

## A- Remarques générales

L'épreuve proposée aborde la plupart des aspects fondamentaux du programme de l'option Informatique : automate, preuve et complexité des algorithmes, programmation en CaML ou Pascal. Seule la logique (modélisation et résolution) n'est pas soumise aux candidats. La forme et le contenu du sujet sont dans la continuité des sujets précédents. La longueur et la difficulté de l'épreuve sont comparables aux années précédentes. Les correcteurs ont noté une amélioration des résultats par rapport à l'année précédente. Par contre, le sujet d'algorithmique reprenait un thème déjà abordé dans une épreuve précédente avec des algorithmes différents plus abordables cette année et les résultats ont été de moins bonne qualité, ce qui dénote quand même une baisse dans la capacité à traiter ce genre de problèmes depuis l'introduction de l'option Informatique.

**La partie I** aborde les automates et langages. L'exercice proposé aux candidats est classique dans le cadre du concours commun polytechnique : étudier un opérateur de transformation d'automates. L'opérateur retenu cette année correspond à la déterminisation. Il s'agit donc d'un sujet abordé explicitement dans le cadre des enseignements qui peut donc être considéré comme une question de cours. L'exercice propose, d'une part, d'étudier l'effet de l'opérateur sur un exemple et, d'autre part, de prouver formellement l'effet remarqué pour tout automate. Les preuves se font par induction sur la structure des mots (ou récurrence sur la taille des mots). Ces deux points correspondent au minimum de ce que les candidats doivent être capables de faire dans ce domaine. Une fois de plus, les correcteurs ont noté que les candidats n'ont pas de problèmes majeurs avec le traitement des exemples mais ne sont pas capables, dans la majorité des cas, de faire correctement la partie preuve. Celle-ci est donc plus discriminante. Les correcteurs regrettent que, malgré les fortes similitudes avec les sujets des années précédentes et la grande proximité avec le contenu des enseignements, la grande majorité des candidats n'aient pas traité cet exercice de manière satisfaisante.

**La partie II** traite de l'algorithmique et la programmation dans le contexte de la géométrie : il s'agit de construire une enveloppe convexe pour un ensemble fini de points en s'appuyant sur des arbres enveloppant pour chaque demi-enveloppe supérieure et inférieure. Cette partie concerne trois éléments du programme : preuve, complexité et programmation. Les structures de données manipulées sont, d'une part, des ensembles de points codés par des listes et, d'autre part, des arbres binaires de points. Le sujet permet ainsi d'évaluer les deux parties du programme d'algorithmique. Une grande partie de ce problème n'aurait pas dû poser de problèmes à la majorité des candidats, ce qui n'a pas été le cas.

Cette seconde partie est significativement plus longue que la première. Les questions posées sont de difficulté croissante partant de questions classiques présentées dans tous les cours jusqu'à des aspects plus neufs pour les candidats. Cette partie a pris une part importante dans la distinction entre candidats.

La longueur du sujet est légèrement plus courte que les années précédentes depuis le passage de trois parties (logique, automate, algorithmique) à deux parties (algorithmique et soit logique, soit automate). Cette année, les correcteurs ont noté une baisse significative dans le nombre de questions traitées sans que la difficulté par rapport aux années précédentes ne puisse le justifier.

Avant de détailler de chaque partie, il est nécessaire de faire quelques remarques globales sur l'attitude des candidats et les changements notables par rapport aux années précédentes :

- l'ensemble des correcteurs ont noté une amélioration par rapport à l'année précédente sans que le niveau des copies ne retrouve le palier constaté auparavant. La qualité de la programmation n'évolue pas. Par contre, la baisse remarquée sur la qualité du traitement formel des questions concernant les automates, la complexité et les preuves reste notable même s'il y a cette année une légère amélioration. Même les questions les plus simples ne sont pas traitées avec la rigueur nécessaire. Et quand elles le sont, la rigueur ne subsiste que rarement au niveau des questions plus complexes. Le cas de l'induction (ou récurrence) devient parfois une caricature. Le raisonnement sur des ... pour montrer qu'il suffit d'appliquer les règles autant de fois que nécessaire apparaît de plus en plus souvent. Le cas d'arrêt est régulièrement négligé. Les correcteurs tiennent encore une fois à insister sur le fait que l'induction est la base de toute formalisation en Informatique et que sa maîtrise est essentielle comme la récurrence l'est en Mathématique. Les difficultés rencontrées par les candidats semblent être exactement les mêmes que les années précédentes dans un sujet qui a exactement la même forme. L'ensemble des correcteurs est toujours extrêmement surpris que les candidats ne se préparent pas mieux à répondre à ce type de questions.
- Le rapport entre le nombre de candidats utilisant CaML et PASCAL reste stable. La différence de qualité l'est malheureusement aussi. Il est une fois de plus nécessaire d'insister sur le fait que CaML est plus adapté que PASCAL au traitement des sujets contenus dans le programme. La sélection de PASCAL au lieu de CaML dans une formation introduit un déséquilibre certains entre les candidats dans les épreuves.
- La qualité de la rédaction des copies est stable cette année. Mais elle fait suite à la forte baisse notée l'année dernière. Les candidats ne semblent toujours pas avoir acquis le fait qu'une réponse correcte mal rédigée sera en général considérée comme fausse en partie, voire totalement, dans certains cas. Les candidats suivent une formation scientifique qui les conduit vers un métier d'ingénieur dans des disciplines scientifiques. Les normes actuelles de qualité dans les différents domaines de l'industrie exigent de plus en plus une documentation précise et sans ambiguïté. Leur futur métier exigera d'eux un changement radical dans la manière dont ils rédigent et présentent actuellement leurs documents.

