Une brève introduction à la théorie effective de l'aléatoire

Laurent Bienvenu, Mathieu Hoyrup

Une brève introduction à la théorie effective de l'aléatoire

Laurent Bienvenu
Chargé de recherche CNRS
LIAFA, Université de Paris 7

Mathieu Hoyrup
Chargé de recherche INRIA
LORIA, Nancy

1 La complexité de Kolmogorov

1.1 Des suites pas vraiment aléatoires

Imaginez qu'un après-midi où vous n'avez rien d'autre à faire, vous décidiez de prendre une pièce de monnaie, et de tirer à pile ou face mille fois de suite. Imaginez de plus que vous obteniez alors le résultat :

$$PPPPPPPPPP\ldots \text{ (mille piles)}$$

Il y a là de quoi s'interroger. La suite obtenue ne semble absolument pas « aléatoire », mais en y réflechissant, du point de vue de la théorie usuelle des probabilités, cette suite a rigoureusement *la même* probabilité d'occurence que n'importe quelle autre suite de la même longueur. Mais quel joueur, fut-il mathématicien, ne conclurait pas immédiatemment devant un tel résultat que la pièce est biaisée ? Nous sommes donc ici face à un paradoxe que la théorie usuelle des probabilités est impuissante à résoudre.

Une réponse à ce paradoxe fut apportée en 1965 par Kolmogorov [Kol65] [1] (déjà à l'origine de la formalisation axiomatique du calcul des probabilités !). L'intuition de Kolmogorov est la suivante : si la suite $PPPPPP\ldots$ nous semble surprenante, c'est qu'elle est simple, c'est-à-dire facile à décrire. Et Kolmogorov de proposer de définir la complexité d'un objet comme étant la longueur de sa plus petite « description ». Il s'agit bien entendu d'être prudent sur ce que l'on appelle « description », car les paradoxes rôdent, notamment le célèbre paradoxe de Berry, qui consiste en l'expression :

Le plus petit entier naturel qui ne peut pas être décrit avec moins de deux cents caractères

1. Des idées similaires furent développées à la même époque par Solomonoff [Sol64] et Chaitin [Cha66, Cha75], ce qui donne lieu, encore aujourd'hui, à d'incessantes querelles de paternité. Nous nous garderons bien d'y prendre part, et renvoyons le lecteur curieux aux articles originaux.

Cette expression comportant moins de deux cents caractères, l'entier n qu'elle décrit est en contradiction même avec sa définition. La solution proposée par Kolmogorov pour éviter ce genre d'éccueil est d'adopter un point de vue inspiré de l'informatique, dans lequel « description » s'entend par « programme informatique ». La *complexité de Kolmogorov* d'un objet fini x (suite de piles/faces, entier naturel, graphe fini, etc) est donc la taille du plus petit programme informatique permettant de générer x. On ne peut s'empêcher de faire l'analogie avec le principe du Rasoir d'Occam, couramment interprété dans la science moderne comme la recherche de l'explication la plus simple à une observation. Ici l'objet observé est x, et sa plus simple « explication » est le plus petit programme informatique qui produit x.

1.2 Un peu de calculabilité

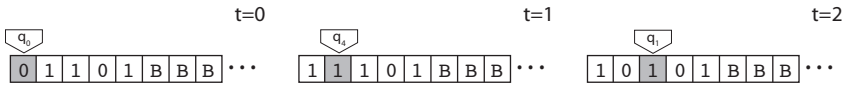
Le lecteur rigoureux ne manquera pas de faire remarquer que la définition de la complexité de Kolmogorov que nous venons de proposer est encore un peu trop vague. Le terme « programme informatique » est très général : il existe divers langages de programmation (Pascal, Caml, C, Java, etc.), et le résultat d'un programme informatique donné dépend également du système d'exploitation (Windows, Unix, etc) sur lequel il est exécuté. Il paraît donc peu aisé de donner une définition *absolue* de la notion de programme informatique.

Ceci peut néanmoins être fait de façon rigoureuse grâce aux travaux d'Alan Turing, datant du milieu des années 1930. Turing propose un modèle mathématique d'ordinateur (et ce bien avant l'apparition des ordinateurs que nous connaissons aujourd'hui !), appelé *machine de Turing*. Une machine de Turing consiste en :

- un ruban de lecture, composé d'une infinité de cases, chaque case pouvant contenir un des trois caractères 0,1, et B (blanc).
- une tête de lecture/écriture, qui se déplace sur le ruban case par case
- un ensemble fini d'états dans lequels peut se trouver la tête, dont deux particuliers, l'état initial q_I, et l'état final q_F.
- une table de transition, qui peut être vue comme une fonction à deux paramètres, prenant en entrée l'état actuel de la tête, et le symbole présent sur la case lue par la tête, et donnant les instructions à effectuer dans ce cas, qui sont au nombre de trois : (1) écrire un nouveau symbole sur la case, (2) déplacer la tête de lecture d'une case vers la droite ou d'une case vers la gauche, (3) changer l'état de la tête de lecture. On peut représenter cette table de transition comme un ensemble de quintuplés ; par exemple, le quintuplé $(q_3, B, 1, \leftarrow, q_5)$ signifie que si la tête est dans l'état q_3 et lit le symbole B, alors elle écrit à la place le symbole 1, se déplce vers la gauche et passe dans l'état q_5.

