La théorie de la calculabilité ou de la cardinalité de la logique en programmation informatique

Irié Hermann Victorien **ZAMBLE**
Université Alassane Ouattara, Côte d'Ivoire
Iriehermannvictorien@gmail.com

***Résumé** :*
Alan Turing (1936), dans son article : On computable Numbers, with an Application to the Entscheidungsproblem, est considéré comme l'acte qui marque la naissance de l'informatique et des ordinateurs. Dans cet article qui fait la description du mécanisme de fonctionnement d'une machine théorique, Turing invente les termes et les concepts de programme et de programmation. Avec le développement de l'informatique dans les années qui suivront, la programmation va impliquer des notions telles que la cohérence, la consistance, la complétude, le langage de programmation etc. Ces notions, fondamentales en informatique, font écho dans la logique et, particulièrement, dans la logique mathématique. En effet, en tant que science formelle, science du raisonnement et de la rectitude, la logique prend pour objet tout ce qui touche à la clarification des énoncés, à la déduction et à la démonstration. Dans cette optique, la consistance attendue des programmes informatiques trouve leur résonnance avec l'essence des systèmes formels en logique. Ce faisant, la logique s'accorde intimement avec la volonté de produire des programmes informatiques dénués d'incohérences, afin d'éviter au mieux les bugs des machines informatiques. C'est donc ce rapport entre la logique et la programmation informatique, que la théorie de la calculabilité a le privilège de mettre en lumière, afin de rendre compte de l'importance, mieux, de la nécessité de la logique pour la programmation informatique mais également pour les informaticiens.
***Mots clés** : Calcul, Logique, Machine, Mathématiques, Programmation.*

The calculability or cardinality theory of logic in computer programming

Abstract :
Alan Turing's (1936) article <u>On computable Numbers, with an Application to the Entscheidungsproblem</u> is considered to mark the birth of computing and computers. In this article, which describes the operating mechanism of a theoretical machine, Turing coined the terms and concepts of program and programming. With the development of computer science in the years that followed, programming would come to imply notions such as coherence, consistency, completeness, programming language and so on. These fundamental notions of computer science are echoed in logic, and particularly in mathematical logic. Indeed, as a formal science, the science of reasoning and rectitude, logic takes as its object everything to do with the clarification of statements, deduction and demonstration. From this perspective, the consistency expected of computer programs resonates with the essence of formal systems in logic. In so doing, logic is closely aligned with the desire to produce computer programs devoid of inconsistencies, so as to avoid computer machine bugs as much as possible. It is therefore this relationship between logic and computer programming that calculability theory has the privilege of highlighting, in order to account for the importance, the need for logic for computer programming but also for computer scientists.
Keywords : *Calculation, Logic, Machine, Mathematics, Programming*

Introduction

Il est difficile d'aborder l'histoire de l'informatique et de l'ordinateur sans faire référence à Alan Matison Turing (1936). Ce mathématicien et logicien de génie est considéré, et à juste titre, comme le père des ordinateurs. La raison d'un tel sacre se situe dans la parution d'un article : On computable Numbers, with an Application to the Entscheidungsproblem[1]. Le titre de cet article, pour le moins évocateur, fait s'interroger sur la possibilité qu'un article qui propose de prendre pour objet des nombres puisse marquer l'acte de naissance des ordinateurs. La compréhension d'un tel fait suppose que nous ayons au préalable connaissance des circonstances qui encadrent la parution d'un tel article. Pour ce faire, il faut porter un regard sur la confluence accentuée qui a lieu, entre le raisonnement et le calcul, au début du XX^e^ siècle par le biais de la logique et des mathématiques.

La logique a fait ses premiers pas dans la Grèce antique. Elle s'illustre dans ses premiers jours comme « la science du raisonnement correct » (P. Wagner, 2007, p. 3). Perçue, à cet effet, comme un instrument pour la direction et la conduite de l'esprit dans ses raisonnements, Aristote qui en est la figure emblématique, dans l'Antiquité, va la conduire sur les chemins du syllogisme. Par ce procédé, il fait de la logique qui était grandement en usage, une science formelle avec pour fonction l'étude du raisonnement. Louée de plusieurs, les efforts pour contribuer à l'amélioration et au perfectionnement de la logique n'auront de cesses de se

[1] On computable Numbers, with an Application to the Entscheidungsproblem = Sur les nombres calculables avec une application au problème de la décision

multiplier. Entre autres efforts, l'une des figures qui va s'avérer une source d'inspiration sans précédent pour le développement de la logique moderne est sans aucun doute celle de René Descartes. Pionnier du XVII^e^ siècle et grandement épris des mathématiques dans lesquelles il voyait une méthode « pour bien conduire la raison » (1996, p. 42), Descartes tentera une approche conjointe de la logique et des mathématiques. C'est ainsi que part le souffle qui se matérialisera pour donner naissance à la logique mathématique.

Au XIX^e^ siècle, avec la crise des fondements des mathématiques engendrée par la théorie des ensembles cantoriens, les rapports entre la logique et les mathématiques se resserrent. On voit alors différentes solutions être proposées afin de pallier ladite crise. C'est dans la quête de ces solutions que se distingue David Hilbert en posant une suite de 23 problèmes qui devraient garantir aux mathématiques leur succès. Cependant, le 20ème problème de cette suite de 23 problèmes, plutôt que d'offrir une réponse mathématique comme le pensait Hilbert, donne une réponse en termes d'algorithmes et de logique. C'est donc dans cette suite logique que parait l'article de Turing qui fait la description d'une machine théorique, dans la suite de laquelle se consolidera une théorie de la calculabilité qui rend compte des concepts de programmes et de programmation, concepts clés et déterminants de l'informatique. Avec le développement de l'informatique, programme et programmation qui pouvaient sembler de simples concepts, donnent leur nom, aujourd'hui, à une activité déterminante dans le développement des machines informatiques.

