

Remarques générales

L'épreuve proposée traite la plupart des points fondamentaux du programme de l'option Informatique : automate, preuve et complexité des algorithmes, programmation en CaML ou Pascal. Seule la logique, que ce soit la modélisation ou la résolution n'est pas proposée aux candidats. La forme et le contenu du sujet sont dans la continuité des sujets proposés les années précédentes. La longueur et la difficulté de l'épreuve sont comparables aux années précédentes. Les correcteurs ont pourtant noté un fléchissement très net des résultats.

La **partie I** traite des automates et langages. Le sujet proposé est commun au sein du concours commun polytechnique : étudier un opérateur de composition d'automates. L'opérateur retenu cette année correspond à l'entrelacement des langages. Il s'agit d'un opérateur utilisé dans la modélisation de systèmes événementiels concurrents. 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 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. Tout en étant classique, l'exercice proposé était un petit peu plus difficile que les années précédentes. Les correcteurs regrettent que, malgré les fortes similitudes avec les sujets des années précédentes, la grande majorité des candidats n'ait pas traité cet exercice de manière satisfaisante.

La **partie II** traite de l'algorithmique et la programmation dans le contexte des structures de données optimales : il s'agit d'étudier les arbres équilibrés d'Adelson-Velskii et Landis. Cette partie concerne trois éléments du programme : preuve, complexité et programmation. Les structures de données manipulées sont, d'une part, des ensembles codés par des listes et, d'autre part, des arbres binaires. Le sujet permet ainsi d'évaluer les deux parties du programme d'algorithmique. Dans une première étape, les candidats manipulent des arbres binaires simples, puis des arbres binaires de recherche avant de s'attaquer aux arbres équilibrés. Cette première étape n'aurait donc dû poser aucun problème particulier, 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 sujets classiques, a priori traités dans tous les cours, jusqu'à des aspects plus novateurs pour les candidats. Cette partie a pris une part importante dans la distinction des différents candidats.

La longueur du sujet est comparable aux années précédentes depuis le passage de trois parties (logique, automate, algorithmique) à deux parties (algorithmique et, soit logique, soit automate). Cette longueur était déjà trop importante les années précédentes. 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 puissent le justifier. L'impression acquise depuis le début de l'introduction des épreuves d'Informatique et de Sciences Industrielles se confirme une fois de plus : chaque année, les candidats vont moins loin dans des sujets de difficulté globalement identique. Le passage de la durée de l'épreuve de deux à trois heures avait permis de compenser ce phénomène.

Avant de rentrer dans le détail de chaque partie, faisons quelques remarques concernant l'attitude globale des candidats et les évolutions remarquées par rapport aux années précédentes :

- Contrairement aux années précédentes où l'ensemble des correcteurs réguliers s'accordait sur le fait que le niveau des copies semblait avoir atteint un palier, un net fléchissement se fait sentir cette année à la fois sur le nombre de questions traitées et sur la qualité du traitement. La qualité de la programmation reste globalement constante. Par contre, il y a une baisse remarquable sur la qualité du traitement formel des questions concernant les automates, la complexité et les preuves. 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 pourtant à insister sur le fait que l'induction est la base de toute formalisation en Informatique et que sa maîtrise est essentielle. 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 réellement surpris que les candidats ne se préparent pas mieux à répondre à ce type de questions et se demandent si, d'une part, ce type d'épreuve, et d'autre part le programme du concours, que ce soit son contenu ou le temps qui lui est consacré, est suffisant pour préparer les candidats actuels.
- Le rapport entre le nombre de candidats utilisant CaML et PASCAL semble rester stable. La différence de qualité reste malheureusement elle aussi stable cette année. Les correcteurs veulent insister une fois de plus sur le fait que CaML est plus adapté que PASCAL au traitement des sujets contenus dans le programme.
- La qualité de la rédaction des copies semble encore une fois être en baisse. Il est très regrettable que les candidats n'aient pas acquis le fait qu'une réponse correcte mal rédigée sera, en général, considérée comme fautive en partie, voire totalement dans certains cas. Les candidats sont engagés dans une formation vers un métier d'ingénieur dans des disciplines scientifiques et techniques. 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

