

Préface

Je suis très heureux et honoré de préfacer l'ouvrage de Narendra Jussien. J'ai eu le plaisir d'initier ce dernier aux joies de la logique, de la calculabilité et de la complexité il y a un certain nombre d'années dans son cursus de formation universitaire à l'Institut de mathématiques appliquées d'Angers. Le chemin parcouru par Narendra depuis, aussi bien en enseignement qu'en recherche, aboutit à la publication d'un livre imposant sur la logique et les fondements théoriques de l'informatique.

Une préface, pour quoi faire ? Le *Trésor de la Langue Française* nous dit : « Texte placé en tête d'un ouvrage pour le présenter et le recommander au lecteur en précisant éventuellement les intentions ou en développant des idées plus générales ». Recommander le livre au lecteur est une évidence ! Quant à développer des idées générales, l'auteur le fait de façon très pertinente à maintes reprises. Je vais tenter de mettre le lecteur en appétit par quelques commentaires.

À qui peut s'adresser ce livre ? À toute personne qui s'intéresse à la logique, aux fondements de l'informatique et aux relations entre ces deux domaines. N'entend-on pas régulièrement des affirmations du type : « Avec les progrès de l'informatique, on pourra tout faire. Il n'y a pas de problème, il n'y a que des solutions. » ? L'ouvrage donnera les points de repère théoriques permettant de répondre à des questions essentielles : « Que peut faire un ordinateur, comment peut-il le faire ? Que signifie raisonner quand on parle d'un ordinateur ? L'informatique peut-elle tout résoudre ? L'informatique a-t-elle des limites et, si oui, quelles sont ces limites ? »

La façon d'aborder le sujet est mathématique. C'est dire que les démonstrations sont présentes tout au long de l'ouvrage. Les concepts abordés nécessitant un entraînement, des exercices corrigés sont proposés de façon à approfondir les notions. Mais la culture générale est loin d'être absente puisqu'un chapitre d'indications biographiques donne les points de repère historiques concernant les différentes notions abordées. Depuis Aristote jusqu'à Turing ou Zadeh en passant par Boole, Gödel, Chomsky ou Robinson. Un joli travail de synthèse et de mise en perspective.

La syntaxe du calcul propositionnel est présentée à l'aide de grammaires formelles (introduites de façon générale en deuxième partie de l'ouvrage). Les personnes qui

apprécient les structures algébriques pourront s'en donner à cœur joie dans les manipulations entre les treillis et anneau de Boole. La partie déductive du calcul propositionnel et du calcul des prédicats est orientée vers l'automatisation du raisonnement. On trouve ici les fondements du raisonnement automatisé et les principes de mise en œuvre.

Après les grammaires formelles et automates finis, les bases de la calculabilité sont données. Qu'entend-on par modélisable, calculable ? Quelle est la définition d'un algorithme ? C'est ici que l'on peut conceptualiser les possibilités de l'informatique. Les diverses approches, allant des grammaires formelles aux machines de Turing en passant par le lambda-calcul, sont équivalentes. Au cours de jolies démonstrations courtes mais très puissantes, on montre, par exemple, à l'aide d'un système de codage adéquat que ce qui est calculable est dénombrable (les entiers naturels) et, en y ajoutant des considérations de cardinalité d'ensembles, que ce qui est non calculable est du domaine du continu (les nombre réels). C'est dire que le volume de ce qui nous échappe est beaucoup plus important que celui des informations que nous pouvons traiter.

À partir des années 1970, la théorie de la complexité des problèmes est apparue et est devenue incontournable. Il ne suffit pas d'avoir un algorithme pour résoudre un problème donné, encore faut-il que le temps de calcul ne soit pas démesuré. La théorie de la complexité a permis de classer les problèmes calculables. Les pionniers dans ce domaine ont été les chercheurs en recherche opérationnelle qui travaillaient sur la méthode la plus efficace de résoudre les problèmes classiques d'optimisation : problème du voyageur de commerce ou du circuit hamiltonien, coloration d'un graphe, optimisation d'emplois du temps, etc. Le théorème de Cook (1971), premier résultat d'importance sur le sujet, montre que le langage SAT des expressions logiques satisfiables est NP-complet. Et la boucle est bouclée ! Nous revenons à la logique, point de départ de l'ouvrage. La notion de NP-complétude est un concept central en théorie de la complexité ; elle est définie à partir des machines de Turing déterministes et non déterministes. En tentant une comparaison avec la démarche humaine, on peut dire que le non-déterminisme s'apparente à l'intuition alors que le déterminisme est du ressort de l'énumération. Une ouverture sur l'intelligence artificielle.