Un calcul d'une machine se déroule de la façon suivante. Initialement, on place une suite finie de symboles 0 ou 1 (la valeur d'entrée du calcul) au début du ruban, que l'on complète par une infinité de B. La tête de lecture/écriture est placée en face de la première case, dans l'état q_I. Puis on exécute les instruc-

tions données par la table de transition. Si à un moment donné la tête de lecture passe dans l'état final q_F, le calcul s'arrête, et le résultat (valeur de sortie) est alors la suite finie de symboles 0 ou 1 qui se trouve sur le ruban (les symboles B sont ignorés).



Une machine de Turing M calcule donc une application (notée également M, par souci de simplicité) de l'ensemble des suites binaires finies, habituellement noté $\{0,1\}^*$, dans lui-même. Cette application est en général partielle car une machine M peut, sur une entrée donnée p, ne jamais atteindre l'état final. Aussi rudimentaire que peut paraître une machine de Turing, on peut montrer que toute fonction calculable (de $\{0,1\}^*$ dans lui-même) par ordinateur l'est par une machine de Turing. On a ainsi un modèle de calcul simple mais suffisamment puissant pour formaliser notre intuition de départ, par la correspondance

ordinateur	=	machine de Turing
programme/description	=	valeur d'entrée du ruban (au début du calcul)
objet décrit	=	valeur de sortie du ruban (à la fin du calcul)

1.3 Vers une définition formelle de la complexité de Kolmogorov

Ainsi, lorsque l'on a un calcul du type $M(p) = x$, p peut-être vu comme une description de la suite binaire finie x *relativement à la machine* M. On peut alors définir la complexité de Kolmogorov relativement à M de x (notée $K_M(x)$) comme étant la longueur du plus petit p tel que $M(p) = x$, formellement :

Définition 1. *La complexité de Kolmogorov de* $x \in \{0,1\}^*$ *relativement à une machine* M *est définie par*

$$K_M(x) = \min\left\{|p| \ : \ M(p) = x\right\}$$

(où $|p|$ *désigne la longueur de* p*), avec la convention* $K_M(x) = +\infty$ *si aucun* p *ne satisfait* $M(p) = x$*.*

Il demeure que la complexité K_M dépend de la machine M, alors que l'on souhaite avoir une notion de complexité de Kolmogorov absolue. Sans pouvoir supprimer totalement cette dépendance, il est possible de la minimiser. En effet, il existe des machines de Turing *universelles*, c'est-à-dire qui peuvent simuler toute autre machine. Plus précisément, il existe une énumération $M_0, M_1, \ldots$ des

machines de Turing [2] et une machine U telle que, sur l'entrée 0^k1p (k symboles 0, suivis de 1 puis de p), U retourne $M_k(p)$.

Proposition 2. *La machine universelle U présentée ci-dessus est* optimale *pour la complexité de Kolmogorov, c'est-à-dire que pour toute machine M, il existe une constante c_M telle que*

$$K_U \leq K_M + c_M$$

Preuve : soit M une machine de Turing, et soit k son indice dans l'énumération des machines. Pour tout $x \in \{0,1\}^*$, s'il existe p tel que $M(p) = x$, on a alors $U(0^k1p) = x$ par définition de U, et donc par définition de la complexité de Kolmogorov, $K_U(x) \leq K_M(x) + (k+1)$. En prenant $c_M = k+1$, on a le résultat voulu.

La proposition précédente permet finalement de définir la complexité de Kolmogorov d'une suite binaire finie x, en posant

$$K(x) = K_U(x)$$

où U est une machine de Turing optimale, fixée une fois pour toutes. Comme nous l'avons dit plus haut, notre définition de la complexité de Kolmogorov n'est pas indépendante de la machine U, mais elle l'est à une constante près. La théorie de complexité de Kolmogorov est donc une théorie *asymptotique* : dire que $K(0001101) = 3$ n'a aucun sens, mais on peut par exemple dire que si x est une suite binaire de longueur n, dont les $n/2$ premiers bits sont des zéros, alors $K(x) \leq n/2 + O(1)$. On notera que la Proposition 2 assure que $K(x) < +\infty$ pour tout x, et même que $K(x) \leq |x| + O(1)$. En effet, si M est la machine calculant la fonction identité (c'est-à-dire la machine qui sur toute entrée s'arrête immédiatement sans rien faire), on a pour tout x, $K_M(x) = |x|$, et donc $K_U(x) \leq |x| + O(1)$ par la Proposition 2. Ceci est clair intuitivement : pour décrire un x donné, une façon de procéder est de décrire x explicitement, c'est-à-dire d'en énoncer les bits un par un.