Le formalisme des automates et des expressions régulières représentant le langage accepté par un automate est maintenant un domaine que les candidats connaissent bien. Les quelques erreurs concernent principalement les états initiaux ou terminaux. Les correcteurs signalent par contre que les preuves sur la structure d'un langage manquent majoritairement de rigueur. Les raisonnements par induction sur la structure du langage (ou par récurrence sur la longueur des mots) sont trop 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, quoique un peu plus long et difficile. 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, le niveau de qualité dans le traitement de cet exercice classique est maintenant en baisse ce qui préoccupe les correcteurs.

Question I.3 (construction des automates)

Dans la grande majorité des cas, les réponses proposées par les candidats sont correctes. Les rares erreurs concernent les états initiaux ou terminaux.

Question I.1, I.2 et I.4 (description et construction des langages)

Le formalisme des expressions régulières est maintenant acquis de manière satisfaisante. Les langages proposés pour la question I.1 sont corrects dans la majorité des cas. En ce qui concerne le I.2, certains cas sont parfois oubliés parmi les 6 mots du résultat. Le I.4 a posé plus de problèmes car l'automate était significativement plus complexe. La plupart des candidats ont oublié des chemins possibles dans l'automate. Les résultats pour les chemins pris en compte étaient par contre corrects dans la plupart des cas. Certains correcteurs ont noté une mauvaise compréhension de la notion d'état terminal. Les candidats concernés arrêtent le parcours dès l'arrivée dans un état terminal sans prévoir qu'il est possible d'en repartir si des transitions adéquates existent.

Question I.5

Cette question classique est souvent traitée de manière incomplète. Il suffit de répondre exactement aux deux points cités dans le sujet : finis et semi-indéterministes. Les candidats répondent souvent à d'autres questions non présentes dans le sujet et ne répondent pas à une des deux. La question et sa réponse sont peut-être trop simples.

Question I.6, I.7 et I.8 (preuve par induction)

Cette année, la preuve d'égalité du langage reconnu par l'automate composé et du langage résultant de la composition des langages reconnus par chaque automate était significativement plus complexe. Le sujet proposait donc une décomposition progressive en 4 questions. Les 3 premières demandaient des preuves par induction de plus en plus sophistiquées. Globalement, ces trois 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.9

La plupart des candidats qui ont traité les questions précédentes répondent correctement. Certains candidats donnent des réponses intuitives sans la moindre preuve. Bien entendu, ces réponses n'ont aucune valeur.

Partie II - Algorithmique et programmation

Le problème proposé consistait en l'étude de la structure d'arbre binaire de recherche équilibré proposée par Adelson-Velskii et Landis dans le but d'optimiser le codage d'un ensemble d'entier vis à vis du test d'appartenance et de l'insertion. Les questions étaient de difficulté croissante en s'appuyant sur plusieurs représentations de l'ensemble par une liste, puis un arbre binaire, puis un arbre de recherche et enfin un arbre équilibré.

Le sujet s'est malheureusement révélé trop long et les candidats n'ont, en général, pas pu traiter la partie arbre équilibré.

Cette année encore, les correcteurs ont noté unanimement que les candidats sont très nettement plus performants 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.

Les correcteurs ont signalé que la plupart des candidats se sont arrêtés au niveau de la partie 2.3.

Certains candidats mélangent les notions de profondeur et de déséquilibre d'un nœud. Les programmes et les preuves qui en découlent sont donc incorrects.

Questions II.1, II.3, II.7, II.8, II.10, II.11, II.14, II.19, II.20, II.23, II.24 (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 même si les candidats ont beaucoup de mal à prendre en compte la contrainte de parcours unique des structures. En général, chaque fonction ne parcourt qu'une fois la structure mais les candidats combinent parfois certaines fonctions et font ainsi plusieurs parcours. Les contraintes d'utilisation des langages que ce soit CaML ou PASCAL ont été très bien respectées.

Globalement, les correcteurs insistent encore sur le fait que les réponses en PASCAL sont plus longues que les réponses en CaML et que les candidats semblent pénalisés en conséquence.

