

Remarques générales

Le sujet proposé préserve la ligne suivie par les épreuves de l'option Informatique du concours commun polytechnique, c'est-à-dire traiter les aspects essentiels du programme dans chaque épreuve : logique, automate, preuve et complexité des algorithmes, programmation en CaML ou PASCAL. Le sujet comporte cette année deux parties indépendantes.

La partie I couvre la logique. Cette partie est très proche de la forme choisie pour les années précédentes : la modélisation logique d'un problème puis sa résolution formelle.

Cette partie comporte deux exercices. Le premier, qui ne contient aucune difficulté, correspond à ce que chaque étudiant devrait être capable de réaliser. Il permet de valider une maîtrise minimale du traitement formel de la logique mais ne se révèle pas très discriminant. Le second pose un problème particulier lors de la modélisation qui a perturbé les candidats. Il s'est donc révélé plus discriminant que prévu car la bonne approche n'a été proposée que par très peu de candidats.

La partie II couvre l'algorithme et la programmation dans le cadre particulier des automates. Elle permet donc de valider quatre aspects du programme. Les structures de données manipulées sont des listes traitées comme des ensembles. Cette partie est nettement plus importante et discriminante. Les questions posées sont de difficulté croissante et couvrent le traitement formel des automates, des algorithmes (preuve et complexité) et la programmation. Cette partie a largement permis de distinguer les candidats.

La fusion des parties automate et algorithmique a permis de réduire la taille du sujet par rapport aux années précédentes. Le sujet se révèle à la correction encore trop long. Par contre, contrairement aux années précédentes, l'ensemble des candidats a pu aborder les différents aspects du sujet.

Avant de détailler chaque partie, notons quelques points par rapport au comportement global des candidats et à son évolution sur les dernières années :

- Le contenu des copies semble confirmer que nous avons atteint un point relativement stable dans la qualité du traitement formel de la logique et des automates. Une part faible mais encore trop importante des candidats répond informellement sans exploiter les techniques formelles présentées en cours et exigées dans le sujet (tables de vérité et/ou formules de De Morgan). Cette part augmente avec la complexité des questions. Cela est notable entre les deux exercices de la partie I : le premier est en général traité correctement ; le second, plus complexe, est souvent abordé en langage naturel par un raisonnement informel. L'utilisation de l'induction, qui est la technique essentielle en Informatique, n'est pas encore maîtrisée alors que les questions qui s'y réfèrent sont extrêmement simples. Une fois de plus, il faut insister sur l'aspect inadmissible de ce comportement. Les difficultés sont maintenant bien identifiées à la fois sur la forme et sur le fond. Les candidats devraient être capables d'appliquer les différents formalismes sans aucune réflexion ni perte de temps. Nous ne devrions plus trouver que des erreurs d'inattention. Cela est loin d'être le cas et il serait nécessaire d'analyser les causes de ce dysfonctionnement.

- Une autre constante se dégage : le rapport entre les copies utilisant PASCAL et celles exploitant CaML pour la programmation. Il faut, une fois de plus, noter que l'usage de PASCAL pose beaucoup plus de problèmes pour les candidats que celui de CaML qui est nettement plus adapté aux objectifs du programme. Tous les correcteurs ont confirmé que cela était notable dans la différence de note entre les copies tout en insistant sur le fait que les copies utilisant PASCAL n'étaient pas moins bonnes mais que le candidat n'avait pas eu le temps de traiter autant de questions que celui qui a utilisé CaML. Lors de l'introduction des options, il avait été évalué qu'une période transitoire serait nécessaire pour passer de PASCAL à CaML. Celle-ci devrait maintenant être révolue sachant que les candidats sont désavantagés quand ils ont pratiqué PASCAL. Il nous semble également nécessaire d'analyser les causes de ce dysfonctionnement et de faire disparaître PASCAL par la contrainte, étant donné que la raison n'a pas eu de succès.
- Pour conclure, l'ensemble des correcteurs s'est encore plaint de la médiocrité de la rédaction des copies. Il devient essentiel d'arriver à faire passer aux candidats qu'une réponse correcte mal rédigée sera en général considérée comme partiellement, voire totalement, fautive. Ceci est particulièrement notable dans la rédaction des preuves et des explications de programme. Il faut insister sur le fait que la production de documents devient une part essentielle du métier de l'ingénieur (citons le cas des normes de qualité telles ISO 9001). Les mauvaises habitudes montrées par les candidats se prolongent lors de leur passage dans les écoles et ceux-ci rencontrent maintenant des problèmes avec la rédaction lors de leur intégration dans le milieu industriel. Ces problèmes nous sont ensuite signalés lors des stages de fin d'études. Nous espérons que ces éléments conduiront enfin à une prise de conscience de la part des candidats et à quelques efforts dans la bonne direction au lieu d'une régression constante.