Nous avons jusqu'à maintenant défini la complexité de Kolmogorov uniquement pour les suites binaires finies. Mais, comme les informaticiens le savent bien, la plupart des objets finis dont l'ensemble est dénombrable peuvent s'encoder facilement par des suites binaires. Ainsi, un entier n peut se représenter par son écriture binaire, ou encore par 0^n, une paire d'entiers (n, m) par 0^n1^m, un graphe fini à n sommets comme une matrice carrée $n \times n$ à coefficients binaires, elle-même représentable par une suite binaire de n^2 élements, etc. On peut donc ainsi, ayant choisi un encodage, définir la complexité de Kolmogorov d'un entier, d'un rationel, d'un graphe fini, etc, comme étant la complexité de Kolmogorov de la suite binaire finie le représentant. Le lecteur attentif (toujours lui !) objectera que la complexité de Kolmogorov de tels objets dépendra de l'encodage choisi. C'est exact, mais seulement à une constante additive près

2. le fait qu'il n'existe qu'une quantité dénombrable de machines de Turing est clair par la définition présentée plus haut

(la complexité de Kolmogorov elle-même étant définie à une constante additive près, ce n'est donc pas un problème !), comme le montre le lemme suivant :

Lemme 3. *Pour toute fonction f calculable, il existe une constante c_f telle que pour $x \in \{0,1\}^*$:*

$$K(f(x)) \leq K(x) + c_f$$

Preuve : soit M la machine de Turing qui sur une entrée p, calcule $f(U(p))$ (où U est notre machine optimale fixée). Une telle machine existe car la composée de deux fonctions calculables est toujours calculable[3]. Soit x une suite binaire finie. Soit p le plus court programme pour U qui produit x (i.e., tel que $|p| = K(x)$). On a par definition $M(p) = f(U(p)) = f(x)$. Donc $K_M(f(x)) \leq K(x)$, et par la Proposition 2, il suit que $K(f(x)) \leq K(x) + c_M$. En prenant $c_f = C_M$ on a le résultat voulu.

Ainsi, si x et y sont deux encodages d'un même objet (par exemple un rationel), en prenant f et g les fonctions permettant de passer d'un codage à l'autre, le Lemme ci-dessus prouve que $\big|K(x) - K(y)\big| = O(1)$. De façon plus générale, le Lemme 3 exprime le fait que l'on ne peut pas créer de la complexité de Kolmogorov par des moyens algorithmiques.

1.4 Des suites finies aléatoires

Ainsi équipés de la notion de complexité de Kolmogorov, nous pouvons enfin formaliser la notion de suite finie aléatoire comme étant une suite n'admettant pas de description plus petite qu'elle-même :

Définition 4. *Une suite binaire finie x est dite aléatoire si $K(x) \geq |x|$. De même, pour un type d'objet fini, on dit qu'un objet x est aléatoire si sa représentation sous forme de suite binaire l'est (l'encodage étant fixé à l'avance). Plus généralement, on dit qu'un objet x est c-aléatoire si $K(x) \geq |x| - c$.*

Notons que pour tout n il existe nécessairement des suites binaires aléatoires de longueur n. En effet, le nombre de programmes de longueur au plus $n-1$ est égal à $2^0 + 2^1 + \ldots + 2^{n-1} = 2^n - 1$; il existe donc moins de descriptions potentielles de longueur au plus $n-1$ que de suites de longueur n, ce qui prouve notre assertion. Le même argument combinatoire montre que parmi les suites binaires de longueur n, au plus 2^{n-c} ne sont pas c-aléatoires (soit une proportion d'au plus 2^{-c} de suites non c-aléatoires). Par exemple, pour $c = 10$, au moins $99,9\%$ des suites d'une longueur n donnée sont c-aléatoires. Ceci correspond bien à l'intuition que les suites finies aléatoires (ou quasi-aléatoires) doivent être nombreuses, de sorte que si l'on tire une suite binaire au hasard, la suite obtenue sera aléatoire avec grande probabilité (ce qui est bien la moindre des choses !)

3. un fait intuitivement évident, mais que nous admettrons ici car il n'est pas si facile de le prouver directement avec les machines de Turing.

Il est en revanche à noter que la notion d'objet aléatoire (ou c-aléatoire) n'est pas robuste. En effet, la complexité de Kolmogorov n'est définie qu'à une constante près (dépendant de la machine optimale U choisie), mais modifier la constante peut changer l'ensemble des suites aléatoires. A bien y réfléchir, le fait de ne pas avoir dans le cas des suites binaire finies une distinction absolue entre « aléatoire » et « non-aléatoire » est sans doute une bonne chose : quel sens aurait l'affirmation « 0000 n'est pas aléatoire » ? On peut en revanche affirmer que pour n suffisamment grand, la suite 0^n n'est pas aléatoire. En effet, soit f la fonction de $\mathbb{N}$ dans $\{0,1\}^*$ qui à n associe 0^n. D'après le Lemme 3, on a

$$K(0^n) = K(f(n)) \leq K(n) + O(1)$$

Or, un entier n peut se représenter par une suite binaire de longueur $\log(n) + O(1)$: son écriture en base 2. Ainsi, pour tout entier n, $K(n) \leq \log(n) + O(1)$. Il suit des deux inégalités précédentes que

$$K(0^n) \leq \log(n) + O(1)$$

Et donc, pour n suffisamment grand, la complexité de 0^n, de l'ordre de $\log(n)$, est beaucoup plus petite que sa longueur, qui est n. Voilà qui apporte une réponse partielle au paradoxe du début de cet article : pour tout n assez grand, la suite $PPPPPP\ldots$ (n fois P) n'est pas aléatoire (on peut en effet identifier P^n à 0^n par l'encodage $P \mapsto 0$ et $F \mapsto 1$).

