La formation des maîtres pour l'option informatique des lycées

Jacques Arsac

LA FORMATION DES MAÎTRES POUR L'OPTION INFORMATIQUE DES LYCÉES

Jacques ARSAC

1. DIFFICULTÉS INTRINSÈQUES

Il est très difficile de parler de formation des maîtres en un domaine ou n'existe encore aucune tradition, ni aucun consensus sur ce que peut être une bonne pédagogie de l'informatique au lycée. Il n'existe pas davantage de tradition universitaire pour la formation des maîtres : rares sont les universités qui y consacrent un peu de leur potentiel d'enseignement, et ceux qui assurent cette formation ont chacun leurs marottes... Je vais donc parler de ce que je fais, en essayant d'expliciter et justifier mes choix. Je dirai aussi ce qu'il me paraîtrait nécessaire de faire, mais que je ne peux réaliser faute de temps. Tout ceci doit être situé dans le contexte des objectifs pédagogiques de l'option. Il devrait être théoriquement possible de former les professeurs à l'informatique, puis de laisser chacun décider de sa propre pédagogie. Mais en raison de l'absence de tradition signalée plus haut, et vue la rareté des documents pédagogiques disponibles, c'est probablement demander l'impossible. Ce que je fais reste contingent, lié à une certaine vision de l'informatique. C'est un moindre mal, que j'espère temporaire...

2. L'INFORMATIQUE AU LYCÉE

Rappelons d'abord les objectifs de l'enseignement de l'informatique au lycée, tels qu'ils sont issus du colloque de Sèvres /1/. Les participants avaient éliminé un certain nombre de fausses pistes :

L'informatique est un phénomène socio-économique important. Il convient donc de l'enseigner au lycée. L'automobile, le réfrigérateur, la haute fidélité sont des phénomènes économiques importants : on ne les enseigne pas pour autant au lycée.

L'informatique a un fort impact sur notre culture. Il faut donc l'enseigner au lycée. Notons d'abord que cet impact n'est pas encore considérable. La télévision ou le cinéma ont un impact beaucoup plus important. On ne les enseigne pas au lycée.

Nous manquons d'informaticiens. L'expérience montre que l'enseignement supérieur suffit à les former, et si tout ne va pas pour le mieux, c'est dans une amélioration de son travail qu'il faut chercher la solution, non en commençant à former des programmeurs au lycée ou à l'école primaire.

Le colloque de Sèvres estimait que l'informatique est une science qui fait maintenant partie de la culture scientifique, et que la pratique de la programmation doit développer chez les élèves des "aptitudes algorithmiques, organisationnelles, opératoires".

Ce sont les objectifs qui ont été repris pour l'option informatique des lycées. D'abord contribuer à la formation de l'esprit : apprendre à lire un texte, en dégager puis formuler le problème qu'il pose, l'analyser, en trouver une méthode de résolution, la rédiger dans un langage de programmation, la mettre en œuvre sur ordinateur. Ensuite, connaître les grandes lignes des outils de l'informatique (matériel, logiciel), de ses applications et de son impact sur la société, afin d'être capable d'un jugement personnel et d'esprit critique : c'est l'insertion de l'informatique dans la culture générale.

Ce qui est frappant dans cette présentation, c'est le fait que l'enseignement de l'informatique n'est pas d'abord transmission de connaissances. Il ne s'agit pas de faire apprendre un savoir technique abondant. Il s'agit bien plutôt de développer des aptitudes intellectuelles :

- la créativité, pour l'invention de nouveaux algorithmes.
- la rigueur de pensée, pour que ces algorithmes résolvent le problème posé.
- la clarté et la précision d'expression, pour que les analyses et programmes soient communicables.
- le sens de l'organisation, car il faut venir à bout de masses de détails dans l'énoncé, ou de données dans les algorithmes.

Ces objectifs ne pourront être atteints que si les élèves pratiquent abondamment la programmation, comme le recommandait le colloque de Sèvres. Ainsi la programmation est au cœur de l'option comme un composant de base que l'on ne peut chercher à éliminer ou réduire sans

détruire l'option elle-même. Alliant ceci au fait que le savoir à transmettre est relativement peu dense. il apparaît fondamental que les professeurs maîtrisent parfaitement une bonne méthodologie de programmation : comment pourraient-ils développer chez les élèves des aptitudes qu'ils n'auraient pas eux-mêmes ? En conséquence, la formation des maîtres va tourner autour de deux pôles : une formation technique, une formation méthodologique.