Toutefois, bien qu'étant une activité fondamentalement rattachée à l'informatique et aux ordinateurs, on parvient encore, à voir se profiler, dans la programmation, une bonne part de logique qui, nous pouvons le dire, contribue, pour sa part, à l'efficacité des machines informatiques. Cela dit, eu égard à la proximité qui semble se nouer entre la logique et la programmation informatique, la question centrale à laquelle nous proposons de répondre à travers le présent travail, s'exprime comme suit : Quel peut bien être l'apport de la logique à la programmation informatique ? À cette question principale se rattache des questions secondaires, indispensables à la résolution et à la compréhension de ce problème : Dans quelle mesure la programmation informatique entretient-elle des rapports avec la logique mathématique ? Cette question qui met en exergue l'idée du rapport entre la logique et la programmation informatique est la même que celle-ci : Quels sont les fondements logiques de la programmation informatique ? A ces questions s'ajoutent les suivantes : Quelles sont les exigences de la programmation informatique, entendues que la programmation informatique se conjugue avec la logique mathématique ? En quoi, la théorie de la calculabilité se pose comme le lieu fondamental qui rend compte, de manière explicite, de la présence de la logique au sein de la programmation informatique ?

Tout l'intérêt de porter une réflexion dans cette direction n'est certainement pas sans fondement. Il suffit de porter un regard sur le monde et son fonctionnement, pour se rendre compte de la place qu'occupent les machines informatiques que sont les ordinateurs. Ces machines qui sont, aujourd'hui, omniprésentes et qui fascinent tous par leurs prouesses, ont considérablement changé le quotidien des individus. Pourtant, au-delà de ces prouesses et de ces progrès dont tout le monde peut témoigner, les machines informatiques entretiennent un lien particulier avec la logique, discipline qui prend ses origines dans l'Antiquité avec la philosophie. Ce lien qui ne transparait pas aux premiers abords, c'est lui que nous proposons de dévoiler sous le prisme de : « La théorie de la calculabilité ou de la cardinalité de la logique en programmation informatique ». Pour y parvenir, nous proposons de partir du supposé que, la programmation informatique et la logique mathématique entretiennent des rapports, quasi indissociables, qui se révèlent à partir de la notion de calcul logique. Mieux, il serait possible d'affirmer que la programmation informatique découlerait, indéniablement, des questions et des problèmes soulevés en logique mathématique. Ce faisant, il est possible qu'un ensemble de théories, aient été indispensables à l'émergence des premiers concepts relatifs à la programmation informatique. Dans cette même perspective, ce lien qui se tisserait entre la logique et les premiers concepts relatifs à la programmation informatique, à partir du calcul, serait à mesure de rendre compte des exigences logiques de la programmation des machines informatiques. Dans cette optique, il serait également possible que, le développement de la notion de calcul, aussi bien en logique qu'en programmation informatique, serait à l'origine de l'émergence d'une théorie de la calculabilité, qui met en exergue ce qui est commun à la logique et aux machines programmables que sont les ordinateurs.

Au regard de tout ce qui précède, l'analyse que nous espérons mener se donne pour finalité de poser le lien qui rattache la logique et, particulièrement, la logique mathématique, à la programmation informatique. Pour ce faire, la théorie de la calculabilité sera le lieu pour rendre clair les rapports qu'entretiennent la logique et l'informatique. Ainsi, il sera question de mettre en exergue ce que nous nommons les fondements logiques de la programmation informatique. Relativement à cet objectif, il sera nécessaire de remonter la marche historique de la logique, depuis Aristote, afin de montrer comment elle finit par embrasser les mathématiques. Un autre objectif est de montrer que, la programmation informatique ne peut pas se conjuguer sans la logique dans chaque part de sa réalisation. Cela dit, l'approche historique et la méthode déductive seront de mise pour la réalisation des objectifs que nous voulons atteindre. Au moyen de la méthode historique, nous pourrons saisir comment la logique formelle, développée par Aristote, rencontre le chemin des mathématiques et, par la suite, celui des machines informatiques. Partant de là, la méthode déductive permettra de rendre raison de l'indispensabilité de la logique en chaque part de la programmation informatique. Toutefois, avant d'y parvenir, il conviendra de mettre en exergue dans ce qui constituera le premier moment de notre argumentaire, les fondements logiques de la programmation informatique. Dans le second moment, nous tâcherons de mettre en lumière quelles sont les exigences de la programmation informatique et, comment l'exigence de logique se profile tout au long du développement informatique. Enfin, nous tâcherons de montrer, dans le troisième moment, comment à travers la théorie de la calculabilité, prend sens l'idée de fondements logiques de la programmation et, par ricochet, celle de cardinalité de la logique en programmation informatique.

