ensembles ne concernent que des problèmes alambiqués et si complexes qu'aucun mathématicien ne s'y intéressera jamais de lui-même : s'interroger sur $s(1\,919)$ est naturel, car cela concerne la partie la plus élémentaire de la théorie mathématique du calcul…

Notons que ce record $n_0 = 1\,919$ est sans doute améliorable, car, comme le savent bien ceux qui tentent de connaître les valeurs de $s(n)$, les difficultés concrètes commencent dès $s(5)$, qui reste inconnu à ce jour. Une bonne marge de travail est donc disponible entre 5 et 1 919, marge que l'on réduira soit en progressant dans le calcul des $s(n)$, soit en réussissant à prouver l'indécidabilité d'énoncés de la forme $s(n) = k$ pour des valeurs encore plus petites que $n_0 = 1\,919$.

## La machine Z ne s'arrête jamais

La méthode utilisée par Adam Yedidia et Scott Aaronson est intéressante. L'idée la plus naturelle serait de mettre en œuvre les techniques d'arithmétisation de la syntaxe de Gödel et donc de construire une machine de Turing M qui recherche une contradiction dans ZFC en énumérant toutes les démonstrations de ZFC et en tentant d'y reconnaître une preuve de l'énoncé « 0 = 1 ». Une telle machine ne trouvera pas de contradiction, et pourtant ZFC ne pourra pas le démontrer (sinon, cela signifierait que ZFC démontre qu'elle n'est pas contradictoire, ce que le second théorème d'incomplétude de Gödel interdit). Cependant, cette méthode par l'arithmétisation de la syntaxe conduirait à une machine de Turing à plusieurs centaines de milliers d'états, voire des millions, et l'on ne démontrerait donc l'indécidabilité de $s(n) = k$ que pour un $n$ très grand.

La méthode utilisée par Scott Aaronson et Adam Yedidia évite cette explosion du nombre d'états de la machine de Turing dont ZFC ne peut prouver le non-arrêt. Elle se fonde sur des travaux récents de Harvey Friedman, de l'université d'État de l'Ohio, qui élabore depuis des années des énoncés d'arithmétiques indécidables dans ZFC ou d'autres théories. L'un d'eux concerne une propriété des graphes et cette propriété, trop abstraite pour être décrite ici, est équivalente à celle exprimant la non-contradiction de ZFC. Cette propriété s'écrit assez facilement en termes arithmétiques et la machine, notée Z, qui cherche un contre-exemple à cet énoncé est donc relativement simple comparée à celle qu'on aurait obtenue par arithmétisation de la syntaxe : elle énumère des séries d'entiers et effectue des tests sur eux. Le fait que Z ne s'arrête pas dans cette recherche de contre-exemple est nécessairement un indécidable de ZFC, sinon, encore une fois, ZFC démontrerait d'elle-même qu'elle est non contradictoire.

Soit $n_0$ le nombre d'états de Z. Puisque pour connaître $s(n_0)$, ZFC doit savoir prouver le non-arrêt de chaque machine à $n_0$ états qui ne s'arrête pas, il en résulte que ZFC ne peut pas démontrer $s(n_0) = k$ pour la bonne valeur $k$. Précisons que la propriété en question des graphes est démontrable dans une théorie un peu plus forte que ZFC, et donc qu'on a toutes les raisons de la croire vraie. Notons aussi qu'une fois leur machine Z programmée, Scott Aaronson et Adam Yedidia l'ont effectivement lancée pour voir si elle s'arrêtait ; au bout de plusieurs heures, elle ne s'était pas arrêtée.

Sans effort particulier, utiliser ce raccourci déduit des travaux de Friedman n'aurait cependant pas permis à lui seul d'arriver aux petites valeurs de $n_0$ aujourd'hui connues. Scott Aaronson et Adam Yedidia ont, et c'est la seconde idée originale de leur travail, conçu un langage informatique nommé Laconic permettant d'exprimer la propriété de Friedman de manière concise. Ils ont aussi écrit et utilisé le traducteur des programmes écrit en Laconic qui en tire des machines de Turing du type considéré par Radó. C'est ainsi qu'ils ont obtenu la machine de Turing Z à 7 918 états qui ne s'arrête jamais, mais dont ZFC ne peut pas démontrer qu'elle ne s'arrête jamais. Le passage de 7 918 états à 1 919 états s'est fait en améliorant le langage et les opérations de transformation des programmes en machines de Turing.

Saura-t-on faire encore mieux et trouver des valeurs encore plus petites que le $n_0 = 1\,919$ d'aujourd'hui ? C'est probable, et peut-être qu'à l'heure où vous lirez ce texte, le record sera déjà battu. Pour le savoir, consultez le blog de Scott Aaronson ! ■


### L'AUTEUR

J.-P. DELAHAYE est professeur émérite à l'université de Lille et chercheur au Centre de recherche en informatique, signal et automatique de Lille (CRISTAL).


### BIBLIOGRAPHIE


S. Aaronson, **The 8000th Busy Beaver number eludes ZF set theory : New paper by Adam Yedidia and me**, blog de mai 2016, www.scottaaronson.com/blog/?m=201605

H. Marxen, **List of the known Busy Beaver values**, 2016, www.drb.insel.de/ffheiner/BB/

P. Michel, **The Busy Beaver competition : A historical survey**, 2016, http://export.arxiv.org/pdf/0906.3749v4

A. Yedidia et S. Aaronson, **A relatively small Turing machine whose behavior is independent of set theory**, 2016, https://arxiv.org/pdf/1605.04343.pdf

F. Soler-Toscano *et al.*, **Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines**, *PLoS One*, vol. 9(5), e96223, 2014.

C. S. Calude et E. Calude, **Evaluating the complexity of mathematical problems : Part 1 & Part 2**, *Complex Systems*, vol. 18, pp. 267-285, 2009 et pp. 387-401, 2010.


Retrouvez la rubrique Logique & calcul sur www.pourlascience.fr