

On comprend sans mal qu'une machine simple, avec peu d'états et peu d'instructions, placée sur un ruban couvert uniquement de 0, va soit s'engager dans un calcul infini (par exemple parcourir le ruban en changeant les 0 en 1, ou osciller entre deux cases sans rien y faire), soit calculer un petit moment puis s'arrêter. Une machine simple produit soit un calcul infini, soit un calcul court, et on verra que c'est justement ce qui pose un défi aux mathématiciens et les expose aux formes les plus élémentaires d'indécidabilité.

En 1962, le mathématicien hongrois Tibor Radó s'est intéressé au nombre maximal, noté $s(n)$, de mouvements qu'exécute une machine de Turing à n états avant de s'arrêter (dans le cas où elle s'arrête).

Les castors affairés

Les machines M à n états qui atteignent ce maximum sont dénommées « castors affairés ». Le nombre $s(n)$ est un nombre parfaitement défini, puisqu'il n'y a qu'un nombre fini de machines à n états n'utilisant que les symboles 0 et 1. Le nombre de machines à n états a d'ailleurs été déterminé : il est égal à $(4n + 4)^{2^n}$.

Pour calculer $s(n)$ pour un n donné, il suffit de faire fonctionner toutes les machines à n états en repérant celles qui calculent le plus longtemps et s'arrêtent. Avec les incroyables ordinateurs dont nous disposons, il semblerait facile d'obtenir rapidement les valeurs de $s(n)$.

Grave erreur ! Aujourd'hui, malgré de considérables efforts, on ne connaît que $s(1) = 1$, $s(2) = 6$, $s(3) = 21$ et $s(4) = 107$. On sait aussi que $s(5) \geq 47\,176\,870$, et que $s(6) \geq 7,4 \times 10^{36\,534}$, mais personne n'est certain que ces valeurs sont définitives. Pourquoi est-il si difficile d'évaluer $s(n)$?

La raison est double. D'abord, le nombre de machines à n états augmente très rapidement quand n augmente et on ne sait guère faire mieux, pour calculer $s(n)$, que de les prendre une à une. Ensuite, et c'est encore plus ennuyeux, les machines qui ne s'arrêtent pas sont délicates à traiter. Il faut en effet raisonner sur leurs instructions ou leur comportement pour être sûr qu'elles ne s'arrêtent

Les castors affairés calculent l'entropie

La connaissance des valeurs de la fonction $s(n)$ de Radó, dont il vient d'être démontré que $s(1919)$ est indécidable, est utile pour certains calculs fondamentaux de la théorie de la complexité. En utilisant la valeur conjecturée comme la bonne pour $n = 5$, $s(5) = 47\,176\,870$, une équipe réunie autour de Fernando Soler-Toscano a calculé l'entropie algorithmique, qui généralise l'entropie statistique de Shannon, des suites courtes de 0 et de 1.

Cette entropie est aussi nommée complexité de Kolmogorov : pour une suite donnée, c'est la taille du plus court programme qui produit la suite. Plus une suite est complexe, plus son entropie algorithmique est grande (voir l'article indiqué en bibliographie de F. Soler-Toscano et al.). La méthode est la suivante. On fait calculer chacune des 26 559 922 791 424 (26 000 milliards) machines de Turing à 5 états jusqu'à ce qu'elle s'arrête où jusqu'à l'étape $s(5)$, ce qui signifie que la machine ne s'arrête jamais. Le nombre k de fois où est obtenue une suite donnée (par exemple la suite $t = 01010$) est directement lié à son entropie algorithmique $K(t)$ par une formule de Leonid Levin,

$K(t) \approx -\log_2[k/N(5)]$, où $N(5)$ désigne le nombre de machines à 5 états, et

donne donc une valeur approchée de l'entropie.

La formule de Leonid Levin est remarquable et très profonde. Elle signifie que les suites complexes (c'est-à-dire celles dont le plus court programme de création est long) sont aussi celles qu'une machine tirée au hasard produit avec la plus faible probabilité.

Quelques valeurs ainsi obtenues sont présentées ci-dessous.

SUITE	FRÉQUENCE	ENTROPIE
1	0,175036	2,51428
0	0,175036	2,51428
11	0,0996187	3,32744
10	0,0996187	3,32744
01	0,0996187	3,32744
00	0,0996187	3,32744
111	0,0237546	5,3962
000	0,0237546	5,3962
110	0,0229434	5,44578
100	0,0229434	5,44578
011	0,0229434	5,44578
001	0,0229434	5,44578
101	0,0220148	5,5038
010	0,0220148	5,5038
1111	0,0040981	7,03083
0000	0,0040981	7,03083
1110	0,00343136	8,187
1000	0,00343136	8,187
0111	0,00343136	8,187
0001	0,00343136	8,187

pas. Pour certaines, c'est facile, mais on tombe rapidement sur des machines dont le comportement est complexe et dont on n'arrive pas à déterminer si elles s'arrêteront et donc si elles interviennent pour la détermination de $s(n)$, ou si elles poursuivent indéfiniment leurs calculs – auquel cas elles sont sans importance pour $s(n)$. On sait par ailleurs qu'aucune méthode algorithmique générale ne permet de répondre à la question. C'est

la fameuse indécidabilité de l'arrêt démontrée par Turing en 1936 : aucun algorithme ne peut déterminer, pour toute machine de ce type placée sur un ruban ne comportant que des 0, si oui ou non elle finira par s'arrêter.

Une conséquence de ce résultat, qui est la raison de l'introduction par Radó de sa fonction $s(n)$, est que $s(n)$ n'est pas calculable par algorithme. Plus intéressant