3. LA FORMATION TECHNIQUE

L'enseignement doit fournir aux élèves une culture générale en informatique, leur permettant de faire preuve d'esprit critique face à tout ce qui est présenté dans les médias. Ils doivent donc savoir ce que peut réellement l'informatique, en connaissant ses outils et ses applications. La formation technique des maîtres porte sur ces points.

3.1 Les outils informatiques

Ce sont d'abord les outils matériels. Il faut savoir ce qu'est un ordinateur, et comment il fonctionne. Le niveau de détails auquel il convient de descendre est encore très incertain. Faut-il entrer dans quelques détails de la Circuiterie (portes, circuits ET, OU...) ou s'en tenir à une vue plus globale qui ne cherche pas à savoir comment sont faits les différents organes ? ne pas dire que le binaire est la base des ordinateurs actuels, faut-il parler des différentes façons de représenter un nombre en machine, de l'arithmétique des ordinateurs, des additionneurs ? Je pense personnellement que l'on ne devrait pas pousser trop loin les détails, parce que l'on n'a pas devant soi des groupes d'élèves cultives en mathématique ou en physique. et que les professeurs eux-mêmes viennent de disciplines diverses, pas toutes scientifiques. On peut comprendre la façon dont marche un ordinateur, et saisir comment ceci a conditionné la programmation que nous connaissons (ce qui est sans doute le point le plus important) en ignorant tout de la circuiterie.

Il me paraît par contre indispensable d'avoir une idée des performances des ordinateurs : capacité de mémoire, temps d'accès, capacité de stockage et temps d'accès des disques ou bandes magnétiques. Il faut avoir une idée de ce qui distingue un micro-ordinateur d'une super machine.

Il faut savoir qu'une machine nue est inutilisable : il faut un logiciel de base avec système d'exploitation (ce qui pose le problème de la

gestion des différents types de périphériques, et pour les gros systèmes, la question de partage de ressources et donc de protection), système de fichiers, compilateurs. Il semble nécessaire de parler un peu de ceux-ci. On pourrait parfaitement les considérer comme des boites noires prenant en entrée un texte en langage X et donnant en sortie un texte de même signification en langage Y. Mais les élèves sont en contact avec des compilateurs, voire en conflit... Il est intéressant de leur expliquer les difficultés à vaincre. En présentant des petits morceaux de compilateur (exemple d'analyse lexicale, c'est-à-dire de reconnaissance d'une unité syntaxique : entier, mot-clef ou identificateur... exemple de compilation d'une expression arithmétique simplifiée) on leur permet de mieux comprendre pourquoi ces programmes signalent que la virgule à tel endroit est une erreur, mais ne la remplacent pas automatiquement par un point, ou pourquoi les messages d'erreur sont si frustrants ! C'est aussi l'occasion de présenter la notion de grammaire formelle, et l'importance de la syntaxe en informatique.

Le marché s'est étendu dans la direction de logiciels puissants couvrant des secteurs assez larges d'activité, c'est-à-dire relativement peu spécialisés : éditeurs de texte ou de graphique, tableurs, gérants de bases de données... Il est intéressant de montrer de telles applications, ce qui suppose que les professeurs les connaissent. Mais il faut rester modeste et prudent. Le professeur n'a pas à se transformer en voyageur de commerce ! La "convivialité" de ces logiciels a été considérablement développée, c'est-à-dire que leur apprentissage est théoriquement facile. Alors pourquoi les enseigner ? En outre leur évolution est rapide. Faut-il investir dans le périssable ? Je suis personnellement très réservé face à ces logiciels. Ils ne me paraissent pas faire partie de la culture générale en informatique. En disant ceci, ne préjuge pas de leur possible utilisation à fins pédagogiques dans l'enseignement de l'option. On a parlé de "langage de quatrième génération" à propos des logiciels de gestion de bases de données. Il est parfaitement possible qu'ils soient utilisables pour le développement des facultés mentionnées plus haut. Ce sont des questions ouvertes, et que je ne saurais figer dans un enseignement alors qu'elles sont encore en pleine évolution. Je crois qu'il y a plus urgent.