1. Les fondements logiques de la programmation informatique

La logique, affirme Pierre Wagner, est « la science du raisonnement correct » (2007, p. 3). Ce point de vue n'est pas grandement distinct de celui qu'avaient les grecs déjà dans l'Antiquité. En effet, dans la Grèce antique, la logique n'était pas perçue en rupture avec le raisonnement. C'est d'ailleurs, son rapport à la pensée et au raisonnement, qui a fait de la logique une discipline célèbre dans le milieu des sciences. Son usage dans l'argumentation et dans la construction des théories lui a accordé, en plus, une place de choix. En effet, en remontant le cours de l'histoire de la philosophie, on peut se rendre compte que, même si Aristote est considéré comme le père de la logique, bien avant lui, on peut retrouver des premières traces de l'usage de la logique. D'ailleurs, Aristote, lui-même, n'affirme pas le contraire, vu qu'il fait de Zénon d'Elée « le fondateur de la dialectique, c'est-à-dire l'art de prouver en partant des principes admis par son interlocuteur » (E. Bréhier, 2001, p. 116). Du nombre des grands méconnues de l'histoire de la philosophie grecque, Zénon d'Elée a, toutefois, marqué de son empreinte, de manière indélébile, l'histoire de la philosophie surtout à travers ses apories. L'une des plus célèbres est sans aucun doute la suivante :

> La tortue piétine devant Achille et celui-ci court pour la rattraper. Si l'on se représente les intervalles que le héros devra parcourir, force est de conclure que jamais il ne rattrapera l'animal. (…) Pour qu'Achille rejoigne la tortue, il devra, d'abord, atteindre le point d'où celle-ci est partie,

> mais pendant ce temps, l'animal aura parcouru une certaine distance. Pour qu'Achille rejoigne ce nouveau point, il lui faudra un nouveau laps de temps durant lequel la tortue aura avancée de quelques mesures. Et ainsi de suite à l'infini. (C. Godin, 2007, p. 35.)

L'objectif visé par Zénon à travers ces apories était de démontrer, à chaque fois, « une proposition, en démontrant que sa négation conduit à une contradiction » (J.-P. Belna, 2014, p. 7). Ce mode de raisonnement, typique du raisonnement dit par l'absurde, démontre de ce que la logique, par la dialectique, était bien présente dans l'Antiquité, et cela bien avant Aristote. C'est donc au moyen de la dialectique que la logique commence à s'offrir comme une science de la déduction. Pour percevoir, au mieux, cet état de fait, il faut simplement regarder à Platon qui s'est fait le champion dans l'art de faire usage de la dialectique. Dans le Timée, il fait remarquer ce qui suit :

> Dieu a inventé et nous a donné la vue, afin qu'en contemplant les révolutions de l'intelligence dans le ciel, nous les appliquions aux révolutions de notre pensée, qui, bien que désordonnées, sont parentes des révolutions imperturbables du ciel, et qu'après avoir étudié à fond ces mouvements célestes et participé à la rectitude naturelle des raisonnements, nous puissions, en imitant les mouvements absolument invariables de la divinité, stabiliser les nôtres, qui sont sujets à l'aberration (Platon, 2018, pp. 105-106).

De ce bref extrait, on peut constater que Platon laisse transparaitre qu'au même titre que les lois et les principes qui régissent les mouvements des planètes, de même nos raisonnements aussi devraient être menés par des lois précises. Ce sont ces lois que Platon retrouve dans l'exercice de la dialectique. C'est la raison pour laquelle la dialectique est au centre de tous les dialogues qu'il a écrit. Ce point de vue de Platon est, toutefois, loin de faire l'unanimité. Et de ceux qui refusent de s'aligner dans la perspective développée par Platon, on retrouve Aristote, le père de la logique formelle et le disciple de Platon.

Aristote considère, en effet, que la dialectique n'est pas en mesure de rendre compte, par la seule nécessité logique, du sens des concepts. Pour ce faire, il lui faut creuser d'autres sillons que ceux de la dialectique. Car, tout raisonnement qui se veut déductif doit, nécessairement, être « une formule d'argumentation dans laquelle, certaines choses étant posées, une chose distincte de celles qui ont été s'ensuit nécessairement, par la vertu même de ce qui a été posé » (Aristote, 2014, p. 413). À l'opposé donc de Platon ou encore de Zénon, qui faisaient usage de la logique de manière empirique et uniquement dans l'argumentation, Aristote percevait dans la logique « un art, celui de construire formellement le raisonnement pour l'appliquer à la science démonstrative » (J.-P. Belna, 2014, p. 11).

Dans la perspective aristotélicienne, la pensée, en effet, se déploie en trois mouvements fondamentaux. Le premier de ces mouvements est ce qu'il a nommé le concept. Le concept désigne ce à quoi l'on pense. Le deuxième mouvement de la pensée est celui du jugement. Le jugement se rapporte à « un énoncé déclaratif grammaticalement correct susceptible d'être vrai ou faux. Il s'agit du deuxième temps de la pensée, dans lequel on affirme ce que l'on pense du concept » (L. D. Brabandere et G. Halpern, 2013, p. 19). Le troisième et dernier mouvement est enfin le raisonnement. Raisonner dans la perspective aristotélicienne, c'est « enchainer de façon nécessaire et méthodique des jugements et en tirer une conclusion » (L. D. Brabandere et G. Halpern, 2013, p. 22).

C'est exactement dans ce troisième lieu de déploiement de la pensée que se matérialise la syllogistique qui supplée, dans la théorie aristotélicienne, la dialectique.

