1(x_k) + \cdots + \alpha_N(x_k). \tag{10}$$

L'application $f : (a_1, \ldots, a_N) \mapsto p_1^{a_1} \cdots p_N^{a_N}$ étant calculable, avec **(K1)** il vient :

$$\begin{aligned} K^{(1)}(x_k) &= K^{(1)}\big(f\big(\alpha_1(x_k), \cdots, \alpha_N(x_k)\big)\big) \\ &\leq 4\big[\ell\big(\alpha_1(x_k)\big) + \cdots + \ell\big(\alpha_N(x_k)\big)\big] + C_f \\ &\leq 4\big[\log_2\big(\max\{1, \alpha_1(x_k)\}\big) + \cdots + \log_2\big(\max\{1, \alpha_N(x_k)\}\big)\big] + 4N + C_f \\ &\leq 4\log_2\big[\alpha_1(x_k) + \cdots + \alpha_N(x_k)\big] + 4N + C_f\ ; \end{aligned}$$

en combinant (9) et (10) on obtient la suite – contradictoire – d'inégalités

$$\log_2 x_k \leq \ell(x_k) \leq K^{(1)}(x_k) \leq 4\log_2(\log_2 x_k)) + 4N + C_f.$$

□

La notion de *complexité algorithmique* de Kolmogorov [Kol65], permet la construction effective d'une famille d'applications $K^{(n)}$ ($n \geq 1$) satisfaisant **(K0)**, **(K1)** et **(K2)**. Pour $T \in \mathfrak{T}\{\star, 0, 1\}$ la T-complexité d'un mot $w \in \mathcal{A}^*$ pour $\emptyset \neq \mathcal{A} \subset \{\star, 0, 1\}^*$ est

$$K_T(w) = \min\Big\{\ell(m)\ ; m \in \mathcal{A}^* \text{ et } T^*([\mathtt{I}]m) = [\mathtt{F}]w\Big\}$$

avec la restriction que $K_T(w) = +\infty$ s'il n'existe pas de mot binaire m tel que $T^*([\mathtt{I}]m) = [\mathtt{F}]w$. Un premier moyen pour contourner la dépendance relativement à la machine T (qui entraîne en particulier que $K_T(w)$ peut-être infini) est de définir le minimum $\underline{K}(w)$ des $K_T(w)$ pour T décrivant l'ensemble des machines dans $\mathfrak{T}\{\star, 0, 1\}$. En considérant la machine identité, il est évident que $\underline{K}(w) \leq \ell(w)$. Cette dernière remarque mène au concept de *nombre compressible* : pour $0 < \gamma < 1$ on dit que w est γ-*compressible* si $\underline{K}(w) \leq \gamma\ell(w)$. Pour un tel w il existe donc une machine T telle $K_T(w) \leq \gamma\ell(w)$; ce critère de

5. Nous utilisons $\ell(w)$ et non $|w|$ – comme d'habitude – afin d'éviter de possibles confusions avec la valeur absolue.

compressibilité ne tient pas compte de la structure interne de la machine[6] T, ni de la longueur d'un calcul du type $T^*([\mathtt{I}]\mathtt{m}) = [\mathtt{F}]w$. Un moyen pour prendre en compte la complexité des machines (mais pas de la longueur du calcul[7]) est de définir un analogue de $\underline{K}(w)$ au moyen d'une machine universelle (voir Définition 8.2 ci-dessous).

Dans l'idée de Kolmogorov, le mot w est *(algorithmiquement) aléatoire* s'il est *peu* compressible. Dans ce paragraphe $U \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ est une machine universelle fixée donnée par le Théorème 6.1. En particulier U permet de simuler les calculs de toute machine $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$. Nous utiliserons aussi le décodeur Δ défini en (7) et $U_0 := \Delta \circ U$.

Définition 8.2. *(i) : La complexité de Kolmogorov (relative à la machine universelle $U_0 = \Delta \circ U$) d'un mot $w \in \mathcal{A}^*$ pour $\emptyset \neq \mathcal{A} \subset \{\star, \mathtt{0}, \mathtt{1}\}^*$ est*

$$K_{U_0}(w) = \min\Big\{\ell(m)\ ; m \in \mathcal{A}^* \text{ et } \ U_0^*([\mathtt{I}]m) = w\Big\} \qquad (< +\infty)\ ;$$

(ii) : pour $n \geq 1$ la complexité de Kolmogorov d'un n-uplet $(x_1, \ldots, x_n) \in \mathbb{N}^n$ ($\equiv \mathcal{N}_2^n$) est

$$K^{(n)}(x_1, \ldots, x_n) := K_{U_0}(x_1 \star \cdots \star x_n).$$

Pour $w \in \{\star, \mathtt{0}, \mathtt{1}\}^*$ donné, il existe un couple optimal machine/mot (T, m) tel que $T^*([\mathtt{I}]m) = w$ et avec $\underline{K}(w) = K_T(w) = \ell(m)$. Comme $U_0^*\big([\mathtt{I}]\mathrm{Pr}(T, [\mathtt{I}]m)\big) = T^*([\mathtt{I}]m) = w$, il vient $K_{U_0}(w) \leq \ell(\mathrm{Pr}(T, [\mathtt{I}]m))$: il est évident que $\underline{K}(w) \leq K_{U_0}(w)$ et raisonnable d'imaginer que $K_{U_0}(w)$ et $\underline{K}(w)$ sont – en un sens – comparables. En fait, si U est bien construite, U_0 doit pouvoir engendrer w de manière optimale en ce sens que U_0 ne peut pas faire beaucoup moins bien qu'une machine non universelle T et optimisée pour engendrer w. Le Lemme suivant précise cette remarque.

Lemme 8.3. *Il existe une machine universelle U (au sens du Théorème 6.1) optimale pour la complexité de Kolmogorov en ce sens que pour toute machine $T \in \mathfrak{T}\{\star, \mathtt{0}, \mathtt{1}\}$ il existe $\theta \geq 1$ (dépendant de U) et $C_T > 0$ (dépendant de U et de T) t.q. pour tout mot $w \in \{\star, \mathtt{0}, \mathtt{1}\}^*$,*

$$K_{U_0}(w) \leq \theta K_T(w) + C_T.$$

Preuve heuristique. Supposons que $K_T(w) < +\infty$ et soit m minimal t.q. $T^*([\mathtt{I}]m) = [\mathtt{F}]w$; en particulier cela signifie que $\ell(m) = K_T(w)$. Par définition de la machine $U_0 = \Delta \circ U$ il vient $U_0^*([\mathtt{I}]\mathrm{Pr}(T, [\mathtt{I}]m) = [\mathtt{F}]w$ et donc $K_{U_0}(w) \leq \ell\big(\mathrm{Pr}(T, [\mathtt{I}]m\big)$. Le détail de la construction de U montre qu'il est possible d'assurer que $\ell\big(\mathrm{Pr}(T, [\mathtt{I}]m\big)$ est une fonction affine de $\ell(m)$: pour une bonne machine universelle U, il existe alors deux constantes $\theta > 1$ (dépendant de U_0) et $C_T > 0$ (dépendant de U_0 et de T) telles que $\ell\big(\mathrm{Pr}(T, [\mathtt{I}]m\big) \leq \theta\ell(m) + C_T$. Les définitions de θ et C_T doivent prendre en compte le système de simulation de U (implémenté sur le ruban de U) ainsi que la description de T dans le programme de simulation de T (inscrit sur le ruban de U) ; il est possible de construire U t.q. $\theta = 2$.

□

6. La machine T en question peut donc être très *complexe* et difficile à trouver : c'est tout le problème de la *compression des données.*

7. Pour prendre en compte la longueur du calcul $T^*([\mathtt{I}]m) = [\mathtt{F}]w$ on pourrait par exemple considérer

$$\sum\nolimits_{k=0}^{\mathrm{Stop}(T,[\mathtt{I}]m)} \ell\big(T^k([\mathtt{I}]m)\big).$$

Proposition 8.4. *Les applications $K^{(n)}$ ($n \geq 1$) vérifient* **(K0)** *; plus précisément, pour tout entier $n \geq 1$, l'ensemble $\{\mathtt{0},\mathtt{1}\}^n$ contient au moins un mot K-aléatoire.*

Preuve. Le nombre de mots dans $\{\mathtt{0},\mathtt{1}\}^*$ de longueur au plus $n-1$ est $1+2+\cdots 2^{n-1} = 2^n-1$. Par suite il existe au moins un mot x, parmi les 2^n mots de $\{\mathtt{0},\mathtt{1}\}^n$, pour lequel $U_0^*([\mathtt{I}]m) = [\mathtt{F}]x$ entraîne $\ell(m) \geq n$: cela signifie que $K^{(1)}(x) \geq \ell(x) = n$.

□

Remarque 8.5. *(1) : La Proposition 8.4 est une porte ouverte sur le monde de la théorie des générateurs de nombres aléatoires (voir* [BH10]*).*

(2) : Soit $w \in \{\star,\mathtt{0},\mathtt{1}\}$ et soit (T,m) un couple machine/mot optimal qui engendre w, i.e. $T^*([\mathtt{I}]m) = [\mathtt{F}]w$ et $\underline{K}(w) = \ell(m)$. Alors $U_0^*([\mathtt{I}]\mathrm{Pr}(T,[\mathtt{I}]m)) = [\mathtt{F}]w$ et donc $K_{U_0}(w) \leq \ell(\mathrm{Pr}(T,[\mathtt{I}]m))$. Il est troublant d'imaginer l'existence d'un mot $x \in \{\star,\mathtt{0},\mathtt{1}\}$ (non prévu par la syntaxe de programmation de U_0) t.q. $U_0^*([\mathtt{I}]x) = [\mathtt{F}]w$: cette possibilité (fonctionnement de U or du ce qui est prévu) est d'autant plus troublante qu'il est possible que $\ell(x) \leq \ell(\mathrm{Pr}(T,[\mathtt{I}]m))$.

Proposition 8.6. *La condition* **(K1)** *est vérifiée, c'est-à-dire, que pour toute fonction $f : \mathbb{N}^p \to \mathbb{N}^q$ Turing-calculable, il existe une constate C_f (dépendant de la machine universelle utilisée) t.q.*

$$K^{(q)}\big(f(x_1,\ldots,x_p)\big) \leq \theta^2\big(\ell(x_1)+\cdots+\ell(x_p)\big) + C_f.$$

(et où la constante $\theta \geq 1$ donnée par le Lemme 8.3 peut prendre la valeur 2).

Preuve. Supposons que $f(x_1,\ldots,x_p) = (y_1,\ldots,y_q)$ et soit $m \in \{\star,\mathtt{0},\mathtt{1}\}$ la plus petite entrée t.q. $U_0^*([\mathtt{I}]m) = [\mathtt{F}]x_1\star\cdots\star x_p$; alors

$$\ell(m) = K^p(x_1,\ldots,x_p) = K_{U_0}(x_1\star\cdots\star x_p)$$

et en notant T_{Id} la machine identité de $\mathfrak{T}\{\star,\mathtt{0},\mathtt{1}\}$ le Lemma 8.3 donne :

$$\ell(m) \leq K_{T_{Id}}(x_1\star\cdots\star x_p) + C_{T_{Id}} = \theta\big(\ell(x_1)+\cdots+\ell(x_p)\big) + p + C_{T_{Id}}. \tag{11}$$

Puisque f est calculable, il existe un machine T_f telle que $T_f^*([\mathtt{I}]x_1\star\cdots\star x_p) = [\mathtt{F}]y_1\star\cdots\star x_q$. Par composition des machines $(T_f\circ U_0)^*([\mathtt{I}]m) = [\mathtt{F}]y_1\star\cdots\star y_q$ et par suite

$$K_{T_f\circ U_0}(y_1\star\cdots\star y_q) \leq \ell(m)\ ;$$

avec la machine $T_f\circ U_0$ et la constante $C_{T_f\circ U_0}$ donnée par le Lemma 8.3

$$\begin{aligned} K^{(q)}\big(f(x_1,\ldots,x_p)\big) &= K_{U_0}(y_1\star\cdots\star y_q) \\ &\leq \theta K_{T_f\circ U_0}(y_1\star\cdots\star y_q) + C_{T_f\circ U_0} \\ &\leq \theta\ell(m) + C_{T_f\circ U_0} \\ &= \theta^2\big(\ell(x_1)+\cdots+\ell(x_p)\big) + \big(\theta(p+C_{T_{Id}}) + C_{T_f\circ U_0}\big). \end{aligned}$$

□

RÉFÉRENCES

[BH10] L. Bienvenu and M. Hoyrup. Une brève introduction à la théorie effective de l'aléatoire. *La Gazette des mathématiciens (SMF)*, 123 :35–47, 2010.

[Bia79] E. Bianco. *Informatique fondamentale : de la machine de Turing aux ordinateurs modernes*. Basel, Boston, Stuttgart ISR 70, 1979.

[Cha66] G. Chaitin. On the lenght of programs for computing finite binary sequences. *J. of Ass. For Computing Machinary*, 13 :547–569, 1966.

[Cha75] G. Chaitin. A theory of program size formally identical to information theory. *J. of Ass. For Computing Machinary*, 22 :329–340, 1975.

[Dav53] M. Davis. Arithmetical problems and recursively enumerable predicates. *Journal of Symbolic Logic*, 18-(1) :33–41, 1953.

[DH73] M. Davis and R. Hersh. Hilbert's tenth problem. *Scientific American*, 229 :84–91, 1973.

[DPR61] M. Davis, H. Putnam, and J. Robinson. The decision problem for exponential Diophantine equations. *Annals of Mathematics, Second Series*, 74-(3) :425–436, 1961.

[Fey99] R. Feynman. *Lectures on Computation*. Penguin Books Ltd (New edition), 1999.

[Hil02] D. Hilbert. Lecture delivered before the International Congress of Mathematicians at Paris in 1900 by Professor David Hilbert, (translated into english by Dr. Maby Winton Newson, with the author's permission). *Bulletin of the American Mathematical Society*, 8 :437–479, 1902.

[Kle36] S. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112 :727–729, 1936.

[Kol65] A. Kolmogorov. Three approaches to quantitative definition of information. *Information and Control*, 1 :1–7, 1965.

[Las98] J. Lassègue. *Turing*. Les Belles Lettres, Paris, 1998.

[Mar13] M. Margenstern. Ce qu'Alan Turing nous a laissé. *La Gazette des mathématiciens (SMF)*, 135 :17–31, 2013.

[Mat70] Y. Matiyasevich. Enumerable sets are Diophantine (in Russian). *Doklady Akademii Nauk SSSR 191 : 279-282. English translation in Soviet Mathematics*, 11-(2) :354–357, 1970.

[Mat99] Yuri Matiyasevich. *Le dixième problème de Hilbert : que peut-on faire avec les équations diophantiennes ?* La recherche de la vérité (M. Serfati ed.). A.C.L. Paris, 1999.

[Min67] Marvin Lee Minsky. *Computation : Finite and Infinite Machines*. Prentice Hall, 1967.

[Oli14] E. Olivier. Qu'est-ce-qu'une machine (II/III). *BIAA*, 98 :45–56, 2014.

[Rob52] J. Robinson. Existential definability in arithmetic. *Transactions of the American Mathematical Society*, 72-(3) :437–449, 1952.

[Tur36] A. Turing. On Computable Numbers, with an application to the Entscheidungsproblem (correction ibid. (1967) 43, p. 544-546). *Proc. Lond. Math. Soc.*, (2)-42 :230–265, 1936.

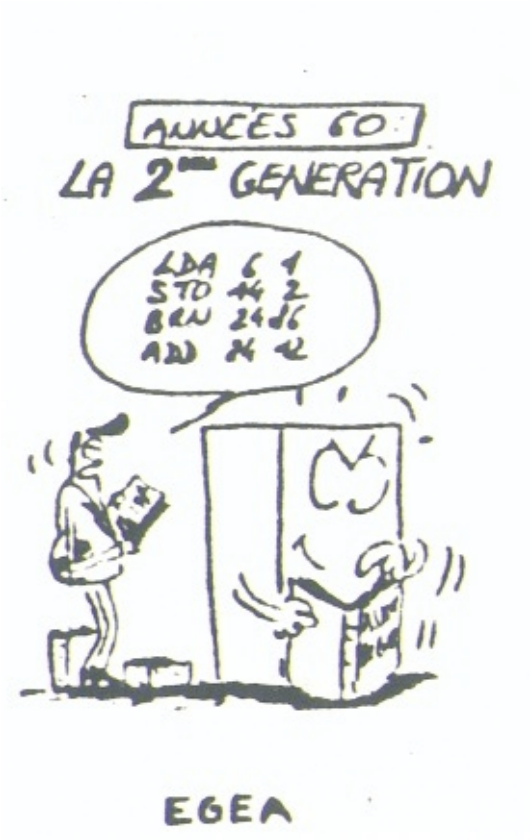
[Yab79] S. Yablonski. *Introduction au machines discrètes*. MIR - Moscou (traduit du russe par D. Embarek), 1979.

VOUZZAVEDIBISAR : 3.75 M (par Michel AVEZARD alias Zevar)

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

Volume 98 – juin 2014



Publication trimestrielle, gratuite, de l'Université d'Aix-Marseille
http://sites.univ-provence.fr/biaa

Impression : juin 2014

ISSN 0291 - 5413

Couverture : dessin de Michel Avezard (Zevar) interprété par la MMIAGe (1985-1987)

(2014) B.I.A.A. no **98** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

ÉDITORIAL : Histoires d'Universités : le travail en miettes

Pierre DUBOIS (le blog "Histoires d'Universités")

Résumé. – Le travail en miettes (ouvrage de Georges Friedmann 1956) me vient en mémoire quand je constate les premiers dégâts de l'arrêté du 31 juillet 2009 sur le référentiel national d'équivalences horaires pour les enseignants-chercheurs. J'y ai déjà consacré plusieurs chroniques critiques, "Enseignants aux forfaits" (2 septembre 2009) et "Suivi du référentiel" (14 novembre 2009). J'écrivais dans cette dernière chronique : "Le comité de suivi du référentiel ne pourra que constater progressivement les dégâts produits par l'arrêté : non mise en application dans certaines universités par refus de leur conseil d'administration, distribution de forfaits très généreux dans les universités "riches". Le pire est donc certainement devant nous".

Le pire est arrivé. Une université, l'université de la Méditerranée (Aix-Marseille 2) [1], a défini "les principes généraux de répartition des services entre les différentes fonctions des enseignants-chercheurs". Deux principes affichés en introduction du texte sont certes louables et excellents : "l'université souhaite que la très grande majorité des enseignants-chercheurs consacre la moitié de son temps de travail annuel à une activité de recherche"... "Il est souhaitable que les enseignants-chercheurs n'assurent qu'un nombre raisonnable d'heures complémentaires au-delà de leur service statutaire".

Les pages suivantes du texte décrivent "le travail en miettes", les activités pour lesquelles l'enseignant-chercheur peut faire valoir une diminution de sa charge de service de 192 heures de travaux dirigés par an : "innovation pédagogique", "activités d'encadrement d'étudiants en formation initiale, continue, dans le cadre de l'apprentissage et de la VAE", "responsabilités de structures ou de missions pédagogiques", "animation, encadrement ou valorisation de la recherche", "activité d'animation de projet scientifique", "autres activités ou activités mixtes". Travail en miettes : le texte attribue, par exemple, 0,1 à 0,2 heure équivalent TD pour l'encadrement d'un stagiaire, multiplié par le nombre de crédits ECTS attribués au stage dans la formation. Il est nécessaire de se munir d'une calculette. Comptes d'apothicaire !

Travail en miettes, mais dans une organisation antérieure à l'organisation scientifique du travail, l'OST taylorienne. Un siècle de retard pour les universités ! L'ouvrier dans une organisation taylorienne se voit prescrire des temps et des méthodes, un résultat. S'il n'y arrive pas, il est "viré". Rien de tel pour les activités listées dans le référentiel : pas d'analyse des temps réels pour les activités exercées, pas d'analyse du travail, pas de contrainte de résultats et de productivité. Un forfait pour le suivi d'un stagiaire, mais aucun contrôle pour prouver que ce suivi a été réalisé et qu'il a apporté une valeur ajoutée à l'étudiant.

