

Les rêves du Lambda-calcul

Soumis par Stephane Desbrosses

En 1932, Alonzo Church, maître spirituel d'Alan Turing, l'un des pères fondateurs de l'Intelligence artificielle) invente le lambda calcul, un langage logique de bas niveau, destiné à décrire de manière ultra-simplifiée les règles de logique, d'opération et de calcul. Church imaginait-il qu'il avait inventée la Pierre de rosette de la logique, le langage qui semblerait décrire nos rêves et nos mathématiques, nos pensées et notre science?

Le Lambda-calcul, langage universel?Ce langage permet, à l'aide trois instructions seulement, de formaliser toute opération logique, comme les algorithmes ou les démonstrations. Trois opérations grammaticales élémentaires pour décrire toute instruction logique : indiquer l'adresse des données, préciser l'adresse des instructions à réaliser sur ses données, et exécuter les instructions. Bien que ce langage fut inventé avant l'arrivée des ordinateurs, dans les années 50, les informaticiens se rendent compte qu'il est la base de toutes les opérations commandées par un ordinateur : sa syntaxe simpliste permet de décrire les programmes et leurs opérations dans un langage, seulement légèrement plus complexe que le langage binaire. Il est parfait pour la programmation, très proche du langage machine mais également aisé à comprendre et à interpréter, pour un humain. Quelques années plus tard, les mathématiciens de l'époque découvrent peu à peu la puissance descriptive du lambda calcul, qui permet également de décrire l'ensemble des mathématiques...

1997 : Jean-Louis Krivine démontre que le lambda calcul permet en fait d'exprimer tous les raisonnements logiques possibles et imaginables, et conséquemment, toutes les structures mathématiques connues.

Dès lors, ce langage devient universel, et devient également une Pierre de Rosette Mathématiques : on décrit des formules d'un langage informatique en lambda calcul, puis on retraduit la traduction en langage mathématique... Un algorithme informatique devient une formule mathématique. Vous ne voyez pas encore la puissance de cette constatation... Traductions et découvertesImaginez un langage, avec lequel de nombreuses choses sont d'ores et déjà démontrées, des concepts, des méthodes. Ce langage a donc permis la constitution d'une immense base de connaissance, mais celle-ci reste limitée : on ne peut l'utiliser qu'avec le langage sus-dit. Grâce au lambda calcul, on peut traduire les vérités d'une base de connaissance propre à un langage, dans un autre. C'est ce qu'a fait le mathématicien français : en traduisant des programmes déjà existant, il a pu trouver des théorèmes, des structures mathématiques, qui n'avaient jamais été découvertes par les mathématiciens. A l'inverse, traduire un théorème ou un corollaire, en langage informatique, donne accès à un programme inconnu jusqu'alors... mais déjà existant, sous forme mathématique, donc démontré et fonctionnel...

Mais cela ne s'arrête pas là. Au rang des système de logique, le neurone se pose là : il s'agit bel et bien, en gros, d'un système binaire. Je m'allume, je ne m'allume pas... Avant d'aller plus loin, il faut évoquer un théorème formidable, véritable bombe mathématique en son temps, et dont on n'a pas fini de s'étonner : Le théorème d'incomplétude de Kurt Gödel. Ce théorème indique que tout système mathématique suffisamment général, et donc puissant, se retrouve incompetent à se décrire lui même. Autrement dit, il existera toujours des vérités que les mathématiques seront incapables de démontrer, et ceci, est prouvé mathématiquement...

L'incomplétude, du ScanDisk aux mécanismes des rêvesCe théorème, formalisé en lambda calcul, puis retraduit en langage informatique, donne un code logique assez remarquable... Le programme obtenu à partir de ce théorème ressemble à s'y méprendre au scandisk, petit programme servant à restaurer les données en cours d'utilisation, lorsque l'ordinateur s'est éteint brusquement : retrouver, parcourir toutes les données antérieurement en cours d'utilisation, puis restaurer celles qui ont subi des dommages...

Le théorème de Gödel, dieu sait par quel miracle (en fait, ce n'en est pas réellement un...) décrirait très justement le programme de sauvegarde de donnée, inventé depuis belle lurette, mais jamais en référence avec une certaine théorie de l'incomplétude... Plus fort encore... le cerveau a-t-il un langage traductible en lambda calcul? une constatation :

- Parcourir les données utilisées avant plantage, les lire, restaurer certaines...
- Revivre les éléments de la journée, revivre les apprentissages et le vécu de la veille, reconsolider ces apprentissages...

Selon J-L Krivine, le théorème d'incomplétude serait une formulation mathématique de nos rêves...

Le cerveau, une machine à calculer? Et ce n'est qu'un exemple... de nombreux programmes pourraient être trouvés par la simple traduction de formules mathématiques.. Inversement, de nombreux théorèmes mathématiques peuvent être démontrés ou même découverts, grâce à la traduction de programmes informatiques... Tout cela pourrait bien être des traductions en langages mathématiques, ou informatiques, de phénomènes de la pensée... Après tout, ces théorèmes et programmes ne viennent bien que d'un endroit, le cerveau de l'homme... C'est d'ailleurs une hypothèse que Krivine défend selon les termes suivants : toute pensée peut être traduite en lambda calcul, le niveau quasi-élémentaire de logique. Au fur et à mesure qu'un mathématicien s'enfoncerait dans l'abstraction, il ne ferait en fait que décrire des suite d'instructions de lambda-calcul de plus en plus complexes, correspondant au fonctionnement de son propre cerveau. Les mathématiques ne seraient donc qu'un descriptif de notre propre système de pensée? tous les programmes et les

démonstrations ne seraient que des traductions, des descriptions du fonctionnement de notre cerveau?
Une bien séduisante théorie...