3.2. Applications de l'informatique

L'objectif de cette partie a été rappelé : permettre au citoyen de savoir comment l'informatique intervient dans nos sociétés, et de faire preuve d'esprit critique en face des simplifications ou exagérations si

fréquentes dans les médias. Je n'ai aucune intention d'être exhaustif, et les exemples qui suivent ne sont donnés qu'à titre de suggestion :

- calcul scientifique (physique et chimie théoriques, études spatiales, aéronautique, météorologie...).
- conduite de processus (surveillance des usines de synthèse en chimie, des centrales nucléaires, des robots de fabrication...).
- traitement du signal (et notamment en médecine pour les scanners, l'échographie, les électrocardiogrammes ou encéphalogrammes...).
- assistance à la conception, à la fabrication.
- gestion (banques, assurances, paie du personnel, gestion de stocks de pièces, facturation... mouvement des professeurs !).
- système de réservation (avions. SNCF...).
- tenue de fichiers (fichiers médicaux, état civil, fisc, police...).

Une énumération d'applications serait sans intérêt. L'important est de savoir les points forts de celles qui sont présentées, pour avoir une idée du possible. Mais il faut en connaître aussi les faiblesses : une chaîne n'est pas plus solide que le plus faible de ses maillons, et l'informatique est trop souvent ce point faible, comme on le constate souvent dans le domaine spatial. Tout ceci est à resituer dans le contexte pédagogique de l'option.

Le problème est celui de la formation des maîtres. L'expérience montre qu'ils sont souvent insuffisamment informés dans ce domaine. Faut-il alors que je passe une partie de mes cours à présenter, ou faire présenter par des spécialistes ces différents types d'application ? J'ai très envie de répondre non. Je pense que ceci relève de l'auto-formation : il faut lire... Je souhaite que se multiplient les documents audiovisuels sur ces points. J'aimerais des vidéocassettes présentant ces grandes applications. Les professeurs commenceraient par les étudier. Si elles soulèvent des questions de fond, on peut en discuter en cours (ainsi par exemple, la conduite d'une fusée par ordinateur pose le problème du temps réel : on reçoit des mesures de la fusée. et l'on dispose d'une fraction de seconde pour les traiter, en déduire quelque chose sur la trajectoire, élaborer la correction à opérer, en déduire les ordres et les envoyer effectivement : comment peut-on tenir les délais ?) Je trouverais particulièrement fastidieux d'avoir à défiler ces applications l'une après l'autre. C'est pour moi typiquement le domaine de l'assistance, peut-être par ordinateur, certainement par magnétoscope.

Une place à part doit être faite à l'intelligence artificielle, parce que c'est un domaine encore beaucoup trop mal connu, et ou le partage entre le scientifique et le philosophique est particulièrement difficile. Il faudrait que les professeurs connaissent les grands thèmes de cette branche de l'informatique (reconnaissance des formes, démonstration automatique, résolution de problèmes, jeu d'échecs, traduction des langues, compréhension de la langue naturelle, systèmes experts), les problèmes scientifiques qu'elle pose (complexité des calculs, relation entre syntaxe et sémantique, représentation des connaissances, méthodes heuristiques). Il faudrait qu'ils aient une idée de ce que cela cache, la meilleure façon d'y parvenir étant probablement de traiter des exemples réduits faisant comprendre les démarches (j'ai tenté ceci dans mon livre de jeux et casse-tête à programmer /4/ avec un exemple d'apprentissage dans un jeu ou plusieurs stratégies sont possibles pour l'ordinateur, un exemple de théorème démontrable sur ordinateur, des méthodes heuristiques pour le jeu des chiffres...).