La théorie du syllogisme défendue par Aristote se justifie dans la mesure où, là où Platon voyait en la dialectique le tout de la philosophie, Aristote y percevait, par contre, « un exercice qui n'apporte pas une certitude, parce qu'elle a égard, non pas aux choses mêmes, mais aux opinions des hommes sur les choses » (E. Bréhier, 2001, p. 308). Aristote faisait aussi remarquer, et plaignait, le fait que « la confusion dans la discussion vient de ce que l'on désigne des choses différentes par un même nom (homonyme) ou une même chose par des noms différents (synonymes) » (E. Bréhier, 2001, p. 309). En d'autres mots, et en termes plus clairs, ce sont les insuffisances de la dialectique qui ont conduit Aristote à rechercher, d'une autre part, les fondements du raisonnement. Par cette volonté, la logique, avec Aristote, commencerait à s'engager sur la voie des sciences formelles. C'est la raison pour laquelle Arthur Robert affirme que « le syllogisme est le signe sensible, il est l'expression du raisonnement » (2009, p. 25). Et, en théorie générale, il « est un discours formé de trois propositions dont l'une appelée conclusion ou conséquent, découle nécessairement des deux autres nommées prémisses ou antécédents » (A. Robert, 2009, p. 25).

Dans la pratique, en effet, les exemples de syllogismes sont très nombreux. Mais, on distingue tout de même chez Aristote des efforts considérables de faire progresser la logique à travers l'usage des syllogismes. Cet effort, on peut le remarquer à travers le syllogisme suivant, énoncé par Aristote lui-même :

> Soit, en effet, « perdre ses feuilles » A, « avoir des feuilles larges » B, « vigne » C. Si donc A appartient à C (car toute vigne a des feuilles larges), A appartient à C et toute vigne perd ses feuilles (Aristote, 2014, p. 360).

Ce syllogisme a quelque chose de particulier. Il met l'accent, en effet, non sur le contenu mais uniquement sur la forme du syllogisme. En faisant usage des lettres, on peut constater que ce n'est pas tant le fond ou le contenu de ce qui est affirmé qui importe, mais plutôt, le rapport de classe qui s'établit entre les différentes propositions qui composent et qui forment le syllogisme. En d'autres mots, cela revient à affirmer qu'à remplacer les lettres, par n'importe quel terme de classes équivalentes, on obtiendra toujours le même le résultat. Aussi, en faisant usage des lettres, Aristote introduit en logique la notion si chère de variable qu'on retrouve bien souvent au sein des mathématiques. C'est par ces différents procédés que transparait l'idée de raisonnement formellement valide, c'est-à-dire logiquement légitime, qui sera exploitée et approfondie par Leibniz et, qui mettra la logique sur le chemin des sciences formalisées. C'est sur ce chemin que la rencontre entre la logique et les mathématiques se fera pour donner naissance à la logique mathématique.

Dans la pratique, en effet, la logique mathématique, comme on la connait aujourd'hui, c'est-à-dire formalisée, prend ses fondements au XVII^e^ siècle avec le logicien et mathématicien Gottfried Wilhelm Leibniz. Penseur éclectique, savant de renom et génie de son époque, il ambitionnait de créer une langue universelle qui puisse servir au raisonnement. Fortement épris des travaux réalisés par Aristote à travers la syllogistique, il tient donc pour fait que, la logique est le point de départ de ses ambitions. Il signifie donc ce qui suit :

> L'invention de la forme des syllogismes est une des plus belles de l'esprit humain, et même des plus considérables. C'est une espèce de mathématique universelle, dont l'importance n'est pas assez connue ; et l'on peut dire qu'un art d'infaillibilité y est contenu, pourvu qu'on sache et qu'on puisse s'en bien servir, ce qui n'est pas toujours permis (G. W. Leibniz, 1765, p. 428).

C'est cet art d'infaillibilité représentatif de la logique que Leibniz va exploiter pour son ambition d'une langue universelle. En effet, il entend faire du raisonnement, un calcul logique qu'il dénomme *''calculus raciocinator''* auquel sera rattaché sa *''lingua caracteristica universalis''*. Même inabouti, ce double projet enclenché par Leibniz est d'une valeur considérable dans ce qui va constituer l'informatique et la programmation des machines que sont les ordinateurs.

Si la logique s'exprimait en langue naturelle et en lien étroit avec la grammaire en usage, la *''lingua caracteristica universalis''* de Leibniz propose, elle, de rompre avec toutes les contraintes des langues usuelles. Cette une langue artificielle dans laquelle les signes et les symboles dont on fera usage, seront à mesure de lever toutes les équivoques et toutes les ambiguïtés. Soit donc, « forger une langue symbolique, constituée de caractères, d'où le nom de caractéristique, qui parle aux yeux et non aux oreilles et plus encore à l'entendement, comme celle des mathématiques » (J-P. Belna, 2014, p. 72). A partir de la constitution de cette nouvelle langue représentative de toutes nos idées, il devient possible de faire du raisonnement un calcul. En d'autres mots, le formalisme introduit par la langue caractéristique permettra « de mettre fin aux controverses en déterminant par le calcul ce qui est logiquement déductible de prémisses posées comme admises ou de connaissances acquises » (P. Wagner, 1998, p. 84). Ainsi convient-il de comprendre le sens que revêt le calcul en logique et le calcul rationnel du projet leibnizien.

