

Remarques générales

L'épreuve proposée traite les points essentiels du programme de l'option Informatique : automate, logique, preuve et complexité des algorithmes, programmation en CaML ou Pascal. La logique intervient en tant que support de l'algorithmique et programmation. Par contre, seule la manipulation de formules du calcul des propositions est abordée. Aucune question de modélisation ou de résolution n'est proposée aux candidats.

La forme et le contenu du sujet sont dans la continuité des sujets précédents. La difficulté de l'épreuve est semblable aux années précédentes. Le sujet d'algorithme est, par contre, plus court ce qui a permis aux candidats d'en traiter une plus grande partie et de remonter significativement la moyenne et l'écart type par rapport à l'année dernière. Les correcteurs n'ont pourtant pas noté un progrès équivalent au niveau des résultats. Les candidats traitent un plus grand nombre de questions. Par contre, la qualité des réponses proposées est en baisse.

La partie I traite des automates et langages. Le sujet proposé est semblable aux questions posées les années précédentes : l'étude d'un opérateur de composition d'automates. Cette année, il s'agit de l'union des langages, un opérateur très classique. L'exercice peut donc être assimilé à une question de cours. Celui-ci 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 peu plus facile que les années précédentes. Les correcteurs regrettent que, malgré les fortes similitudes avec les sujets précédents, la grande majorité des candidats n'aient pas traité cet exercice de manière satisfaisante.

La partie II traite de l'algorithmique et programmation dans le contexte des structures de données optimisées : il s'agit d'étudier la structure de tas binomial. L'étude est précédée d'un prélude concernant l'implantation en électronique numérique binaire des opérations arithmétiques (incrémentation et addition). Les algorithmes d'ajout dans un tas et d'union de deux tas sont identiques aux algorithmes d'incrémentation et d'addition pour les entiers binaires. Cette étude préliminaire permet de traiter une partie du programme de logique, principalement la manipulation de formules du calcul des propositions et la construction de circuits électroniques numériques correspondant à ces formules. En ce qui concerne l'algorithmique, cette partie concerne trois éléments du programme : preuve, complexité et programmation. Les structures de données manipulées sont, d'une part, des suites de bits codés par des listes de booléens et, d'autre part, des arbres d'entiers n-aires et des listes d'arbres d'entiers n-aires. Le sujet permet ainsi d'évaluer les deux parties du programme d'algorithmique.

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.

Le sujet est plus court que les années précédentes. Il poursuit ainsi un objectif débuté par le passage de trois parties (logique, automate, algorithmique) à deux parties (algorithmique et soit logique, soit automate). Cette année, les correcteurs ont noté que le nombre de questions traitées était plus important par rapport aux années précédentes et attribuent ceci à la réduction de la taille du sujet.

Les correcteurs tiennent quand même à signaler que certains candidats ont traité l'intégralité des questions et obtiennent des notes supérieures à 18.

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 :

- La remarque effectuée pour la session 2005 correspondant à un fléchissement dans le niveau des copies, aussi bien sur le nombre de questions traitées que sur la qualité du traitement par rapport au pallier atteint dans les sessions précédentes, se confirme cette année. La qualité de la programmation reste globalement constante. Par contre, il y a une baisse remarquée 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 avec une légère baisse de PASCAL. La différence de qualité reste malheureusement 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. Le choix d'enseigner PASCAL au lieu de CaML est donc la certitude de réduire les chances des candidats dans les épreuves.
- 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 fausse 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.
- Enfin, les correcteurs tiennent à rappeler que les candidats doivent se comporter honnêtement, c'est-à-dire que lorsqu'ils ne savent pas faire une preuve, il est plus simple de ne pas perdre de temps sur la question et de passer à la suite, plutôt que d'essayer d'escroquer les correcteurs. De toute façon, la réponse sera bien sûr considérée comme fausse, et le temps aura été perdu pour rien.

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 souvent 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, pour un opérateur encore plus simple. 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.1, I.2 et I.3 (description et construction des langages)

Le formalisme des expressions régulières est maintenant acquis de manière satisfaisante. Les langages proposés pour la question I.1 et I.3 sont corrects dans la majorité des cas.

Certains correcteurs ont noté que les candidats exploitent parfois le résultat de la question I.6, dont ils ont l'intuition, pour répondre à la question I.3.

Question I.4

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

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

Contrairement à l'année précédente, la preuve d'égalité du langage reconnu par l'automate composé et du langage résultant de la composition des langages reconnus par chaque automate est très classique et proche d'une question de cours. Globalement, ces deux questions n'ont pas été traitées de manière satisfaisante étant donné leur simplicité. Souvent, les questions étaient tout simplement ignorées. Parfois, les candidats ne voyaient pas le lien entre les deux questions et n'utilisaient pas le résultat de la première dans la seconde. 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. Les correcteurs ont également noté un retour en force des preuves par ..., au lieu de faire une récurrence rigoureuse, les candidats écrivent qu'il suffit d'appliquer les règles autant de fois que le mot a de symboles. Bien entendu, ce type de réponse est jugée intégralement fautive.

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. Bien entendu, ces réponses n'ont aucune valeur.