Certains candidats proposent des réponses très longues sans fonctions intermédiaires et sans explications. 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 que pour la question II.25. Une fonction correctement programmée suffit si le sujet ne demande pas d'explications spécifiques. 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. Dans tous les cas, les commentaires détaillés exposés au début du sujet ne doivent être proposés que pour les questions qui le demandent explicitement.

Question II.5, II.6, II.13, II.16, II.17, II.18, II.21, II.22 (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.

II.6 : De nombreux candidats connaissent par cœur la formule calculant la profondeur d'un arbre en fonction du nombre de nœud mais ne sont pas capables de retrouver sa preuve. Le caractère minimal ou maximal n'est souvent pas justifié.

II.13 : La preuve est souvent incomplète. Les cas $E(a) = v$ et $D = 0$ sont souvent oubliés.

II.17 : La plupart des candidats refont une preuve complète pour les rotations doubles alors qu'il suffit de s'appuyer sur la composition de deux rotations simples.

Question II.25 (explication des programmes)

Cette question n'a été traitée que par un très petit nombre de candidats à cause de sa position vers la fin du sujet. Les réponses proposées suivent le schéma décrit en début de sujet mais ne sont pas très satisfaisantes. Les correcteurs font l'hypothèse que les candidats ont manqué de temps pour donner des réponses de bonne qualité.

Questions II.2, II.4, II.9, II.12, II.15, II.26, II.27, II.28, II.29, II.30 (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.

Lors du parcours d'une liste avec un traitement en $O(n)$ pour chaque élément, certains candidats proposent une complexité en $O(2n)$ donc $O(n)$ au lieu de $O(n*n)$.

Les questions sur les arbres étaient par contre plus complexes et les explications fournies par les candidats étaient en général très insuffisantes.

Certains candidats ne prennent en compte que les appels récursifs à la fonction en cours et oublient la complexité des sous-fonctions appelées.

Les dernières questions de synthèse sur la complexité des arbres équilibrés n'ont pas été traitées par les candidats par manque de temps.

Conclusion

Le sujet composé de deux parties indépendantes a permis de mesurer les capacités des candidats sur une grande partie du programme.

La qualité des réponses proposées confirme le pallier noté les années précédentes et montrent même une certaine régression au niveau de la qualité des preuves dans le traitement formel des automates et des algorithmes. La faible difficulté des premières questions dans les deux parties aurait 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.

Les nouveaux correcteurs qui n'avaient encore jamais participé à ce type de concours ont tous été surpris par la manière dont la majorité des candidats traitent les questions faisant appel à des arguments formels. L'avis global est, une fois de plus, que les candidats ne semblent pas avoir conscience qu'un problème d'Informatique est avant tout un problème scientifique comparable aux Mathématiques et que, la même rigueur que pour la résolution d'un problème de Mathématiques, doit être mise en place.

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

La baisse de la moyenne confirme les remarques effectuées les années précédentes. Il semble que la durée, et parfois la difficulté, de l'épreuve ne soit plus en adéquation avec les capacités actuelles des candidats. Deux solutions sont alors envisageables, soit l'augmentation de la durée de l'épreuve, soit un changement radical dans la forme du sujet proposé.

La durée de l'épreuve a déjà été augmentée par le passé et quelques années plus tard, il semble que les candidats traitent en trois heures la même chose qu'ils traitaient auparavant en deux heures. Cette solution n'est donc pas raisonnable sur du long terme.

La seconde solution consiste à réduire la couverture du programme et à ne plus traiter complètement un sujet mais à poser des questions relativement distinctes, de difficulté croissante dont le seul objectif est l'évaluation et la comparaison des candidats. Cette forme d'évolution devra a priori être expérimentée dans les années suivantes.

Nous concluons en insistant sur le fait que les recommandations proposées dans les rapports depuis la création des épreuves optionnelles n'ont, pour l'instant, toujours pas produits d'effets significatifs observables dans les copies des candidats. Il semble même qu'une certaine régression soit visible cette année. Ces éléments doivent absolument être pris en compte pour les sessions suivantes.