Bref, ce référentiel d'équivalences horaires et sa première application détaillée dans une université sont pitoyables. Il renvoie les universités à une organisation obsolète depuis plusieurs décennies. Il est un cache-sexe inefficace pour masquer la détérioration

1. Note de l'éditeur : Les trois universités d'Aix Marseille, soient : l'Université de Provence (Aix-Marseille I), l'Université de la Méditerranée (Aix-Marseille II) et l'Université Paul Cézanne (Aix-Marseille III), ont fusionné en janvier 2012 et sont devenues l'Université d'Aix-Marseille.

du statut salarial des enseignants-chercheurs. Les enseignants qui accepteraient de faire valoir telle ou telle activité pour avoir une petite décharge de service d'enseignement se rangeraient sûrement dans une profession en voie de profonde déqualification.

Ils seraient même infiniment plus "cocus" que les ouvriers du temps de Taylor. Ceux-là avaient au moins bénéficié d'augmentations de salaires parce que l'organisation du travail était devenue nettement plus productive. Pourquoi "cocus" ? Les universités, passées aux responsabilités et compétences élargies (RCE), n'ont même pas la capacité financière de faire face à de trop nombreuses décharges de services d'enseignement. De notre informateur local : "l'université de la Méditerranée vient de se rendre compte qu'elle n'a pas de quoi payer ; elle a ouvert l'enveloppe que le ministère lui avait envoyé, et il s'avère qu'elle était vide… Pour ceux qui ont utilisé le référentiel pour remplir leur service 2009-2010, pas de problème, on ne peut plus leur demander de compléter, et on n'osera pas leur demander le complément l'an prochain. Pour les braves cons qui se sont portés volontaires pour compléter des enseignements, ils se passeront de leurs heures complémentaires : le sarkozysme appliqué à l'université, c'est travailler plus pour gagner pareil".

Présidents d'université, enseignants-chercheurs, syndicalistes représentant les enseignants, refusez le référentiel national des tâches ! Revendiquez de meilleures classifications et donc de meilleurs salaires pour les enseignants-chercheurs, revendiquez la place qu'ils méritent, devraient avoir dans la société ! Ils devraient être l'élite de la Nation, ne les abaissez pas à se défendre, heure de TD par heure TD, contre la déqualification ! Les enseignants-chercheurs payés correctement s'investissent dans toutes les tâches de leur fonction. Ne les obligez pas à entrer dans des comptes d'apothicaires ! Le référentiel national des tâches, c'est la fin de la profession d'enseignant-chercheur, le début de frustrations individuelles contre-productives dans les universités, leur recul vers une organisation pré-taylorienne obsolète. Le pire n'est cependant jamais certain ! Action !

(2014) B.I.A.A. no **98** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

Qu'est-ce qu'une machine ? (II/III)

Eric OLIVIER [1] [2]

Résumé. – La théorie des machines de Turing reformule et clarifie un certain nombre de questions portant sur les fondements (logiques) des mathématiques. Ainsi la question *"Qu'est-ce qu'une machine ?"* est-elle équivalente à la question *"Qu'est-ce qu'un calcul ?"*. Richard Feynman résume cela en affirmant que *n'importe quelle procédure de calcul à laquelle on pourrait penser, est équivalente au calcul d'une machine de Turing – les fonctions récursives générales sont Turing-calculables et vice-versa – et on peut donc prendre "Turing-calculable" pour un synonyme effectif de "calculable"*. Notons enfin que le calcul *automatique* (i.e. le calcul effectué par une machine de Turing) distingue la notion de *proposition démontrable* de celles de *proposition vraie, décidable, indécidable* : cela éclaire les travaux révolutionnaires de Gödel sur la *complétude* et la *consistance* des théories mathématiques.

1. CALCULABILITÉ : UN APERÇU HISTORIQUE

La mécanisation du calcul est un vieux problème : les premiers bouliers ainsi que leurs ancêtres, les abaques, sont déjà utilisés plusieurs siècles avant notre ère. Plus récemment, les calculs algébriques prennent la forme abstraite d'*algorithme* avec Al-Khawarizmi, où se matérialisent sous la forme mécanique d'une *machine à calculer*, avec Pascal (1642-1645) et Leibniz [Lei10]. La formalisation moderne du problème est amorcée par Hilbert puis par Gödel [Göd31] et Herbrand [Her31] ; enfin Church introduit le λ-calcul et la notion de fonction *effectivement calculable* [3] [Chu36] ; c'est Stephen Kleene (élève de Church), qui le premier parle de *Thèse de Church*. Il est maintenant admis (avec Gödel, Church et Kleene) de parler de *Thèse de Church-Turing* du fait de l'importance théorique du concept de machine inventé par Turing (voir Théorème 5.4 ci-dessous). Toute l'ambiguité du statut mathématique de cette *"thèse "* vient du fait que la notion de calcul effectif ne peut être complètement (définitivement) formalisée : la raison en est que l'effectivité d'un calcul dépend des *moyens mécaniques/physiques* qui sous-tendent le modèle abstrait. Ainsi, un calcul est un événement du monde réel dont le résultat est obtenu par une mesure au sens de la science physique. L'*incertitude* inhérente aux processus de mesure physique – en particulier dans le domaine quantique – posent des questions quant aux limites *physiques/techniques* de la calculabilité : la *Thèse de Church-Turing* ne se réduit donc pas à un problème interne aux mathématiques ! La notion de *Turing-calculabilité* est ce qui – jusqu'à présent – se rapproche le plus de ce qu'on peut imaginer être *l'effectivement calculable*.

1. GDAC-I2M UMR 7373 CNRS Université d'Aix-Marseille
2. eric.olivier@univ-amu.fr

3. Dans [Chu36] Church écrit : *"L'objectif du présent article est de proposer une définition de la caculabilité effective dont on pense qu'elle correspond de manière satisfaisante à la notion vague et intuitive au moyen de laquelle les problèmes de cette classe sont souvent énoncés et de montrer, à l'aide d'un exemple, que les problèmes de cette sorte ne sont pas tous résolubles."*

2. FONCTIONS PRIMITIVEMENT RÉCURSIVES

Par convention $\mathcal{A}(\mathbb{N}^0,\mathbb{N}) \equiv \{0\}$; lorsque $n \geq 1$, l'ensemble $\mathcal{A}(\mathbb{N}^n,\mathbb{N})$ est formé des applications/fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$. Par convention, $\mathbb{N}^m \times \mathbb{N}^n$ est identifié à $\mathbb{N}^{m+n}$ et

$$\big((x_1,\dots,x_m),(y_1,\dots,y_n)\big) \equiv (x_1,\dots,x_m,y_1,\dots,y_n).$$

La *fonction successeur* $\mathrm{Succ} : \mathbb{N} \to \mathbb{N}$ est telle que

$$\mathrm{Succ}(x) = x+1 \ ;$$

pour tout entier $n \geq 1$, la *fonction nulle* $O_n \in \mathcal{A}(\mathbb{N}^n,\mathbb{N})$ avec

$$O_n(x) = 0 \ ;$$

pour tout entier $n \geq 1$ et tout $1 \leq i \leq n$, la *projection* $P_{n,i} \in \mathcal{A}(\mathbb{N}^n,\mathbb{N})$ t.q.

$$P_{n,i}(x_1,\dots,x_p) = x_i \ ;$$

la *composition* : pour tout $(G,H_1,\dots,H_p) \in \mathcal{A}(\mathbb{N}^p,\mathbb{N}) \times \mathcal{A}(\mathbb{N}^q,\mathbb{N}) \times \cdots \times \mathcal{A}(\mathbb{N}^q,\mathbb{N})$ l'application $F = \mathrm{Comp}(G,H_1,\dots,H_p) \in \mathcal{A}(\mathbb{N}^q,\mathbb{N})$ est telle que

$$F(x_1,\dots,x_q) := G\Big(H_1(x_1,\dots,x_q),\dots,H_p(x_1,\dots,x_q)\Big) \ ;$$

la *récurrence simple* : pour tout $(G,H) \in \mathcal{A}(\mathbb{N}^n,\mathbb{N}) \times \mathcal{A}(\mathbb{N}^{n+1},\mathbb{N})$ l'application $F = \mathrm{Rec}(G,H) \in \mathcal{A}(\mathbb{N}^{n+1},\mathbb{N})$ est définie par les conditions suivantes, soient

$$\begin{aligned} F(0,x_1,\dots,x_n) &= G(x_1,\dots,x_n) \quad (= 0 \text{ si } n=0) \\ F(x_0+1,x_1,\dots,x_n) &= H\big(F(x_0,\dots,x_n),x_1,\dots,x_n\big) \quad (= H\big(F(x_0)\big) \text{ si } n=0). \end{aligned}$$

Définition 2.1. *La famille des fonctions primitivement récursives est la plus petite famille de fonctions dans $\bigcup_{n=0}^{\infty} \mathcal{A}(\mathbb{N}^n,\mathbb{N})$ vérifiant les conditions suivantes :*

(i) : les fonctions Succ*, O_n (avec $n \geq 1$) et $P_{n,i}$ (avec $1 \leq i \leq n$) sont primitivement récursives.*

(ii) : la famille des fonctions primitivement récursives est close pour les schémas de composition et de récurrence simple.

Exemple 2.2. *(0) : L'application identité $n \mapsto \mathrm{Id}(n) = n$ est primitivement récursive ; elle peut être définie en posant* $\mathrm{Id} = \mathrm{Rec}(0,\mathrm{Succ})$

$$\begin{aligned} \mathrm{Id}(0) &= 0 \\ \mathrm{Id}(n+1) &= \mathrm{Succ}(\mathrm{Id}(n)) = \mathrm{Id}(n)+1. \end{aligned}$$

Notons aussi deux autres fonctions élémentaires, soient $(m,n) \mapsto \mathrm{Eq}(m,n) = 1$ si $m=n$ et 0 si non) et $n \mapsto \mathrm{Pred}(n)$ telle que $\mathrm{Pred} \circ \mathrm{Succ} = \mathrm{Id}$ *et* $\mathrm{Pred}(0) = 0$ *(exercice : définir ces fonctions comme primitivement récursives).*

(1) : L'application somme $(m,n) \mapsto \mathrm{Sum}(n,m) = n+m$ est primitivement récursive ; elle peut être définie en posant

$$\begin{aligned} \mathrm{Sum}(0,m) &= P_{2,2}(0,m) = m \\ \mathrm{Sum}(n+1,m) &= \mathrm{Succ}(\mathrm{Sum}(n,m)) \end{aligned}$$

(2) L'application soustraction $(n,m) \mapsto \mathrm{Sub}(n,m) = n - m$ *si* $n \geq m$ *et* $\mathrm{sub}(n,m) = 0$ *sinon est aussi primitive récursive ; elle peut être définie en posant*

$$\begin{aligned} \mathrm{Sub}(n,0) &= n \\ \mathrm{Sub}(n,m+1) &= \mathrm{Pred}(\mathrm{sub}(n,m)) \end{aligned}$$

(3) : L'application produit $\mathrm{prod} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *t.q.* $\mathrm{prod}(n,m) = n * m$ *est primitivement récursive ; elle peut être définie en posant*

$$\begin{aligned} \mathrm{prod}(0,m) &= 0 \\ \mathrm{prod}(n+1,m) &= \mathrm{Sum}(\mathrm{prod}(n,m),m) \end{aligned}$$

(4) : L'application puissance $\mathrm{Pow} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *t.q.* $\mathrm{Pow}(n,m) = m^n$ *est primitivement récursive ; elle peut être définie en posant*

$$\begin{aligned} \mathrm{Pow}(0,m) &= 1 \\ \mathrm{Pow}(n+1,m) &= \mathrm{prod}(\mathrm{Pow}(n,m),m) \end{aligned}$$

(5) : L'application logarithme $\log : \mathbb{N}^2 \to \mathbb{N}$ *est définie de sorte que (convention)* $\log(0,b) = 0$ *et pour* $a \geq 1$, $\log(a,b)$ *est le plus grand entier* k *tel que* $b^k \leq a$. *On définie cette application par un schéma de récurrence, en posant :*

$$\begin{aligned} \log(0,b) &= 0 \\ \log(a+1,b) &= \log(a,b) + \mathrm{Eq}\Big(\mathrm{Pow}\big(b, 1+\log(a,b)\big), a+1\Big) \end{aligned}$$

Pour $b \geq 2$ *un entier donné, le logarithme de base* b *est l'application partielle* $a \mapsto \log_b(a) = \log(a,b)$ *(avec* $\log_b(0) = 0$*). Ainsi, la récurrence définissant* $\log(a,b)$ *peut se lire*

$$\log_b(a+1) = \log_b(a) + \mathrm{Eq}\Big(b^{1+\log_b(a)}, a+1\Big).$$

Une remarque intuitive – et importante en pratique – est que

$$\ell_b(a) := 1 + \log_b(a) \tag{1}$$

est la longueur de la représentation b-addique de a.

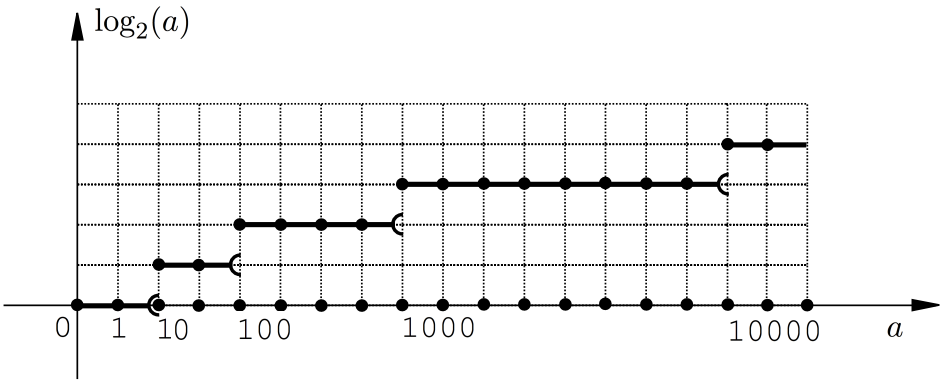


FIGURE 3. Logarithme binaire.

3. FONCTION D'ACKERMANN

Dans les années 1920, Wilhelm Ackermann et Gabriel Sudan étudient – sous la direction de David Hilbert – les fondements de la calculabilité. Sudan est le premier à donner un exemple de fonction *effectivement calculable* mais non primitivement récursive. Peu après (1928), Ackermann publie indépendamment son propre contre-exemple (une fonction dépendant de trois variables). Dans son article [Hil26] portant sur la construction des nombres transfinis, Hilbert conjecture que la fonction d'Ackermann n'est pas primitivement récursive. Cette conjecture est établie par Ackermann dans [Ack28]. Une fonction semblable de seulement deux variables – dérivée de celle d'Ackermann par Rózsa Péter et Raphael Robinson – est aujourd'hui connue sous le nom de *fonction d'Ackermann* : considérons la suite $A_0, A_1, \dots$ de fonctions primitivement récursives (exercice) t.q.

$$\begin{aligned} A_0(q) &= q+1 \\ A_{p+1}(0) &= A_p(1) \\ A_{p+1}(q+1) &= A_p\big(A_{p+1}(q)\big). \end{aligned}$$

Par définition, la *fonction d'Ackermann* est l'application $(p,q) \mapsto A_p(q) =: \mathrm{Ack}(p,q)$, de sorte qu'elle vérifie les relations récursives

$$\mathrm{Ack}(p+1,0) = \mathrm{Ack}(p,1) \quad \text{et} \quad \mathrm{Ack}(p+1,q+1) = \mathrm{Ack}\big(p, \mathrm{Ack}(p+1,q)\big). \tag{2}$$

Les valeurs successives de $\mathrm{Ack}(p,q)$ sont effectivement calculables ; les premières valeurs de la fonction se trouvent dans le tableau suivant.

	0	1	2	3	…	q	…
0	1	2	3	4	…	$q+1$	…
1	2	3	4	5	…	$q+2$	…
2	3	5	7	9	…	$2q+3$	…
3	5	13	29	61	…	$2^{q+3}-3$	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	…
p	$\mathrm{Ack}(p,0)$	$\mathrm{Ack}(p,1)$	$\mathrm{Ack}(p,2)$	$\mathrm{Ack}(p,3)$	…	$\mathrm{Ack}(p,q)$	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Lemme 3.1. *Pour tout $p \geq 0$, les propriétés suivantes sont satisfaites*

(i) : L'application partielle $\mathrm{Ack}(p,\cdot)$ est croissante ;

(ii) : L'application partielle $\mathrm{Ack}(\cdot,p)$ est croissante ;

(iii) : Pour tout $q \geq 0$ on a : $\mathrm{Ack}(p,q) \geq \mathrm{Ack}(0,q) = q+1 > q$;

(iv) : Pour tout $q \geq 0$ on a : $\mathrm{Ack}(p,q+1) \leq \mathrm{Ack}(p+1,q)$;

(v) : Pour tout $q,q,r \geq 0$ on a : $\mathrm{Ack}(p,r) + \mathrm{Ack}(q,r) \leq \mathrm{Ack}(\max\{p,q\}+4,r)$.

Preuve. (i)-(ii)-(iii) : Exercices.

(iv) : Pour $p \geq 0$ arbitrairement fixé, on montre par récurrence sur $q \geq 0$ que

$$\mathrm{Ack}(p,q+1) \leq \mathrm{Ack}(p+1,q).$$

La propriété est vraie pour $q = 0$ par définition, puisque la première relation de (2) donne

$$\mathrm{Ack}(p+1,0) = \mathrm{Ack}(p,0+1).$$

Pour la récurrence, soit $q \geq 0$ t.q. $\mathrm{Ack}(p, q+1) \leq \mathrm{Ack}(p+1, q)$; l'inégalité

$$q + 2 = (q+1) + 1 = \mathrm{Ack}(0, q+1) \leq \mathrm{Ack}(p, q+1)$$

entraîne par croissance que

$$\begin{array}{rcl} \mathrm{Ack}(p, q+2) & = & \mathrm{Ack}\big(p, \mathrm{Ack}(p, q+1)\big) \\ \text{(récurrence et croissance)} & \leq & \mathrm{Ack}\big(p, \mathrm{Ack}(p+1, q)\big) = \mathrm{Ack}(p+1, q+1) \end{array}$$

(v) : On utilise (i) (ii) et le fait que $\mathrm{Ack}(2, r) = 2r + 3$, de sorte que pour $p, q \geq 0$:

$$\begin{array}{rcl} \mathrm{Ack}(p, r) + \mathrm{Ack}(q, r) & \leq & 2\mathrm{Ack}\Big(\max\{p, q\}, r\Big) \\ \text{(croissance)} & \leq & 2\mathrm{Ack}\Big(\max\{p, q\}, r\Big) + 3 \\ & = & \mathrm{Ack}\Big(2, \mathrm{Ack}\big(\max\{p, q\}, r\big)\Big) \\ \text{(croissance)} & \leq & \mathrm{Ack}\Big(\max\{p, q\} + 2, \mathrm{Ack}\big(\max\{p, q\}, r\big)\Big) \\ \text{(croissance)} & \leq & \mathrm{Ack}\Big(\max\{p, q\} + 2, \mathrm{Ack}\big(\max\{p, q\} + 3, r\big)\Big) \\ & = & \mathrm{Ack}\Big(\max\{p, q\} + 3, r + 1\Big) \\ \text{(d'après (iv))} & \leq & \mathrm{Ack}\Big(\max\{p, q\} + 4, r\Big) \end{array}$$

□

À première vue, la définition de la fonction $\mathrm{Ack}(\cdot, \cdot)$ ne rentre pas dans le schéma de la récurrence simple déjà introduit. En fait (Théorème 3.3) cette fonction n'est pas primitivement récursive : la démonstration de ce résultat (théoriquement important) découle facilement de la proposition suivante.

