

4. Tableau de résultats

Hector Zenil a fait fonctionner les 11019960576 machines de Turing à quatre états et en a déduit, par un énorme calcul sur ordinateur, la fréquence des séquences finies produites. Il a obtenu des valeurs proches de la mesure de Solomonoff-Levin $SL(s)$ reliées à la complexité de Kolmogorov par la formule $K(s) = -\log_2 SL(s)$. Puis H. Zenil a classé par complexité croissante les valeurs approchées de $K(s)$ des séquences courtes.	0	2,285794	1111	8,644805	00000	11,791904	10110	12,755680
	1	2,285794	0001	9,019632	11111	11,791904	01010	12,830138
	00	3,287643	0111	9,019632	00001	12,513528	10101	12,830138
	01	3,287643	1000	9,019632	01111	12,513528	00110	12,946395
	10	3,287643	1110	9,019632	10000	12,513528	01100	12,946395
	11	3,287643	0101	9,029529	11110	12,513528	10011	12,946395
	000	5,735457	1010	9,029529	00010	12,554899	11001	12,946395
	111	5,735457	0010	9,040613	01000	12,554899	00101	12,979816
	001	5,791928	0100	9,040613	10111	12,554899	01011	12,979816
	011	5,791928	1011	9,040613	11101	12,554899	10100	12,979816
	100	5,791928	1101	9,040613	00100	12,694473	11010	12,979816
	110	5,791928	0110	9,263820	11011	12,694473	00011	13,177551
	010	5,867773	1001	9,263820	01001	12,755680	00111	13,177551
	101	5,867773	0011	9,276355	01101	12,755680	11000	13,177551
	0000	8,644805	1100	9,276355	10010	12,755680	11100	13,177551

six, le calcul de $SL(s)$ mené en faisant fonctionner toutes les machines de Turing à trois états et en examinant ce qu'elles produisent indique que les deux séquences les plus probables (donc les moins complexes) sont à égalité 000000 et 111111, ce qu'on attendait. Viennent ensuite dans l'ordre les séquences suivantes :

- 000001 à égalité avec 100000, 111110 et 011111 ;
- 000100 à égalité avec 001000, 110111 et 110111 ;
- 001001 à égalité avec 100100, 110110 et 011011 ;
- 010110 à égalité avec 101001, 100101 et 011010.

Un tel classement est assez subtil, mais naturel : placer un 1 derrière cinq 0 est un peu plus simple que de placer un 1 avec trois 0 devant et deux derrière. Si vous en doutez, comparez la longueur des deux descriptions que je viens de formuler :

- « un 1 derrière cinq 0 »
- « un 1 avec trois 0 devant et deux derrière ».

Cependant, le classement obtenu avec les machines de Turing à trois états ne donne de résultats que pour un petit nombre de séquences, celles qui sont produites par au moins une machine. Il y a 128 séquences différentes ainsi produites. Cela laisse toutes les autres sans probabilité, ce qui revient à considérer que ces autres séquences sont infiniment complexes.

Pour obtenir plus de précision et surtout pour qu'un plus grand nombre de séquences soient classées, il faut sauter un pas et faire fonctionner toutes les 11 019 960 576 machines de Turing à quatre états.

Pour chacune, on doit déterminer ce qu'elle fait en simulant son fonctionnement. Il y a deux possibilités : soit la machine produit une séquence de 0 et de 1 et s'arrête, et alors elle contribuera au calcul de l'approximation de $SL(s)$ (comme un tirage de loterie) ; soit elle ne s'arrête jamais et la machine ne sert à rien. Malgré l'utilisation de plusieurs astuces pour écourter le calcul, les programmes de H. Zenil ont dû fonctionner neuf jours durant, à raison de plusieurs milliers de machines par seconde, pour obtenir cette nouvelle version approchée de $SL(s)$.

Calcul colossal et castors affairés

Le fait que de nombreuses machines calculent indéfiniment sans jamais rien produire a pu être maîtrisé grâce à la connaissance qu'on a du *Castor affairé* à quatre états. On dénomme *Castor affairé* à n états, la machine de Turing à n états qui calcule le plus longtemps avant de s'arrêter. Le *Castor affairé* à trois états calcule durant 21 étapes et donc, toute machine ayant trois états calcule moins de 22 étapes, ou alors calcule indéfiniment. Le *Castor*

affairé à quatre états calcule durant 107 étapes. Le *Castor affairé* à cinq états n'a pas encore été identifié avec certitude, mais on pense qu'il calcule durant 47 176 870 étapes.

Comme le *Castor affairé* à quatre états calcule 107 étapes, toute machine de Turing à quatre états qui ne s'est pas arrêtée au bout de 107 étapes ne s'arrêtera jamais. Lors du calcul de $SL(s)$, on peut interrompre le fonctionnement d'une telle machine, puisqu'il est certain qu'elle ne produira aucune séquence. Cette prise en compte est évidemment essentielle pour ne pas perdre de temps en faisant fonctionner inutilement des machines qui ne contribuent pas au résultat recherché.

Les résultats obtenus avec les machines à quatre états concernent cette fois une volumineuse famille de séquences binaires. Précisément 1 824 séquences différentes sont produites par les machines de Turing à quatre états et ce sont donc 1 824 séquences dont on a su évaluer la complexité par utilisation de l'approximation ainsi calculée de $SL(s)$ et de la formule $-\log_2 SL(s)$ qui détermine leur complexité de Kolmogorov approchée. Notons que cette complexité de Kolmogorov pour séquences courtes est un nombre réel pas forcément entier. C'est en fait un avantage, puisque cela permet d'avoir un classement plus fin avec moins d'*ex aequo*.