## Remarques particulières

### Partie I - Automates et langages

Les automates et expressions régulières représentant le langage accepté par un automate sont maintenant un domaine que les candidats connaissent bien. Les correcteurs signalent, par contre, que la qualité des preuves sur la structure d'un langage est en baisse très significative. Les raisonnements par induction sur la structure du langage (ou par récurrence sur la longueur des mots) sont de plus en plus souvent remplacés par des argumentations informelles ou des manipulations de « $a^1 \dots a^N$ » pour indiquer que le mot est composé d'un nombre quelconque de caractères et par des allusions à «appliquer les règles autant de fois que nécessaire». Ce comportement est inacceptable. L'induction est la technique de preuve la plus utilisée en Informatique et elle doit être maîtrisée par tous les candidats sans exception. L'exercice proposé aux candidats est très proche de ceux posés les années précédentes et s'appuie directement sur le contenu des enseignements (déterminisation des automates). L'étude consciencieuse des épreuves précédentes doit leur permettre de traiter cet exercice sans la moindre difficulté, par un simple mimétisme, même sans maîtrise complète des techniques. Les correcteurs ont également noté un oubli régulier des conditions d'arrêt chez les candidats, que la forme du raisonnement soit satisfaisante ou pas.

Après avoir atteint un pic jusqu'à l'année dernière, la régression du niveau de qualité, dans le

traitement de cet exercice classique se confirme.

#### Question I.2 (construction des automates)

Dans la grande majorité des cas, les réponses proposées par les candidats sont correctes.

#### Question I.1 et I.3 (description et construction des langages)

Le formalisme des expressions régulièrement est maintenant acquis de manière satisfaisante. Les langages proposés pour la question I.1 sont corrects dans la majorité des cas. Le I.3 a posé plus de problèmes car l'automate était significativement plus complexe. La majorité des candidats n'a pas répondu ce qui est une surprise par rapport aux années précédentes

#### Question I.4

Cette question classique est souvent traitée de manière incomplète. Il suffit de répondre exactement aux deux points cités dans le sujet : finis et déterministes. Les candidats répondent souvent à d'autres questions non présentes dans le sujet et ne répondent pas à une des deux.

#### Question I.5 et I.6 (preuve par induction)

Cette année, la preuve d'égalité des langages reconnus par l'automate initial et l'automate déterminisé était quasiment une question de cours qui devait être adaptée aux notations de l'énoncé. Globalement, ces questions n'ont pas été traitées de manière satisfaisante. Souvent, les questions étaient tout simplement ignorées. La grande majorité des solutions proposées n'étaient pas rédigées correctement. Certains correcteurs ont noté une mauvaise interprétation du mot vide ou des inductions qui prenaient comme cas d'arrêt le mot composé d'un seul symbole.

#### Question I.7

La plupart des candidats qui ont traité les questions précédentes répondent correctement. Certains candidats donnent des réponses intuitives sans la moindre preuve.

## Partie II - Algorithmique et programmation

Le problème soumis aux candidats était l'étude de la structure d'arbre enveloppant dont le but est la construction de l'enveloppe convexe d'un ensemble fini de points par union des arbres enveloppant des demi-enveloppes inférieures et supérieures. Cette structure permet de réduire le nombre d'opérations nécessaires pour construire une enveloppe convexe dans le cas où le nombre de points sur le contour est très petit par rapport au nombre de points total. Les questions étaient de difficulté croissante en s'appuyant sur les représentations de l'ensemble des points par une liste et un arbre binaire de recherche à deux dimensions s'appuyant sur un ordre lexicographique (ordonnée puis abscisse). Les candidats n'ont en général pas pu traiter l'application de la structure d'arbre enveloppant à la construction de l'enveloppe convexe (section 4).