J'ai mis tout ceci au conditionnel, parce qu'il n'est pas sûr que ce soit actuellement faisable, dans les temps impartis à la formation des maîtres. Au demeurant, il ne faut pas perdre de vue les objectifs : l'intelligence artificielle est un des secteurs ou il est le plus fait appel à l'esprit critique du citoyen, parce que les présentations faites par les médias sont trop trompeuses. Certains minimisent trop. D'autres, anticipant allègrement sur l'avenir, annoncent que la machine sera sous peu plus intelligente que l'homme. Il faut permettre à l'élève de s'y retrouver, et de choisir son camp, parce que rien de ceci n'est innocent. C'est l'homme qui est en question : ou bien il est machine et la machine le dépassera (point de vue réductionniste), ou bien il est esprit et d'une autre nature que l'ordinateur (point de vue spiritualiste). Ce n'est pas une affaire neutre.

3.3. Implications de l'informatique

Ceci nous met de plain-pied dans le troisième volet de la culture informatique. Cette science interagit avec la culture et la société. Elle pose des problèmes épistémologiques : qu'est-elle, quel est son objet, quelles sont ses méthodes ? Les questions de l'intelligence artificielle appellent une réflexion à ce niveau.

L'informatique modifie notre relation au savoir, par exemple dans l'enseignement assisté par ordinateur, mais aussi dans l'informatique documentaire ou par les banques de données. Il faut en parler aux élèves,

ce qui suppose que les professeurs y aient réfléchi, et donc qu'on leur en ait parlé...

L'informatique modifie les conditions de travail, agissant sur sa pénibilité (robots, traitement de texte), mais en introduisant d'autres nuisances (travail devant l'écran cathodique). Qu'en est-il du travail à domicile (télétraitement...) ? Agit-elle sur les communications (messagerie électronique à travers les réseaux d'ordinateurs, télématique) ? Agit-elle sur l'emploi (parallèle avec l'affaire des canuts) ? Est-elle une menace pour les libertés ? Détruit-elle la notion de propriété du créateur (piratage des logiciels) ?... Toutes ces questions sont importantes. Comment les professeurs s'y préparent-ils ? En cours ? Là encore je souhaite une importante assistance par des moyens audiovisuels...

4. LA FORMATION MÉTHODOLOGIQUE

Le professeur doit amener l'élève à être capable de créativité, tout en gardant la maîtrise intellectuelle de sa création. Il doit l'aider à avoir une pensée claire et structurée, et à savoir s'organiser. Il n'y a pas une réponse unique à ce défi pédagogique. Mais je crois que toutes les réponses reposent sur la même base : une solide méthodologie de la programmation.

Les tenants de l'empirisme disent qu'il faut faire confiance aux aptitudes spontanées des élèves. On leur enseigne un langage de programmation, le reste étant affaire de bon sens /2/. J'avais entendu dire que 15 % des enfants ont en fait ces aptitudes spontanées qui leur permettent d'être créatifs en programmation. Je viens de lire un article de Donald Knuth, très célèbre professeur à Stanford, qui ramène ce chiffre à 2% /3/.

Si l'on ne peut se fonder sur les aptitudes naturelles, alors il faut les développer par la pratique d'une méthode, à la façon dont la gymnastique développe les aptitudes corporelles. Pour les élèves, l'apprentissage d'une méthodologie ne peut résulter de cours théoriques. Il en va un peu différemment avec les professeurs. Je peux attendre d'eux qu'ils acceptent un discours plus formel sur une approche méthodique de la programmation, ce qui a pour effet d'en raccourcir la présentation. Je dis ce qu'elle est, j'en donne quelques exemples. Je suppose les professeurs assez grands pour comprendre qu'il faut qu'ils pratiquent abondamment et fassent des exercices, en s'appuyant sur la méthode formelle qui leur a été proposée.

Personnellement, je m'appuie sur la méthode que j'ai mise au point : l'action se déduit de la connaissance des situations de départ et d'arrivée. Résoudre un problème, c'est expliciter la situation des données (départ), celle des résultats (arrivée). Si la distance entre les deux est trop grande, on jalonne le chemin de situations intermédiaires, qu'il faut inventer : là est le processus créatif. Mais on a des guides, qui limitent le champ de recherche, quelque chose comme les heuristiques de l'intelligence artificielle. Ce ne sont pas des méthodes infaillibles : il n'y a pas d'algorithme universel de résolution de problèmes ! Ce sont des suggestions, tirées de l'expérience, qui valent toujours la peine d'être essayées. Quand il leur arrive d'échouer, l'analyse de l'échec fournit des indications sur une autre façon de repartir.

