

Quelle informatique enseigner au lycée ?

Gilles Dowek(*)

Résumé. Enseigner l'informatique au lycée nous paraît être une nécessité, du fait de la place de cette discipline aussi bien dans notre économie et dans notre société que parmi les outils qui nous permettent de comprendre le monde. Nous proposons l'idée que l'enseignement de l'informatique au lycée devrait avoir comme but principal l'apprentissage d'un langage de programmation et d'algorithmes de base, avec l'objectif de savoir écrire un programme au moment de passer son baccalauréat. L'apprentissage de la programmation et de l'algorithmique est de nature à apporter beaucoup aux lycéennes et lycéens dans leur développement intellectuel, car il permet un travail par projets et demande de mettre en application des connaissances acquises. Et également car il permet de construire un pont entre le langage et l'action et montre l'utilité de la rigueur scientifique.

Enseigner l'informatique

Enseigner l'informatique au lycée apparaît comme une nécessité à beaucoup d'entre nous. Cette nécessité prend plusieurs formes.

Tout d'abord, on peut constater que nos étudiants ont un faible niveau en informatique, comparés à leurs camarades indiens et chinois, bien entendu, mais aussi européens. Notre population active a également un niveau global insuffisant en informatique, y compris, malheureusement, les ingénieurs. Enfin, peut-être le plus regrettable est le niveau très faible en informatique de certains de nos concitoyens, qui sont, de ce fait, laissés à l'extérieur des transformations qui s'opèrent dans nos sociétés. Ce faible niveau est à l'origine ou bien d'une confiance démesurée en les ordinateurs, ou bien de craintes injustifiées. Il est aussi à la source d'une certaine difficulté à faire émerger des débats publics pourtant essentiels à la vie de nos sociétés, sur la brevetabilité du logiciel, par exemple, ou sur l'opportunité, pour l'Administration, d'utiliser des logiciels propriétaires ou libres.

Le lycée semble également la dernière occasion pour enseigner largement l'informatique aux jeunes filles qui ont trop souvent tendance à considérer cette discipline comme un « jeu de garçons ». Cette fausse impression pénalise les femmes sur le marché de l'emploi. De plus, elle participe à la faible culture générale en informatique des citoyens, dont la moitié sont des citoyennes.

À côté de ces motivations pour enseigner l'informatique relatives à notre compétitivité industrielle ou à la vie de notre société, une autre série de motivations vient de ce que l'informatique a, ces dernières décennies, profondément bouleversé notre manière de voir le monde. Par exemple, une discipline aussi ancienne que la grammaire a été transformée quand on a pris conscience du fait que l'ensemble des

(*) École polytechnique. <http://www.lix.polytechnique.fr/~dowek>

règles de grammaire d'une langue définissait un algorithme qui décide si une phrase est grammaticale ou non. Et la manière de formuler ces règles de grammaire en a été profondément affectée. De même, la compréhension des mécanismes de l'œil et du cerveau qui nous permettent de construire une image à partir de stimuli visuels s'est grandement renouvelée quand on a compris que ces mécanismes devaient être algorithmiques et, de plus, massivement parallèles. Pour prendre un exemple plus récent, nous avons vu apparaître ces dernières années de nouvelles manières de décrire le fonctionnement de la cellule, issues de la théorie des langages distribués. Même si certains de ces changements de point de vue doivent encore faire la preuve de leur pertinence et de leur utilité, ils montrent que l'informatique est devenue, à l'instar des autres disciplines scientifiques, un outil essentiel pour appréhender le monde.

Les trois informatiques

Pour dépasser ces quelques généralités, il est nécessaire de commencer par se mettre d'accord sur la signification du mot « informatique ». Dans notre langue, ce mot a essentiellement trois significations. Selon la première de ces significations, on connaît l'informatique dès que l'on sait utiliser un ordinateur, par exemple dès que l'on sait utiliser un logiciel de traitement de texte ou un navigateur. Selon la deuxième, on connaît l'informatique quand on sait écrire un programme, ce qui demande la connaissance d'au moins un langage de programmation et d'algorithmes élémentaires. Selon la troisième enfin, quand on connaît les différentes théories – calculabilité, lambda-calcul, réécriture, complexité, théorie algorithmique des nombres, théorie des automates, théorie des systèmes distribués, théorie des graphes, ... – qui forment une science dont l'objet d'étude est le calcul sous toutes ses formes.

Transposée dans un autre domaine du savoir, cette situation est comparable à celle d'une langue qui utiliserait la même expression pour « savoir conduire une locomotive à vapeur », « savoir fabriquer une locomotive à vapeur » et « connaître la thermodynamique ». Cette confusion que l'on retrouve par exemple dans les différentes significations du mot « chimie » a certains avantages - par exemple celui de permettre à nos anciens doctorants de se définir comme « informaticiens » quand ils cherchent un emploi, de même qu'un docteur en chimie peut se déclarer « chimiste », alors que le mot « physicien » est moins polysémique – mais aussi de nombreux inconvénients, en particulier celui d'introduire beaucoup de malentendus dans les discussions relatives à l'enseignement de l'informatique.