Proposition 3.2. *Pour toute fonction $F : \mathbb{N}^p \to \mathbb{N}$ primitivement récursive il existe $K_F \geq 0$ t.q.*

$$F(x_1, \ldots, x_p) < \mathrm{Ack}(K_F, x_1 + \cdots + x_p). \tag{3}$$

Preuve. Soit $\mathcal{R}_0$ le sous ensemble de $\bigcup_{p=0}^{\infty} \mathcal{A}(\mathbb{N}^p, \mathbb{N})$, constitué des fonctions Succ, O_n (avec $n \geq 1$) et $P_{n,i}$ (avec $1 \leq i \leq n$). On définit par induction (sur $n \geq 0$) la suite

$$\mathcal{R}_0 \subset \mathcal{R}_1 \subset \cdots \subset \mathcal{R}_n \cdots$$

de sous-ensembles de $\bigcup_{n=1}^{\infty} \mathcal{A}(\mathbb{N}^n, \mathbb{N})$, par les conditions suivantes,

$$\begin{array}{rcl} (i) & : & \forall p, q \geq 1,\ \forall (G, H_1, \ldots, H_p) \in \mathcal{A}(\mathbb{N}^p, \mathbb{N}) \times \mathcal{A}(\mathbb{N}^q, \mathbb{N}) \times \cdots \times \mathcal{A}(\mathbb{N}^q, \mathbb{N}), \\ & & \quad (G, H_1, \ldots, H_p) \in \mathcal{R}_n \times \cdots \times \mathcal{R}_n \Rightarrow \mathrm{Comp}(G, H_1, \ldots, H_p) \in \mathcal{R}_{n+1}\ ; \\ (ii) & : & \forall p \geq 1,\ \forall (G, H) \in \mathcal{A}(\mathbb{N}^p, \mathbb{N}) \times \mathcal{A}(\mathbb{N}^{p+1}, \mathbb{N}), \\ & & \quad (G, H) \in \mathcal{R}_n \times \mathcal{R}_n \Rightarrow \mathrm{Rec}(G, H) \in \mathcal{R}_{n+1}. \end{array}$$

de sorte que F est primitivement récursive si et seulement si il existe $n \geq 0$ t.q. $F \in \mathcal{R}_n$. On vérifie (exercice) que oute fonction F dans $\mathcal{R}_0$ satisfait (3) avec $k = 1$. Dans la suite $(*)$ est l'hypothèse de récurrence sur $n \geq 0$ assurant que toute fonction F dans $\mathcal{R}_n$ satisfait (3) pour l'entier K_F (dépendant de F). Pour le schéma de composition considérons

$$F = \mathrm{Comp}(G, H_1, \ldots, H_p)$$

avec

$$(G, H_1, \ldots, H_p) \in \mathcal{A}(\mathbb{N}^p, \mathbb{N}) \times \mathcal{A}(\mathbb{N}^q, \mathbb{N}) \times \cdots \times \mathcal{A}(\mathbb{N}^q, \mathbb{N})$$

et $G, H_1, \ldots, H_p \in \mathcal{R}_n$; alors il vient successivement :

$$\begin{array}{rcl}
F(x_1, \ldots, x_q) & = & G\Big(H_1(x_1, \ldots, x_q), \ldots, H_p(x_1, \ldots, x_q)\Big) \\
\text{(récurrence } (*) \text{ sur } G) & < & \mathrm{Ack}\Big(K_G, \sum_{i=1}^{p} H_i(x_1, \ldots, x_q)\Big) \\
\text{(récurrence } (*) \text{ sur les } H_i \text{ et croissance}) & \leq & \mathrm{Ack}\Big(K_G, \sum_{i=1}^{p} \mathrm{Ack}(K_{H_i}, x_1 + \cdots + x_q)\Big) \\
\text{(Lemme 3.1-v et } K = \max\{K_{H_1}, \ldots, K_{H_p}\} + 4p) & \leq & \mathrm{Ack}\Big(K_G, \mathrm{Ack}(K, x_1 + \cdots + x_q)\Big) \\
(K_* = \max\{K, K_G\} \text{ et croissance}) & \leq & \mathrm{Ack}\Big(K_*, \mathrm{Ack}(K_*, x_1 + \cdots + x_q)\Big) \\
\text{(croissance)} & \leq & \mathrm{Ack}\Big(K_*, \mathrm{Ack}(K_* + 1, x_1 + \cdots + x_q)\Big) \\
& = & \mathrm{Ack}\Big(K_* + 1, x_1 + \cdots + x_q + 1\Big) \\
\text{(Lemme 3.1-(iv))} & \leq & \mathrm{Ack}\Big(K_* + 2, x_1 + \cdots + x_q\Big)
\end{array}$$

Il reste à considérer le schéma de récurrence : soit alors $F = \mathrm{Rec}(G, H)$, avec

$$(G, H) \in \mathcal{A}(\mathbb{N}^p, \mathbb{N}) \times \mathcal{A}(\mathbb{N}^{p+1}, \mathbb{N})$$

et $G, H \in \mathcal{R}_n$; en l'hypothèse de récurrence $(*)$ sur G donne :

$$F(0, x_1, \ldots, x_p) = G(x_1, \ldots, x_p) < \mathrm{Ack}\big(K_G, x_1 + \cdots + x_p\big) \ ;$$

soit alors $(**)$ l'hypothèse de récurrence sur $m \geq 0$, assurant que

$$F(m, x_1, \ldots, x_p) < \mathrm{Ack}(K_*, m + x_1 + \cdots + x_p),$$

avec

$$K_* := \max\{K_G, K_H\} + 4$$

(ce qui donne en particulier $K_H + 3 \leq K_* - 1$), il est alors possible d'écrire :

$$
\begin{array}{rcl}
F(m+1,x_1,\ldots,x_p) & = & H\Big(F(m,x_1,\ldots,x_p),x_1,\ldots,x_p\Big)\\
\text{(récurrence } (*) \text{ sur } H) & < & \mathrm{Ack}\Big(K_H,F(m,x_1,\ldots,x_p)+x_1+\cdots+x_p\}\Big)\\
\text{(récurrence } (**)) & \leq & \mathrm{Ack}\Big(K_H,\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)+x_1+\cdots+x_p\Big)\\
(\mathrm{Ack}(p,q)>q \text{ et croissance}) & \leq & \mathrm{Ack}\Big(K_H,2\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)\Big)\\
\text{(croissance)} & \leq & \mathrm{Ack}\Big(K_H,2\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)+3\Big)\\
(\mathrm{Ack}(2,q)=2q+3) & = & \mathrm{Ack}\Big(K_H,\mathrm{Ack}\Big(2,\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)\Big)\Big)\\
\text{(croissance)} & \leq & \mathrm{Ack}\Big(K_H+1,\mathrm{Ack}\Big(K_H+2,\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)\Big)\Big)\\
& = & \mathrm{Ack}\Big(K_H+2,\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)+1\Big)\\
\text{(Lemme 3.1-(iv))} & \leq & \mathrm{Ack}\Big(K_H+3,\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)\Big)\\
(K_*:=\max\{K_G,K_H\}+4) & \leq & \mathrm{Ack}\Big(K_*-1,\mathrm{Ack}\big(K_*,m+x_1+\cdots+x_p\big)\Big)\\
& = & \mathrm{Ack}\Big(K_*,m+x_1+\cdots+x_p+1\Big)\\
& = & \mathrm{Ack}\Big(K_*,(m+1)+x_1+\cdots+x_p\Big)
\end{array}
$$

□

Théorème 3.3 (Ackermann-Péter-Robinson)**.** $\mathrm{Ack}(\cdot,\cdot)$ *n'est pas primitivement récursive.*

Preuve. Si $\mathrm{Ack}(\cdot,\cdot)$ est primitivement récursive alors la fonction

$$f=\mathrm{Comp}\Big(\mathrm{Ack}\big(\cdot,\cdot\big),P_{1,1},P_{1,1}\Big):\mathbb{N}\to\mathbb{N}\quad\text{(i.e. } f(n)=\mathrm{Ack}(n,n))$$

est aussi primitivement recursive. Donc (Proposition 3.2) il existe K t.q. $f(n)<\mathrm{Ack}(K,n)$ pour tout n : en particulier $\mathrm{Ack}(K,K)=f(K)<\mathrm{Ack}(K,K)$: c'est contradictoire.

□

4. Fonctions récursives

Dans son cours de 1934 [Göd65], Kurt Gödel reprend les idées introduites dans son article de 1931 (ainsi que celles développées en parallèles par Herbrand dans [Her31]). En particulier il y précise la notion de fonctions primitivement récursives (qu'il appelle récursives) et en donne une généralisation (qu'il appelle les fonctions récursive générales). Stephen Kleene (élève de Church) est un des rédacteurs du cours de Gödel ; ce travail le mène à son article fondamental de 1936 [Kle36b] (pour plus de détails, c.f. [Cha10, Chap. 15]) ; il y définit plusieurs schémas donnant la même classe de fonctions effectivement calculables appelées *fonctions récursives*. Cette classe contient les fonctions primitivement récursives, mais aussi la fonction d'Ackermann : elle coïncide avec la classes des fonctions récursives générales de Gödel mais aussi avec les fonctions Turing-calculables.

Le *schéma de minimisation* introduit par Kleene [Kle36a], est le chainon manquant entre récursivité primitive et récursivité. Pour toute application $G \in \mathcal{A}(\mathbb{N}^{n+1}, \mathbb{N})$, on note $\mathrm{Min}(G)$ l'application $F \in \mathcal{A}(\mathbb{N}^n, \mathbb{N} \sqcup \{\omega\})$ (avec $\omega \notin \mathbb{N}$) telle que

$$F(x) = \begin{cases} \min\big\{y \ ; \ G(y,x) = 0\big\} & \text{si } \{y \ ; \ G(y,x) = 0\} \neq \emptyset \\ \omega & \text{si non} \end{cases}$$

On pourra écrire dans la suite $F(x) = \mathrm{Min}\big(G(\cdot, x)\big)$.

Définition 4.1 (Kleene). *La famille des fonctions récursives est la plus petite famille de fonctions dans $\bigcup_{n=1}^{\infty} \mathcal{A}(\mathbb{N}^n, \mathbb{N} \sqcup \{\omega\})$ qui vérifie les conditions suivantes.*

(i) : les fonctions primitivement récursives sont récursives.

(ii) : la famille des fonctions récursives est close pour les schémas de composition, de récurrence simple et de minimisation.

Remarque 4.2. *(1) : Par définition $F = (F_1, \ldots, F_q) : \mathbb{N}^p \to \mathbb{N}^q$ (resp. $\mathbb{N}^q \cup \{\omega\}$) est primitivement récursive (resp. récursive) si et seulement si chaque composante $F_1, \ldots, F_q$ l'est.*

(2) : Les schémas de composition, récurrence et minimisation ont été définis pour les fonctions à valeurs dans $\mathbb{N}$ (ou $\mathbb{N} \cup \{\omega\}$) ; ces définitions s'étendent pour les fonctions à valeurs dans $\mathbb{N}^q$ (ou $\mathbb{N}^q \cup \{\omega\}$, pour tout $q \geq 1$.

(3) : Le schéma de minimisation intervient de manière essentielle comme caractérisation de l'arrêt d'une machine de Turing : voir (6) dans la démonstration du Théorème 5.4 (c.f. infra) établissant l'identification entre fonctions récursives et Turing-calculables.

5. Récursivité et Turing-calculabilité

Afin de démontrer (Théorème 5.4) l'égalité entre la classe des fonctions récursives (voir Définition 4.1) et celle des fonctions Turing calculables (voir Définition 5.2), nous aurons besoin de quelques notions intermédiaires concernant des questions de numérations. L'algorithme glouton (greedy algorithm) binaire détermine la bijection $\mathrm{Gr}_2 : \mathbb{N} \to \mathcal{N}_2 := \{\mathtt{0}\} \cup \mathtt{1}\{\mathtt{0},\mathtt{1}\}^*$ qui a tout entiers associe sa représentation binaire standard. De même que pour l'application Gr_2 donnant l'écriture binaire d'un entier, on définit la bijection $\mathrm{Gr}_3 : \mathbb{N} \to \mathcal{N}_3 := \{\mathtt{0}\} \cup \bigcup_{\mathtt{i}=\mathtt{1},\mathtt{2}} \mathtt{i}\{\mathtt{0},\mathtt{1},\mathtt{2}\}^*$, qui a tout entier n associe sa représentation ternaire. Nous aurons aussi besoin des applications $\mathrm{Rep}_3 : \{\mathtt{0},\mathtt{1},\mathtt{2}\}^* \to \mathbb{N}$ et $\mathrm{Mir} : \{\mathtt{0},\mathtt{1},\mathtt{2}\}^* \to \{\mathtt{0},\mathtt{1},\mathtt{2}\}^*$ définies pour tout mot $a_0 \cdots a_k \in \{\mathtt{0},\mathtt{1},\mathtt{2}\}^*$ en posant

$$\begin{aligned} \mathrm{Rep}_3(a_0 \cdots a_k) &= a_0 3^k + \cdots + a_k 3^0 \\ \mathrm{Mir}(a_0 \cdots a_k) &= a_k \cdots a_0. \end{aligned}$$

Nous utilisons la convention $\mathrm{Rep}_3(\phi)$, de sorte que par exemple,

$$\mathrm{Gr}_3 \circ \mathrm{Rep}_3(\mathtt{00110201}) = \mathtt{110201} \quad \text{et} \quad \mathrm{Gr}_3 \circ \mathrm{Rep}_3 \circ \mathrm{Mir}(\mathtt{00110201}) = \mathtt{10201100}$$

L'analogues binaires de Rep_3 est l'application $\mathrm{Rep}_2 : \{\mathtt{0},\mathtt{1},\mathtt{2}\}^* \to \mathbb{N}$ telle que

$$\mathrm{Rep}_2(a_0 \cdots a_k) = a_k 2^0 + a_{k-1} 2^1 + \cdots + a_0 2^k.$$

(Remarquer que Rep_2 est définie sur $\{\mathtt{0},\mathtt{1},\mathtt{2}\}^*$ et non $\{\mathtt{0},\mathtt{1}\}^*$, mais que pour tout $w \in \mathcal{N}_2$ on a $\mathrm{Gr}_2 \circ \mathrm{Rep}_2(w) = w$). Nous noterons aussi $\mathrm{mod}_3(n)$ le reste de la division euclidienne

de n par 3 ; enfin, l'application $\mathrm{Shift}_3 : \mathbb{N} \to \mathbb{N}$ est définie en posant

$$\mathrm{Shift}(n) = \begin{cases} 0 & \text{si } n\{0,1,2\} \\ \mathrm{Rep}_3(a_1 \cdots a_k a_{k+1}) & \text{si } \mathrm{Gr}_2(n) = a_1 \cdots a_k \text{ pour } k \geq 1 \end{cases}$$

(exercice : les fonctions ainsi définies sont primitivement récursives).

Lemme 5.1. *L'application* $\mathrm{Rep}_2 \circ \mathrm{Gr}_3 : \mathbb{N} \to \mathbb{N}$ *est primitivement récursive.*

Définition 5.2. *Un application (totale)* $F : \mathbb{N}^k \to \mathbb{N}$ *(i.e. le domaine de* F *est* $\mathbb{N}^k$ *tout entier) sera dite Turing calculable, s'il existe*[4] $T \in \mathfrak{T}\{0,1,2\}$ *telle que pour tout* $(x_1, \cdots, x_k) \in \mathbb{N}^k$,

$$F(x_1, \cdots, x_k) = y \iff T^*(\texttt{[I)}\,\mathrm{Gr}_2(x_1)2\cdots 2\mathrm{Gr}_2(x_k)) = \texttt{[F)}\,\mathrm{Gr}_2(y).$$

Remarque 5.3. *(1) : Toute partie* A *de* $\mathbb{N}^k$ *est associée à la fonction indicatrice* $\mathbf{1}_A : \mathbb{N}^k \to \mathbb{N}$ *telle que* $\mathbf{1}_A(n) = 1$ *si* $n \in A$ *et* $\mathbf{1}_A(n) = 0$ *si* $n \notin A$ *; alors le langage* $\mathcal{L} := \{\mathrm{Gr}_2(n)\ ;\ n \in A\}$ *est récursif (on dit aussi que* A *est récursif : voir* [Oli14, § 5]*) si et seulement si l'indicatrice* $\mathbf{1}_A$ *est Turing-calculable.*

(2) : Soit $F : A \to \mathbb{N}$ *définie sur une partie (infinie non vide)* A *de* $\mathbb{N}^k$*. Alors* A *est récursif si et seulement si il existe une application* $G = (G_1, \ldots, G_k) : \mathbb{N} \to \mathbb{N}^k$ *(application totale) Turing calculable et telle que* $G(0), G(1), \ldots$ *est la liste complète et ordonnée des éléments de*[5] *de* A*. L'application* $F : A \to \mathbb{N}$ *est alors Turing calculable si et seulement si* A *est une partie récursive de* $\mathbb{N}^k$ *et si de plus, l'application totale* $F \circ G : \mathbb{N} \to \mathbb{N}$ *est Turing-calculable.*

Théorème 5.4. $F \in \mathcal{A}(\mathbb{N}^n, \mathbb{N} \sqcup \{\omega\})$ *est récursive si et seulement si* F *est Turing-calculable.*

Preuve du Théorème 5.4. De par la puissance de calcul offerte par les machines de Turing, il est facile de se convaincre que les fonctions récursives sont Turing-calculables. La difficulté est de démontrer la réciproque. Pour cela, il suffit de considérer une machine de Turing, soit $T \in \mathfrak{T}\{0,1,2\}$ (ici 2 remplace $\star$), calculant une application $F \in \mathcal{A}(\mathbb{N}^n, \mathbb{N})$, i.e.

$$(\mathbf{CAL}) \ : \qquad F(x_1, \ldots, x_n) = y \iff T^*(\texttt{[I)}\,\mathrm{Gr}_2(x_1)2\cdots 2\mathrm{Gr}_2(x_n)) = \mathrm{Gr}_2(y)$$

et de montrer que F est récursive. Tout d'abord, l'ensemble $\mathcal{Q}$ des états internes de T est identifié à une partie finie de $\mathcal{N}_3$ (les entiers ternaires) avec la convention $(\texttt{I}, \texttt{F}) = (1, 0)$. Pour $w_0 = \mathrm{Gr}_2(x_1)2\cdots 2\mathrm{Gr}_2(x_n)$, nous notons $T^k(\texttt{[I)}\,w_0) = v_k\,\texttt{[}\mathcal{Q}_k\texttt{)}\,w_k$ le k-ème cycle de T calculant $F(x_1, \ldots, x_n)$, où par construction v_k et w_k sont des mots dans $\{0,1,2,\}^*$ et $\mathcal{Q}_k$ ($\in \mathcal{N}_3 \subset \{0,1,2,\}^*$) est l'état interne de T. Remarquer que $v_k\,\texttt{[}\mathcal{Q}_k\texttt{)}\,w_k = \texttt{[F)}\,w_{k_0}$ – i.e. $(v_k, \mathcal{Q}_k, w_k) = (\phi, \texttt{F}, w_{k_0})$ – pour tout $k \geq k_0 := \mathrm{Stop}(T, \texttt{[I)}\,w_0)$: nous posons

$$\phi\big(k, \mathrm{Rep}_3 \circ \mathrm{Mir}(w_0 2)\big) := \Big(\mathrm{Rep}_3(2v_k), \mathrm{Rep}_3(\mathcal{Q}_k), \mathrm{Rep}_3 \circ \mathrm{Mir}(w_k 2)\Big) \in \mathbb{N}^3. \tag{4}$$

L'introduction du digit 2 dans (4) se justifie comme suit : par exemple (et de même pour pour w_k) $\mathrm{Rep}_3(v_k) = \mathrm{Rep}_3(0\cdots 0 v_k)$, ce qui engendre des ambiguïtés quand v_k (resp. w_k) est vides (du fait de la convention $\mathrm{Rep}(\phi) = 0$) ; l'ajout du digit 2 dans la première et la troisième composante de ϕ évite ces ambiguïtés (i.e. $\mathrm{Rep}_3(2) \neq \mathrm{Rep}_3(20\cdots 0)$ lorsque $v_k = \phi$) ; en particulier, il est ainsi toujours possible de retrouver (sous forme d'un entier dans $\{0,1,2\}$) le digit à gauche de (resp. lu par) la tête de lecture/écriture, soient

4. Pour pouvoir effectuer des calculs arithmétiques sans effectuer de re-codage, le symbole $\star$ représentant habituellement la case vide (voir [Oli14]) est remplacé par 2.