Il me semble nécessaire de faire comprendre aux professeurs pourquoi l'informatique a du faire face à un problème nouveau en matière de résolution de problèmes. Il y a plus de 2 000 ans que les mathématiciens en résolvent. Pourquoi a-t-il fallu innover ? Un peu de structure des ordinateurs, et d'histoire de l'informatique, est indispensable ici. Après quoi, je montre comment l'instruction d'affectation est une conséquence directe de cette structure. J'en montre la nature transformationnelle : elle change la valeur d'une variable ; il y a une situation avant Son exécution, et une autre, différente, après. Le programme dit la suite de transformations qui fait passer de la situation initiale à la situation finale. Il ne se comprend qu'à travers cette histoire des situations. En conséquence, il ne peut vraiment se construire qu'en s'appuyant sur cette histoire.

Tout ceci, je peux le dire à des professeurs, et je crois qu'il faut le leur dire. Comment pourraient-ils enseigner proprement quelque chose dont ils n'auraient pas compris la raison d'être ? Bien entendu, ce n'est pas ainsi que ce doit (ou peut) être présenté aux élèves. D'où la nécessité de parler de pratiques pédagogiques.

Il faut aussi élargir l'horizon. Il n'y a pas qu'une méthode d'analyse. Il y a trois modes de programmation : la programmation constructive (ou impérative, celle qui utilise l'instruction d'affectation), la construction implicite (ou récursive, ou fonctionnelle, ou applicative...), la programmation logique. Ce sont trois modes de pensée différents. Pour chaque mode, il y a différentes méthodes pour élaborer un programme. Il semble important de mettre les professeurs en face de cette multiplicité, afin de leur offrir un choix. Chacun, suivant ses charismes, choisira la voie de l'efficacité maximum.

Mais on ne peut empêcher le formateur d'avoir ses propres charismes, ou ses propres manies ou obsessions. Peut-être parce que je ne suis pas très inventif (je ne "voyais" pas la solution des problèmes de géométrie au lycée), je privilégie les méthodes constructives. Il me parait inutile, voire néfaste, de donner des algorithmes tout faits aux élèves. Si on ne peut pas leur expliquer d'où ils viennent, comment on les a inventés, pourquoi ils résolvent le problème posé, comment ils répondent à une sorte de nécessité, alors il vaut mieux ne pas les présenter. C'est ce que je m'efforce de faire comprendre aux professeurs.

Tout ceci se fait par un mélange d'informatique (j'étudie devant eux des problèmes en essayant de décortiquer le chemin de leur résolution) et de méta-informatique, réflexion sur les activités mises en jeu dans ces résolutions, sur les choix qui sont faits (tout choix est le refus d'autre chose).

5. VALIDATION

Jusqu'à présent, le problème de la validation des maîtres pour l'enseignement de l'informatique ne s'est pas pose de façon grave. Le nombre de lycées engagés dans l'option était petit. Nous ne formions chaque année qu'un petit nombre de professeurs. Le contrôle "personnel" suffisait : je connaissais les professeurs qui travaillaient avec moi, et je pouvais témoigner de leur aptitude. Mais l'option a été banalisée. Elle se répand. Il y a des professeurs formés hors des circuits usuels : pourquoi ne seraient-ils pas capables d'enseigner l'option ? Dans le même temps, une illusion de formation règne encore de façon dangereuse : quiconque a pu écrire 3 ou 4 petits programmes en BASIC est capable d'enseigner l'option. Qui pourra convaincre les proviseurs que ce n'est pas vrai, que l'enseignement de l'option demande d'autres qualités, et qu'en particulier tout professeur de mathématiques n'est pas ipso-facto apte à enseigner l'option ? On a déjà vu sur le terrain des phénomènes aberrants. Il faut un mécanisme régulateur.

