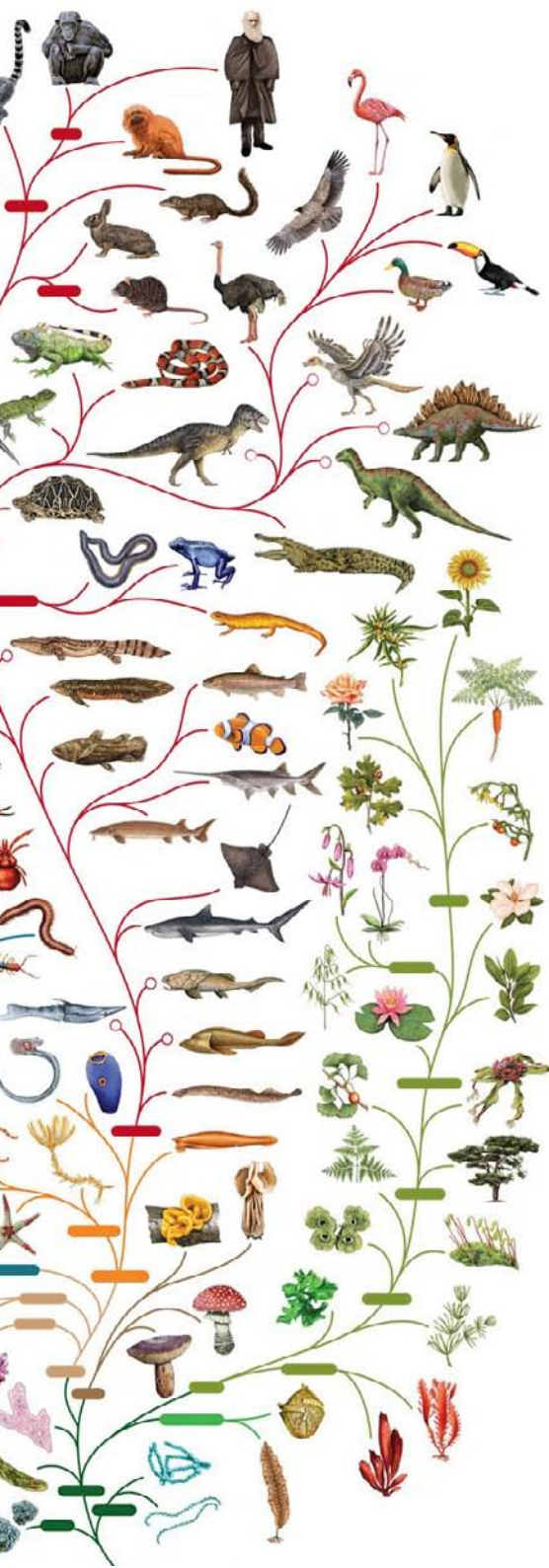




profondeur logique d'un système en évolution vérifie la loi de croissance lente (d'après laquelle la complexité structurelle ne peut augmenter rapidement), ce qui constitue un argument fort en faveur de sa définition. Cette idée s'appliquerait en particulier à l'évolution des organismes vivants.

démontrée en 1936 par Alan Turing et qui concerne l'arrêt des programmes : aucune méthode d'analyse ne permet de savoir à l'avance si un programme va se mettre à boucler.

Cependant, il ne faut pas tirer de cette impossibilité d'un calcul exact systématique pour tout s de $P(s)$ la conclusion qu'on ne peut pas évaluer $P(s)$. Les algorithmes de compression de données sont des outils qui calculent des versions approchées de s^* et leur utilisation permet l'évaluation de $P(s)$.

De la même façon, la complexité de Kolmogorov de la suite s , qui est la longueur (mesurée en bits) de s^* , n'est pas calculable, mais, grâce aux algorithmes de compression de données, elle est évaluable. Cette possibilité a permis des avancées dans la classification automatique des textes, des partitions musicales et des séquences génétiques. La non-calculabilité n'est pas un obstacle rédhibitoire : nous faisons des mathématiques tout en sachant qu'il n'y a pas de système parfait d'axiomes qui élimine toutes les propositions indécidables.

Complexité aléatoire et complexité structurelle

La complexité de Kolmogorov d'une suite, notion très importante en informatique théorique (et maintenant en physique et en biologie), ne pouvait pas mesurer la complexification évoquée en cosmologie et en biologie. En effet, une suite aléatoire où chaque élément est tiré à pile ou face a une complexité de Kolmogorov maximale (parmi les suites de même longueur) alors que, nous l'avons déjà remarqué, aucune structure n'y est présente en général.