Au delà de cet exemple simple, la notion de suite binaire aléatoire est riche d'applications, notamment à l'algorithmique et à la combinatoire. On peut en effet, pour donner une une borne inférieure sur complexité d'un algorithme, utiliser les suites aléatoires comme « pire cas », et estimer la complexité de l'algorithme sur ces suites seulement.

Il arrive également que l'on puisse utiliser la complexité de Kolmogorov pour prouver la correction d'algorithmes probabilistes. En plus de son entrée x, un algorithme probabiliste possède une seconde entrée, qui peut être vue comme une chaîne de bits aléatoires r, que l'algorithme utilise lors de son exécution. Pour prouver que l'algorithme est correct, il est d'usage de prouver que pour toute entrée x, la probabilité que r fasse que l'algorithme retourne la bonne réponse est haute. Une autre façon de procéder est de prouver que si r est une suite aléatoire (ou quasi-aléatoire), alors l'algorithme est correct. Cette technique a récemment été utilisée de façon spectaculaire par Moser [Mos09] pour obtenir un résultat majeur en combinatoire, à savoir une version constructive (via un algorithme probabiliste) du lemme local de Lovász (voire [For09] pour une bonne "vulgarisation" de ce résultat).

Il n'est pas possible, dans ces quelques pages, de donner un exposé plus détaillé de tels arguments. Cependant, afin de comprendre le type de techniques qu'ils peuvent mettre en jeu, nous allons donner une preuve alternative d'un théorème élémentaire bien connu en utilisant la complexité de Kolmogorov : il existe une infinité de nombres premiers. Supposons en effet qu'il en existe

seulement un nombre fini $p_1 < \ldots < p_k$. On a alors une surjection (calculable) f de $\mathbb{N}^k$ dans $\mathbb{N} \setminus \{0\}$ définie par

$$f(\alpha_1, \ldots, \alpha_k) = p_1^{\alpha_1} \ldots p_k^{\alpha_k}$$

Soit maintenant un entier n aléatoire, c'est-à-dire que sa représentation binaire est une suite aléatoire. Cette suite a pour taille $\log(n) + O(1)$, et comme elle est aléatoire on a par définition $K(n) = \log(n) + O(1)$. Soit alors $(\alpha_1, \ldots, \alpha_k)$ le k-uplet tel que $f(\alpha_1, \ldots, \alpha_k) = n$. D'après le Lemme 3, on a $K(n) \leq K(\alpha_1, \ldots, \alpha_k) + O(1)$. Comme les nombres premiers p_i sont tous au moins égaux à 2, les α_i sont tous d'ordre $O(\log n)$, et donc représentés par une suite binaire de longueur $O(\log \log n)$. Donc :

$$\log(n) + O(1) = K(n) \leq K(\alpha_1, \ldots, \alpha_k) = O(k \log \log n)$$

ce qui est bien sûr une contradiction pour n suffisamment grand.

Dans cette première partie, nous avons vu comment donner un sens à la notion d'aléatoire pour les suites binaires finies (et les objets qui peuvent être représentés ainsi), via la complexité de Kolmogorov. Insistons une nouvelle fois sur le fait que la notion de complexité de Kolmogorov est plus quantitative que qualitative, et ne permet pas de dire si une suite de dix mille « piles » obtenue en jouant à pile ou face est aléatoire ou non. Supposons maintenant que l'on lance une pièce équilibrée une infinité de fois, et que l'on obtienne la suite $PPP\ldots$ (uniquement des piles). On a envie dans ce cas de dire que cette suite *n'est pas* aléatoire. On peut donc ainsi espérer définir un critère qualitatif d'aléatoire sur l'ensemble des suites binaires infinies c'est-à-dire, définir de façon absolue quelles suites sont aléatoires et quelles suites ne le sont pas. C'est en effet possible, non seulement dans le cas des suites binaires infinies, mais dans de nombreux espaces de probabilité, comme nous allons le voir maintenant.

2 Eléments aléatoires et pseudo-aléatoires dans les espaces de probabilités

Si l'on tente de définir mathématiquement ce que pourrait vouloir dire, pour un élément d'un espace de probabilité, qu'être « aléatoire », on s'attend à ce qu'un tel élément soit typique vis-à-vis de la mesure de probabilité sous-jacente, c'est-à-dire qu'il vérifie « la plupart » des propriétés vraies avec probabilité 1, ou qu'il soit dans « la plupart » des ensembles de mesure 1. Sauf dans le cas où l'espace de probabilité est discret, on ne peut évidemment pas exiger d'un élément qu'il soit dans tous les ensembles de mesure 1. Alors quel sens donner à l'expression « la plupart » ? La théorie de la calculabilité (qui permet de formaliser la notion d'algorithme, et que l'on a présentée ci-dessus via les machines de Turing), et plus généralement l'*analyse calculable* (la théorie qui

permet de manipuler les objets, structures de l'analyse mathématique avec des algorithmes) fournissent des réponses satisfaisantes à cette question.