Plusieurs formules ont été proposées. On a évoqué la possibilité d'options informatiques dans les CAPES et agrégations de différentes disciplines. C'est une possibilité intéressante, même si elle soulève quelques difficultés. Qui appréciera l'aptitude à enseigner ? Ce devrait être un informaticien. Quel sera son statut dans les jurys ? Trouvera-t-on assez d'informaticiens disponibles pour un nombre peut-être grand de jurys siégeant longtemps ? On peut aussi créer un diplôme universitaire

national d'aptitude à l'enseignement de l'informatique. C'est beaucoup plus simple à faire fonctionner, même si la création passe par un peu d'administration. Au demeurant, c'est une question secondaire. L'urgent est d'instituer un mécanisme de régulation.

La forme de l'examen est plus critique. S'il s'agissait d'un enseignement où l'important est le transfert de connaissances il suffirait de contrôler les connaissances des professeurs : vivent les questions de cours ! Si l'on veut se placer dans la perspective de cette nouvelle relation au savoir évoquée plus haut, on organisera une épreuve sur documents, demandant au professeur de faire preuve d'un savoir synthétisé.

Mais une large part de l'aptitude à enseigner, la plus critique en tous cas, c'est la capacité à développer chez les élèves de bonnes aptitudes mentales. On ne le vérifiera pas par des questions de cours. On peut envisager des épreuves sur le style des exercices dans les concours de mathématiques ou physique. On fournira au professeur un texte de travail (je pèse mes mots : pas un énoncé de problème, ce sera au candidat d'énoncer le problème). Il disposera de plusieurs heures (4 ou 6) pour étudier le texte, formuler le problème, l'analyser, le résoudre, le programmer, vraisemblablement l'essayer sur ordinateur. Le jury lui demandera alors d'en exposer non la solution, mais la façon dont il l'a trouvée, les choix qu'il a faits, comment la solution proposée fournit la réponse cherchée, et pourquoi c'est la bonne réponse. Exercice difficile, mais faisable. Ceci respecte les méthodes personnelles d'analyse. Il faut du temps. Mais il n'y a pas encore trop de candidats, et l'affaire parait jouable.

Personnellement, j'ai expérimenté une autre formule. J'ai proposé aux professeurs qui ont travaillé avec moi cette année (et dont certains le faisaient dans le cadre du DEA de didactique des disciplines scientifiques, à PARIS VII) un pseudo-texte d'examen pour élève de terminale, avec une pseudo-copie de lycéen. Le professeur devait corriger la copie, puis le sujet (quant à sa formulation, sa validité face aux objectifs de l'option, le niveau de l'épreuve, etc.). Les professeurs ont passé un peu moins de 4 heures sur cette épreuve, et m'ont remis leurs copies.

J'ai été surpris de voir combien cette épreuve était significative. Les qualités personnelles des professeurs y étaient fort manifestes : rapidité de lecture d'un programme, capacité à en voir les erreurs, attitude en face d'une analyse mal conduite et mal présentée (en tant que pseudo-élève, j'avais mis quelque malice en ce point, réunissant en une seule copie des bêtises glanées dans les copies de l'épreuve libre 1985).

Leurs préoccupations stylistiques étaient tout aussi évidentes. Le sujet comportait une épreuve de culture générale. Là aussi, leur compétence se manifestait.

Quant à la méthodologie, l'épreuve fut en un sens trop probante : il était clair que je n'ai pas atteint cette année l'objectif que je m'étais fixé. Après coup, il me semble que j'aurais dû faire plutôt l'effort que l'EPI m'a demandé : mettre noir sur blanc les problèmes de la formation des maîtres. Cela aide à voir clair. Peut-être le ferai-je l'an prochain. L'espoir fait vivre...

Jacques ARSAC
École Normale Supérieure

BIBLIOGRAPHIE

/1/ *Séminaire sur l'enseignement de l'informatique à l'école secondaire*. Publications de l'OCDE, Paris, 1970.

/2/ Michel LACROIX - *La micro-informatique reflet des mœurs*. Esprit, février 1985.

/3/ Donald E. KNUTH *Algorithmic thinking and mathématical thinking American Mathematical Monthly*, Vol.92 n° l, Janvier 1985, p. 170-181.

/4/ Jacques ARSAC - *Jeux et casse-tête à programmer*. Paris, Dunod, 1985.