Les erreurs de raisonnements suivant la perspective du calcul rationnel se rapportent, par conséquent, à des erreurs de calcul. Par analogie aux raisonnements arithmétiques, il devient également possible, par des méthodes de preuves, de déterminer, pour un raisonnement quelconque, s'il est correct ou non. Mais faisons-le remarquer, une telle ambition n'est réalisable que si l'on pose, au préalable, la langue caractéristique qui est censée porter le calcul rationnel. Ce point de vue grandement influencé par le succès des mathématiques et particulièrement de l'algèbre, Leibniz le partage avec George Boole.

Considéré comme l'un des pères de la logique moderne, Boole a été grandement influencé par Leibniz. Voici en quelques mots en quoi consiste la tâche qu'il s'est assigné :

> Montrer que nous raisonnons conformément à des lois qui sont celles de l'algèbre et, reprenant l'idée leibnizienne de calculus raciocinator sans lui associer celle de caractéristique universelle, il entend appliquer à la logique le calcul algébrique, considéré comme purement formel (J-P. Belna, 2014, p. 88).

A travers ce but, Boole est celui qui sonne le début de la logique mathématique. Son calcul, l'algèbre de Boole, réalise une approche algébrique de la logique, en termes de variables, de fonctions, d'opérateurs sur les variables logiques. En d'autres mots, son algèbre ne dissocie pas la logique des concepts et des méthodes propres à l'algèbre. Mais comment les symboles et les opérations de l'algèbre peuvent-ils apporter une solution à des questions d'ordre logique ?

Boole conçoit qu'il est en effet possible de parvenir à la connaissance des lois et des principes qui guident le raisonnement. Ces principes, il estime qu'ils s'apparentent aux principes de l'algèbre. De ce fait, en ayant connaissance des principes qui guident le raisonnement, on peut assez aisément parvenir à leur symbolisation puis à l'élaboration d'une méthode qui puisse aider dans les calculs mathématiques. Ainsi, on peut déduire que la quête d'une méthode logique va de soi avec l'étude et la connaissance des lois de la pensée. Par conséquent, en plus d'être un moyen de calcul des inférences, l'intérêt particulier de la logique réside en « la lumière qu'elle projette sur nos facultés intellectuelles » (G. Boole, 1992, p. 22). Mais comment les concepts et les opérations algébriques peuvent-ils être une solution aux problèmes que soulève la logique ?

La réponse à cette question suggère de considérer qu'il est possible de mener, à partir d'un ensemble de signes bien déterminés, les différentes opérations du raisonnement. A ce propos, le formalisme développé par la logique des prédicats demeure l'un des systèmes les plus complets en mesure de rendre compte de la possibilité des opérations à mener sur le raisonnement. La richesse de la syntaxe ainsi que celle des différents procédés dont elle fait usage accorde à la logique de réaliser ce qui pouvait être un projet irréalisable.

De ce qui précède, il ressort que si la logique leibnizienne et la logique booléenne ont retenu notre attention, c'est parce que nous les considérons comme des références, d'un point de vue logique, pour ce qui, ultérieurement donnera naissance aux machines informatiques. Ce faisant, nous retenons les traits caractéristiques suivants de ces logiques :

- La logique leibnizienne est indissociable de l'idée d'une caractéristique universelle et d'une mise en ordre de toutes les connaissances humaines.
- Qu'il s'agisse de la logique leibnizienne ou de la logique booléenne, toutes deux prennent la forme d'un calcul opératoire avec pour fin la production de résultats.

Le calcul revêt donc une place de choix dans la logique de Leibniz et dans la logique de Boole. Il est ce qui détermine le succès du raisonnent correct et avec lui, le développement de la science logique. Faisons noter que, c'est également à partir de l'idée de calcul rationnel présente chez Leibniz et Boole que se dégage les premiers moments de la logique mathématique. Cela dit, comme le suggère la dénomination de programmation informatique, qui laisse paraitre l'idée de planification à travers le terme de programmer, n'est-ce pas parce que la programmation en informatique revêt des exigences semblables à celles du calcul logique ?

2. Des exigences de la programmation informatique

Avant de pouvoir rendre compte des exigences requises par la programmation informatique, portons premièrement un regard sur ce qu'est la programmation elle-même.

> La programmation d'un ordinateur consiste, en effet, à « expliquer » en détail à une machine ce qu'elle doit faire, en sachant d'emblée qu'elle ne peut pas véritablement « comprendre » un langage humain, mais seulement effectuer un traitement automatique sur des séquences de caractères. Un programme n'est rien d'autre qu'une suite d'instructions, encodées en respectant de manière très stricte un ensemble de conventions fixées à l'avance que

l'on appelle un langage informatique (G. Swinnen, 2009, p. 1).

De manière générale donc, la programmation informatique se rapporte à l'ensemble des activités qui permettent l'écriture des programmes informatiques. Elle désigne, également, un moment important dans l'activité de développement des logiciels en informatique. Ainsi, de manière pratique, suivant la définition donnée par Gérard Swinnen, la programmation est le moyen par lequel des instructions claires et précises sont données aux machines informatiques, c'est-à-dire les ordinateurs. De ce fait, on peut remarquer, de manière inductive que, l'activité des ordinateurs est conditionnée par leur programmation qui est, elle-même, conditionnée par les instructions que reçoivent les machines. Par conséquent, on déduit également que les instructions données aux machines sont le premier moment qui commence la programmation informatique. Et ce procédé qui consiste à donner des instructions ou « des ordres précis aux ordinateurs, de façon à ce qu'ils fassent ce qu'on attend d'eux s'appelle « écrire des algorithmes » (J-P. Delahaye, 2006, p. 74).