5. Ici nous avons combiné la définition de la récursivité d'un langage (voir [Oli14, Definition 5.1]) avec la notion de Turing calculabilité : cela assure l'existence de l'application G.

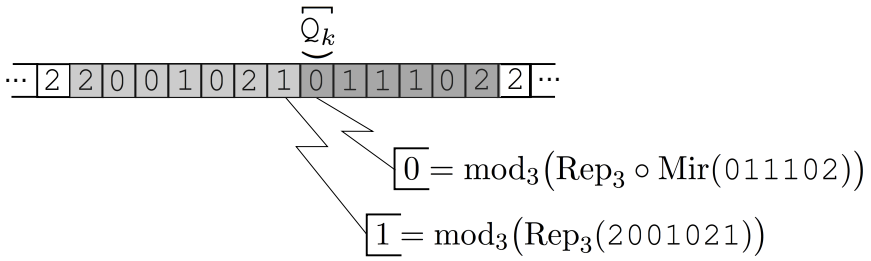


FIGURE 4. Ici, nous supposons que l'état de la machine T à l'étape k du calcul est $v_k\,[\mathtt{Q}_k)\,w_k$ avec $v_k = \mathtt{001021}$ et $w_k = \mathtt{01110}$: il est alors facile d'obtenir par le calcul la *valeur* de la case lue par la tête de lecture, ainsi que celle de la case juste à sa gauche.

respectivement $\mathrm{mod}_3\big(\mathrm{Rep}_3(2v_k)\big)$ et $\mathrm{mod}_3\big(\mathrm{Rep}_3 \circ \mathrm{Mir}(w_k 2)\big)$ (voir Figure 4). Nous allons maintenant étendre ϕ en une application totale $\phi = (\phi_1, \phi_2, \phi_3) : \mathbb{N}^2 \to \mathbb{N}^3$, puis prouver sa (primitive) récursivité. Pour cela, considérons la table de transition de T comme l'application $\tau : \mathcal{Q} \times \{0,1,2\} \to \mathcal{Q} \times \{0,1,2\} \times \{0,1,2\}$, où $\mathtt{L}, \mathtt{R}$ et $\mathtt{S}$ sont respectivement identifiés aux entiers $0, 1$ et 2 et où $\mathcal{Q}$ ($\subset \mathcal{N}_3$) est identifié à $\mathrm{Rep}_3(\mathcal{Q}) \subset \mathbb{N}$; la table τ est alors étendue en une application $\tau = (\tau_1, \tau_2, \tau_3) : \mathbb{N}^2 \to \mathbb{N}^3$ telle que[6] en posant $\tau_i(p,q) = 0$ pour les couples d'entiers (p,q) pour lesquels la table de transition n'est pas définie (la fonction τ est primitivement récursive). D'une part, $T^0(\mathtt{[I)}\,w_0) = \mathtt{[I)}\,w_0$, ce qui donne $(2v_0, \mathtt{Q}_0, w_0 2) = \big(2, \mathtt{1}, w_0 2\big)$ (avec la convention $\mathtt{I} = \mathtt{1}$) et en suivant (4), il est alors cohérent de poser pour tout entier[7] q

$$\phi(0,q) = \big(2,1,q\big) \tag{5}$$

D'autre part, afin de définir $\phi(p+1,q)$, nous introduisons les fonctions intermédiaires

$$\mathrm{Read}(p,q) := \mathrm{mod}_3\Big(\phi_3(p,q)\Big) \quad \text{et} \quad \mathrm{New}(p,q) := \tau_2\Big(\phi_2(p,q), \mathrm{Read}(p,q)\Big)$$

qui représentent respectivement le digit de la case courant lu par la tête de lecture/écriture, ainsi que le digit écrit dans cette même case avant le déplacement de la tête, ce déplacement étant obtenu comme

$$\mathrm{Mov}(p,q) := \tau_3\Big(\phi_2(p,q), \mathrm{Read}(p,q)\Big).$$

6. Si p est l'état interne de la machine et q le symbole lu par la tête de lecture/écriture alors la machine écrit $\tau_2(p,q)$ sur la case courante, se déplace dans la direction $\tau_3(p,q)$ en se plaçant dans l'état interne $\tau_1(p,q)$.

7. Pour le calcul de F, les seuls entiers qui sont mis en jeux sont de la forme

$$q = \mathrm{Rep}_3 \circ \mathrm{Mir}\Big(\mathrm{Gr}_2(x_1)2\cdots 2\mathrm{Gr}_2(x_n)\Big).$$

Du fonctionnement des machines de Turing et de (4) il vient nécessairement que [8]

$$\begin{aligned}
\phi_1(p+1,q) &= \begin{pmatrix} 3\phi_1(p,q)+\mathrm{New}(p,q) \\ \mathrm{Shift}(\phi_1(p,q)) \\ \phi_1(p,q) \end{pmatrix} \cdot \begin{pmatrix} \mathrm{Eq}\big(\mathrm{Mov}(p,q),\mathtt{R}\big) \\ \mathrm{Eq}\big(\mathrm{Mov}(p,q),\mathtt{L}\big) \\ \mathrm{Eq}\big(\mathrm{Mov}(p,q),\mathtt{S}\big) \end{pmatrix} \\
\phi_2(p+1,q) &= \tau_1\Big(\phi_2(p,q),\mathrm{Read}(p,q)\Big) \\
\phi_3(p+1,q) &= \begin{pmatrix} \mathrm{Shift}(\phi_3(p,q)) \\ 3\phi_3(p,q)+\mathrm{New}(p,q) \\ \phi_3(p,q) \end{pmatrix} \cdot \begin{pmatrix} \mathrm{Eq}\big(\mathrm{Mov}(p,q),\mathtt{R}\big) \\ \mathrm{Eq}\big(\mathrm{Mov}(p,q),\mathtt{L}\big) \\ \mathrm{Eq}\big(\mathrm{Mov}(p,q),\mathtt{S}\big) \end{pmatrix}
\end{aligned}$$

Cela permet d'obtenir la primitive récursivité de ϕ en écrivant (à l'aide du schéma de récurrence) $\phi(p+1,q)=\psi\big(\phi(p,q)\big)$, avec $\psi=(\psi_1,\psi_2,\psi_3):\mathbb{N}^3\to\mathbb{N}^3$ t.q.

$$\begin{aligned}
\psi_1(x,y,z) &= \begin{pmatrix} 3x+\tau_2\big(y,\mathrm{mod}_3(z)\big) \\ \mathrm{Shift}(x) \\ x \end{pmatrix} \cdot \begin{pmatrix} \mathrm{Eq}\big(\tau_3\big(y,\mathrm{mod}_3(z)\big),\mathtt{R}\big) \\ \mathrm{Eq}\big(\tau_3\big(y,\mathrm{mod}_3(z)\big),\mathtt{L}\big) \\ \mathrm{Eq}\big(\tau_3\big(y,\mathrm{mod}_3(z)\big),\mathtt{S}\big) \end{pmatrix} \\
\psi_2(x,y,z) &= \tau_1\big(x,\mathrm{mod}_3(z)\big) \\
\psi_3(x,y,z) &= \begin{pmatrix} \mathrm{Shift}(x) \\ 3+\tau_2\big(y,\mathrm{mod}_3(z)\big) \\ x \end{pmatrix} \cdot \begin{pmatrix} \mathrm{Eq}\big(\tau_3\big(y,\mathrm{mod}_3(z)\big),\mathtt{R}\big) \\ \mathrm{Eq}\big(\tau_3\big(y,\mathrm{mod}_3(z)\big),\mathtt{L}\big) \\ \mathrm{Eq}\big(\tau_3\big(y,\mathrm{mod}_3(z)\big),\mathtt{S}\big) \end{pmatrix}
\end{aligned}$$

La conclusion arrive avec le *schéma de minimisation de Kleene* dont le rôle *clef* est de caractériser la *convention d'arrêt des machines de Turing* (i.e. $T(w\,\mathtt{[Q)}\,m)=w\,\mathtt{[Q)}\,m$ si et seulement si $\mathtt{Q}=\mathtt{F}$). Ayant pris soin de poser $\mathtt{F}=\mathtt{0}$ et d'identifier $\mathcal{Q}$ (les états internes de T) avec $\mathrm{Rep}_3(\mathcal{Q})$, nous pouvons définir

$$K(q):=\phi_3\Big(\mathrm{Min}\big(\phi_2(\cdot,q)\big),q\Big)=\phi_3\Big(\min\{k\,;\,\phi_2(k,q)=0\},q\Big), \tag{6}$$

de sorte qu'en faisant $q=\mathrm{Rep}_3\circ\mathrm{Mir}\big(\mathrm{Gr}_2(x_1)2\cdots 2\mathrm{Gr}_2(x_n)2\big)=:H(x_1,\ldots,x_n)$,

$$F(x)=\mathrm{Rep}_2\circ\mathrm{Gr}_3\circ\mathrm{Sub}\Big(K\circ H(x_1,\ldots,x_n),2\cdot\mathrm{Pow}\Big(\ell_3\big(K\circ H(x_1,\ldots,x_n)\big),3\Big)\Big)$$

où pour tout $n\in\mathbb{N}$, $\ell_3(n)$ est la longueur ternaire de $\mathrm{Gr}_3(n)$ définie en (1). On conclut (Lemme 5.1) du fait que $\mathrm{Rep}_2\circ\mathrm{Gr}_3:\mathbb{N}\to\mathbb{N}$ est (primitivement) récursive.

□

RÉFÉRENCES

[Ack28] W. Ackermann. Zum Hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99 :118–33, 1928.

[Cha10] J.-L. Chabert. *Histoire d'algorithmes. Du caillou à la puce*. Belin, 2010.

[Chu36] A. Church. A note on the Entscheidungsproblem (correction pp. 101-102). *Journal of Symbolic Logic*, 1 :40–41, 1936.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der Principia Mathematica und verwandter Systeme, i. *Monatshefte für Mathematik und Physik*, 38 :173–198, 1931.

8. Rappelons que $\mathrm{Eq}(p,q)\in\{0,1\}$ représente l'égalité entre p et q, i.e. $\mathrm{Eq}(p,q)=1$ si et seulement si $p=q$; nous utilisons aussi la notation pratique en produit scalaire sur des vecteurs colonnes à trois composantes.

[Göd65] Kurt Gödel. On undecidable propositions of formal mathematical systems, (1934) lecture notes taken by Kleene and Rosser at the Institute for Advanced Study : reprinted in M. Davis (ed.), 1965.

[Her31] J. Herbrand. Sur la non-contradiction de l'arithmétique. *Journal fur die reine und angewandte Mathematik*, 166 :1–8, 1931.

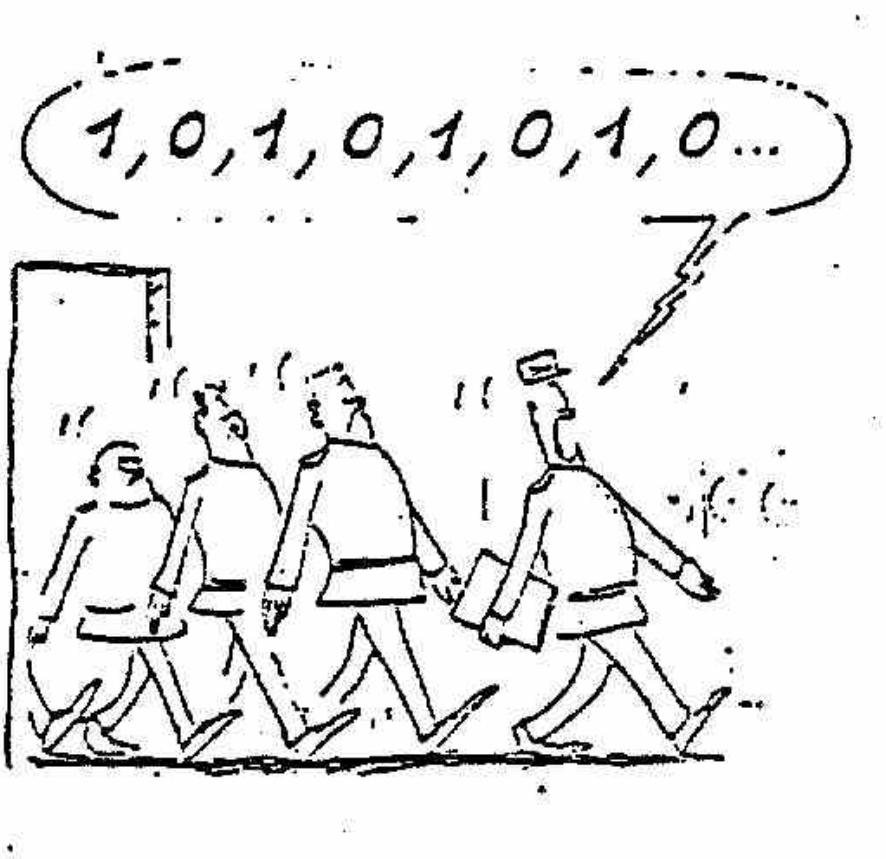
[Hil26] D. Hilbert. Sur l'infini. *Math. Annal.*, 95 :161–190, 1926.

[Kle36a] S. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112 :727–729, 1936.

[Kle36b] S. Kleene. Lambda-definability and recursiveness. *Duke Mathematical Journal*, 2 :340–353, 1936.

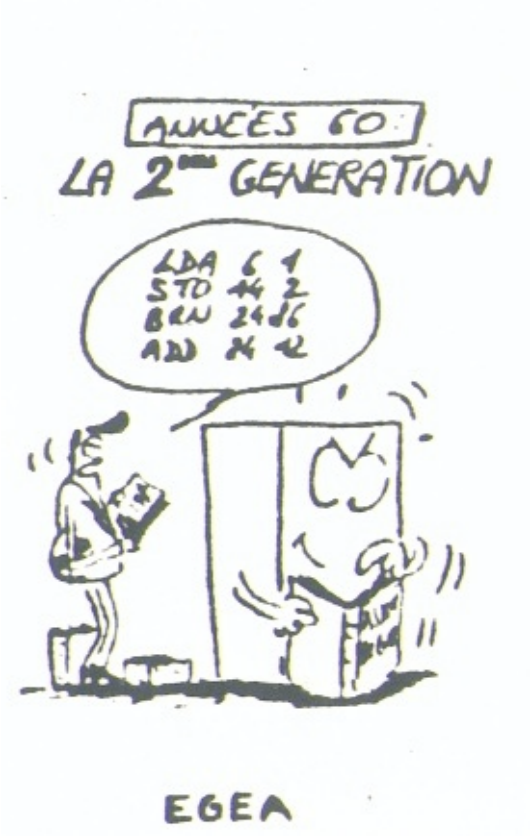
[Lei10] G. Leibniz. Brève description de la machine arithmétique (in latin). *Miscellanea Berolinensia*, pages 317–319, 1710.

[Oli14] E. Olivier. Qu'est-ce-qu'une machine ? (I/III). *Bull. Info. Appr. et Appl.*, 97 :27–38, 2014.



VOUZZAVEDIBISAR : 1,0,1,0,1,0,1,0,... (par Michel AVEZARD alias Zevar)

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
COMPUTATION - INFORMATION
Volume 99 – décembre 2014



Publication trimestrielle, gratuite, de l'Université d'Aix-Marseille
http://sites.univ-provence.fr/biaa

Impression : décembre 2014 ISSN 0291 - 5413

Couverture : dessin de Michel Avezard (Zevar) interprété par la MMIAGe (1985-1987)

(2014) B.I.A.A. no **99** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

ÉDITORIAL : Notre singularité

Serge HALIMI (Le Monde Diplomatique no 682, janvier 2011)

Résumé. – La presse écrite chancelle. La plupart des journaux sont confrontés à un tassement de leur lectorat. Ce qui fragilise leur équilibre économique, menace leur survie et remet en question le rôle qu'ils jouent dans la bataille des idées. Plusieurs facteurs contribuent à cette difficulté d'ensemble.

Le nombre de personnes connectées à la Toile pour y puiser des informations, souvent gratuites et abondantes, ne cesse d'augmenter. Les frais d'abonnement au réseau, auxquels s'ajoutent ceux du téléphone portable, absorbent l'essentiel du budget de communication des ménages. Du coup, certains ont cessé d'acheter des journaux, persuadés que l'information, dorénavant offerte, avait perdu toute valeur. Et que celle qu'il fallait encore payer était devenue hors de prix... S'il s'agit de suivre l'actualité heure par heure, d'agréger des informations éparses produites çà et là, d'accélérer la circulation des nouvelles, de réagir sur-le-champ et de briser le monopole des grands médias sur l'information et le commentaire, l'apport d'Internet est irrécusable. Il est moins évident pour qui veut comprendre un chaos d'événements, hiérarchiser les développements de l'actualité, prendre le temps de réfléchir au sens de l'histoire, à l'enchaînement des causes et des conséquences. Grâce à ses enquêtes, à ses reportages, aux regards croisés de ses centaines de collaborateurs, occasionnels et réguliers, français et étrangers, spécialistes reconnus ou francs-tireurs, Le Monde diplomatique peut, lui, aller au-delà du recopiage des priorités et des idées produites et reproduites partout ailleurs. En France, la propriété des grands médias est concentrée entre les mains de quelques groupes industriels et financiers, tous proches de l'Elysée, dont deux fabricants d'armes - Lagardère (via Hachette) et Dassault (via la Socpresse) - et un groupe de BTP, Bouygues. De son côté, l'homme le plus riche de France, M. Bernard Arnault, propriétaire de LVMH, a acheté Les Echos en 2007 avec l'appui déclaré du chef de l'Etat. Enfin, deux des principaux quotidiens *gratuits* de l'Hexagone appartiennent à M. Vincent Bolloré, lui aussi proche de M. Nicolas Sarkozy. Mais la crise de la presse ne se résume pas à la concurrence d'Internet et des *gratuits* ou à l'identité de ceux qui la possèdent. Elle est aussi celle d'un regard social étriqué, d'une offre uniformisée. Faire court, faire *people*, faire vite, faire parler les mêmes et faire comme les autres est devenu synonyme de faire du journalisme. S'abonner, au moment où nous livrons une guerre médiatique asymétrique face aux géants de la communication, constitue par conséquent un acte de résistance à l'uniformisation de l'information. C'est aussi une marque d'intérêt et de soutien en faveur d'une presse irrespectueuse, curieuse, indépendante. Différente, tout simplement.

(2014) B.I.A.A. no **99** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

Qu'est-ce qu'une machine ? (III/III)

Eric OLIVIER [1] [2]

Résumé. – La théorie des machines de Turing reformule et clarifie un certain nombre de questions portant sur les fondements (logiques) des mathématiques. Ainsi la question *"Qu'est-ce qu'une machine ?"* est-elle équivalente à la question *"Qu'est-ce qu'un calcul ?"*. Richard Feynman résume cela en affirmant que *n'importe quelle procédure de calcul à laquelle on pourrait penser, est équivalente au calcul d'une machine de Turing – les fonctions récursives générales sont Turing-calculables et vice-versa – et on peut donc prendre "Turing-calculable" pour un synonyme effectif de "calculable"*. Notons enfin que le calcul *automatique* (i.e. le calcul effectué par une machine de Turing) distingue la notion de *proposition démontrable* de celles de *proposition vraie, décidable, indécidable* : cela éclaire les travaux révolutionnaires de Gödel sur la *complétude* et la *consistance* des théories mathématiques.