L'idée est donc de définir un élément aléatoire comment un élément qui vérifie toutes les propriétés de mesure 1 qui peuvent être « décrites par un algorithme ». Tout le problème est donc de donner un sens précis à cette dernière expression et nous verrons qu'il existe plusieurs notions non équivalentes, ayant chacune son intérêt propre.

Pour définir comment un algorithme peut décrire un ensemble mesurable, nous partirons du résultat classique de la théorie de la mesure.

Théoreme 5. *Sur tout espace métrique X, toute mesure de probabilité borélienne P est régulière, c'est-à-dire que pour tout borélien $A \subseteq X$ et tout $\epsilon > 0$, il existe un fermé F et un ouvert G tels que $F \subseteq A \subseteq G$ et $P(G \setminus F) < \epsilon$.*

Une manière de décrire un borélien est donc de fournir une suite de fermés et une suite d'ouverts approchant l'ensemble par l'intérieur et l'extérieur respectivement.

Nous travaillons ici sur l'ensemble $\mathbb{R}$ des nombres réels muni de la σ-algèbre $\mathscr{B}$ des boréliens. Tout peut se généraliser aux espaces métriques séparables.

2.1 L'approche de Martin-Löf

L'approche de Martin-Löf s'appuie sur une version calculable de la régularité des mesures.

Définition 6. *Un ensemble A est* **Martin-Löf P-mesurable** *s'il existe une machine de Turing qui, sur l'entrée n, décrit un fermé F_n et un ouvert U_n tels que $F_n \subseteq A \subseteq U_n$ et $P(U_n \setminus F_n) < 2^{-n}$.*

Décrire un ouvert $U \subseteq \mathbb{R}$, c'est produire une liste (infinie) d'intervalles ouverts à bornes rationnelles dont l'union est U. Décrire un fermé F, c'est décrire l'ouvert $\mathbb{R} \setminus F$.

La définition 6 est une généralisation d'une définition due à Martin-Löf, qui s'est limité dans [ML66] aux ensembles de mesure nulle (l'algorithme n'ayant alors pas à fournir les fermés F_n). Faisons quelques observations à propos de cette notion :

Remarques.

– *La description produite par l'algorithme est partielle dans le sens où elle ne caractérise pas l'ensemble A. En fait, l'algorithme décrit tout ensemble A' vérifiant*

$$\bigcup_n F_n \subseteq A' \subseteq \bigcap_n U_n. \tag{1}$$

Ainsi, la description fournie par l'algorithme peut éventuellement distinguer deux boréliens P-équivalents : il existe (sauf dans le cas d'une mesure discrète) des boréliens A' qui sont P-équivalent à A mais ne satisfont pas la double-inclusion (1) : il suffit par exemple d'enlever un élément de $\bigcup_n F_n$ à A. Ainsi, l'information donnée par l'algorithme sur A est plus fine qu'une description de la classe d'équivalence de A.

– *En réalité, si A est Martin-Löf P-mesurable, il existe des boréliens A' qui sont P-équivalents à A mais qui ne sont pas Martin-Löf P-mesurables.*

Théoreme 7 (Martin-Löf [ML66])**.** *L'intersection de tous les ensembles Martin-Löf P-mesurables de mesure 1 est Martin-Löf P-mesurable.*

On note M_P cette intersection et on qualifie de **Martin-Löf *P*-aléatoires** *ses éléments.*

Notons $\mathscr{M}_P$ la collection des ensembles Martin-Löf P-mesurables. Ce n'est pas une σ-algèbre en général, mais

1. C'est une algèbre booléenne
2. Si pour tout $i \in \mathbb{N}$, A_i est Martin-Löf P-mesurable et s'il existe un *même* algorithme qui sur l'entrée i décrit A_i et si $P(\bigcup_i A_i)$ est un nombre calculable, alors $P(\bigcup_i A_i)$ est Martin-Löf P-mesurable, et de même pour $\bigcap_i A_i$.

Nous étions partis d'un espace de probabilité $(\mathbb{R}, \mathscr{B}, P)$. Considérons l'ensemble M_P, la collection $\mathscr{M}_P$ de parties de M_P, et la restriction de P à $\mathscr{M}_P$. $(\mathrm{M}_P, \mathscr{M}_P, P)$ n'est pas un espace de probabilité car $\mathscr{M}_P$ n'est pas une σ-algèbre. Néanmoins, $\mathscr{M}_P$ a une structure proche, qui permet, dans certaines limites, de « faire des probabilités » sur cette structure. En voici un exemple.

Lemme 8 (Borel-Cantelli)**.** *Soient A_i des ensembles uniformément Martin-Löf P-mesurables tels que $\sum_i P(A_i) < \infty$. Alors*

i) $\limsup A_i = \bigcap_i \bigcup_{j>i} A_j$ *est Martin-Löf P-mesurable, et de mesure nulle.*

ii) $M_P \subseteq (\limsup A_i)^c$.

(on utilise ci-dessus la notation E^c pour désigner le complémentaire d'un évènement E). Ainsi, la conclusion n'est pas que *P-presque tout* élément n'appartient qu'à un nombre fini de A_i, mais que cela est vrai de *tout élement* de M_P. Ainsi chacun des points Martin-Löf P-aléatoires se comporte bien de manière typique pour cette propriété.