Le livre de Narendra Jussien est une référence pour toute personne qui s'intéresse aux fondements de l'informatique et qui désire l'aborder de façon approfondie. Il ne peut qu'être vivement conseillé aux étudiants en mathématiques, mathématiques appliquées, informatique et aux élèves ingénieurs.

Yvon L'HOSPITALIER¹

1. Yvon L'Hospitalier est ancien directeur de l'IMA (Institut de mathématiques appliquées d'Angers) et est actuellement professeur à l'ESSCA (École supérieure des sciences commerciales d'Angers).

Avant-propos

L'informatique est partout mais l'informatique pose question. S'agit-il d'une science ? S'agit-il d'une technique ? Qu'étudient les chercheurs en informatique ? Et d'ailleurs, y-a-t-il même une recherche en informatique ? Quelles sont les compétences d'un informaticien ? Ces questions sont fréquentes. Les réponses dépendent souvent de la personne à qui on les pose. Essayons d'éclaircir un peu les choses.

Ce que n'est pas l'informatique

L'informatique n'est pas la science des ordinateurs² : les informaticiens (et encore moins les docteurs en informatique) ne sont pas voués à la réparation des ordinateurs, ils ne sont pas non plus nécessairement des « as » de l'électronique, ni des spécialistes des branchements et des câbles. L'informaticien n'est pas forcément la personne la plus à même de vous guider dans le choix d'un appareil photo numérique, d'un ordinateur portable ou d'une imprimante. En effet, l'informatique peut s'étudier avec uniquement une feuille de papier et un crayon (sans ordinateur).

L'informatique n'est pas non plus la science des logiciels. L'informaticien n'est pas toujours au courant de la dernière macro miracle de tel ou tel traitement de texte du commerce. L'informaticien n'est pas non plus forcément en train de tester un jeu vidéo ou de chercher une bogue, etc.

Le métier de l'informaticien, et donc ses compétences, ne tient ni du matériel ni du logiciel. Il s'agit de quelque chose de plus abstrait entre les deux. C'est aussi quelque chose de plus fondamental sans quoi ni ordinateurs, ni logiciels ne pourraient fonctionner.

2. C'est pourtant ce que pourrait laisser penser le vocable anglais : *computer science*.

Vers une définition de l'informatique

L'informatique est de plus en plus souvent abordée par l'utilisation de logiciels d'usage courant : traitements de texte, tableurs, navigateurs Internet, jeux, etc. L'ordinateur est alors souvent vu comme une imposante machine à calculer, passant des heures à réaliser des calculs complexes. Mais l'informatique n'est pas uniquement cela. Elle peut être utilisée aussi par exemple en robotique : du robot aspirateur autonome à la machine à laver en passant par le drone. C'est encore des utilisations en médecine : séquençement du génome, imagerie médicale, opérations à distance, etc. Ses utilisations sont à la fois nombreuses et variées.

Mais, les applications de l'informatique, et *a fortiori* les programmes sous-jacents, ont tous deux points communs :

- tout programme manipule des données qui peuvent être de différents types : caractères alphanumériques, nombres, symboles mathématiques, dessins, sons, images, données factuelles, etc. Toutes ces données sont discrètes (distinctes les unes des autres et énumérables) ;

- tout programme réalise des traitements sur ces données : copier, coller, effacer, déplacer, mettre en page, calculer, écrire et résoudre des équations, faire des graphiques, appliquer des règles (d'un jeu), stocker en mémoire et rechercher des données, etc. Tous ces traitements sont effectifs (exprimables par un algorithme³).

