Plus précis, le second théorème d'incomplétude de Gödel établit que l'énoncé affirmant que la théorie T est non contradictoire est l'un des indécidables de T : une théorie n'a pas le pouvoir de démontrer d'elle-même qu'elle est non contradictoire… sauf si elle est contradictoire, auquel cas elle n'a aucun intérêt.

Ces formidables résultats de Gödel établissent que l'indécidabilité est toute proche des questions que l'on considère comme fondamentales, car la non-contradiction d'une théorie T est bien sûr la première des questions qu'on doit se poser quand on envisage cette théorie.

Cependant, ces énoncés ne sont simples que par leur sens : la non-contradiction d'une théorie est toujours assez complexe. On l'exprime souvent sous la forme : « Jamais une suite de formules obtenues en suivant les règles de raisonnement de la théorie ne conduira à la formule 0 = 1. » Or traduire une telle affirmation en un énoncé arithmétique exige qu'on traduise d'abord les règles de raisonnement de T, ce qui demande un travail délicat et long dénommé arithmétisation de la syntaxe et dont l'invention est due à Gödel.

Indécidabilité avec des énoncés simples ?

Pour évaluer la proximité de l'indécidabilité, les logiciens ont donc recherché des énoncés simples dont on puisse prouver l'indécidabilité. Récemment, de nouveaux résultats permettent d'évaluer le risque d'une rencontre avec un indécidable que certains mathématiciens considéraient comme improbable. Nous allons voir plus loin que l'énoncé concernant la fonction $s(n)$ de Radó, qui permettrait d'en connaître la valeur pour l'entier 1 919, est indécidable dans la puissante théorie classique des ensembles : $s(1\,919)$ est définitivement inconnaissable pour le mathématicien qui voudrait mener tout son travail avec la théorie classique des ensembles.

Pour expliquer ces nouveaux résultats, nous utiliserons les machines introduites par Alan Turing en 1936. Une machine de Turing est un automate qui peut se trouver dans un nombre fini d'états, par exemple l'état A, B ou C si ce nombre est 3. L'automate dispose d'un ruban découpé en cases. Sur le ruban, la tête de lecture-écriture de la machine lit et écrit des symboles qui viennent éventuellement effacer des symboles déjà inscrits. La machine n'a accès qu'à une seule case à la fois et déplace sa tête de lecture-écriture d'une case dans un sens ou dans l'autre. Elle parcourt ainsi le ruban pour y opérer un calcul.

Le programme de la machine est une liste finie d'instructions. Chacune est du type [A 0→1 droite B] qui signifie : « Quand je suis dans l'état A, que je lis 0 dans la case que je suis en train d'examiner, alors j'écris 1 dans cette case, je déplace la tête de lecture-écriture d'une case vers la droite et je passe dans l'état B. »

Le castor affairé à 3 états

Un castor affairé à *n* états est une machine de Turing à *n* états qui fonctionne le plus longtemps possible (pour une machine à *n* états) et qui s'arrête. Le programme du castor affairé à 3 états comporte 6 instructions. Il a été trouvé par Tibor Radó. Il écrit cinq « 1 » successifs et fonctionne 21 fois avant de s'arrêter. Aucune machine de Turing à 3 états ne fonctionne plus de 21 fois avant de s'arrêter, sauf si elle ne s'arrête jamais.

Les 6 instructions du programme du castor affairé à 3 états sont les suivantes :

[A 0→1 droite B],
[A 1→1 droite ARRÊT],
[B 0→1 gauche B],
[B 1→0 droite C],
[C 0→1 gauche C]
[C 1→1 gauche A].