1. SYSTÈMES FORMELS

Soit $T \in \mathfrak{T}(\mathcal{A})$ une machine de Turing où $\mathcal{A} \supset \{\star, \mathtt{0}, \mathtt{1}\}$ et $\star$ désigne le symbole de la case vide. L'ensemble $\mathcal{Q}$ des états internes de T sera toujours défini (par recodage si nécessaire) de sorte que $\mathcal{Q} \sqcup \{\mathtt{[}, \mathtt{)}\}$ soit disjoint du l'alphabet d'exécution $\mathcal{A}$. Dans [Oli14a] l'ensemble $\mathrm{St}(T)$ des *états courants* de T est défini comme l'ensemble des mots de la forme $w\,\mathtt{[Q)}\,m$, où w et m sont dans $\mathcal{A}^*$ et $\mathtt{Q} \in \mathcal{Q}$. Par abus de notation $T : \mathrm{St}(T) \to \mathrm{St}(T)$ est l'application telle que $T(w\,\mathtt{[Q)}\,m)$ soit l'état courant obtenu par application d'un cycle de T initialement dans l'état courant $w\,\mathtt{[Q)}\,m$; plus précisément :

$$
\begin{array}{rcll}
T(w\,\mathtt{[Q)}\,\mathtt{a}m') & = & w\mathtt{b}\,\mathtt{[P)}\,m' & \text{lorsque } T(\mathtt{Q},\mathtt{a}) = (\mathtt{P},\mathtt{b},\mathtt{R}) \\
T(w\,\mathtt{[Q)}\,\mathtt{a}m') & = & w\,\mathtt{[P)}\,\mathtt{b}m' & \text{lorsque } T(\mathtt{Q},\mathtt{a}) = (\mathtt{P},\mathtt{b},\mathtt{S}) \\
T(w'\mathtt{c}\,\mathtt{[Q)}\,\mathtt{a}m') & = & w'\,\mathtt{[P)}\,\mathtt{cb}m' & \text{lorsque } T(\mathtt{Q},\mathtt{a}) = (\mathtt{P},\mathtt{b},\mathtt{L}) \\
T(\,\mathtt{[Q)}\,\mathtt{a}m') & = & \,\mathtt{[P)}\star\mathtt{b}m' & \text{lorsque } T(\mathtt{Q},\mathtt{a}) = (\mathtt{P},\mathtt{b},\mathtt{L}) \\
T(w\,\mathtt{[Q)}\,) & = & \mathtt{wb}\,\mathtt{[P)} & \text{lorsque } T(\mathtt{Q},\star) = (\mathtt{P},\mathtt{b},\mathtt{R}) \\
T(w'\mathtt{c}\,\mathtt{[Q)}\,) & = & w'\,\mathtt{[P)}\,\mathtt{cb} & \text{lorsque } T(\mathtt{Q},\star) = (\mathtt{P},\mathtt{b},\mathtt{L})
\end{array}
\tag{1}
$$

avec $w = w'\mathtt{c}$ pour $\mathtt{c} \in \mathcal{A}$, lorsque $w \neq \phi$ et $m = \mathtt{a}m'$ lorsque $m \neq \phi$. L'ensemble des entrées de T est le sous-ensemble de $\mathrm{St}(T)$ formé des états courants de la forme $w\,\mathtt{[I)}\,m$. L'ensemble des sorties de T est noté $\mathrm{Out}(T)$ et $\mathrm{Stop}(T, w\,\mathtt{[Q)}\,m)$ ($\in \{0, 1, \ldots\} \cup \{+\infty\}$) est le temps d'arrêt de l'état courant $w\,\mathtt{[Q)}\,m$. L'application $T^* : \mathrm{St}(T) \to \mathrm{Out}(T) \cup \{\omega\}$ est aussi définie avec la convention $T^*(m\,\mathtt{[Q)}\,m) = \omega$ lorsque $\mathrm{Stop}(T, w\,\mathtt{[Q)}\,m) = +\infty$. Soient maintenant $\mathcal{A}$ et $\mathcal{X}$ deux alphabets finis disjoints et n, N deux entiers arbitrairement donnés ; une *Post-production* P (d'ordre [3] 1) est une *règle d'inférence* du type

$$u_0 X_1 u_1 \ldots u_{n-1} X_n u_n \xrightarrow{R} v_0 X_{\varphi(1)} v_1 \ldots v_{N-1} X_{\varphi(N)} v_N$$

1. GDAC-I2M UMR 7373 CNRS Université d'Aix-Marseille
2. eric.olivier@univ-amu.fr
3. Il existe une notion de Post-production d'ordre $r \geq 1$.

où les u_i et les v_j (resp. les X_k) sont dans $\mathcal{A}^*$ (resp. $\mathcal{X}$) et où $\varphi : \{1, \dots, N\} \to \{1, \dots, n\}$ est une application quelconque. Les X_k sont des *variables de production* représentant des mots de $\mathcal{A}^*$: pour $w, m \in \mathcal{A}^*$, on écrit $R : w \to m$ (ou que m *dérive* de w par R), lorsque

$$w = u_0 x_1 u_1 \dots u_{n-1} x_n u_n, \quad \text{et} \quad m = v_0 x_{\varphi(1)} v_1 \dots v_{N-1} x_{\varphi(N)} v_N.$$

(ici les x_k sont dans $\mathcal{A}^*$). Le couple $S = (\mathcal{O}, (P_1, \dots, P_N))$ est un *système formel* sur $\mathcal{A}$ lorsque $\mathcal{O}$ (l'ensemble des *S-axiomes*) est une partie finie ou infinie et récursive [4] de $\mathcal{A}^*$ et que $P_1, \dots, P_N$ sont N Post-productions sur $\mathcal{A}^*$. Le système S détermine un langage noté $\mathrm{Form}(S)$ dont les éléments sont appelés des *S-formules* ; $\mathrm{Form}(S)$ est défini comme le plus petit des langages $\mathcal{F}$ t.q.

$$\mathcal{O} \subset \mathcal{F} \subset \mathcal{A}^* \quad \text{et} \quad \forall w \in \mathcal{F}, \quad w \xrightarrow{P_i} w \implies w \in \mathcal{F}$$

ou encore que $\mathrm{Form}(S)$ est l'ensemble des mots de $m \in \mathcal{A}^*$ qui *dérivent d'un axiome* w, en ce sens qu'il existe une suite finie de mots $w_1, \dots, w_n \in \mathcal{A}^*$ avec $w_n = m$ et une suite du Post-production $P_{i_1}, \dots, P_{i_n}$ du système S t.q.

$$\mathcal{O} \ni w \xrightarrow{P_{i_1}} w_1 \ \cdots \ w_{n-1} \xrightarrow{P_{i_n}} w_n = m \tag{2}$$

Une séquence de Post-production telle que (2) signifie que m dérive de l'axiome w : dans la suite la notation $w \xrightarrow{*} m$ permet d'éviter de faire la liste des Post-productions mises en jeu. Pour un système formel S donné, le langage $\mathrm{Form}(S)$ est récursivement énumérable : la réciproque (moins facile) est aussi vraie.

Théorème 1.1. *[admis] Un langage $\mathcal{L}$ est récursivement énumérable si et seulement si il existe un système formel S tel que $\mathcal{L} = \mathrm{Form}(S)$.*

Une Post production (sur $\mathcal{A}$) est dite *normale* lorsqu'elle de la forme $uX \longrightarrow Xv$ où $u, v \in \mathcal{A}^*$ et X est une variable de production. Un système formel est dit normal si toutes les Post-productions sont normales : on montre (voir [Bia79]) que tout système formel S possède une forme normale S' de sorte que $\mathrm{Form}(S') = \mathrm{Form}(S)$.

Une *Thue-production* P sur un alphabet $\mathcal{A}$ est un cas particulier de Post-production dont l'ensemble des variables de production se réduit à $\mathcal{X} = \{X, Y\}$ et prenant la forme

$$XwY \xrightarrow{P} XmY$$

(avec w et m deux mots donnés dans $\mathcal{A}^*$). Pour simplifier la notation, on pourra aussi définir la Thue-production P sous forme de *substitutions* en écrivant

$$w \xrightarrow{P} m$$

Les systèmes formels munis de Thue-productions (on parlera de *systèmes formels de Thue*) pourraient sembler engendrer une classe plus restreinte de langages : il n'en est rien.

Théorème 1.2. *[admis] Un langage $\mathcal{L}$ est récursivement énumérable si et seulement si il existe un système formel de Thue S tel que $\mathcal{L} = \mathrm{Form}(S)$.*

4. Dans le cas où $\mathcal{O}$ est infini et récursif, on peut étendre l'ensembles des règles d'inférence pour se ramener au cas où $\mathcal{O}$ est fini ; il est cependant pratique de conserver dans la définition la possibilité que $\mathcal{O}$ soit infini et récursif.

1.1. **Le système MIU.** Dans le système MIU introduit par Post (et repris par Douglass Hofstadter dans [Hof79]), on prend $\mathcal{A} := \{\mathtt{M}, \mathtt{I}, \mathtt{U}\}$ et $S := (\mathcal{O}, (P_1, P_2, P_3, P_4))$, avec $\mathcal{O} = \{\mathtt{MI}\}$ et les Post-productions :

$$\begin{aligned} x\mathtt{I} &\xrightarrow{P_1} x\mathtt{IU} && \text{(inflation)} \\ \mathtt{M}x &\xrightarrow{P_2} \mathtt{M}xx && \text{(inflation)} \\ x\mathtt{III}y &\xrightarrow{P_3} x\mathtt{U}y && \text{(déflation)} \\ x\mathtt{UU}y &\xrightarrow{P_4} xy && \text{(déflation)} \end{aligned}$$

(où x et y sont des variables de production). Nous allons voir que $\mathtt{MU} \notin \mathrm{Form}(S)$. En effet, considérons le nombre de $\mathtt{I}$ se trouvant dans un mot de $\mathrm{Form}(S)$. Les productions P_1 et P_4 le laissent inchangé. La production P_3 diminue le nombre de $\mathtt{I}$ de 3 et ne change pas sa divisibilité par 3. La production P_2 double le nombre de $\mathtt{I}$; comme $2n$ ne peut être divisé par 3 que si n est divisible par 3, la production P_2 ne donnes pas de multiple de 3. Donc aucune des productions ne donne de multiple de 3. L'ensemble des axiomes étant réduit au singleton $\mathcal{O} = \{\mathtt{MI}\}$ et $\mathtt{MI}$ ne contenant qu'un nombre de $\mathtt{I}$ non multiple de 3, aucun mot de $\mathrm{Form}(S)$ ne peut contenir un nombre de $\mathtt{I}$ qui soit un multiple de 3 (et en particulier lorsque ce nombre est zéro). Ainsi, on a démontré que le mot $\mathtt{MU}$ n'est pas dans $\mathrm{Form}(S)$; cependant cette démonstration (utilisant la partition des entiers modulo 3) ne peut être déduite d'un nombre fini de S-dérivations du système MIU (essentiellement du fait que $\mathrm{Form}(S)$ est infini).

Exercice 1.3. *Le langage des formules du système MIU est-il récursif ?*

1.2. **La conjecture de Syracuse.** Notons $\mathbb{N}_* := \mathbb{N}\backslash\{0\}$ et $s : \mathbb{N}_* \to \mathbb{N}_*$ la transformation telle que $s(n) = n/2$ si n est pair et $s(n) = 3n + 1$ si n est impair. La *conjecture de Syracuse* affirme que pour tout n non nul, il existe un rang N tel que $s^N(n) = 1$. En *inversant* s, il est possible de reformuler cette conjecture en terme de système de Post. Considérons l'alphabet $\mathcal{O} = \{\mathtt{I}\}$ et le système de Post $S = (\mathcal{O}, (P_1, P_2))$ où les Post-productions sont données comme suit :

$$\begin{aligned} x &\xrightarrow{P_1} xx && \text{(inflation)} \\ xx\mathtt{I}xx\mathtt{I}xx\mathtt{II} &\xrightarrow{P_2} xx\mathtt{I} && \text{(déflation)} \end{aligned}$$

(où x est une variable de production). La conjecture de Syracuse équivaut à l'affirmation que $\mathrm{Form}(S) = \{\mathtt{I}^{(n)} \ ; \ n \geq 1\}$ (ici on pose $\mathtt{I}^{(0)} := \phi$ avec l'induction $\mathtt{I}^{(n+1)} := \mathtt{I}^{(n)}\mathtt{I}$ pour tout entier $n \geq 0$). La difficulté de cette conjecture montre à quel point il peut être difficile de cerner $\mathrm{Form}(S)$ pour un système formel S donné.

2. Grammaires formelles et hiérarchie de Chomsky

Dans les années 50, le linguiste Noam Chomsky dégage la notion de *grammaire formelle* et introduit une classification des langages maintenant appelée *hiérarchie de Chomsky* [Cho56] (voir aussi [GL67]). Soient $\mathcal{A}$ et $\mathcal{N}$ deux alphabets finis disjoints et $\mathtt{S}$ une élément spécial de $\mathcal{N}$. Une *grammaire formelle* est un système formel

$$G = (\mathcal{O} = \{\mathtt{S}\}, (P_1, \cdots, P_N))$$

sur $(\mathcal{A} \sqcup \mathcal{N})^*$ où les Post-productions sont des Thue-productions d'un type particulier, appelées *règles de production* : ainsi P_i est de la forme (substitutive)

$$u_i \mathtt{X}_i v_i \xrightarrow{P_i} w_i.$$

où $u_i, v_i w_i$ sont des mots dans $(\mathcal{A} \cup \mathcal{N})^*$ et où $\mathtt{X}_i \in \mathcal{N}$. Les deux alphabets $\mathcal{A}$ et $\mathcal{N}$ ne jouent pas un rôle symétrique : ils sont appelés respectivement l'*alphabet terminal* et l'*alphabet auxiliaire* (ou *non-terminal*) de la grammaire. Le langage engendré par G (en tant que grammaire) est par définition

$$\mathrm{Lang}(G) = \mathrm{Form}(G) \cap \mathcal{A}^*.$$

Le symbole spécial $\mathtt{S}$ qui est le seul axiome du système formel G est appelé le *point d'entrée de la grammaire formelle associée à* G. Considérons, par exemple, la grammaire $G = (\{\mathtt{S}\}, (P_1, P_2))$, avec $\mathcal{A} = \{\mathtt{a}, \mathtt{b}\}$, $\mathcal{N} = \{\mathtt{S}\}$ et les deux règles de production

$$\begin{aligned} \mathtt{S} &\xrightarrow{P_1} \mathtt{aSb} \\ \mathtt{S} &\xrightarrow{P_2} \phi \end{aligned}$$

(où ϕ représente le mot vide) ; alors $\mathrm{Lang}(G)$ est formé des mots de la forme $\mathtt{a}^{(n)}\mathtt{b}^{(n)}$ (avec $n \geq 0$). Soit maintenant $G = (\{\mathtt{S}\}, (P_1, P_2, P_3, P_4))$, avec $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $\mathcal{N} = \{\mathtt{S}, \mathtt{B}\}$ et les règles de production suivantes :

$$\begin{aligned} \mathtt{S} &\xrightarrow{P_1} \mathtt{aBSc} \\ \mathtt{S} &\xrightarrow{P_2} \mathtt{abc} \\ \mathtt{Ba} &\xrightarrow{P_3} \mathtt{aB} \\ \mathtt{Bb} &\xrightarrow{P_4} \mathtt{bb} \end{aligned}$$

On peut montrer que cette grammaire définit le langage

$$\mathrm{Lang}(G) = \left\{ \mathtt{a}^{(n)}\mathtt{b}^{(n)}\mathtt{c}^{(n)} \ ; \ n \geq 1 \right\}.$$

Théorème 2.1. *[admis] Un langage $\mathcal{L}$ est récursivement énumérable si et seulement si il existe une grammaire formelle G telle que $\mathcal{L} = \mathrm{Lang}(G)$.*

La *hiérarchie de Chomsky* définit quatre types de langages récursivement énumérables classés suivant certaines caractéristiques des grammaires formelles qui les engendrent. Soit $G = (\{\mathtt{S}\}, (P_1, \cdots, P_N))$ une grammaire formelle d'alphabet terminal $\mathcal{A}$ et d'alphabet auxiliaire $\mathcal{N}$. Le langage $\mathrm{Lang}(G)$ est dit de *type 0* lorsqu'il est récursivement énumérable : d'après le Théorème 2.1 le type 0 recouvre tous les langages possiblement engendrés par grammaire formelle. $\mathrm{Lang}(G)$ est de *type 1*, ou encore *contextuel* (*context sensitive* en anglais), lorsque les règles de production de G sont de la forme

$$u\mathtt{A}v \to uwv$$

pour $\mathtt{A} \in \mathcal{N}$ et $u, v, w \in (\mathcal{A} \cup \mathcal{N})^*$ avec $w \neq \phi$.

Proposition 2.2. *Les langages contextuels sont récursifs.*

Le langage $\mathrm{Lang}(G)$ est de *type 2*, ou *langages algébriques*, ou encore *hors-contexte* (*context free* en anglais), lorsque les règles de production de G sont de la forme

$$\mathtt{A} \to u$$

pour $\mathtt{A} \in \mathcal{N}$ et $u \in (\mathcal{A} \cup \mathcal{N})^*$ (de tels langages peuvent être considérés comme des langages contextuels à contexte vide). Le langage $\mathrm{Lang}(G)$ est de *type 3*, ou *rationnels* (ou *régulier*) lorsqu'il est engendré, soit par une *grammaire linéaire à gauche*, et dans ce cas les règles de production de G sont de la forme (pour $\mathtt{A}, \mathtt{B} \in \mathcal{N}$ et $\mathtt{a} \in \mathcal{A}$)

$$\begin{aligned} \mathtt{A} &\to \mathtt{Ba} \\ \mathtt{A} &\to \mathtt{a} \end{aligned}$$

($\mathtt{A}, \mathtt{B} \in \mathcal{N}$ et $\mathtt{a} \in \mathcal{A}$), soit par une *grammaire linéaire à droite*, et dans ce cas les règles de production de G sont de la forme

$$\begin{aligned} \mathtt{A} &\to \mathtt{aB} \\ \mathtt{A} &\to \mathtt{a} \end{aligned}$$

Théorème 2.3 (Kleene)**.** *Les langages rationnels sont les langages reconnus par automates finis.*

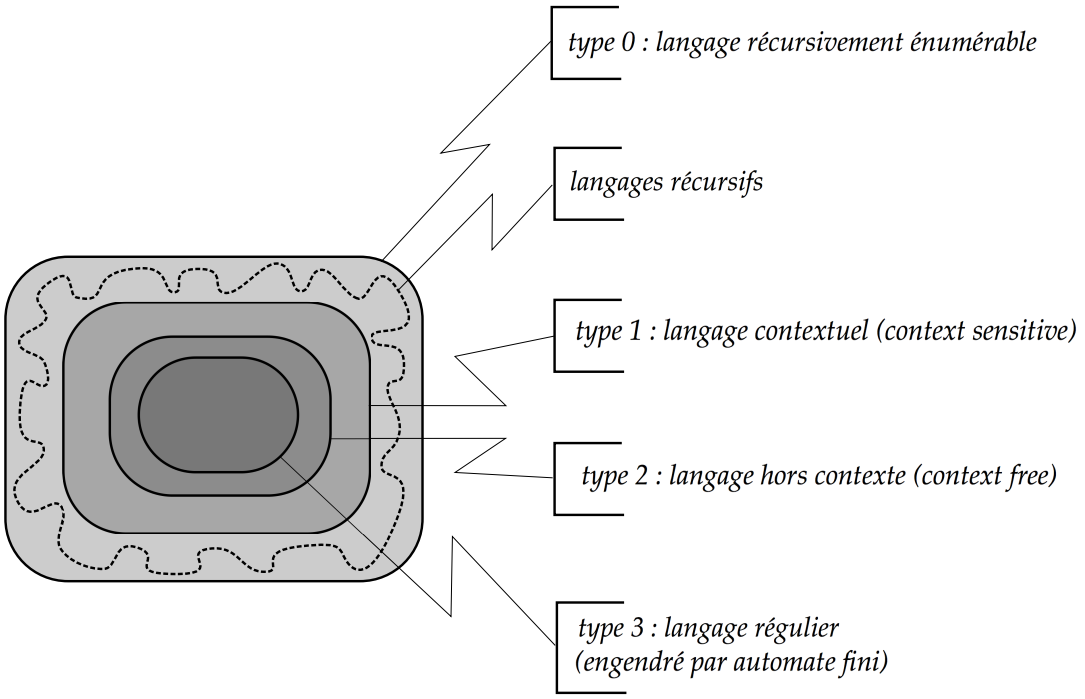


FIGURE 5. Hiérarchie des langages de Chomsky.