C'est ces deux points qui caractérisent l'informatique et qui rendent caduque la traditionnelle distinction entre matériel et logiciel. En effet, les données (évidemment stockées au niveau matériel) sont manipulées par les traitements en provenance du logiciel. Et c'est là la compétence principale de l'informaticien. On peut ainsi définir l'informatique comme la science de tous les traitements effectifs applicables à des données discrètes.

Concrètement l'informatique se décline sur un spectre très large : de l'étude papier-crayon de la complexité intrinsèque d'un problème à la programmation d'un jeu. Mais la démarche de l'informaticien confronté à un problème est toujours la même :

- trouver un équivalent discret et numérique aux données réelles du problème ;
- exprimer sous forme d'algorithme les conditions de modifications de ces données permettant d'aboutir au résultat recherché.

3. On peut définir informellement un algorithme – le nom vient du mathématicien persan al-Khwarizmi – comme une méthode systématique permettant de résoudre à coup sûr et en un nombre fini d'étapes une certaine classe de problèmes ou de répondre à une certaine classe de questions.

Fondements de l'informatique

Dans cet ouvrage, nous allons aborder trois thèmes qui constituent les bases théoriques de l'informatique :

- le premier thème concerne la logique dite formelle. Il s'agit d'explorer l'univers fascinant des outils permettant de manipuler les composants de bases des données d'un ordinateur : les bits (une donnée ne possédant que deux états possibles : vrai ou faux, zéro ou un, allumé ou éteint, etc.). Nous pousserons notre exploration jusqu'aux concepts nécessaires à l'automatisation de raisonnements logiques et nous ferons même une incursion vers d'autres logiques que les logiques classiques ;
- le deuxième thème concerne les notions de langage formel et d'automate. Ces outils mathématiques sont à la base de la théorie des langages (de programmation) qui permettent de développer des algorithmes au jour le jour par des millions d'informaticiens. Nous manipulerons les principes de bases de ces outils et verrons comment, malgré leur simplicité, ils ont encore de nombreuses applications directes ;
- enfin, le troisième thème concerne les limites de l'informatique. En effet, il faut être conscient que l'informatique ne peut pas tout et que certaines choses sont plus difficiles (complexes) que d'autres à réaliser. Nous verrons là encore les outils théoriques nécessaires à la délimitation entre ce que peut et ce que ne peut pas faire un ordinateur.

Information

Le symbole « \leftrightarrow » signale dans le texte une figure historique importante du domaine pour laquelle des informations biographiques sont fournies dans l'annexe A. Le symbole « \blacktriangleright », quant à lui, signale un exercice dont une correction est proposée dans l'annexe B.

La plupart des résultats (théorèmes, propositions, etc.) présentés dans cet ouvrage sont démontrés formellement. Quelques-uns, ceux pour lesquels une démonstration formelle dépasserait le cadre de ce livre, ne le sont pas mais des indications pour en retrouver une démonstration sont fournies.

Enfin, des supports spécifiques pour l'enseignement sont disponibles en ligne sur le site : njussien.e-constraints.net/11c. En effet, cet ouvrage est issu d'un enseignement dispensé dans différents cours :

- « *Logique(s), Langages, Algorithmes* », cours de 47 heures aux étudiants de troisième année de l'Institut de mathématiques appliquées d'Angers ;
- « *Fondements de l'informatique* », cours de 15 heures aux élèves de première année de l'École des Mines de Nantes ;
- « *Programmation logique* », cours de 30 heures aux élèves de troisième année de l'École des Mines de Nantes [JUS 03].

Remerciements

Ce livre doit beaucoup à Yvon L'Hospitalier : pour avoir accepté de le préfacier bien sûr, mais aussi et surtout pour la qualité de son enseignement qui vous permet, quinze ans après, d'en feuilleter les fruits.

Merci enfin à tous ceux qui ont consacré du temps à la relecture des premières épreuves de cet ouvrage et plus particulièrement à Claire-Marie Cheval, Catherine Jussien et Guillaume Rochart.