Remarquons que dans le lemme classique, l'analogue de (i) n'est pas explicité puisqu'il se déduit immédiatement du fait que l'on dispose d'une σ-algèbre.

2.2 L'approche de Schnorr

Montrer qu'un ensemble de mesure 1 est Martin-Löf P-mesurable, c'est l'approcher de l'intérieur par des fermés dont on contrôle la mesure. Dans bien des cas, on connaît précisément la mesure de ces fermés, ou tout au moins, on peut concevoir un programme qui va estimer leurs mesures avec n'importe quelle précision. C'est par exemple le cas dans la loi forte des grands nombres.

La définition suivante, plus forte que la Définition 6, requiert explicitement que l'on puisse estimer à l'aide d'un algorithme les mesures des fermés et ouverts approchant le borélien. On dira qu'une machine de Turing *décrit* un réel x si elle produit une suite de rationnels q_n (encodés sous forme de suites binaires finies) vérifiant $|q_n - x| < 2^{-n}$. Un réel x est dit *calculable* s'il peut être décrit par une machine de Turing. L'ensemble des réels calculables est dénombrable puisque l'ensemble des algorithmes l'est.

Définition 9. *Un ensemble A est* **Schnorr P-mesurable** *s'il existe un algorithme qui, sur l'entrée n, décrit un fermé F_n, un ouvert U_n et leurs mesures respectives, tels que $F_n \subseteq A \subseteq U_n$ et $P(U_n \setminus F_n) < 2^{-n}$.*

De la même façon, notons $\mathscr{S}_P$ la collection des ensembles Schnorr P-mesurables. Par définition, la Schnorr P-mesurabilité est plus forte que la Martin-Löf P-mesurabilité, c'est-à-dire que $\mathscr{S}_P \subseteq \mathscr{M}_P$. Encore une fois, $\mathscr{S}_P$ n'est pas une σ-algèbre en général mais possède les même propriétés que $\mathscr{M}_P$:

1. C'est une algèbre booléenne
2. Si pour tout $i \in \mathbb{N}$, A_i est Schnorr P-mesurable et qu'il existe un *même* algorithme qui sur l'entrée i décrit A_i et si $P(\bigcup_i A_i)$ est un nombre calculable, alors $P(\bigcup_i A_i)$ est Schnorr P-mesurable, et de même pour $\bigcap_i A_i$.

Cette notion plus forte induit une autre notion d'élément aléatoire, plus faible.

Définition 10. *On note S_P l'intersection de tous les ensembles Schnorr P-mesurables de mesure 1, et on appelle* **Schnorr P-aléatoires** *ses éléments.*

La situation est alors bien différente, puisque l'analogue du Théorème 7 n'est pas vrai en général, comme le montre le théorème suivant (la mesure P est dite non-atomique si elle n'a pas de point de concentration, c'est-à-dire de point $x \in [0,1]$ tel que $P(\{x\}) > 0$).

Théoreme 11. *Si la mesure P est non-atomique, S_P n'est pas Schnorr P-mesurable.*

Ce résultat est un corollaire du résultat suivant :

Théoreme 12. *Tout ensemble Schnorr P-mesurable de mesure non nulle contient un élément calculable.*

En effet, pour obtenir le Théorème 11 à partir du Théorème 12, il suffit de remarquer que si un point calculable est de P-mesure nulle, il n'est pas Schnorr aléatoire. Ainsi si P n'a pas d'atome, aucun point Schnorr P-mesurable n'est calculable, donc S_P, de mesure 1, ne peut pas être Schnorr P-mesurable.

Notons qu'inversement, ce dernier théorème n'a pas d'analogue dans le cadre de Martin-Löf, puisque M_P, Martin-Löf P-mesurable et de mesure 1, ne contient pas de point calculable (toujours dans le cas où P n'a pas d'atome).

Le Théorème 12 est intéressant parce qu'il permet de simuler l'aléatoire, et fonctionne comme un générateur pseudo-aléatoire : l'élément dont l'existence est assurée se comporte comme un point aléatoire vis-à-vis de la propriété considérée, mais n'est pas aléatoire puisqu'il peut être engendré par un programme, processus déterministe, prédictible, reproductible. Ce théorème est de plus constructif : de la description d'un ensemble Schnorr P-mesurable A de mesure non nulle, la preuve du théorème fournit automatiquement un algorithme calculant un élément de A.

Nous avons mentionné que la loi forte des grands nombres entrait dans le cadre de Schnorr. En fait, on peut prouver un analogue du lemme de Borel-Cantelli dans ce cadre, en ajoutant une hypothèse.

Lemme 13 (Borel-Cantelli)**.** *Soient A_i des ensembles uniformément Schnorr P-mesurables tels que $\sum_i P(A_i)$ est fini et calculable. Alors*

i) $\limsup A_i = \bigcap_i \bigcup_{j>i} A_j$ *est Schnorr P-mesurable, et de mesure nulle*

ii) $S_P \subseteq (\limsup A_i)^c$.