La complexité de Kolmogorov mesure le « contenu en information » d'une suite s , c'est-à-dire la quantité minimale d'information qu'il faut mémoriser pour être capable de reproduire s (même si cette information est sans intérêt, comme c'est le cas pour une suite de tirages à pile ou face). La complexité structurelle, que la profondeur logique de Bennett veut évaluer, est liée à la richesse d'organisation, au nombre de corrélations et d'interactions apparentes ou cachées entre parties de s , à la quan-

tité d'interdépendances des parties, toutes choses absentes dans une suite aléatoire de 0 et de 1.

Cette distinction que nous faisons tous sans en avoir nécessairement conscience entre complexité aléatoire (ou contenu incompressible d'information que mesure la complexité de Kolmogorov, la taille de s^*) et complexité structurelle (celle que Ch. Bennett veut évaluer, et qui en première approximation est le temps de calcul de s^*) est essentielle pour que puisse progresser une théorie générale de la complexité. C'est faute de l'avoir comprise plus tôt et de s'être aperçu que l'impossibilité d'un calcul exact des versions formelles de ces notions (complexité de Kolmogorov et profondeur logique) n'interdisait pas de les utiliser, que l'on a cherché – et que l'on cherche encore parfois – à formuler des définitions particulières de la complexité qui ne s'appliquent que dans des contextes limités.

Des versions mathématiques plus avancées de la profondeur logique de Bennett ont été proposées. Elles ne se contentent pas de faire intervenir le programme le plus court produisant s , mais tous les programmes produisant s qui ne peuvent être compressés de plus de k bits ou même tous les programmes produisant s , mais en accordant plus de poids aux programmes les plus courts qui sont considérés comme des origines plus probables de l'objet s . Ces versions ont de meilleures propriétés que $P(s)$, dont celle d'autoriser la démonstration d'une « loi de croissance lente » [la complexité structurelle ne peut pas croître rapidement]. Malheureusement, ces définitions sont plus délicates à mettre en application que la définition de base $P(s)$.

Une série d'avancées dans la compréhension et l'utilisation de la profondeur logique de Bennett ont eu lieu depuis sa définition en 1977. Il faut les voir comme la confirmation que l'idée de Ch. Bennett est pertinente et que, même si rien n'est très facile dans ce domaine soumis à de nombreuses « indécidabilités », il faut poursuivre son étude. La profondeur de Bennett, ne serait-ce que d'une manière abstraite, nous aide à mieux cerner la complexité dont notre compréhension intuitive

Évaluation expérimentale de $K(s)$ et $P(s)$

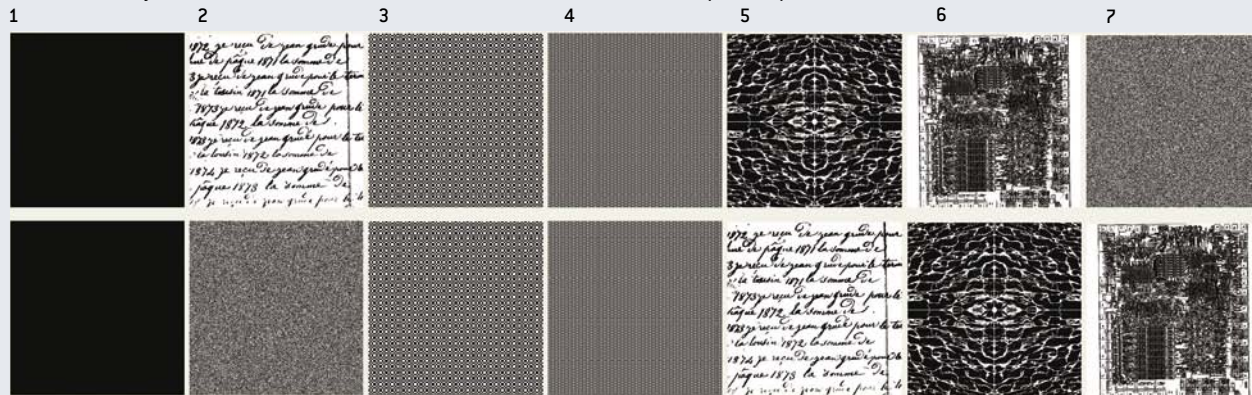
Un programme de compression de données permet d'évaluer la complexité de Kolmogorov et la profondeur logique de Bennett. La version compressée d'un fichier doit être vue comme un court programme engendrant le fichier : sa taille indique une valeur approchée de la complexité de Kolmogorov, et le temps nécessaire à la décompression du fichier est assimilable au temps de calcul pour reproduire le fichier initial, donc à une évaluation de sa profondeur logique. Les expériences ont été réalisées par Hector Zenil, Cédric Gaucherel et l'auteur (J.-P. Delahaye).