Voici comment il faut lire ce programme et le faire fonctionner. La machine *(comme celle schématisée page ci-contre)* est posée sur un ruban recouvert de 0 doublement infini (vers la droite et vers la gauche). Supposons-la dans l'état A, et que sa tête de lecture-écriture lit un 0. L'instruction à appliquer est donc la première instruction [A 0→1 droite B] de son programme, qui indique que le 0 lu doit être remplacé par un 1, que la tête de lecture-écriture doit se déplacer d'une case vers la droite, et que le nouvel état de la machine doit être B. On a ainsi, en supposant que la machine débute sa lecture à la quatrième case et en marquant en rouge la position de la tête de lecture-écriture :

État initial
A - 000**0**0000000000000000000000...,
État atteint
B - 0001**0**0000000000000000000000...
La machine est maintenant dans l'état B et lit un 0. Elle applique donc la 3e instruction [B 0→1 gauche B], ce qui conduit à la situation :
B - 000**1**10000000000000000000000...

Le calcul se poursuit alors. Voici l'ensemble du calcul jusqu'à l'arrêt :

A - 000**0**0000000000000000000000...
B - 0001**0**0000000000000000000000...
B - 000**1**10000000000000000000000...
C - 0000**1**0000000000000000000000...
A - 0000**1**0000000000000000000000...
B - 0001**1**0000000000000000000000...
C - 0001**0**0000000000000000000000...
C - 0001**0**1000000000000000000000...
C - 000**1**11000000000000000000000...
A - 000**1**11000000000000000000000...
B - 001**1**11000000000000000000000...
C - 0010**1**1000000000000000000000...
A - 0010**1**1000000000000000000000...
B - 0011**1**1000000000000000000000...
C - 0011**0**1000000000000000000000...
A - 0011**0**1000000000000000000000...
B - 0011**1**1000000000000000000000...
C - 0011**1**0000000000000000000000...
C - 0011**1**0100000000000000000000...
C - 0011**1**1100000000000000000000...
A - 0011**1**1100000000000000000000...
ARRÊT - 0011**1**1100000000000000000...

Considérons la machine M à 2 états A et B, et dont les instructions sont [A 0→0 droite A], [A 1→0 droite A] et [A 2→0 gauche B]. Quand on place la tête de lecture-écriture de M au début d'un ruban comportant par exemple 0101101020010..., alors M parcourt le ruban de gauche à droite en remplaçant les 1 par des 0, puis, arrivée au 2, elle recule d'une case et s'arrête, car il n'y a aucune instruction lui indiquant ce qu'elle doit faire quand elle se trouve dans l'état B.

Si on place cette machine sur un ruban infini ne comportant que des 0 et des 1, elle le parcourt vers la droite sans jamais s'arrêter en remplaçant les 1 par des 0. La machine de Turing peut ainsi s'engager dans un calcul infini.

L'état A dans lequel on place la machine au démarrage est l'« état de départ ». Un état comme l'état B de notre exemple, auquel ne correspond aucune instruction, est l'« état d'arrêt ».

Quand on compte les états d'une machine de Turing, on convient de ne pas compter l'état d'arrêt. La machine que nous avons donnée en exemple est, selon cette convention, une machine à un état. Sauf exception, nous ne considérerons que des calculs de machines de Turing commençant sur un ruban (potentiellement infini) entièrement couvert de 0, et nous supposerons que les machines n'utilisent que les deux symboles 0 et 1.

Les machines de Turing qui calculent le plus longtemps

Pour chaque entier positif *n*, il existe une machine de Turing à *n* états record, c'est-à-dire telle qu'aucune autre machine à *n* états n'utilisant que les symboles 0 et 1 ne calcule plus longtemps, sauf celles qui ne s'arrêtent jamais. Le nombre de mouvements de ce calcul record est noté *s*(*n*).

Les résultats connus à ce jour sont :
$s(1) = 1$; $s(2) = 6$; $s(3) = 21$ (Lin et Radó, 1965) ;
$s(4) = 107$ (Brady, 1975) ; $s(5) \geq 47\,176\,870$ (Marxen et Buntrock, 1990) ;
$s(6) \geq 7{,}4 \times 10^{36\,534}$ (Kropitz, 2010) ;

$s(7) > 10^{10^{10^{10^{10^{7}}}}}$ (Wythagoras, 2014).