Un algorithme est, par définition, « une série d'instructions permettant d'obtenir un résultat » (D. Cardon, 2015, p. 4). Cela dit, les algorithmes se rapportent à des suites finies et non ambiguës d'instructions et d'opérations avec pour but la résolution de problèmes ou la réalisation de taches bien déterminées. Ainsi, si écrire un algorithme c'est donner un ordre ou une instruction à exécuter à une machine, alors il apparait qu'une instruction mal formulée ne puisse pas permettre à la machine qui la reçoit d'exécuter ou de produire le résultat souhaité. Dès lors, on comprend qu'écrire un algorithme impose à celui qui l'écrit de l'écrire sans confusion tout en faisant preuve de logique : « Chaque étape d'un algorithme doit être définie précisément, les actions à transposer doivent être spécifiées rigoureusement et sans ambiguïtés pour chaque cas » (F. G. Pansier, 2019, p. 21). Ce réquisit étant posé l'algorithme peut être alors traduit dans un langage de programmation.

Un langage de programmation est un langage dont on se sert en informatique pour mettre en formule des algorithmes et, par le même fait, écrire des programmes informatiques qui appliqueront ces algorithmes. De ce fait, comme tout langage, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de règles de significations qui rendent compréhensibles les différentes formules écrites à partir du langage. En pratique, les langages de programmation sont donc le moyen par lequel les instructions produites par le programmeur humain deviennent compréhensibles par la machine. Ils décrivent les structures de données qui seront utilisées par les machines. Au même titre que les systèmes formels[2] en logique, les langages de programmation sont constitués d'un ensemble de règles, de symboles qui constituent une syntaxe à laquelle est rattachée une sémantique qui donne sens aux différentes formulations des différentes instructions qui sont produites. De ce fait, comme tout bon traducteur, les langages de programmation sont logiquement construits de sorte à transcrire dans le langage machine, de manière précise, les différentes instructions données à la machine. C'est l'ensemble de ces instructions, une fois traduites, qui

[2] Un système formel est une modélisation mathématique d'un langage en général spécialisé.

constitue le programme à exécuter par la machine. Par conséquent, une mauvaise traduction dans le langage de programmation, c'est-à-dire un mauvais codage, ne saurait permettre au programme de s'exécuter convenablement. Comme quoi, « écrire un code informatique supporte rarement l'ambiguïté » (A. Jean, 2019, p. 60).

Développer et en particulier programmer est une activité fondamentale dans le développement des capacités des machines informatiques. C'est à partir de ces activités convenablement menées que l'on arrive à déterminer de l'efficacité et de l'utilité des logiciels, des applications et des ordinateurs dont nous faisons usage. Pour ce faire, le développement informatique nécessite en chaque part de sa constitution de faire abstraction des ambiguïtés diverses et des enchainements non logiques. C'est particulièrement ce recours à la logique, en chaque part du développement informatique, qui nous mène à examiner les rapports qui lient la logique au développement informatique à travers la théorie de la calculabilité.

3. La calculabilité, trait fondamental de la logique en programmation informatique

Le radical du vocable « calculabilité » est le mot « calcul ». Le mot calcul, d'un point de vue général, peut s'entendre comme « une suite d'opérations sur des signes » (P. Wagner, 1998, p. 158). Sous le prisme de cette définition, le calcul se rapporte aux différentes opérations usuelles qui ont cours en arithmétique ou en algèbre. Toutefois, au-delà de cette acception générale, le calcul peut être appréhendé, aussi, sous deux sens bien distincts : le calcul au sens logique du terme et le calcul au sens fonctionnel. Quand il est en rapport avec la logique, le calcul ressort l'idée suivant laquelle :

> Dans le cours d'une démonstration, on considère les relations entre les énoncés ou les expressions quelconques d'un point de vue strictement formel, en prenant en considération seulement les aspects morphologiques et syntaxiques des suites de signes (P. Wagner, 1998, p. 158).

Le calcul logique peut donc être saisie comme un procédé qui vise la formalisation des démonstrations mais, également, comme une suite d'opérations sur des signes, au même titre que le calcul effectué en arithmétique ou en algèbre. Ainsi, le procédé de calcul logique ne saurait être dissocié du procédé de démonstration parce que, la démonstration se laisse appréhender comme une suite finie de formules dont chacune des formules, qui composent la suite, peut être une conséquence immédiate des formes qui précèdent en vertu des règles d'inférences qui auront été appliquées. Par conséquent, on peut déduire que, démontrer désigne, relativement au calcul logique, une opération proprement syntaxique qui vise la transformation de formules vraies en d'autres formules vraies, au moyen de règles de constructions. C'est par opposition à cette acception qu'on peut situer le calcul fonctionnel, objet particulier de la théorie de la calculabilité. Car, si en logique, effectuer un calcul revient à formaliser une démonstration, le sens et la portée du calcul fonctionnel est tout autre, même si ces deux types de calculs sont fondamentalement identiques. Ainsi, le calcul d'un point de vue fonctionnel désigne :

> Une suite d'opérations sur les signes, déterminée par les règles d'une procédure effective qui prend la forme d'une machine de Turing, d'un lamda-terme, d'une fonction récursive ou encore, par exemple, d'un programme écrit

> dans un langage informatique (P. Wagner, 1998, p. 159).