3. HEURISTIQUE DES THÉORIES FORMELLES BOOLÉENNES

Depuis Saussure [Sau79, Sau02] – et l'introduction des concepts de *signifiant* et de *signifié* – l'étude des langages amène à distinguer la *syntaxe* de la *sémantique*. La syntaxe est l'objet de la théorie des *grammaires formelles* et des diverses formes de *règles d'écriture*. L'aspect sémantique est une question plus difficile et plus profonde, qui touche aussi bien la *linguistique* que la *logique mathématique*. En logique, la *théorie des modèles* de Tarski [Tar44], introduit une forme de sémantique basée sur le concept de *vérité mathématique*. Il n'est pas envisageable ici de donner une approche systématique de ces questions, mais plutôt d'en

fournir des éléments *heuristiques*. Pour simplifier, commençons par définir une classe de langages possédant une *symétrie* dont l'origine est la négation logique (contraposition) des propositions (et donc la dualité *vrai/faux*). Dans la suite, le langage $\mathcal{L} \subset \mathcal{A}^*$ est appelé *booléen* lorsqu'il existe un symbole spécial dans $\mathcal{A}$ – notons le $\$$ – pour lequel $\$\mathcal{L} = \mathcal{L}$ ($\$\mathcal{L}$ est l'ensembles des mots $\$w$ obtenus comme la concatenation du symbole $\$$ avec un mot $w \in \mathcal{L}$) : nous dirons dans ce cas que $\mathcal{L}$ est *$-booléen*.

Définition 3.1 (Heuristique)**.** *Soit $\mathcal{A}$ ($\ni \$$) un alphabet ; alors le couple $\mathcal{T} = (\mathrm{Ass}(\mathcal{T}), \mathrm{Th}(\mathcal{T}))$ formées de deux sous-langages (non vides) de $\mathcal{A}^*$ est une* $\$$-théorie formelle *si les deux axiomes suivants sont satisfaits, soient : (i) :* $\mathrm{Ass}(\mathcal{T})$ *(les* $\mathcal{T}$-assertions*) est un langage $\$$-booléen récursif ; (ii) :* $\mathrm{Th}(\mathcal{T})$ *(les* $\mathcal{T}$-théorèmes*) est un langages récursivement énumérable t.q.*

$$\mathrm{Th}(\mathcal{T}) \subset \mathrm{Ass}(\mathcal{T}) \text{ et } \$\$\mathrm{Th}(\mathcal{T}) = \mathrm{Th}(\mathcal{T}).$$

De plus, (iii) : $\mathcal{T}$ est cohérente *(où* consistante *ou encore* non-contradictoire *), lorsque*

$$\mathrm{Th}(\mathcal{T}) \cap \$\mathrm{Th}(\mathcal{T}) = \emptyset$$

(iii) : $\mathcal{T}$ est complète *lorsque*

$$\mathrm{Th}(\mathcal{T}) \cup \$\mathrm{Th}(\mathcal{T}) = \mathrm{Ass}(\mathcal{T}).$$

Les assertions qui sont dans $\mathrm{Th}(\mathcal{T})$ (resp. $\$\mathrm{Th}(\mathcal{T})$) sont dites *$\mathcal{T}$-démontrables* (resp. *$\mathcal{T}$-réfutables*) ; les assertions dans $\mathrm{Ass}(\mathcal{T})\backslash\big(\mathrm{Th}(\mathcal{T}) \cup \$\mathrm{Th}(\mathcal{T})\big)$ sont dites *$\mathcal{T}$-indécidables* : ainsi l'existence d'assertions $\mathcal{T}$-indécidables équivaut à l'incomplétude de $\mathcal{T}$.

3.1. **Un premier exemple.** Nous allons construire une $\$$-théorie formelle $\mathcal{T}$ décrivant la divisibilité des entiers naturels par 2 et 3. Pour cela commençons par définir un système formel permettant de coder les assertions de la théorie, soit $S = (\mathcal{O}, (P_1, P_2))$, avec $\mathcal{O} = \{\mathtt{2D}, \mathtt{3D}\}$ et les Post-productions

$$\begin{aligned} x\mathtt{D}y &\xrightarrow{P_1} x\mathtt{DI}y \\ x\mathtt{D}y &\xrightarrow{P_2} \$x\mathtt{D}y \end{aligned}$$

Il est évident que

$$\mathrm{Form}(S) = \Big\{ \$^{(p)}\mathtt{2DI}^{(q)}, \$^{(p)}\mathtt{3DI}^{(q)} \ ; \ p, q \geq 0 \Big\} =: \mathrm{Ass}(\mathcal{T})$$

est bien un langage $\$$-booléen. La deuxième étape consiste à définir le système formel Σ qui va donner les $\mathcal{T}$-théorèmes : ici, $\mathrm{Th}(\mathcal{T}) := \mathrm{Form}(\Sigma)$ avec

$$\Sigma = (\mathcal{O}, (P_+, P_-, R_1, R_2, R_3, R_4, R_5))$$

et les Post-productions

$$\begin{aligned} x\mathtt{D}y &\xrightarrow{P_+} \mathtt{\$\$}x\mathtt{D}y \\ \mathtt{\$\$}x\mathtt{D}y &\xrightarrow{P_-} x\mathtt{D}y \\ x\mathtt{2D}y &\xrightarrow{R_1} x\mathtt{2DII}y \\ x\mathtt{3D}y &\xrightarrow{R_2} x\mathtt{3DIII}y \\ \mathtt{2D}y &\xrightarrow{R_3} \mathtt{\$2DI}y \\ \mathtt{3D}y &\xrightarrow{R_4} \mathtt{\$3DI}y \\ \mathtt{3D}y &\xrightarrow{R_5} \mathtt{\$3DII}y \end{aligned}$$

Proposition 3.2. $\mathcal{T} = (\mathrm{Ass}(\mathcal{T}), \mathrm{Th}(\mathcal{T}))$ *est cohérente et complète.*

3.2. **Addition des entiers positifs.** Le système formel S codant les assertions de $\mathcal{T}$ est $S = (\mathcal{O}, (P_1, \ldots, P_4))$, avec $\mathcal{O} = \{\mathtt{+=}\}$ et les Post-productions

$$\begin{aligned} x\mathtt{=}y &\xrightarrow{P_1} x\mathtt{=I}y \\ x\mathtt{=}y &\xrightarrow{P_2} x\mathtt{I=}y \\ x\mathtt{+}y &\xrightarrow{P_3} x\mathtt{I+}y \\ x\mathtt{=}y &\xrightarrow{P_4} \mathtt{\$}x\mathtt{=}y \end{aligned}$$

de sorte que

$$\mathrm{Form}(S) = \left\{ \mathtt{\$}^{(p)}\mathtt{I}^{(a)}\mathtt{+I}^{(b)}\mathtt{=I}^{(c)} \ ; \ (p,a,b,c,d) \in \mathbb{N}^4 \right\} =: \mathrm{Ass}(\mathcal{T})$$

est un langage $\mathtt{\$}$-booléen. Les $\mathcal{T}$-théorèmes de $\mathcal{T}$ sont définis en posant $\mathrm{Th}(\mathcal{T}) := \mathrm{Form}(\Sigma)$ avec le système formel $\Sigma = (\mathcal{O}, (P_+, R_1, \ldots, R_6))$ dont les Post-productions sont :

$$\begin{aligned} u &\xrightarrow{P_+} \mathtt{\$\$}u \\ u\mathtt{+}v\mathtt{=}w &\xrightarrow{R_1} u\mathtt{I+}v\mathtt{=}w\mathtt{I} \\ u\mathtt{II+}v\mathtt{=}w\mathtt{II} &\xrightarrow{R_2} u\mathtt{I+}v\mathtt{=}w\mathtt{I} \\ u\mathtt{+}v\mathtt{=}w\mathtt{II} &\xrightarrow{R_3} u\mathtt{I+}v\mathtt{=}w\mathtt{I} \\ u\mathtt{II+}v\mathtt{=}w &\xrightarrow{R_4} u\mathtt{I+}v\mathtt{I=}w \\ u\mathtt{+}v\mathtt{II=}w &\xrightarrow{R_5} u\mathtt{I+}v\mathtt{I=}w \\ u\mathtt{+}v\mathtt{=}w &\xrightarrow{R_6} \mathtt{\$I}u\mathtt{+}v\mathtt{=}w \end{aligned}$$

Ainsi, pour tout entier $a, b, c \geq 1$ et $n \geq 0$ on a $\mathtt{\$}^{(2n)}\mathtt{I}^{(a)}\mathtt{+I}^{(b)}\mathtt{=I}^{(c)} \in \mathrm{Form}(\Sigma)$ si et seulement si $a + b = c$ et $\mathtt{\$\$}^{(2n)}\mathtt{I}^{(a)}\mathtt{+I}^{(b)}\mathtt{=I}^{(c)} \in \mathrm{Form}(\Sigma)$ si et seulement si $a + b \neq c$. Par suite, $\mathrm{Form}(\Sigma) \cap \mathtt{\$}\mathrm{Form}(\Sigma) = \emptyset$ et $\mathrm{Form}(\Sigma) \cup \mathtt{\$}\mathrm{Form}(\Sigma) = \mathrm{Ass}(\mathcal{T})$.

Proposition 3.3. $\mathcal{T} = (\mathrm{Ass}(\mathcal{T}), \mathrm{Th}(\mathcal{T}))$ *est cohérente et complète.*

3.3. **PGCD.** Soit $a \wedge b$ le PGCD (Plus Grand Commun Diviseur) des deux entiers non nuls a et b. Nous allons définir une théorie formelle $\mathcal{T} = (\mathrm{Ass}(\mathcal{T}), \mathrm{Th}(\mathcal{T}))$ permettant de décider la validité des équations du type $a \wedge b = c$. Le système formel codant les assertions

de $\mathcal{T}$ est $S = (\mathcal{O}, (P_1, P_2, P_3, P_4))$, avec $\mathcal{O} = \{\mathtt{I}\wedge\mathtt{I}=\mathtt{I}\}$ et les Post-productions

$$\begin{aligned} x{=}y &\xrightarrow{P_1} x{=}\mathtt{I}y \\ x{=}y &\xrightarrow{P_2} x\mathtt{I}{=}y \\ x{\wedge}y &\xrightarrow{P_3} x\mathtt{I}{\wedge}y \\ x &\xrightarrow{P_4} \$x \end{aligned}$$

de sorte que

$$\mathrm{Form}(S) = \left\{ \$^{(p)}\mathtt{I}^{(a)}{\wedge}\mathtt{I}^{(b)}{=}\mathtt{I}^{(c)} \ ; \ p \geq 0, \ a, b, c \geq 1 \right\} =: \mathrm{Ass}(\mathcal{T})$$

est bien un langage $\$$-booléen. Le système formel Σ tel que $\mathrm{Form}(\Sigma) = \mathrm{Th}(\mathcal{T})$ est $\Sigma = (\mathcal{O}; (R_1, R_2, R_3, R_4, R_5, R_+))$ avec les Post-productions suivantes

$$\begin{aligned} u{\wedge}u{=}u &\xrightarrow{R_1} \mathtt{I}u{\wedge}\mathtt{I}u{=}\mathtt{I}u \\ \mathtt{I}u{\wedge}v{=}w &\xrightarrow{R_2} v{\wedge}\mathtt{I}u{=}w \\ u{\wedge}v{=}w &\xrightarrow{R_3} u{\wedge}vu{=}w \\ u{\wedge}v{=}w &\xrightarrow{R_4} \$u{\wedge}v{=}w\mathtt{I} \\ u{\wedge}v{=}w\mathtt{II} &\xrightarrow{R_5} \$u{\wedge}v{=}w\mathtt{I} \\ u &\xrightarrow{P_+} \$\$u \end{aligned}$$

Par exemple, l'équation $6 \wedge 8 = 2$ s'obtient par la dérivation suivante :

$$\begin{aligned} \mathtt{I}{\wedge}\mathtt{I}{=}\mathtt{I} &\xrightarrow{R_1} \mathtt{II}{\wedge}\mathtt{II}{=}\mathtt{II} \\ &\xrightarrow{R_3} \mathtt{II}{\wedge}\mathtt{IIII}{=}\mathtt{II} \\ &\xrightarrow{R_3} \mathtt{II}{\wedge}\mathtt{IIIIII}{=}\mathtt{II} \\ &\xrightarrow{R_2} \mathtt{IIIIII}{\wedge}\mathtt{II}{=}\mathtt{II} \xrightarrow{R_3} \mathtt{IIIIII}{\wedge}\mathtt{IIIIIIII}{=}\mathtt{II} \end{aligned}$$

Plus généralement (exercice), pour tout entiers $a, b, c \geq 1$ et $n \geq 0$,

$$\begin{aligned} \$^{(2n)}\mathtt{I}^{(a)}{\wedge}\mathtt{I}^{(b)}{=}\mathtt{I}^{(c)} \in \mathrm{Form}(\Sigma) &\iff a \wedge b = c \\ \text{et} \quad \$\$^{(2n)}\mathtt{I}^{(a)}{\wedge}\mathtt{I}^{(b)}{=}\mathtt{I}^{(c)} \in \mathrm{Form}(\Sigma) &\iff a \wedge b \neq c \end{aligned}$$

Proposition 3.4. $\mathcal{T} = (\mathrm{Ass}(\mathcal{T}), \mathrm{Th}(\mathcal{T}))$ *est cohérente et complète.*

4. Problème de l'arrêt et incomplétude

Le langage $\mathrm{Ass}(\mathcal{T})$ des assertions d'une $\$$-théorie formelle $\mathcal{T}$ est toujours supposé récursif : si non, on ne pourrait pas savoir – dans tous les cas et en temps fini – si un mot donné est une $\mathcal{T}$-assertion, ce qui est absurde [5]. La même objection ne s'applique pas au langage $\mathrm{Th}(\mathcal{T})$ des théorèmes, qui peut donc être strictement récursivement énumérable. Supposons alors, par exemple, que $\mathcal{P}$ soit un langage strictement récursivement énumérable $\mathcal{P}$ tel que $\$\$\mathcal{P} = \mathcal{P}$ et $\$\mathcal{P} \cap \mathcal{P} = \emptyset$ (exercice : exhiber un tel lalngage). Alors pour tout langage récursif $\mathcal{Q}$ qui est $\$$-booléen et contenant $\mathcal{P}$, la $\$$-théorie formelle $\mathcal{T} = (\mathcal{Q}, \mathcal{P})$ est cohérente et nécessairement incomplète.

5. *"Le moins qu'on puisse demander à une statue, c'est qu'elle ne bouge pas !"*(Salvador Dalí)

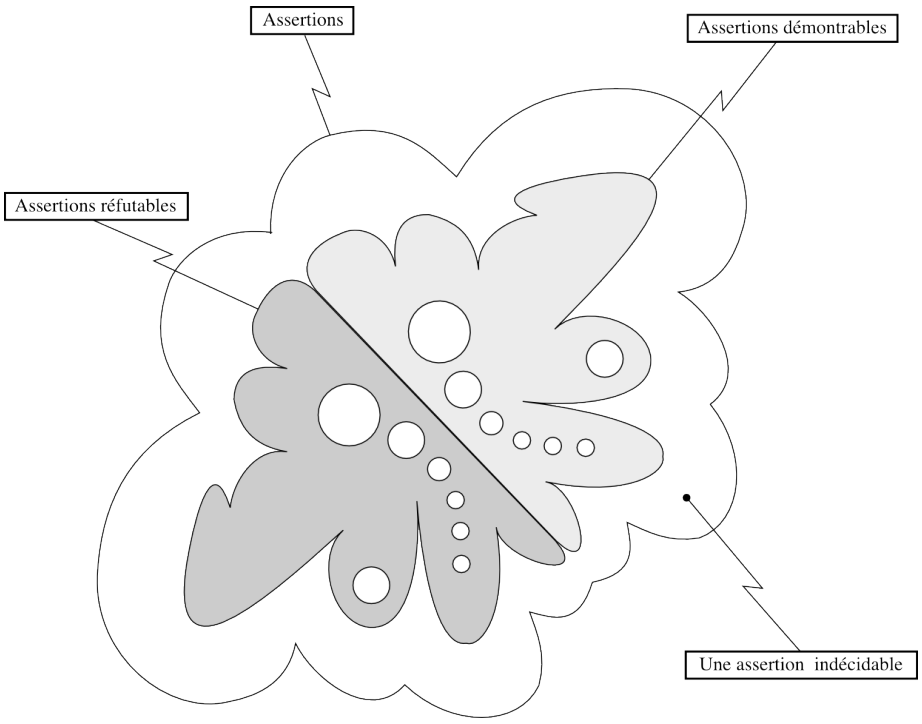


FIGURE 6. Une théorie mathématique cohérente mais incomplète.

Le deuxième problème posé par Hilbert à la *Conférence internationales des mathématiciens de Paris* en 1900 [Hil02] concerne la *consistance de l'arithmétique*. Une réponse négative à ce problème est donnée par le premier théorème d'incomplétude de Gödel [Göd31]. Dans une note de 1963 ajoutée à son article de 1931, Gödel écrit "*On peut démontrer rigoureusement que dans tout système formel consistant contenant une théorie des nombres finitaire relativement développée, il existe des propositions arithmétiques indécidables et que, de plus, la consistance d'un tel système ne saurait être démontrée à l'intérieur de ce système.*" (Voir [GNNG89] pour une présentation générale des travaux de Gödel sur l'incomplétude.)

L'idée d'une formalisation des théories mathématiques qui agite le début du 20-ème siècle est parallèle à une réflexion sur la mécanisation du calcul (voir la question de la thèse de Church dans [Oli14b]). En 1928 Hilbert et Ackermann [HA28] posent la question de la *décidabilité algorithmique du calcul des prédicats du premier ordre* aussi connue comme le *problème de la décision (Entscheidungsproblem)*. Pour une théorie formelle cohérente et complète, il existe (par définition) une machine Turing qui démontre ou réfute toute $\mathcal{T}$-assertion donnée [6]. La réponse négative au problème de la décision, proposée simultanément par Church et Turing [Chu36][Tur36] est basées sur le type d'incomplétude étudié par Gödel dans son article de 1931. Nous illustrons ce point avec l'idée de Turing, en présentant le *problème de l'arrêt* sous la forme d'une `$`-théorie formelle cohérente et incomplète. Pour cela, soit U une machine de Turing universelle pouvant simuler toutes les machines de $\mathfrak{T}\{\star, 0, 1\}$ (voir [Oli14a]) ; rappelons que $\mathrm{Pr}(T, \texttt{[I)}\, w)$ est le programme

6. Il demeure cependant que le temps d'attente nécessaire pour obtenir la réponse à une question donnée est, en général, impossible à estimer : on saura un jour, mais on ne sait pas quand !

de simulation du calcul de T sur l'entrée $\texttt{[I)}\,w$ et que

$$\mathcal{E}_U := \Big\{ \mathrm{Pr}(T, \texttt{[I)}\,w)\ ;\ (T,w) \in \mathfrak{T}\{\star,\texttt{0},\texttt{1}\} \times \{\texttt{0},\texttt{1}\}^* \Big\}$$

est un langage récursif (par définition de U). Par suite, l'ensemble des $\mathcal{T}$-assertions, soit $\mathrm{Ass}(\mathcal{T}) := \bigcup_{n=0}^{\infty} \$^{(n)}\mathcal{E}_U$, est un langage $\$$-booléen qui est lui même récursif. Afin de définir les $\mathcal{T}$-théorèmes on commence par remarquer que l'ensemble

$$\mathcal{F}_U := \Big\{ \mathrm{Pr}(T, \texttt{[I)}\,w) \in \mathcal{E}_U\ ;\ \mathrm{Stop}(T, \texttt{[I)}\,w) < \infty \Big\}$$

est strictement récursivement énumérable [Oli14a, Théorème 7.2]. Il est alors assez naturel de poser $\mathrm{Th}(\mathcal{T}) := \bigcup_{n=0}^{\infty} \$^{(2n)}\mathcal{F}_U$, de sorte que (i) : $\$\$\mathrm{Th}(\mathcal{T}) = \mathrm{Th}(\mathcal{T})$; (ii) : $\mathrm{Th}(\mathcal{T}) \cap \$\mathrm{Th}(\mathcal{T}) = \emptyset$ et (iii) : $\mathrm{Th}(\mathcal{T}) \cup \$\mathrm{Th}(\mathcal{T}) \subset \mathrm{Ass}(\mathcal{T})$. Le langage $\mathcal{F}_U$ étant strictement récursivement énumérable, il en est de même de $\mathrm{Th}(\mathcal{T}) \cup \$\mathrm{Th}(\mathcal{T})$: l'inclusion $\mathrm{Th}(\mathcal{T}) \cup \$\mathrm{Th}(\mathcal{T}) \subset \mathrm{Ass}(\mathcal{T})$ est stricte car $\mathrm{Ass}(\mathcal{T})$ est une langage récursif.