Nous avons indiqué ci-dessous les programmes de quelques machines record, que l'on dénomme « castors affairés ». En écrivant un programme d'exécution sur ordinateur qui simule une machine de Turing, on vérifie facilement le nombre d'étapes (égal au nombre de déplacements de la tête de lecture-écriture de la machine de Turing).

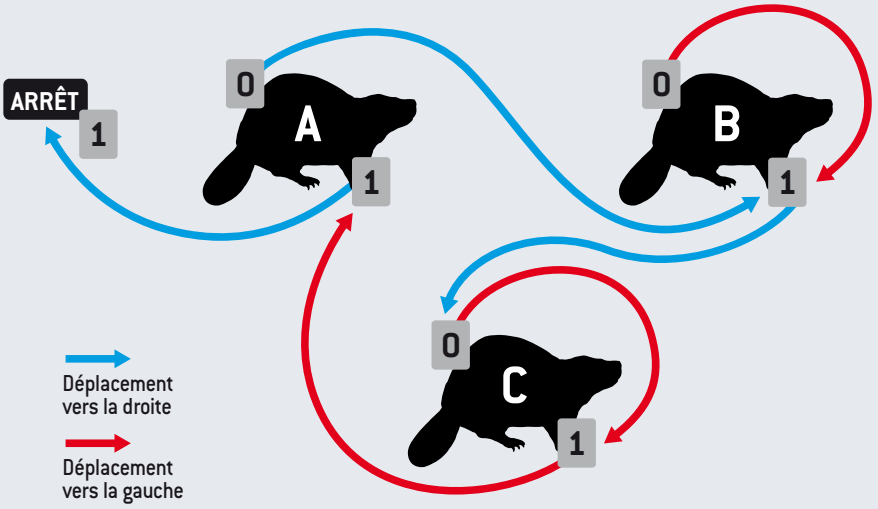
Le castor affairé à 2 états s'arrête en 6 étapes après avoir écrit quatre « 1 ». Son programme :
[A 0→1 droite B],
[A 1→1 gauche B],
[B 0→1 gauche A],
[B 1→1 droite ARRÊT].

Le castor affairé à 3 états s'arrête en 21 étapes après avoir écrit cinq « 1 ». Son programme :
[A 0→1 droite B],
[A 1→1 droite ARRÊT],
[B 0→1 gauche B],
[B 1→0 droite C],
[C 0→1 gauche C],
[C 1→1 gauche A].

Le castor affairé à 4 états s'arrête en 107 étapes après avoir écrit treize « 1 ». Son programme :
[A 0→1 droite B],
[A 1→1 gauche B],
[B 0→1 gauche A],
[B 1→0 gauche C],
[C 0→1 droite ARRÊT],
[C 1→1 gauche D],
[D 0→1 droite D],
[D 1→0 droite A].

Le castor affairé à 5 états, record actuel, s'arrête en 47 176 870 étapes après avoir écrit 4 098 « 1 ». Son programme :
[A 0→1 gauche B], [A 1→1 droite C], [B 0→1 gauche C],
[B 1→1 gauche B], [C 0→1 gauche D], [C 1→0 droite E],
[D 0→1 droite A], [D 1→1 droite D], [E 0→1 gauche ARRÊT],
[E 1→0 droite A].

En mai et juin 2016, une série de travaux ont conduit à la conclusion que le nombre *s*(1 919) était indécidable dans la théorie des ensembles. Cela signifie qu'aucun raisonnement mené dans cette théorie ne permet de démontrer la formule $s(1\,919) = k$ avec la bonne valeur de *k*, laquelle est pourtant parfaitement et clairement définie.



REPRÉSENTATION GRAPHIQUE de l'algorithme du castor affairé à 3 états.

© PLS/Cassandra Vion