Partie II - Algorithmique et programmation

Le problème proposé consistait en l'étude de la structure de tas binomial qui permet d'optimiser le coût de la recherche d'un entier dans un ensemble ainsi que les différentes opérations de manipulation de ces ensembles. Le sujet se limitait à l'étude des opérations d'ajout d'un élément et d'union de deux ensembles. L'approche proposée s'appuyait sur une étude préliminaire des algorithmes d'incrémentation et d'addition pour des entiers codés en binaire. L'objectif était de montrer que les algorithmes sur les entiers et sur les tas binomiaux étaient les mêmes.

Les questions étaient de difficulté croissante. Les deux structures de données principales du programme étaient exploitées : d'une part, les listes pour représenter les entiers en codage binaire et les tas binomiaux, et d'autre part, les arbres pour représenter les arbres binomiaux. La longueur du sujet s'est révélée relativement adéquate et l'ensemble des questions a été traité par une partie non négligeable des candidats. L'épreuve aurait pu être légèrement plus courte pour permettre à un plus grand nombre de candidats de traiter un plus grand nombre de questions. Mais toutes les parties ont été abordées, ce qui n'était pas le cas dans la plupart des épreuves des années précédentes.

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.

Questions II.4 (programmation simple), II.6, II.8, II.12, II.16, II.34 (structure de liste), II.18, II.21, II.23, II.30 (structure d'arbre)

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. Notons qu'il y a eu un progrès par rapport au sujet de la session précédente qui a introduit ce type de contraintes. 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 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'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.

Questions II.6 et II.8 : Les candidats ont parfois proposé des solutions beaucoup plus complexes que nécessaire avec des accumulateurs ou des fonctions puissance de 2. Certains candidats ont mélangé les valeurs binaires 0 et 1 et le codage booléen proposé. Cela n'a pas été jugé sévèrement par les correcteurs.

Certains correcteurs ont noté que l'ordre des bits dans le codage binaire n'était pas conforme avec l'ordre suggéré dans le sujet. La conséquence était des opérations de codage et décodage plus complexes. Le sujet aurait dû être plus précis sur ce point là. Certains candidats n'ont pas utilisé le même ordre lors du codage et du décodage.

II.18 : Certains candidats n'ont considéré que des arbres binaires au lieu des arbres n-aires. Les réponses aux questions étaient donc trop simples et fausses.

Question II.7 (complexité)

Suite à la faible qualité des réponses à ce type de question les années précédentes, une seule question a été posée cette année. Cette question était très simple. Les réponses fournies par les candidats qui l'ont traitée étaient convenables mais elle n'a été traitée que par un petit nombre de candidats.

Questions II.1, II.2, II.3 (logique des propositions)

Les questions posées étaient extrêmement simples. Il s'agit sans ambiguïté de questions de cours. La grande majorité des candidats y a répondu de manière tout à fait satisfaisante.

Questions II.10 et II.14 (logique et circuit)

Il s'agissait de traduire les résultats des réponses formelles des questions II.9 et II.13 sous la forme de circuits logiques. Les candidats ont, en général, répondu de manière très satisfaisante. Et cela même s'ils n'avaient pas traité les questions II.9 et II.13 ce qui a beaucoup surpris les correcteurs. En fait, cela confirme le fait que les candidats acquièrent une bonne connaissance intuitive du sujet qui leur permet de répondre aux questions pratiques mais qu'ils ont beaucoup de mal à formaliser cette intuition, ce qui est pourtant une caractéristique essentielle chez un futur ingénieur.

Question II.11, II.15 et II.24 (exemples)

Ces questions ont globalement été bien traitées par les candidats. Cela correspond bien à la manière dont ils abordent les problèmes par l'exemple plutôt que par la formalisation. Cela correspond principalement à la programmation dirigée par les tests (ou programmation extrême) qui connaît un grand succès actuellement chez les jeunes programmeurs. Cette approche ne peut pas remplacer le traitement rigoureux car elle ne s'applique pas de manière satisfaisante sur des systèmes complexes. Il est donc essentiel que les candidats sachent toujours aborder les problèmes de manière rigoureuse en suivant un cheminement déductif de type mathématique. La seule manière d'assurer la correction d'un algorithme est d'en faire la preuve. Toutes les autres approches sont des heuristiques limitées qui ne passent pas à l'échelle des systèmes complexes. Il est essentiel que les candidats le comprennent.

Questions II.5, II.9, II.13 (arithmétique et logique), II.17, II.19, II.20, II.22, II.25, II.26, II.27, II.28, II.29, II.31, II.32, II.33, II.35, II.36 (formalisation et 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.5 : L'encadrement proposé est rarement justifié.

II.9 et II.13 : Certains correcteurs ont noté que la forme de ces questions a perturbé les candidats (relation entre v_m et m , v_p et p) qui n'ont pas vraiment compris la réponse attendue.

II.17, II.22, II.29 : Certains candidats oublient le cas $S(a) = \{\}$.

II.19, II.20, II.22, II.29 et II.33 : De nombreux candidats oublient de quantifier universellement sur $S(a)$ dans les définitions formelles des propriétés de a .

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 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 réponse aux questions les plus complexes.

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, une fois de plus, 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.

Cette année, la moyenne est de 11,62 et l'écart type de 2,77. La remontée par rapport à l'année dernière est significative, plus particulièrement au niveau de l'écart type. Cela découle principalement de la réduction de la longueur de l'épreuve. Le résultat est toutefois loin d'être satisfaisant même dans le cadre d'un concours.