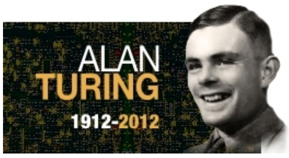

Logique et calculabilité
L'influence de Turing aujourd'hui

Laurent Bienvenu
(LIAFA – CNRS & Université Paris 7)

J'ai découvert les travaux de Turing il y a environ dix ans, dans le cours de calculabilité et logique (sobrement intitulé "Décidabilité") que donnait Jacques Mazoyer à l'Ecole Normale Supérieure de Lyon. J'avais, avant de suivre ce cours, développé un goût pour les résultats "négatifs" en mathématiques, comme par exemple l'impossiblité de la quadrature du cercle avec une règle et un compas ou l'impossibilité de la résolution par radicaux des équations algébriques de degré 5 (sans en connaître les preuves à l'époque). Prouver mathématiquement les limites des mathématiques... quel triomphe de l'esprit ! Parmi les résultats de ce type, celui qui me fascinait par dessus tout était le théorème d'incomplétude de Gödel de 1931 : pour toute théorie mathématique T "raisonnable", on peut prouver qu'il existe un énoncé φ tel que dans la théorie T, on ne peut prouver ni φ ni son contraire. En d'autres termes, certains problèmes sont, et demeureront, insolubles (comme par exemple l'hypothèse du continu : existe-t'il un ensemble strictement plus gros que l'ensemble des entiers, et strictement plus petit que l'ensemble des réels ?).

Et voilà que dans ce cours, j'apprends que Turing a demontré en 1936 : (1) que l'on ne peut pas en général vérifier à l'aide d'un programme informatique si un autre programme donné s'arrête... comment, on aurait donc également des résultats négatifs en informatique ?! et (2) que ce résultat de Turing est en fait intrinsèquement lié au théorème de Gödel ! (on peut même prouver le théorème de Gödel de façon quasi-directe à partir du théorème de Turing). Avec le recul, j'irais même aujourd'hui plus loin : pour vraiment comprendre le théorème de Gödel, il faut avant tout comprendre les idées de Turing sur la formalisation du calcul. Il paraît d'ailleurs presque surréaliste que ce lien profond entre la prouvabilité mathématique et le calcul informatique ait été découvert si tôt, à une époque où les ordinateurs (au sens où nous l'entendons aujourd'hui) n'existaient pas encore.

Loin d'être isolée, l'idée de calculabilité developpée par Turing a un impact considérable sur toute la logique mathématique. Au fil du temps, de nombreuses interactions ont émergé : avec la théorie des modèles, la théorie des ensembles, la théorie de la preuve ou les mathématiques constructives, pour ne citer qu'elles. Plus surprenant encore, ses applications s'étendent bien au-delà de la logique. Dans les années 1960, Kolmogorov, à qui l'on doit la formalisation mathématique de la théorie des probabilités, se demande : comment expliquer, si l'on tire une pièce à pile ou face cent fois, que le résultat `PPPPPPPPPPPPPPPPP...` (cent fois pile) nous apparaisse "non-aléatoire" alors qu'un résultat de type `PPFFFPFPFPFFFPFPPF...`, pourtant de même probabilité a priori, semble lui parfaitement aléatoire a posteriori ? La solution proposée par Kolmogorov, connue aujourd'hui sous le nom de *complexité de Kolmogorov* et à laquelle je consacre l'essentiel de mes recherches, s'appuie directement... sur la notion de calculabilité !

En cette année célébrant le centenaire de Turing, des dizaines de conférences ont eu ou auront lieu dans le monde, traduisant la très grande vitalité de la communauté des chercheurs en calculabilité. On ne peut, par contraste, que se désoler que cette théorie si riche et élégante soit à ce point négligée dans l'enseignement de l'informatique dispensé dans nos universités.