Sept images de même format ont été compressées avec un algorithme de compression sans perte : l'image retrouvée après décompression est exactement celle avant compression. La première série indique le classement des images par ordre croissant de taille du fichier compressé, donc par complexité de Kolmogorov $K(s)$ croissante. Sans surprise, on trouve que l'image toute noire (image 1) a le contenu en information le plus petit, et que l'image composée de pixels tirés aléatoirement (image 7) est celle qui exige la plus grande quantité de mémoire ($K(s)$ maximal). Les autres

images sont classées à peu près comme on s'y attend : le manuscrit (image 2) demande assez peu de mémoire, car il y a beaucoup de blanc ; un réseau périodique (image 3) ; une courbe de Peano (image 4) ; une image irrégulière, mais avec deux axes de symétrie (image 5) ; un microprocesseur (image 6).

La seconde série d'images reprend les mêmes sept images, mais cette fois par ordre croissant de temps de décompression, ce qui donne des valeurs approchées de la profondeur logique de Bennett $P(s)$, donc un classement par complexité structurelle croissante.

L'image 7 du premier classement, qui est parfaitement aléatoire, est maintenant parmi les premières, conformément à l'idée qu'un hasard parfait est totalement sans structure. Le microprocesseur est bien identifié comme le plus complexe structurellement. La courbe de Peano est, sans surprise, toujours considérée comme contenant des structures un peu plus riches que le motif périodique. Le texte écrit et le motif symétrique changent de position, donnant au total un classement compatible à ce qui, intuitivement, correspond à une complexité structurelle croissante.



doit maintenant devenir mathématique et scientifique.

Parmi les tentatives de définition d'une mesure de complexité structurelle, certaines, issues de la théorie du calcul, sont différentes de celle de Ch. Bennett. Elles sont en compétition avec l'idée de Ch. Bennett et il est donc nécessaire de comprendre s'il s'agit de notions liées.

La notion de « sophistication » introduite par l'Israélien Moshe Koppel et le Français Henri Atlan, et la notion de « complexité effective » des Américains Seth Lloyd et Murray Gell-Mann se fondent sur l'idée d'une séparation possible entre la composante aléatoire et la composante organisée d'un objet. L'idée des définitions proposées par ces chercheurs sous des formes mathématiques qui en masquent parfois le côté naturel est la suivante. Quand on examine

un programme produisant un objet, il est souvent possible d'y reconnaître deux parties : l'une représente ce qui fixe et décrit la structure de l'objet, l'autre est ce qui habille la structure par des éléments de moindre importance, voire aléatoires.

La partie non aléatoire

Considérons par exemple une suite finie de paires de chiffres binaires 00 ou 11 dont la succession est aléatoire :

$s = 00\ 11\ 11\ 00\ 00\ 00\ 11\ 00\ 11\ 00\ 00\ 11\ 11$. Le plus court programme s^* produisant cette suite comportera deux parties assez faciles à identifier, car il sera de la forme : « Imprimer, en doublant chaque chiffre, la suite 011000101001011. »

La structure régulière de s est exprimée par la partie « Imprimer en doublant chaque

chiffre ». La composante désordonnée se trouve, quant à elle, dans la partie descriptive et incompressible de 0 et de 1 qui termine le programme.

Lorsqu'on aura affaire à une suite s plus structurée que celle de notre exemple, la longueur de la partie non aléatoire sera supérieure. C'est cette longueur qui, selon M. Koppel, H. Atlan, S. Lloyd et M. Gell-Mann, mesure la richesse en organisation, la complexité structurelle de la suite s .

La partie du programme où l'on a inscrit la séquence 011000101001011 est d'autant plus longue que la séquence comporte beaucoup d'éléments aléatoires. C'est la partie aléatoire de la séquence et sa longueur est, en gros, la complexité de Kolmogorov de s .

Des difficultés techniques rendent délicate la formulation mathématique de cette idée de séparation en deux parties d'un pro-