Voici quelques observations sur les résultats obtenus :

— Pour les séquences prises en exemple au début de l'article, tout se passe bien. La séquence 0000000 est plus probable que 0001000, qui elle-même est plus probable que 1001101. Les complexités déduites des probabilités pour ces trois séquences sont conformes à ce que l'intuition exigeait.

— Chaque suite ayant une longueur comprise entre 1 et 8 est produite au moins une fois et le calcul approché avec les machines de Turing à quatre états lui attribue une probabilité non nulle, donc une complexité finie.

— Pour les 128 séquences déjà classées par les machines à trois états, le classement obtenu avec les machines à quatre états reste le même, sauf dans 17 cas. En général, les déplacements opérés entre les deux classements conduisent à des positions plus conformes à l'intuition. En considérant des machines à quatre états, on améliore ce qu'on avait obtenu avec les machines à trois états : c'est une validation de la méthode.

— Les séquences produites les plus longues ont pour longueur 16. En voici une : 1101010101010101. Bien sûr, ce sont des suites assez peu complexes, car les suites de longueur 16 les plus complexes ne sont pas produites par une machine de Turing à quatre états.

— En général, plus une séquence est longue, plus sa complexité est grande, mais ce n'est pas une règle absolue : par exemple, la séquence 0000000000 (dix fois le 0) est évaluée moins complexe que la séquence de neuf bits 000011100.

— De petites imperfections persistent. En passant de trois à quatre états, on a notablement progressé, mais pour avoir un classement parfait incluant par exemple toutes les séquences de longueur allant jusqu'à dix, il faudrait maintenant mener un calcul complet avec les machines à cinq états, à la limite de ce qu'on peut envisager aujourd'hui. On espère l'entreprendre prochainement. Bien sûr, il ne correspondra qu'à une marche de plus dans une quête infinie.

Pour disposer d'une méthode pratique qui attribue une complexité à toute séquence finie, il faut accepter d'associer plusieurs idées. Trois idées semblent suffire :

(a) Pour les séquences courtes (jusqu'à huit ou dix bits, peut-être un peu plus), la méthode des machines de Turing ;

(b) Pour les séquences de longueur moyenne (de dix bits à quelques centaines de bits) un découpage en tranches ramenant au cas précédent ;

(c) Pour les séquences plus longues, une méthode à base d'algorithmes de compression.

Recollement délicat

Le travail est délicat et on tombera sur des cas où l'évaluation sera peu satisfaisante. Mais le principe général semble acquis et tout n'est plus qu'une question de perfectionnement et d'efforts : prises en compte de machines de Turing ayant plus d'états ; utilisation simultanée d'une multitude d'algorithmes de compression, etc. Ces efforts n'auront jamais de fin, mais nous disposons maintenant de moyens pour le calcul de la complexité de Kolmogorov des séquences binaires de toutes les longueurs, moyens qu'on adaptera aux séquences plus générales de chiffres décimaux, de lettres, de nombres entiers, etc.

Aujourd'hui, une grande variété de méthodes spécifiques, fondées sur des principes particuliers et bien souvent *ad hoc* sont utilisées pour calculer et comparer des complexités d'objets discrets (voir deux exemples sur la figure 4). C'est le cas en informatique dans le domaine des algorithmes pour suites pseudo-aléatoires, en traitement d'images, en génétique, en linguistique, en psychologie, etc. Les avancées du programme de recherche décrit ici mettront à disposition de tous une méthode générale de calcul de la complexité de Kolmogorov qui, en plus d'être homogène (car l'unité de mesure est toujours le bit d'information), sera fondée sur ce qui est reconnu comme la théorie universelle de l'information. Cette théorie longtemps considérée comme inutilisable (du fait de l'existence de nombreux énoncés indécidables concernant $K(s)$) se révèle au fur et à mesure des années praticable et utile. Il est certain pourtant qu'on est loin d'en avoir tiré toutes les applications possibles. ■

L'AUTEUR



Jean-Paul DELAHAYE est professeur à l'Université de Lille et chercheur au Laboratoire d'informatique fondamentale de Lille (LIFL).

✓ BIBLIOGRAPHIE

H. Zenil, Pages Internet donnant accès aux articles cités ci-dessous : <http://www.mathrix.org/zenil/>

H. Zenil et J.-P. Delahaye, An algorithmic information theoretic approach to the behaviour of financial markets, themed issue on « Nonlinearity, Complexity and Randomness », *Journal of Economic Surveys*, 2011.

H. Zenil, Compression-based investigation of the dynamical properties of cellular automata and other systems, *Journal of Complex Systems*, vol. 19(1), 2010 : <http://arxiv.org/abs/0910.4042>

J.-P. Delahaye et H. Zenil, A glance into the structure of algorithmic complexity : Numerical evaluation of the complexity of short strings by small Turing machines, 2011 : <http://arxiv.org/pdf/1101.4795>

J.-P. Delahaye et H. Zenil, On the algorithmic nature of the world, dans G. Dodig-Crnkovic et M. Burgin (éds.), *Information and Computation*, World Scientific, 2010.

J.-P. Delahaye et H. Zenil, On the Kolmogorov-Chaitin complexity for short sequences, *Randomness and Complexity : From Leibniz to Chaitin* (C.S. Calude, éd.), World Scientific, pp. 123-130, 2007.

J.-P. Delahaye, Classer musiques, images, textes et génomes, *Pour la Science*, pp.90-95, mars 2004.