Cette année encore, les correcteurs ont noté unanimement que les candidats produisent de meilleures réponses dans les questions de programmation que dans le traitement rigoureux des algorithmes, que ce soit la preuve ou le calcul de complexité. Ceci confirme le défaut habituel qui est de plus en plus incompatible avec un métier d'ingénieur en Informatique. Les systèmes complexes réalisés actuellement dans un cadre professionnel sont soit critiques pour l'être humain (logiciels embarqués dans des systèmes de transport ou des outils médicaux, ...) ou pour son environnement (systèmes bancaires, ...). Seul un traitement rigoureux permet d'obtenir des résultats satisfaisants. Il est essentiel de faire comprendre au plus tôt que l'Informatique est devenu un domaine industriel comme les autres et que la programmation n'est plus qu'une partie d'un ensemble, et qu'il est nécessaire de comprendre et de maîtriser l'ensemble des techniques. De plus, la complexité croissante des systèmes font que la pratique industrielle se tourne de plus en plus vers

la génération automatique de programmes à partir de la conception et la vérification des solutions exprimées de manière plus abstraites (cadre de l'ingénierie dirigée par les modèles).

Les correcteurs ont signalé que la plupart des candidats se sont arrêtés au niveau de la section 4.

#### Questions II.1, II.2, II.4, II.5, II.7, II.11, II.15, II.19, II.20, II.22, II.23 (programmation)

Les correcteurs ont signalé qu'ils s'agissaient des questions qui ont largement été le plus et le mieux traitées par les candidats. Les solutions proposées sont globalement correctes. Les contraintes d'utilisation des langages que ce soit CaML ou PASCAL ont été globalement bien respectées.

Les correcteurs insistent une fois de plus sur le fait que les réponses en PASCAL sont plus longues que les réponses en CaML car le langage est beaucoup moins bien adapté à l'exploitation des structures de liste et d'arbre. Les candidats sont ensuite pénalisés en conséquence par un temps moindre disponible pour traiter les autres questions.

Certains candidats proposent des réponses très longues sans fonction intermédiaire et sans explication. La pertinence de la forme d'un programme est prise en compte dans l'évaluation de sa qualité. Il serait important d'insister sur ce point pour que les candidats décomposent mieux leurs fonctions en sous-fonctions plus simples dans les cas où cela n'est pas directement imposé par le sujet.

Par contre, d'autres candidats passent beaucoup de temps à expliquer systématiquement chaque fonction en détail alors que le sujet ne l'exige pas. Une fonction correctement programmée suffit si le sujet ne demande pas d'explication spécifique. En cas de fonctions particulièrement complexes, le candidat peut fournir des éléments d'explication mais, en général, sa fonction devrait être plutôt décomposée en sous-fonctions plus simples et compréhensibles sans commentaires.

#### Question II.3, II.8, II.9, II.10, II.12, II.13, II.14, II.16, II.17, II.18 (preuve algorithme)

Globalement, les correcteurs ont noté que les candidats ne répondent pas à ce type de question avec la rigueur nécessaire à une preuve mathématique de correction de l'algorithme proposé par le sujet. Ceux-ci s'appuient souvent sur une paraphrase du sujet et négligent de nombreux cas. Ces questions se révèlent donc une fois de plus très discriminantes. Les correcteurs signalent que cette partie du programme est essentielle et qu'il n'est pas acceptable d'exploiter un programme sans que celui-ci soit validé. Or, la meilleure technique de validation est la preuve que l'algorithme est correct et que sa complexité est compatible avec son contexte de fonctionnement. Ces approches prennent une part de plus en plus importante dans les contextes industriels critiques. Il est donc essentiel que les candidats soient aptes à répondre correctement à ce type de questions.

#### Questions II.6, II.10, II.21, II.24 (complexité)

Les questions posées en début de sujet étaient très simples et ont été traitées de façon convenable par les candidats. Les exemples proposés étaient en général pertinents. Les dernières questions n'ont pas été traitées probablement par manque de temps.

## Conclusion

Le sujet comportant deux parties indépendantes a permis d'évaluer les compétences des candidats sur la majeure partie du programme. La qualité des réponses soumises est en léger progrès par rapport à l'année précédente mais confirment le pallier remarqué les années précédentes. La régression notée au niveau de la qualité des preuves dans le traitement formel des automates et des algorithmes ne s'est pas aggravée mais aucune amélioration notable n'est visible. La faible difficulté

des premières questions dans les deux parties auraient dû permettre à tous les candidats de fournir des réponses satisfaisantes. La progressivité des questions suivantes aurait dû permettre de distinguer les candidats selon leur vitesse et leur capacité de répondre aux questions les plus complexes. Un progrès est notable mais les résultats ne sont pas satisfaisants.

Une fois de plus, les correcteurs ont été surpris par la manière dont la majorité des candidats traitent les questions faisant appel à des arguments formels. L'avis global est que les candidats ne semblent pas avoir conscience qu'un problème d'Informatique est avant tout un problème scientifique et doit être traité avec la même rigueur qu'un problème de mathématiques.

Les correcteurs ont globalement notés que la programmation et la manipulation d'exemples étaient satisfaisantes mais que le traitement formel devait encore très nettement progresser.