Proposition 4.1. *La $\$$-théorie formelle $\mathcal{T} = (\mathrm{Ass}(\mathcal{T}), \mathrm{Th}(\mathcal{T}))$, modélisant le problème de l'arrêt, est cohérente et incomplète.*

5. Le calcul des propositions de Lukasiewicz

Le *calcul des propositions* (ou encore *logique classique*) date de l'antiquité ; il est classiquement présentée sous sa forme sémantique, où la validité (vrai/faux) des propositions est déterminée à l'aide des tables de vérité. Nous présentons ici la théorie formelle [7] $\mathcal{P} = (\mathbf{Ass}(\mathcal{P})), \mathbf{Th}(\mathcal{P})$ du calcul des propositions proposée par Lukasiewicz [8] [Luk29].

Définition 5.1. *Le langage des $\mathcal{P}$-assertions (usuellement appelées Expressions Bien Formées ou EBF [9]) est $\mathbf{Ass}(\mathcal{P}) = \mathrm{Lang}(G)$, pour la grammaire formelle $G = (\mathcal{O} = \{\texttt{S}\}, (P_1, \ldots, P_7))$ d'alphabet terminal (resp. auxiliaire) $\mathcal{A} = \{\texttt{C}, \texttt{N}, \texttt{[}, \texttt{i}, \texttt{]}\}$, (resp. $\mathcal{N} = \{\texttt{S}, \texttt{A}, \texttt{B}\}$) et dont les Thue-productions sont :*

$$\begin{aligned}
\texttt{S} &\xrightarrow{P_1} \texttt{[A]}\\
\texttt{A} &\xrightarrow{P_2} \texttt{iA}\\
\texttt{A} &\xrightarrow{P_3} \phi\\
\texttt{S} &\xrightarrow{P_4} \texttt{B}\\
\texttt{B} &\xrightarrow{P_5} \texttt{NB}\\
\texttt{B} &\xrightarrow{P_6} \texttt{CBB}\\
\texttt{B} &\xrightarrow{P_7} \texttt{S}
\end{aligned}$$

Un rôle spécial est joué par les mots $\texttt{[]}, \texttt{[i]}, \texttt{[ii]}, \ldots$: ils représentent la suite des variables propositionnelles. En pratique, $\texttt{[]} =: \texttt{[0]}$, $\texttt{[i]} =: \texttt{[1]}$, $\texttt{[ii]} =: \texttt{[2]}, \ldots$, le rôle crucial des variables propositionnelles $\texttt{[}p\texttt{]}$ ($p \geq 0$) étant justifié par la propriété de substitution. Supposons que $W_0\,\texttt{[}p\texttt{]}\,W_1$ soit une EBF ; cela signifie que

$$\texttt{S} \longrightarrow W_0\texttt{S}W_1 \xrightarrow{*} W_0\,\texttt{[}p\texttt{]}\,W_1$$

7. Ici la notion de théorie formelle est plus générale que celle des § 3 et § 4 : nous utilisons une fonte grasse pour $\mathbf{Ass}(\cdot)$ et $\mathbf{Th}(\cdot)$ afin de marquer la différence avec $\mathrm{Ass}(\cdot)$ et $\mathrm{Th}(\cdot)$.

8. C'est dans ce formalisme que Lukasiewicz introduit la *notation polonaise*.

9. En anglais on dit WFF (prononcer wouf !) pour Well Formed Formulae.

Si X est une autre EBF, alors $\mathtt{S} \stackrel{*}{\longrightarrow} X$ et W_0XW_1 est aussi une EBF, puisque par composition des productions

$$\mathtt{S} \longrightarrow W_0\mathtt{S}W_1 \stackrel{*}{\longrightarrow} W_0XW_1$$

La forme d'une représentation syntaxique du calcul des propositions est déterminée par la *sémantique sous-jascente*. Dans la représentation de Lukasievicz, les variables propositionnelles `[]`, `[i]`, `[ii]`,... sont susceptibles de prendre la valeur 0 (le faux) ou 1 (le vrai). Ainsi, `N[`p`]` représente la négation de `[`p`]` et `C[`p`][`q`]` représente l'implication `[`p`]` implique `[`q`]`. Nous nous contenterons ici d'une présentation heuristique du calcul des propositions de Lukasiewicz. Pour commencer rappelons les deux *tables de vérité* de la négation et de l'implication, soient :

`[`p`]`	`N[`p`]`
0	1
1	0

`[`p`]`	`[`q`]`	`C[`p`][`q`]`
0	0	1
0	1	1
1	0	0
1	1	1

On dira d'une EBF E qu'elle est n-aire si elle s'exprime en *fonction* de n variables, disons `[`p_1`]`, ..., `[`p_n`]` (deux à deux distinctes) et de ces n variables seulement. Ainsi,

$$E = W_0\,\mathtt{[}p_{\varphi(1)}\mathtt{]}\,W_1\,\mathtt{[}p_{\varphi(2)}\mathtt{]}\cdots W_{m-1}\,\mathtt{[}p_{\varphi(m)}\mathtt{]}\,W_m$$

où les W_i sont des mots dans $\{\mathtt{C},\mathtt{N}\}^*$ et où $\varphi : \{1,\ldots,m\} \to \{1,\ldots,n\}$ est une application surjective. On écrit abusivement que $E = E(x_1,\cdots,x_n)$, ce qui revient à identifier E avec l'application $E : (x_1,\cdots,x_n) \mapsto E(x_1,\cdots,x_n)$, définie sur $\{0,1\}^n$ et à valeur dans $\{0,1\}$ et qu'on appelle la table de vérité de E. Les valeurs de la table de vérité d'une EBF s'obtiennent par un calcul récursif utilisant les tables de vérité de l'implication et de la négation. Par exemple `C[`p`]CN[`p`][`q`]` est une EBF binaire et on peut calculer $E(0,1)$ comme suit [10] :

$$\begin{aligned} E(0,1) &= \mathtt{C0CN01} \\ &= \mathtt{C0C(N0)1} \\ &= \mathtt{C0C11} \\ &= \mathtt{C0(C11)} \\ &= \mathtt{C01} \\ &= 1 \end{aligned}$$

de même pour l'EBF ternaire $E = \mathtt{CC[}p\mathtt{][}q\mathtt{]CC[}q\mathtt{][}r\mathtt{]C[}p\mathtt{][}r\mathtt{]}$, on a :

$$\begin{aligned} E(0,0,1) &= \mathtt{CC00CC01C01} \\ &= \mathtt{C(C00)C(C01)(C01)} \\ &= \mathtt{C1C11} \\ &= \mathtt{C1(C11)} \\ &= \mathtt{C11} \\ &= 1 \end{aligned}$$

10. Les parenthèses ne sont introduites que pour faciliter la lecture du calcul : en principe, elles ne devraient pas figurer

Une EBF n-aire $E = E(x_1, \cdots, x_n)$ est appelée une *tautologie* (resp. *antilogie*) si

$$E(x_1, \ldots, x_n) = 1 \quad \text{(resp. } E(x_1, \ldots, x_n) = 0\text{)}$$

pour tout $(x_1, \ldots, x_n) \in \{0,1\}^n$. Si $E(x_1, \ldots, x_n) = 1$ pour au moins une valeur de $(x_1, \ldots, x_n)$, on dit que E est *satisfaisable* ; E est dite *contingente* si elle n'est ni une tautologie ni une antilogies. Dans la suite $\mathrm{Taut}(\mathcal{P})$ désigne l'ensemble des tautologies.

Définition 5.2. *(i) : L'ensemble des axiomes du système de Lukasiewicz est*

$$\mathcal{O} = \Big\{ \mathtt{CC}\,[p]\,[q]\,\mathtt{CC}\,[q]\,[r]\,\mathtt{C}\,[p]\,[r]\,,\ \mathtt{CCN}\,[p]\,[p]\,[p]\,,\ \mathtt{C}\,[p]\,\mathtt{CN}\,[p]\,[q]\ ;\ p,q,r \in \mathbb{N} \Big\}.$$

(on vérifie – exercice – que chacun des trois axiomes est une EBF).

(ii) : L'ensembles $\mathbf{Th}(\mathcal{P})$ *des* $\mathcal{P}$*-théorèmes est défini récursivement grâce à deux règles d'inférence : ainsi, un* $\mathcal{P}$*-théorème est une EBF qui est soit un axiome, soit déduite des axiomes par les règles suivantes :* $R1$ *est la règle de substitution : si dans un* $\mathcal{P}$*-théorème du système on substitue une EBF à une occurrence d'une variable, on obtient un nouveau* $\mathcal{P}$*-théorème : en d'autres termes, pour tout entier* $p \geq 0$*,*

$$R_1 \ :\ \textit{si}\ (x\,[p]\,y, z) \in \mathbf{Th}(\mathcal{P}) \times \mathrm{Lang}(G)\ \textit{alors}\ xzy \in \mathbf{Th}(\mathcal{P})$$

$R2$ *est la règle du modus ponens ou du détachement : si* x *et* $\mathtt{C}xy$ *sont des* $\mathcal{P}$*-théorèmes alors* y *est un théorème du système, soit encore,*

$$R_2 \ :\ \textit{si}\ (\mathtt{C}xy, x) \in \mathbf{Th}(\mathcal{P}) \times \mathbf{Th}(\mathcal{P})\ \textit{alors}\ y \in \mathbf{Th}(\mathcal{P}).$$

Proposition 5.3. *Les axiomes du système de Lukasiewicz sont des tautologies.*

Preuve. Le calcul des tables de vérité de chacun des trois axiomes dans $\mathcal{O}$ donne :

$[p]$	CCN $[p]$ $[p]$ $[p]$
0	1
1	1

$[p]$	$[q]$	C $[p]$ CN $[p]$ $[q]$
0	0	1
0	1	1
1	0	1
1	1	1

$[p]$	$[q]$	$[r]$	CC $[p]$ $[q]$ CC $[r]$ $[r]$ C $[p]$ $[r]$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

□

Remarque 5.4. *(1) : La règle* R_2 *est une Post-production d'ordre* 2 *(voir* [Bia79] *pour la définition généralisant l'ordre 1). Par contre, la règle* R_1 *n'est une Post-production d'ordre* r *pour aucun* $r \geq 1$ *; en particulier, son domaine de définition n'est pas réduit à* $\mathbf{Th}(\mathcal{P}) \times \mathbf{Th}(\mathcal{P})$*) : en fait* R_2 *met implicitement en jeu une infinité de règles.*

(2) Les axiomes (tautologiques) de Lukasiewicz s'interprètent dans le système de Russel-Whitehead/Tarski qui est couramment utiliser de nos jours (voir [RW27]*) : si on note (p) la variable propositionnelle correspondant à `[p]` et $\sim(p)$ (Russel-Whitehead), ou encore $\neg(p)$ (Tarski) la négation `N[p]` de `[p]`.*

Lukasiewicz	*Russel/Tarski*
`CC[p][q]CC[q][r]C[p][r]`	$((p) \Rightarrow (q)) \Rightarrow (((q) \Rightarrow (r)) \Rightarrow ((p) \Rightarrow (r)))$
`CCN[p][p][p]`	$((\neg(p)) \Rightarrow (p)) \Rightarrow (p)$
`C[p]CN[p][q]`	$(p) \Rightarrow ((\neg(p)) \Rightarrow (q))$

(3) : Le premier axiome est la loi du syllogisme hypothétique, le second est la loi de Clavius (le raisonnement par l'absurde), le troisième axiome étant connu comme la loi de Dun Scott.

Proposition 5.5. *Les $\mathcal{P}$-théorèmes sont des tautologies (i.e.* $\mathbf{Th}(\mathcal{P}) \subset \mathrm{Taut}(\mathcal{P})$*).*

Preuve. Exercice.

□

Post et Lukasievicz ont simultanément démontrer le théorème suivant.

Théorème 5.6 (Post-Lukasiewicz)**.** *La théorie formelle $\mathcal{P}$ décrivant le calcul des propositions est cohérente et complète.*

6. Appendice : langage d'exécution d'une machine

Soit $T \in \mathfrak{T}(\mathcal{A})$ et $\mathcal{Q}$ l'ensemble des états internes de T ; pour une syntaxe consistante, on suppose que $\{\texttt{[}, \texttt{)}, \texttt{L}, \texttt{S}, \texttt{R}\}$ et $\mathcal{Q}$ sont disjoints de $\mathcal{A}$. Rappelons que l'ensemble des états courants $\mathrm{St}(T)$ de la machine T est constitué des mots de la forme $w\,\texttt{[Q)}\,m$ avec $\texttt{Q} \in \mathcal{Q}$ et $w, m \in \mathcal{A}^*$. Le langage d'exécution de T sur un langage $\mathcal{L} \subset \mathcal{A}^*$ est le sous-langage (récursivement énumérable) de $\mathrm{St}(T)$ défini par :

$$\mathrm{Exec}(T, \mathcal{L}) := \bigcup_{n=0}^{\infty} T^n(\texttt{[I)}\,\mathcal{L}\backslash\{\phi\}).$$

Le langage d'exécution d'une machine de Turing sur un langage récursif coïncide avec le langage d'un système de Post. Pour voir cela considérons $T \in \mathfrak{T}(\mathcal{A})$ dont l'ensemble des états internes est $\mathcal{Q}$ (avec $\star \in \mathcal{A}$ et $\{\texttt{L}, \texttt{S}, \texttt{R}, \texttt{[}, \texttt{)}\} \cup \mathcal{Q}$ disjoint de $\mathcal{A}$) et $\mathcal{R} \subset \mathrm{St}(T)$ un langage récursif. On définit alors le système de Post $S_T = (\mathcal{R}, (P_1, \ldots, P_N))$ où les Post-productions P_i sont tirées de (1) : ainsi, avec les variables de production w et m on a :

$$\begin{array}{rcll} w\,\texttt{[Q)}\,\texttt{a}m & \longrightarrow & w\texttt{b}\,\texttt{[P)}\,m & \text{lorsque } T(\texttt{Q},\texttt{a}) = (\texttt{P},\texttt{b},\texttt{R}) \\ w\,\texttt{[Q)}\,\texttt{a}m & \longrightarrow & w\,\texttt{[P)}\,\texttt{b}m & \text{lorsque } T(\texttt{Q},\texttt{a}) = (\texttt{P},\texttt{b},\texttt{S}) \\ w\texttt{c}\,\texttt{[Q)}\,\texttt{a}m & \longrightarrow & w\,\texttt{[P)}\,\texttt{cb}m & \text{lorsque } T(\texttt{Q},\texttt{a}) = (\texttt{P},\texttt{b},\texttt{L}) \end{array} \tag{3}$$

(où $\texttt{a}, \texttt{b}, \texttt{c} \in \mathcal{A}$). Il est alors facile de vérifier que $\mathrm{Form}(S_T) = \mathrm{Exec}(T, \mathcal{R})$. Enfin, remarquons que les Post-productions (3) du système formel S_T sont en fait des Thue-productions ; en effet, il est possible d'écrire (3) sous la forme substitutive suivante

$$\begin{array}{rcll} \texttt{[Q)a} & \longrightarrow & \texttt{b[P)} & \text{lorsque } T(\texttt{Q},\texttt{a}) = (\texttt{P},\texttt{b},\texttt{R}) \\ \texttt{[Q)a} & \longrightarrow & \texttt{[P)b} & \text{lorsque } T(\texttt{Q},\texttt{a}) = (\texttt{P},\texttt{b},\texttt{S}) \\ \texttt{c[Q)a} & \longrightarrow & \texttt{[P)cb} & \text{lorsque } T(\texttt{Q},\texttt{a}) = (\texttt{P},\texttt{b},\texttt{L}) \end{array} \tag{4}$$

RÉFÉRENCES

[Bia79] E. Bianco. *Informatique fondamentale : de la machine de Turing aux ordinateurs modernes*. Basel, Boston, Stuttgart ISR 70, 1979.

[Cho56] N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2-(2) :113–123, 1956.

[Chu36] A. Church. A note on the Entscheidungsproblem (correction pp. 101-102). *Journal of Symbolic Logic*, 1 :40–41, 1936.

[GL67] M. Gross and A. Lentin. *Notions sur les grammaires formelles*. Gauthier-Villars, 1967.

[GNNG89] K. Gödel, E. Nagel, J. Newman, and J.-Y. Girard. *Le Théorème de Gödel*. Seuil, 1989.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der Principia Mathematica und verwandter Systeme, i. *Monatshefte für Mathematik und Physik*, 38 :173–198, 1931.

[HA28] D. Hilbert and W. Ackermann. *Grundz"uge der theoretischen Logik (Principles of Mathematical Logic)*. Springer-Verlag, 1928.

[Hil02] D. Hilbert. Lecture delivered before the International Congress of Mathematicians at Paris in 1900 by Professor David Hilbert, (translated into english by Dr. Maby Winton Newson, with the author's permission). *Bulletin of the American Mathematical Society*, 8 :437–479, 1902.

[Hof79] D. Hofstadter. *Gödel, Escher, Bach : les brins d'une guirlande éternelle*. Dunod, 1979.

[Luk29] J. Lukasiewicz. *Elements of Mathematical Logic*. MacMillan, New York, 1929.

[Oli14a] E. Olivier. Qu'est-ce-qu'une machine ? (I/III). *Bull. Info. Appr. et Appl.*, 97 :27–38, 2014.

[Oli14b] E. Olivier. Qu'est-ce-qu'une machine ? (II/III). *Bull. Info. Appr. et Appl.*, 98 :45–56, 2014.

[RW27] B. Russel and A. N. Whitehead. *Principia Mathematica*. Second Ed. University Press, Cambridge, 1927.

[Sau79] F. De Saussure. *Cours de linguistique générale (édition originale 1916)*. Payot, Paris, 1979.

[Sau02] F. De Saussure. *Écrits de linguistique générale*. Bibliothèque de Philosophie, Gallimard, Éd. S. Bouquet, R. Engler, A. Weil, 2002.

[Tar44] A. Tarski. The semantic conception of truth and the foundations of semantics. *Philosophy and Phenomenological Research*, 4 :341–376, 1944.

[Tur36] A. Turing. On Computable Numbers, with an application to the Entscheidungsproblem (correction ibid. (1967) 43, p. 544-546). *Proc. Lond. Math. Soc.*, (2)-42 :230–265, 1936.

7. POSTSCRIPTUM : TEXTES CHOISIS

René Descartes (Discours de la Méthode)

Et enfin, comme ce n'est pas assez, avant de commencer à rebâtir le logis où on demeure, que de l'abattre, et de faire provision de matériaux et d'architectes, ou s'exercer soi-même à l'architecture, et outre cela d'en avoir soigneusement tracé de dessin, mais qu'il faut aussi s'être pourvu de quelque autre où on puisse être logé commodément pendant le temps qu'on y travaillera ; ainsi, afin que je ne demeurasse point irrésolu en mes actions, pendant que la raison m'obligerait de l'être en mes jugements, et que je ne laissasse pas de vivre dès lors le plus heureusement que je pourrais, je me formai une morale par provision, qui ne consistait qu'en trois ou quatre maximes dont je veux bien vous faire part.