Le calcul d'un point de vue fonctionnel est donc une généralisation de la notion de calcul numérique. Et comme le suggère la définition donnée par Pierre Wagner, le calcul fonctionnel met directement en lumière ses rapports à des notions logiques telles que la procédure effective ou le lambda-terme, mais également informatique, telles que la machine de Turing. Ce sont ces rapports que rend possible le calcul fonctionnel entre la logique et les machines informatiques que se fait d'approfondir la théorie de la calculabilité.

> Pour toute recherche qui porte sur les relations les relations entre les machines informatiques et la logique, la théorie de la calculabilité se présente à première vue comme le domaine le plus significatif. D'une part, elle trouve son origine dans la confluence historique, en 1936, de méthodes et de solutions relatives à l'une des principales difficultés posées par la logique mathématique au début de ce siècle, le problème de la décision, et d'autre part, elle définit, élabore et analyse un grand nombre des concepts fondamentaux pour l'étude théorique des machines informatiques. Elle traite précisément de ce qui est commun aux calculs logiques et aux calculateurs programmables que sont les ordinateurs. (P. Wagner, 1998, p. 15.)

Parvenir à la bonne saisie de ce qu'est la théorie de la calculabilité et comment est-ce qu'elle pose un lien substantiel entre la logique et l'informatique, nécessite de situer, avant tout, son contexte d'émergence. À ce propos, on pose les fondements de la théorie de la calculabilité dans le prolongement de la réponse du dixième problème posé par David Hilbert lors du congrès des mathématiciens en 1900.

Plus connu sous la dénomination du problème de la décision, il pose la question de savoir « s'il existe une procédure effective permettant de déterminer, pour un élément quelconque d'un surensemble donné, s'il fait partie de cet ensemble » (P. Wagner, 1998, p. 23). Dans ce problème, une chose est fondamentale : la notion de procédure effective. En effet, la préoccupation de Hilbert vise à démontrer s'il est possible de déterminer, par avance, pour n'importe quel élément d'un ensemble, s'il appartient à cet ensemble, ou plus simplement s'il est calculable. Cette question qui caractérise l'un des problèmes majeurs du formalisme, programme de David Hilbert, est celle qui aussi déterminera le sens de la théorie de la calculabilité. C'est pour répondre à cette question qu'Alan Turing écrit son célèbre article. Voici la description qui est faite de cette machine :

> Une machine de Turing est un mécanisme qui possède un nombre fini d'étapes intérieurs et qui, selon l'état où il se trouve et selon ce qu'il lit sur le ruban, efface la case du ruban qui est sous sa tête de lecture-écriture, y écrit un symbole et se déplace vers la gauche ou vers la droite. Le programme de la machine est une suite finie d'instructions du type : « si je suis dans l'état E_3 et si je lis un 0 sur le ruban, alors je le remplace par 1, je me déplace d'une case vers la droite et je passe dans l'état E_2 » ; en abrégé, on note une telle instruction ($E_30 \rightarrow E_21$ D) (J.-P. Delahaye, 2003, p. 10).

De cette description de la machine de Turing et de son fonctionnement, on comprend que pour chaque procédure de calcul entamé par la machine, il existe une suite finie de signes et d'instructions de sorte qu'il est possible de distinguer clairement chaque étape du calcul. Pour chaque moment du calcul, la réaction de la machine est déterminée en fonction des signes et des informations lus sur le ruban qui compose la machine. La première configuration du ruban caractérise l'état initial et, la dernière celle dans laquelle la machine ne réalise plus aucune tâche, l'état final. Ainsi, une opération ou un calcul est achevé lorsque la machine ne réalise plus aucune tâche, c'est-à-dire lorsqu'elle arrive à l'état final. Toutefois, il n'est aucune garantie qui assure qu'une machine de Turing arrivera nécessairement à l'état final. Ce que la machine de Turing est, par contre, assurée de produire, c'est la procédure de calcul en fonction des différentes données inscrits sur le ruban. Par conséquent, le résultat d'un calcul s'entend sous la forme d'une fonction, soit totale ou partielle, qui se rapporte à l'inscription initiale. Une fonction pour laquelle il existe une machine de Turing qui en est la procédure effective est donc dite Turing-calculable.

Le calcul avec les machines de Turing, premier modèle théorique des ordinateurs, devient symbolique et généralisé. Le calcul est symbolique parce qu'il ne se présente plus nécessairement sous la forme numérique mais se fait également sur des suites de signes quelconques. Il est aussi généralisé dans la mesure où toutes les procédures de calcul usuelles peuvent être exécutées par une machine de Turing. Dans cette mesure, une fonction est dite calculable lorsque les valeurs de cette fonction peuvent être déterminées à partir de leurs paramètres par un processus mécanique fini. Ainsi, on passe d'un problème de logique mathématique à travers le problème de la décision, à la caractérisation de ce qu'est le calcul à travers les machines de Turing. Et, c'est de cette caractérisation de la notion de calcul à travers les machines de Turing que se posent les fondements directs de la calculabilité mais également des liens entre la logique et les calculateurs universels que sont les ordinateurs.

Les origines de la calculabilité ayant été retracées, on comprend aisément comment Leibniz et Boole peuvent être considérés, à juste titre, comme les précurseurs, d'un point de vue logique, de l'informatique et des ordinateurs contemporains. La dénomination anglaise du mot ordinateur, c'est-à-dire *computer,* qui traduit l'idée littéralement de « calculateur » nous enjoint de nous rappeler que les ordinateurs ont vocation à opérer des calculs pour la production de résultats. Ce sont ces calculs qu'opèrent les ordinateurs qui se rapportent à l'idée de calcul rationnel présente chez Boole et Leibniz. Ce faisant, à travers l'idée de caractéristique universelle et de mise en ordre des connaissances humaines, on peut également percevoir les ordinateurs en tant que machines intelligentes qui mettent à disposition un savoir encyclopédique.