Apprendre à écrire des programmes

L'idée que je voudrais proposer est que l'enseignement de l'informatique au lycée doit prendre le mot « informatique » dans la deuxième acception décrite ci-dessus : l'enseignement en informatique au lycée doit avoir comme objectif premier que les lycéens sachent écrire un programme au moment de passer leur baccalauréat.

On peut concevoir un enseignement de l'informatique en trois temps. L'apprentissage de l'utilisation d'un ordinateur, qui relève sans doute de l'école primaire ou des premières années du collège, l'apprentissage de la programmation et des algorithmes

élémentaires au lycée, et l'apprentissage de la science du calcul dans l'enseignement supérieur. Cependant, le point de vue que je défends n'est pas uniquement motivé par une telle vision globale. Il est aussi motivé par le fait que l'apprentissage d'un langage de programmation et des algorithmes élémentaires me semble une expérience propre à apporter beaucoup aux lycéens, quant à leur développement personnel et dans leur relation à la connaissance.

Concevoir et accomplir un projet

L'enseignement de l'informatique accorde une part importante au développement de projets. Le premier apport de l'apprentissage de l'informatique, pour les lycéens, me semble être celui du développement de leur capacité à définir les objectifs d'un projet et à se donner les moyens de les accomplir.

Écrire un programme est un projet qui se déroule sur une longue période, qui va d'un après-midi à quelques semaines dans le cas d'un programme pédagogique, et qui peut aller jusqu'à plusieurs années dans le cas d'un programme professionnel. La première phase d'un tel projet consiste à définir des objectifs qui doivent être attrayants, mais réalistes. En organisant des ateliers d'initiation à l'informatique pour des adolescents avec l'*Association Nationale Sciences et Techniques Jeunesse*, il m'est arrivé de rencontrer des jeunes qui pensaient pouvoir écrire en quelques jours un programme capable de battre le champion du monde aux échecs, sans manifestement trop se soucier du principe de réalité. Il m'est arrivé, à l'inverse, de rencontrer des jeunes qui se fixaient des objectifs très modestes par crainte de ne pas parvenir à les réaliser. Par sa nature, l'enseignement de l'informatique favorise l'apprentissage du juste niveau auquel il faut fixer ses objectifs, de leur nécessaire redéfinition au cours du développement du projet, de la persévérance que demande la réalisation d'un tel projet, ... Connaissances qui sont, bien entendu, également utiles hors du champ propre de l'informatique.

Naturellement, cette organisation de l'enseignement des sciences par projets n'est pas nouvelle, et je me réjouis qu'elle soit revenue à la mode, par exemple, avec l'opération *La main à la pâte*. Mais l'informatique, parce qu'elle permet de construire des objets complexes avec un support physique minimum permet, peut-être plus que d'autres disciplines, de proposer un tel enseignement par projets, alors que la difficulté de mener à bien, de manière autonome, une expérience de physique des particules, par exemple, a réduit de manière bien inévitable les possibilités d'enseigner par projets d'autres champs du savoir, au lycée et dans les premières années des études supérieures.

Mettre en pratique les connaissances acquises

Dans l'apprentissage de l'informatique, il est important de mettre en pratique, par l'écriture de programmes, les connaissances acquises en cours. Nous savons qu'il y a un certain fossé entre le fait de comprendre une notion et celui d'être capable de l'utiliser, par exemple entre le fait de comprendre la fonction sinus en cours de mathématiques et celui d'être capable de l'utiliser dans un exercice de physique. De même, les écoliers apprennent à lire des poèmes de Jacques Prévert bien avant de savoir composer une rédaction. Ce passage de la compétence à la performance est

d'ailleurs l'une des principales difficultés que rencontrent les étudiants quand ils commencent leur apprentissage de l'informatique.

Naturellement, ces connaissances mises en pratique sont celles acquises lors des cours d'informatique, mais aussi des autres matières. Ainsi, un lycéen qui écrit un programme pour calculer les mille premières décimales du nombre π met en application aussi bien des connaissances acquises lors de ses cours d'informatique que de mathématiques. Cela peut également donner un supplément de finalité à la définition du nombre π par une série, finalité qui peut, sans une telle mise en pratique, paraître ténue à un lycéen de niveau moyen.

Cette situation de l'informatique est assez singulière. Nos collègues physiciens, par exemple, nous ont convaincu qu'il est possible d'apprendre la thermodynamique sans fabriquer de locomotive à vapeur. En informatique, en revanche, même s'il est possible, en principe, d'apprendre la théorie des fonctions récursives, sans avoir jamais écrit un programme de sa main – ce qui était d'ailleurs le cas des créateurs de la théorie des fonctions récursives dans les années trente –, il semble qu'il ne soit pas souhaitable de former des informaticiens qui connaîtraient tout sur l'algorithmique des graphes, mais n'auraient jamais programmé un algorithme de parcours de graphe à un moment ou à un autre de leur vie. Dans l'apprentissage de l'informatique, la théorie et les travaux pratiques – l'écriture de programmes – vont donc ensemble. Je ne sais pas si cela est dû à la relative jeunesse de notre discipline, ou si c'en est un trait permanent, qui s'explique autrement.