René Descartes (Discours de la Méthode)

Et je m'étais ici particulièrement arrêté à faire voir que s'il y avait de telles machines qui eussent les organes et la figure extérieure d'un singe ou de quelque autre animal sans raison, nous n'aurions aucun moyen pour reconnaître qu'elles ne seraient pas en tout de même nature que ces animaux ; au lieu que s'il y en avait qui eussent la ressemblance de nos corps, et imitassent autant nos actions que moralement il serait possible, nous aurions toujours deux moyens très certains pour reconnaître qu'elles ne seraient point pour cela de vrais hommes : dont le premier est que jamais elles ne pourraient user de paroles ni d'autres signes en les composant, comme nous faisons pour déclarer aux autres nos pensées : car on peut bien concevoir qu'une machine soit tellement faite qu'elle profère des paroles, et même qu'elle en profère quelques-unes à propos des actions corporelles qui causeront quelque changement en ses organes, comme, si on la touche en quelque endroit, qu'elle demande ce qu'on lui veut dire ; si en un autre, qu'elle crie qu'on lui fait mal, et choses semblables ; mais non pas qu'elle les arrange diversement pour répondre au sens de tout ce qui se dira en sa présence, ainsi que les hommes les plus hébétés peuvent faire. Et le second est que, bien qu'elles fissent plusieurs choses aussi bien ou peut-être mieux qu'aucun de nous, elles manqueraient infailliblement en quelques autres, par lesquelles on découvrirait qu'elles n'agiraient pas par connaissance, mais seulement par la disposition de leurs organes : car, au lieu que la raison est un instrument universel qui peut servir en toutes sortes de rencontres, ces organes ont besoin de quelque particulière disposition pour chaque action particulière ; d'où vient qu'il est moralement impossible qu'il y en ait assez de divers en une machine pour la faire agir en toutes les occurrences de la vie de même façon que notre raison nous fait agir. Or, par ces deux mêmes moyens, on peut aussi connaître la différence qui est entre les hommes et les bêtes. Car c'est une chose bien remarquable qu'il n'y a point d'hommes si hébétés et si stupides, sans en excepter même les insensés, qu'ils ne soient capables d'arranger ensemble diverses paroles, et d'en composer un discours par lequel ils fassent entendre leurs pensées ; et qu'au contraire il n'y a point d'autre animal, tant parfait et tant heureusement né qu'il puisse être, qui fasse le semblable.

René Descartes (Discours de la Méthode)

J'avais décrit après cela l'âme raisonnable, et fait voir qu'elle ne peut aucunement être tirée de la puissance de la matière, ainsi que les autres choses dont j'avais parlé, mais qu'elle doit expressément être créée ; et comment il ne suffit pas qu'elle soit logée dans le corps humain, ainsi qu'un pilote en son navire, sinon peut-être pour mouvoir ses membres, mais qu'il est besoin qu'elle soit jointe et unie plus étroitement avec lui, pour avoir outre cela des sentiments et des appétits semblables aux nôtres, et ainsi composer un vrai homme.

René Descartes (Les Principes de la philosophie)

Il n'y a donc qu'une même matière en tout l'univers, et nous la connaissons par cela seul qu'elle est étendue ; pour ce que toutes les propriétés que nous apercevons distinctement en elle, se rapportent à ce qu'elle peut être divisée et mue selon ses parties, et qu'elle peut recevoir toutes les diverses dispositions que nous remarquons pouvoir arriver par le mouvement de ses parties.

René Descartes (Traité de l'Homme)

Tous les mouvements que nous faisons sans que notre volonté y contribue (comme il arrive souvent que nous respirons, que nous marchons, que nous mangeons, et enfin que nous faisons toutes les actions qui nous sont communes avec les bêtes) ne dépendent que de la conformation de nos membres et du cours que les esprits excités par la chaleur du cœur, suivent naturellement dans le cerveau, dans les nerfs et dans les muscles, en même façon que le mouvement d'une montre est produit par la seule force de son ressort et la figure de ses roues.

René Descartes (Traité de l'Homme)

Je désire que vous considériez, après cela, [...] que toutes les fonctions que j'ai attribuées à cette machine, comme la digestion des viandes, le battement du cœur et des artères, la nourriture et la croissance des membres, la respiration, la veille et le sommeil ; la réception de la lumière, des sons, des odeurs, des goûts, de la chaleur et de telles autres qualités, dans les organes des sens extérieurs ; l'impression de leurs idées dans l'organe du sens commun et de l'imagination, la rétention ou l'empreinte de ces idées dans la mémoire, les mouvements intérieurs des appétits et des passions [...] je désire, dis-je, que

vous considériez que ces fonctions suivent toutes naturellement en cette machine, de la seule disposition de ses organes, ni plus ni moins que font les mouvements d'une horloge, ou autre automate, de celle de ses contrepoids et de ses roues ; en sorte qu'il ne faut point à leur occasion concevoir en elle aucune autre âme végétative, ni sensitive, ni aucun autre principe de mouvement et de vie, que son sang et ses esprits, agités par la chaleur du feu qui brûle continuellement dans son cœur, et qui n'est point d'autre nature que tous les feux qui sont dans les corps inanimés.

René Descartes (Méditation métaphysique)

La nature m'enseigne aussi par ces sentiments de douleur, de faim, de soif, etc, que je ne suis pas seulement logé dans mon corps, ainsi qu'un pilote en son navire, mais, outre cela, que je lui suis conjoint très étroitement et tellement confondu et mêlé, que je compose comme un seul tout avec lui. Car, si cela n'était, lorsque mon corps est blessé, je ne sentirais pas pour cela de la douleur, moi qui ne suis qu'une chose qui pense, mais j'apercevrais cette blessure par le seul entendement, comme un pilote aperçoit par la vue si quelque chose se rompt dans son vaisseau.

René Descartes (Traité des passions)

Il est besoin aussi de savoir que, bien que l'âme soit jointe à tout le corps, il y a néanmoins en lui quelque partie en laquelle elle exerce ses fonctions plus particulièrement qu'en toutes les autres. Et on croit communément que cette partie est le cerveau, ou peut-être le cœur : le cerveau, à cause que c'est à lui que se rapportent les organes des sens ; et le cœur, à cause que c'est comme en lui qu'on sent les passions. Mais, en examinant la chose avec soin, il me semble avoir évidemment reconnu que la partie du corps en laquelle l'âme exerce immédiatement ses fonctions n'est nullement le cœur, ni aussi tout le cerveau, mais seulement la plus intérieure de ses parties, qui est une certaine glande fort petite, située dans le milieu de sa substance, et tellement suspendue au-dessus du conduit par lequel les esprits de ses cavités antérieures ont communication avec ceux de la postérieure, que les moindres mouvements qui sont en elle peuvent beaucoup pour changer le cours de ces esprits, et réciproquement que les moindres changements qui arrivent au cours des esprits peuvent beaucoup pour changer les mouvements de cette glande.

Blaise Pascal (Les pensées)

Il y a beaucoup de différence entre l'esprit de Géométrie et l'esprit de finesse. En l'un les principes sont palpables, mais éloignez de l'usage commun, de sorte qu'on a peine à tourner la teste de ce côté là manque d'habitude ; mais pour peu qu'on s'y tourne on voit les principes à plein ; et il faudrait avoir tout à fait l'esprit faux pour mal raisonner sur des principes si gros qu'il est presque impossible qu'ils échappent.

Mais dans l'esprit de finesse les principes sont dans l'usage commun, et devant les yeux de tout le monde. On n'a que faire de tourner la teste ni de se faire violence. Il n'est question que d'avoir bonne vue : mais il faut l'avoir bonne ; car les principes en sont si déliés et en si grand nombre, qu'il est presque impossible qu'il n'en échappe. Or l'omission d'un principe mène à l'erreur : ainsi il faut avoir la vue bien nette, pour voir tous les principes ; et ensuite l'esprit juste, pour ne pas raisonner faussement sur des principes connus.

Tous les géomètres seraient donc fins, s'ils avaient la vue bonne ; car ils ne raisonnent pas faux sur les principes qu'ils connaissent : et les esprits fins seraient géomètres, s'ils pouvaient plier leur vue vers les principes inaccoutumés de Géométrie.

Ce qui fait donc que certains esprits fins ne sont pas géomètres, c'est qu'ils ne peuvent du tout se tourner vers les principes de Géométrie : mais ce qui fait que des géomètres ne sont pas fins, c'est qu'ils ne voient pas ce qui est devant eux, et qu'étant accoutumés aux principes nets et grossiers de Géométrie, et à ne raisonner qu'après avoir bien vu et manié leurs principes, ils se perdent dans les choses de finesse, où les principes ne se laissent pas ainsi manier. On les voit à peine : on les sent plutôt qu'on ne les voit : on a des peines infinies à les faire sentir à ceux qui ne les sentent pas d'eux-mêmes : ce sont choses tellement délicates et si nombreuses, qu'il faut un sens bien délicat et bien net pour les sentir, et sans pouvoir le plus souvent les démontrer par ordre comme en Géométrie, parce qu'on n'en possède pas ainsi les principes, et que ce serait une chose infinie de l'entreprendre. Il faut tout d'un coup voir la chose d'un seul regard, et non par progrès de raisonnement, au moins jusqu'à un certain degré. et ainsi il est rare que les géomètres soient fins, et que les fins soient géomètres ; à cause que les géomètres veulent traiter géométriquement les choses fines, et se rendent ridicules, voulant commencer par les définitions, et ensuite par les principes, ce qui n'est pas la manière d'agir en cette sorte de raisonnement. Ce n'est pas que l'esprit ne le fasse ; mais il le fait tacitement, naturellement, et sans art ; car l'expression en passe tous les hommes, et le sentiment n'en appartient qu'à peu.

Et les esprits fins au contraire ayant ainsi accoutumé de juger d'une seule vue, sont si étonnez quand on leur présente des propositions où ils ne comprennent rien, et où pour entrer il faut passer par des définitions et des principes stériles et qu'ils n'ont point accoutumé de voir ainsi en détail, qu'ils s'en rebutent et s'en dégoûtent. Mais les esprit faux ne sont jamais ni fins ni géomètres.

Les géomètres qui ne sont que géomètres ont donc l'esprit droit, mais pourvu qu'on leur explique bien toutes choses par définitions et par principes ; autrement ils sont faux et insupportables ; car ils ne sont droits que sur les principes bien éclaircis. Et

les fins qui ne sont que fins ne peuvent avoir la patience de descendre jusqu'aux premiers principes des choses spéculatives et d'imagination qu'ils n'ont jamais vues dans le monde et dans l'usage.

Blaise Pascal
(La Machine d'Arithmétique – 1642-1645)

Au reste, si quelquefois tu as exercé ton esprit à l'invention des machines, je n'aurai pas grand-peine à te persuader que la forme de l'instrument, en l'état où il est à présent, n'est pas le premier effet de l'imagination que j'ai eue sur ce sujet : j' avais commencé l'exécution de mon projet par une machine très différente de celle-ci et en sa matière et en sa forme, laquelle (bien qu'en état de satisfaire à plusieurs) ne me donna pas pourtant la satisfaction entière ; ce qui fit qu'en la corrigeant peu à peu j'en fis insensiblement une seconde, en laquelle rencontrant encore des inconvénients que je ne pus souffrir, pour y apporter le remède, j'en composai une troisième qui va par ressorts et qui est très simple en sa construction. C'est celle de laquelle, comme j'ai déjà dit, je me suis servi plusieurs fois, au vu et su d'une infinité de personnes, et qui est encore en état de servir autant que jamais. Toutefois, en la perfectionnant toujours, je trouvai des raisons de la changer, et enfin reconnaissant dans toutes, ou de la difficulté d'agir, ou de la rudesse aux mouvements, ou de la disposition à se corrompre trop facilement par le temps ou par le transport, j'ai pris la patience de faire jusqu'à plus de cinquante modèles, tous différents, les uns de bois, les autres d'ivoire et d'ébène, et les autres de cuivre, avant que d'être venu à l'accomplissement de la machine que maintenant je fais paraître ; laquelle, bien que composée de tant de petites pièces différentes, comme tu pourras voir, est toutefois tellement solide, qu'après l'expérience dont j'ai parlé ci-devant, j'ose te donner assurance que tous les efforts qu'elle pourrait recevoir en la transportant si loin que tu voudras, ne sauraient la corrompre ni lui faire souffrir la moindre altération.

Gottfried Wilhelm Leibniz
Nouveaux Essais sur l'entendement humain
Livre deuxième – Chap XIV

§ 4. *Philalèthe*. Dans les nombres les idées sont et plus précises et plus propres à être distinguées les unes des autres que dans l'étendue, où on ne peut point observer ou mesurer chaque égalité et chaque excès de grandeur aussi aisément que dans les nombres, par la raison que dans l'espace nous ne saurions arriver par la pensée à une certaine petitesse déterminée au-delà de laquelle nous ne puissions aller, telle qu'est l'unité dans le nombre.

Théophile. Cela se doit entendre du nombre entier. Car autrement le nombre dans sa latitude, comprenant le rompu, le sourd, le transcendant et tout ce qui se peut prendre entre deux nombres entiers, est proportionnel à la ligne, et il y a là aussi peu de minimum que dans le continu. Aussi cette définition, que le nombre est une multitude d'unités, n'a lieu que dans les entiers. La distinction précise des idées dans l'étendue ne consiste pas dans la grandeur : car pour reconnaître distinctement la grandeur, il faut recourir aux nombres entiers, ou aux autres connus par le moyen des entiers, ainsi de la quantité continue il faut recourir à la quantité discrète pour avoir une connaissance distincte de la grandeur. Ainsi les modifications de l'étendue, lorsqu'on ne se sert point des nombres, ne peuvent être distinguées par la figure, prenant ce mot si généralement qu'il signifie tout ce qui fait que deux étendus ne sont pas semblables l'un à l'autre.

§ 5. *Philalèthe*. En répétant l'idée de l'unité et la joignant à une autre unité, nous en faisons une idée collective que nous nommons deux. Et quiconque peut faire cela et avancer toujours d'un de plus à la dernière idée collective, à laquelle il donne un nom particulier, peut compter, tandis qu'il a une suite de noms et assez de mémoire pour la retenir.

Théophile. Par cette manière seule on ne saurait aller loin. Car la mémoire serait trop chargée s'il fallait retenir un nom tout à fait nouveau pour chaque addition d'une nouvelle unité. C'est pourquoi il faut un certain ordre et une certaine réplication dans ces noms, en recommençant suivant une certaine progression. [...]

Gottfried Wilhelm Leibniz
Nouveaux Essais sur l'entendement humain
Livre troisième – Chap. II

§ 1*Philalèthe.* Maintenant, les mots étant employés par les hommes pour être signes de leurs idées, on peut demander d'abord comment ces mots y ont été déterminés ; et l'on convient que c'est non par aucune connexion naturelle qu'il y ait entre certains sons articulés et certaines idées (car en ce cas il n'y aurait qu'une langue parmi les hommes), mais par une institution arbitraire en vertu de laquelle un tel mot a été volontairement le signe d'une telle idée.

Théophile. Je sais qu'on a coutume de dire dans les écoles et partout ailleurs que les significations des mots sont arbitraires (ex instituto) et il est vrai qu'elles ne sont point déterminées par une nécessité naturelle, mais elles ne laissent pas de l'être par des raisons tantôt naturelles, où le hasard a quelque part, tantôt morales, où il y entre du choix [...]

Karl Marx (Le capital)

Un homme qui ne dispose d'aucun loisir, dont la vie tout entière, en dehors des simples interruptions purement physiques pour le sommeil, les repas, etc., est accaparée par son travail pour le capitaliste, est moins qu'une bête de somme. C'est une simple machine à produire la richesse pour autrui, écrasée

physiquement et abrutie intellectuellement. Et pourtant, toute l'histoire moderne montre que le capital, si on n'y met pas obstacle, travaille sans égard ni pitié à abaisser toute la classe ouvrière à ce niveau d'extrême dégradation.

Kurt Gödel
(note de 1963 ajouté à son article de 1931)

On peut démontrer rigoureusement que dans tout système formel consistant contenant une théorie des nombres finitaire relativement développée, il existe des propositions arithmétiques indécidables et que, de plus, la consistance d'un tel système ne saurait être démontrée à l'intérieur de ce système.

Kurt Gödel
(note de 1963 ajouté à son article de 1931)

Grâce à certains travaux qui ont suivi cet article [article de 1931], notamment ceux de A.M. Turing, nous disposons désormais d'une définition sûre, précise et adéquate du concept de système formel [...] dont la propriété est qu'en son sein et en principe, le raisonnement peut-être entièrement remplacé par des règles mécaniques.

Alan Turing

L'identification des fonctions 'effectivement calculables' avec les fonctions calculables est peut-être plus convaincante que l'identification avec les fonctions λ-définissables ou récursives générales. Pour ceux qui adoptent ce point de vue, la démonstration formelle de l'équivalence fournit une justification du calcul de Church et permet de remplacer les 'machines' qui engendrent des fonctions calculables par les λ-définitions qui sont plus commodes.

Alan Turing

Hydra ressemble à une anémone de mer mais vit en eau douce et possède cinq à dix tentacules. Si l'on coupe une partie de *Hydra*, cette partie se réorganise pour former un nouvel organisme complet. Lors de ce processus, l'organisme prend la forme d'un tube, ouvert et légèrement évasé du côté de la tête, et fermé de l'autre côté. Le tout possède encore une symétrie circulaire. Ultérieurement, la symétrie disparaît au point qu'une tache spécifique se révèle, celle-ci étant à l'origine d'un certain nombre de plaques du côté de la tête. Ces plaques se manifestent aux endroits où apparaîtront les tentacules.

Gershom Scholem
(Lecture at Weizmann Institute on June 17, 1965)

Once upon a time, there was a great Rabbi in Prague. His name was Rabbi Jehuda Loew ben Bezalel and he is known in Jewish tradition as the Maharal of Prague. A famous scholar and mystic, he is credited by Jewish popular tradition with the creation of a Golem – a creature produced by the magical power of man and taking on human shape. Rabbi Loew's robot was made of clay and given a sort of life by being infused with the concentrated power of the Rabbi's mind. This great human power is, however, nothing but a reflection of God's own creative power, and therefore, after having gone through all the necessary procedures in building his Golem, the Rabbi finally put a slit of paper into its mouth with the mystic and ineffable Name of God written on it. So long as this seal remains in his mouth, the Golem was alive – if you can call such a state alive. For the Golem could work and do the biding of his master and perform all kinds of chores for him, helping him and the Jews of Prague in many ways. But the poor creature could not speak. He could respond to orders and could sort them out, but no more than that. [...]

Now, THIS IDEA of the Golem is deeply ingrained in the thinking of the Jewish mystics of the Middle Ages known as the Kabbalists. I want to give you an inkling of what lies behind the idea. It may be far remove from what the modern electronic engineer and applied mathematician have in mind when they concoct their own species of Golem – and yet, all theological trappings notwithstanding, there is straight line linking to the two developments. As a matter of fact, the Golem – a creature created by human intelligence and concentration, which is controlled by its creator and performs tasks set for him, but which at the same time may have a dangerous tendency to outgrow that control and develop destructive potentialities – is nothing but a replica of Adam, the first Man himself. God could create Man from a heap of clay and invest him with a spark of His divine life force and intelligence (this, in the last analysis, is the "divine image" in which man was created). Without this intelligence and the spontaneous creativity of the human mind, Adam would have been nothing but a Golem – as, indeed, he is called in some of the old rabbinic stories interpreting the Biblical account. [...]

The universe, so the Kabbalists tell us, is built essentially on the prime elements of numbers and letters, because letters of God's language reflected in human language are nothing but concentrations of His creative energy. Thus, by assembling these elements in all their possible combinations and permutations, the Kabbalist who contemplates the mysteries of Creation radiates some of this elementary power into the Golem. The creation of a Golem is then in some way an affirmation of the productive and creative power of Man. It repeats, on however small a scale, the work of creation.

Isaac Asimov : "The Three Laws of Robotics"

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey any orders given to it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

DUNOD

(ouvrage technique)

VOUZZAVEDIBISAR : DUNOD (par Michel AVEZARD alias Zevar)