Ce résultat admet un joli corollaire concernant les nombres normaux. Rappelons qu'un nombre réel x est normal dans une base b si dans la suite des chiffres de l'expansion de x en base b, tout motif de longueur k apparaît avec une fréquence b^{-k} ; ainsi, dans l'écriture décimale d'un nombre x normal, le bloc de décimales 7251 apparaît avec une fréquence égale à $1/10000$. La loi des grands nombres assure que pour une base donnée b, l'ensemble des nombres normaux en base b est de mesure 1 mais paradoxalement, on connaît peu d'exemples explicites de nombres normaux ; par exemple, la normalité de π en base 10, bien que très vraisemblable (une étude statistique des milliards de décimales de π déjà calculées le laisse penser), n'a pas été démontrée.

Corollaire 14. *Il existe un réel calculable absolument normal, c'est-à-dire normal dans toutes les bases.*

En effet, en utilisant le lemme 13 et les propriétés de $\mathscr{S}_P$, on montre facilement que l'absolue normalité est une propriété Schnorr λ-mesurable, où λ est la mesure de Lebesgue. Puisque cette propriété a probabilité 1, le théorème 12 fournit un algorithme calculant un réel absolument normal.

Le théorème 12 a d'autres conséquences inattendues : il permet de « faire des probabilités » sur l'ensemble des points calculables qui est, rappelons-le, dénombrable. Nous partons d'un espace de probabilité $(\mathbb{R}, \mathscr{B}, P)$, puis nous considérons l'ensemble $\mathbb{R}_c$ des réels calculables et la collection $\mathscr{B}_c$ des restrictions des ensembles Schnorr P-mesurable à $\mathbb{R}_c$. En utilisant le théorème 12, il est facile de montrer que la fonction P_c qui à $A_c = A \cap \mathbb{R}_c$ (où A est Schnorr P-mesurable) associe $P(A)$ est bien définie. En effet, si A et B sont Schnorr P-mesurables et $A_c = B_c$, alors $A \triangle B$ est Schnorr P-mesurable ($\mathscr{S}_P$ est une algèbre booléenne) et $(A \triangle B)_c = A_c \triangle B_c = \emptyset$ donc $P(A \triangle B) = 0$.

Ainsi on peut « faire des probabilités » sur $(\mathbb{R}_c, \mathscr{B}_c, P_c)$, ce qui a une conséquence intéressante. Un ensemble dénombrable n'admet pas de mesure uniforme. Mais si P est la mesure de Lebesgue λ sur l'intervalle réel $[0,1]$, λ_c est donc une « pseudo-mesure » uniforme sur l'ensemble des réels calculables : chaque point calculable est de poids nul pour P_c mais l'ensemble des réels calculables de $[0,1]$, bien que dénombrable, est de poids 1 pour P_c.

2.3 Les réels aléatoires via la complexité de Kolmogorov

Nous avons jusqu'à présent proposé deux approches de la théorie effective de l'aléatoire : une pour les suites binaires finies (la complexité de Kolmogorov) et une pour les réels (la théorie effective de la mesure). Est-il possible de concilier les deux ? Après tout, un réel $\alpha \in [0,1]$ peut être identifié avec la suite $a_0 a_1 a_2 a_3 \ldots$ des chiffres de son expansion binaire. Ne pourrait-on pas alors dire que α est

aléatoire si tous les segments initiaux de la suite des a_i ont une complexité de Kolmogorov maximale ? Plus précisément, on pourrait dire que α est aléatoire si

$$K(a_0a_1a_2\dots a_n) \geq n - O(1) \tag{2}$$

Hélas, ainsi que l'a démontré Martin-Löf, cette définition ne convient pas, puisqu'aucune suite binaire infinie $a_0a_1a_2\dots$ ne satisfait la condition (2), ce qui peut sembler rendre impossible une caractérisation de l'aléatoire (pour la mesure de Lebsegue) sur les réels par la complexité de Kolmogorov. Cependant, il est tout de même possible d'obtenir une telle caractérisation, mais en utilisant une variante de la complexité de Kolmogorov, appelée *complexité de Kolmogorov préfixe*. La complexité de Kolmogorov préfixe (notée KP) est définie de la même manière que la complexité de Kolmogorov classique, à ceci près que l'on impose une restriction aux machines de Turing.

Définition 15. *Une machine de Turing M est dite* préfixe *si pour toute paire de suite binaires finies x, y telles que x est un préfixe de y, et que M s'arrête sur l'entrée x, alors M s'arrête également sur l'entrée y et $M(x) = M(y)$.*

Ici, préfixe s'entend au sens usuel ; par exemple 00 est un préfixe de 0010. Il est possible de montrer un analogue de la Proposition 2 pour les machines de Turing préfixes, à savoir l'existence d'une machine de Turing préfixe V, optimale dans la classe des machines préfixes. On peut alors définir pour tout x :

$$KP(x) = K_V(x) = \min\left\{|p| \ : \ V(p) = x\right\}$$

La complexité de Kolmogorov préfixe est en général plus grande que la complexité simple, et en réalité n'en diffère qu'à un terme logarithmique près. Mais elle se trouve être parfaitement adaptée pour caractériser l'aléatoire au sens de Martin-Löf, comme le montre le théorème suivant.