Le développement de cette faculté à mettre ses connaissances en pratique est certainement un apport important de l'apprentissage de l'informatique.

Un pont entre le langage et l'action

Un autre apport pour les lycéens de l'apprentissage de l'informatique me semble être celui de la construction d'un pont entre le langage et l'action.

On sait que les lycéens perçoivent parfois un certain décalage entre un savoir heureusement livresque qui leur est enseigné au lycée et l'idéologie à laquelle ils sont soumis à l'extérieur du lycée qui valorise l'action et l'initiative. L'idée d'une dichotomie entre l'univers du langage et celui de l'action est particulièrement préjudiciable pour les lycéens quand ils en tirent la conclusion que ce qu'ils disent importe peu ou que la pensée est peu utile à l'action.

Un programme informatique a la première qualité d'appartenir à un langage, c'est-à-dire d'être un texte. Mais ce texte a une seconde qualité qui est d'être exécutable, c'est-à-dire d'être l'agent d'une action. L'apprentissage de l'informatique enseigne que dire « $x = 1$ » ou « $x = 2$ » n'est absolument pas la même chose. Et également que l'action ne peut pas s'effectuer sans la construction préalable d'un discours.

Une école de rigueur

Le dernier apport de l'apprentissage de l'informatique pour les lycéens me semble être celui du développement de la rigueur. Naturellement, la rigueur est une qualité partagée par l'ensemble des sciences et l'informatique n'est pas une science plus rigoureuse que les autres. Cependant, le rôle de la rigueur est différent dans

l'apprentissage de l'informatique et dans l'apprentissage des autres sciences. Un ingénieur qui manque de rigueur dans la construction d'un pont, voit celui-ci s'effondrer : c'est la Nature qui sanctionne son manque de rigueur. En revanche, ce sont souvent les enseignants, et non la Nature, qui sanctionnent le manque de rigueur des lycéens. Les lycéens peuvent en tirer la conclusion, évidemment erronée, que cette rigueur, imposée par une autorité, est arbitraire, et mal percevoir sa finalité et sa fonction.

En informatique, les erreurs des élèves ne sont pas sanctionnées par un enseignant, mais d'abord par un ordinateur inanimé et donc impartial. Un programme qui comporte une erreur de syntaxe est simplement rejeté par le compilateur. Un programme qui comporte une erreur de sémantique ne donne pas le bon résultat. La rigueur n'apparaît donc pas comme une norme imposée de l'extérieur, mais simplement comme une condition d'une bonne communication entre le lycéen et la machine.

Sur ce point, on peut noter que plusieurs expériences pédagogiques, en particulier une expérience menée par le Département de Mathématiques de l'Université de Chambéry, ont montré que l'utilisation de programmes qui sanctionnent des erreurs de logique dans les démonstrations mathématiques, comme le programme Coq, développé à l'*Institut National de Recherche en Informatique et en Automatique*, à l'École polytechnique et à l'Université de Paris-Sud, ou le programme Phox développé à Chambéry, peuvent être utilisés, avec succès, pour enseigner les règles de manipulation des quantificateurs et la notion de démonstration par récurrence aux étudiants du premier cycle universitaire. Même s'il est sans doute trop tôt pour généraliser ce type d'expériences au lycée, on peut remarquer la convergence de points de vue : l'apprentissage de la rigueur passe, dans un cas comme dans l'autre, par une mécanisation de la sanction.

Cet apprentissage de la rigueur et du sens du détail est souvent très nouveau pour les lycéens et les étudiants et constitue l'une des difficultés de l'apprentissage de l'informatique. Il est parfois surprenant pour les débutants qu'un programme juste à quatre-vingt-dix-neuf pour cent donne un résultat faux à cent pour cent. Le mode d'évaluation des connaissances à l'école et au lycée privilégie la compréhension de l'idée centrale. Certains élèves en déduisent, à tort, que la compréhension des idées générales suffit. L'apprentissage de l'informatique montre que les ordinateurs, comme la Nature, sont moins bienveillants que les enseignants et qu'un programme peut être faux parce que l'idée est mal comprise, mais aussi parce qu'un détail a été négligé.

Pour conclure, je voudrais revenir sur cette urgence que je perçois à commencer, dès le lycée, l'enseignement de l'informatique, c'est-à-dire d'un langage de programmation et des algorithmes élémentaires, avec l'objectif de savoir écrire un programme au moment de passer son baccalauréat. Cette urgence n'est pas uniquement due à la demande du marché de l'emploi, mais avant tout au fait que cet apprentissage est de nature à apporter beaucoup aux lycéens aussi bien quant à leur développement personnel que dans leur rapport à la connaissance et dans la vision du monde qu'ils se construisent.