Relativement à l'activité de développement en informatique, s'il est que les ordinateurs sont des calculateurs universels, alors posons que le développement et le fonctionnement des ordinateurs repose sur l'exécution de fonctions récursives, c'est-à-dire l'exécution de fonctions calculables. Comme quoi, « l'ensemble des capacités que manifestent ou pourraient manifester les machines (…) reposent sur des possibilités de calcul » (P. Wagner, 1998, p. 203). Le calcul, logiquement déterminé, est donc d'une importance capitale dans la programmation informatique en particulier et dans le développement général. D'autant plus que, c'est à partir de calcul au moyen de la théorie de la calculabilité que se pose de manière efficiente les rapports entre la logique et l'informatique. La logique, science de la rectitude qui conduit toute opération est donc d'une importance capitale pour le développement informatique.

Conclusion

La théorie de la calculabilité ou théorie de la récursion définie ce qu'il y a de commun entre la logique et l'informatique. Ce lien commun qui rapproche ces deux domaines de savoir qui pouvaient sembler distincts, c'est le calcul. En logique, il s'exprime dans ses premiers moments avec Leibniz sous la forme d'une caractéristique universelle c'est-à-dire « une sorte d'alphabet idéographique » (L. Couturat, 1901, p. 54) à partir de laquelle pourra se réaliser un calcul rationnel en mesure d'« entrainer la découverte mécanique de la vérité » (F. Chenique, 2006, p. 38). Boole dans la même veine que Leibniz, relativement à l'idée de calcul rationnel, entreprend d'associer à la logique les moyens de l'algèbre. Ainsi, la logique entame-t-elle son ascension vers la mathématisation dans sa quête de rectitude.

Cette quête de la rectitude à travers le calcul, on la retrouve chez David Hilbert lorsqu'il pose la question de l'existence d'un procédé effectif à mesure de déterminer, pour n'importe quelle fonction donnée, si elle est calculable. C'est à partir de cette question, premièrement logique, que vont naitre les premières réflexions sur l'informatique théorique et les calculateurs universels que sont les ordinateurs. La théorie de la calculabilité qui voit le jour dans la suite du problème de la décision posé par Hilbert, vient rendre compte et mettre à jour les liens fondamentaux qui rapportent la logique à l'informatique en prenant pour objet le calcul. Déterminant dans la réalisation des tâches et le traitement de l'information, le calcul, logiquement mené, s'avère être le fondement du fonctionnement des machines et, par conséquent, la clé de la programmation et du développement informatique. Cela dit, c'est ce rapport qu'entretient le calcul avec la logique qui fait se poser la logique comme cardinale pour le développement informatique.

Références bibliographiques

ARISTOTE, 2014, Œuvres complètes, *Seconds analytiques,* Paris, Flammarion, 5701 p.

BELNA Jean-Pierre, 2014, *Histoire de la logique,* Paris, Seuil, 170 p.

BOOLE George, 1992, *Les lois de la pensée,* Paris, Vrin, 414 p.

BRABANDERE Luc De et HALPERN Gabrielle, 2013, *La logique, l'art de bien utiliser le mot donc,* Paris, Eyrolles, 46 p.

BREHIER Emile, 2001, *Histoire de la philosphie, Tome 1, Antiquité et Moyen Age,* Paris, PUF, 712 p.

CARDON Dominique, 2015, *À quoi rêvent les algorithmes. Nos vies à l'heure des big data,* Paris, Seuil, 108 p.

CHENIQUE François, 2006, *Éléments de logique classique,* Paris, L'Harmattan, 390 p.

COUTURAT Louis, 1901, *La logique de Leibniz,* Paris, Ancienne librairie Germer Baillère, 608 p.

DELAHAYE Jean-Paul, 2003, *Logique, informatique et paradoxes,* Paris, Belin, 158 p.

DELAHAYE Jean-Paul, 2006, *Complexité. Aux limites des mathématiques et de l'informatique,* Paris, Belin, 256 p.

DESCARTES René, 1996, *Discours de la méthode,* Paris, Bordas, 192 p.

GODIN Christian, 2007, *La philosophie pour les nuls,* Paris, First, 720 p.

JEAN Aurélie, 2019, *De l'autre côté de la machine. Voyage d'une scientifique au pays des algorithmes,* Paris, Humensis, 177 p.

LEIBNIZ Georg Wilhelm, 1765, *Nouveaux essais sur l'entendement humain,* Trad. E. Grevin, Paris, Flammarion, 441 p.

PANSIER Frédéric-Jérôme., 2019, *iJuge, vers une justice prédictive,* Paris, Louis Gouyon Matignon, 95 p.

PLATON, 2018, *Timée*, Québec, BeQ, 204 p.

ROBERT Arthur, 2009, *Leçon de logique,* Québec, Fondation littéraire Fleur de Lys, 236 p.

SWINNEN Gérard, 2009, *Apprendre à programmer avec python,* Paris, Eyrolles, 341 p.

WAGNER Pierre, 1998, *La machine en logique*, Paris, PUF, 239 p.

WAGNER Pierre, 2007, *La logique,* Paris, PUF, 131 p.