Théoreme 16 (Levin-Schnorr)**.** *Soit α un élément de $[0,1]$. Soit $0, a_0a_1a_2\dots$ l'écriture de α en base 2. On a l'équivalence entre les assertions suivantes.*
(i) α est aléatoire au sens de Martin-Löf pour la mesure de Lebesgue
(ii) $KP(a_0a_1\dots a_n) \geq n - O(1)$
(iii) $\lim_n KP(a_0a_1\dots a_n) - n = \infty$

Remarques.

– *Le théorème ci-dessus peut en fait se généraliser à l'aléatoire au sens de Martin-Löf pour une mesure μ quelconque, en remplaçant (ii) par*

$$(ii') \quad KP(a_0a_1\dots a_n) \geq -\log \mu(X_{a_1a_2\dots a_n}) - O(1)$$

où $X_{a_1a_2\dots a_n}$ est l'ensemble des réels dont l'expansion binaire commence par $a_0\dots a_n$, et en remplaçant similairement (iii) par

$$(iii') \quad \lim_n KP(a_0a_1\dots a_n) + \log \mu(X_{a_1a_2\dots a_n}) = \infty$$

– *Malgré l'équivalence $(ii) \Leftrightarrow (iii)$, la borne $n - O(1)$ de l'assertion (ii) est la meilleure possible (voir [Bie08]).*

– *La complexité de Kolmogorov (préfixe ou non) est en revanche inadaptée pour caratériser l'aléatoire au sens de Schnorr. Ainsi, il existe des réels Schnorr aléatoires (pour la mesure de Lebesgue) dont l'expansion binaire a une complexité de Kolmogorov très petite, tandis que certains réels ont une expansion binaire de complexité de Kolmogorov préfixe quasi-maximale mais ne sont pas Schnorr-aléatoires.*

Pour en savoir plus

Nous espérons à travers ces quelques pages avoir donné envie au lecteur d'en apprendre plus sur cette riche théorie qu'est celle de l'aléatoire effectif. Pour ce qui est de la complexité de Kolmogorov, le livre de référence est celui de Li et Vitanyi [LV08]. Pour les liens entre calculabilité, complexité, et aléatoire, on pourra consulter le livre de Nies [Nie09], ainsi que celui de Downey et Hirschfeldt [DHar] qui paraîtra très prochainement. Signalons également une remarquable introduction à la complexité de Kolmogorov, plus complète que le présent article, de Grigorieff et Ferbus [GF04]. Pour plus de détails sur une théorie générale de l'aléatoire effectif dans des espaces de probabilité, et ses interactions avec la théorie ergodique et les systèmes dynamiques, se référer aux thèses de doctorat de Hoyrup [Hoy08] et Rojas [Roj08]. Enfin, une autre approche possible de la théorie effective de l'aléatoire (que nous n'avons pas présentée dans cet article) est possible via la théorie des jeux et martingales (proposée par Ville [Vil39] et Schnorr [Sch71]) ; cette dernière est traitée en détail dans la thèse de doctorat de Bienvenu [Bie08].

Références

[Bie08] Laurent Bienvenu. *Caractérisation de l'aléatoire par les jeux : imprédictibilité et stochasticité*. Thèse de doctorat, Université de Provence, 2008.

[Cha66] Gregory Chaitin. On the length of programs for computing finite binary sequences. *Journal of the Association for Computing Machinery*, 13(547-569), 1966.

[Cha75] Gregory Chaitin. A theory of program size formally identical to information theory. *Journal of the Association for Computing Machinery*, 22 :329–340, 1975.

[DHar] Rodney Downey and Denis Hirschfeldt. *Algorithmic randomness and complexity*. Springer, to appear.

[For09] Lance Fortnow. Computational complexity blog : A Kolmogorov complexity proof of the Lovász local lemma (post du 2 juin 2009). `http ://blog.computationalcomplexity.org`, 2009.

[GF04] Serge Grigorieff and Marie Ferbus. Is randomness native to computer science ? *Current Trends in Theoretical Computer Science*, 2 :141–179, 2004.

[Hoy08] Mathieu Hoyrup. *Calculabilité, aléatoire et théorie ergodique sur les espaces métriques*. Thèse de doctorat, Université de Paris VII, 2008.

[Kol65] Andrei Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1 :1–7, 1965.

[LV08] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Texts in Computer Science. Springer-Verlag, New York, 3rd edition, 2008.

[ML66] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9 :602–619, 1966.

[Mos09] Robin Moser. A constructive proof of the Lovász local lemma. In *Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 343–350. ACM, 2009.

[Nie09] André Nies. *Computability and randomness*. Oxford Logic Guides. Oxford University Press, 2009.

[Roj08] Cristóbal Rojas. *Randomness and ergodic theory : an algorithmic point of view*. Thèse de doctorat, Ecole Polytechnique, Paris, 2008.

[Sch71] Claus Schnorr. *Zufälligkeit und Wahrscheinlichkeit*, volume 218 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin-Heidelberg-New York, 1971.

[Sol64] Ray Solomonoff. A formal theory of inductive inference i and ii. *Information and Control*, 7 :1–22 and 224–254, 1964.

[Vil39] Jean Ville. *Etude critique de la notion de collectif*. Gauthiers-Villars, Paris, 1939.