L'ensemble des correcteurs a noté que les questions suivantes ont été les plus discriminantes :

- le second exercice de la partie I ;
- les preuves et les explications des algorithmes dans la partie II ;
- la fin de la partie II, probablement à cause de la longueur du sujet.

Remarques particulières

Partie I – Logique et calcul des propositions

En général, la manipulation formelle des formules et la construction des tables de vérité sont des techniques bien maîtrisées. La modélisation, c'est-à-dire le passage entre un énoncé en langage naturel et une formule équivalente, pose plus de problèmes. Le traitement du second exercice demandait l'introduction d'une variable intermédiaire pour représenter le fait qu'une partie de l'énoncé en langage naturel était effacée. Très peu de candidats ont pensé à cette approche et n'ont donc pas pu traiter la suite de cet exercice. Face à cette difficulté, les candidats ont en général oublié l'aspect formel et essayé de raisonner en langage naturel ce qui était, bien entendu, jugé comme faux étant donné que le sujet imposait un traitement formel.

Notons que certains candidats ne respectent pas le plan introduit par le sujet :

- a) modélisation du texte en langage naturel par une formule,
- b) résolution avec une technique formelle imposée,
- c) exploitation du résultat de la résolution.

Notons que certains candidats fournissent comme résultat des formules qui ne sont pas suffisamment simplifiées ou mélangent les deux formes de résolution (De Morgan et tables de vérité) pour obtenir les résultats souhaités.

Questions I.1 et I.2 : Réponses correctes pour la plupart des candidats. Quelques erreurs de transformation de $A \Rightarrow B$.

Question I.3 : Réponses correctes pour la majorité des candidats. Certains n'utilisent toujours pas les formalismes présentés en cours et requis par le sujet.

Question I.4 : Extrêmement peu de réponses correctes car les candidats n'ont pas pensé à introduire une variable pouvant prendre les valeurs possibles pour la partie effacée de l'inscription mais essaient plutôt de trouver une formule exprimant la même chose ce qui n'était pas réalisable.

Question I.5 : Le fait que la question précédente n'ait pas été traitée correctement n'a pas permis aux candidats d'obtenir une note maximale même si la manipulation formelle était correcte. D'assez nombreux candidats ayant proposé une formule très complexe pour la question précédente ont opté pour le langage naturel au lieu de construire les tables de vérité comme le sujet le demandait.

Partie II – Algorithmique et programmation

Séparons les remarques en deux parties selon qu'elles concernent les deux notions évaluées dans le problème : les automates ou les techniques de programmation.

Automates et langage

La plupart des candidats n'a aucun problème pour manipuler les automates et les expressions régulières. Par contre, les preuves par induction structurée sur un mot (récurrence sur la longueur du mot) sont encore souvent incomplètes voire incorrectes. L'utilisation de "..." en lieu et place d'un mot quelconque, accompagnée d'un raisonnement informel et incorrect à partir des "..." se retrouvent encore dans beaucoup de copies. Ce type de question est maintenant classique et les erreurs effectuées par la plupart des candidats sont inexcusables.

Questions II.1 (expression régulière), **II.19**, **II.25** et **II.29** (construction des automates) : Réponses correctes quand les candidats ont eu le temps de les traiter.

Questions II.14, **II.21**, **II.27** (preuves) : La rigueur nécessaire à la rédaction d'une preuve correcte ne se retrouve que chez très peu de candidats. L'ensemble des correcteurs a noté une tendance à la paraphrase du sujet et de la question de la part de candidats qui exploitent souvent le résultat attendu comme une hypothèse et concluent sans autre forme de justification. Ces questions se révèlent ainsi très discriminantes.

L'induction est une technique essentielle dans le traitement formel de l'algorithmique. Tous les candidats doivent être capables de l'utiliser, ce qui est loin d'être le cas. Beaucoup de copies exploitent des "..." pour notifier "application d'une même transformation autant de fois que nécessaire". Beaucoup de candidats oublient l'aspect inductif et se limitent à un traitement par cas et un tour de passe-passe en guise de conclusion, ayant oublié les hypothèses d'induction.

Les candidats ont souvent oublié que le mot vide (λ) pouvait être composé d'une suite de transitions arbitraires (ϵ) et qu'il était donc nécessaire de faire une induction sur le nombre de transitions arbitraires pour le cas du mot vide.

L'ensemble des correcteurs souhaite une fois de plus rappeler que les aspects formels de l'algorithmique sont une part essentielle non seulement du programme des enseignements mais également de l'Informatique elle-même, aspect qui distingue de plus en plus les métiers de techniciens et d'ingénieurs. Il est donc critique que les candidats puissent construire des preuves correctes sans masquer la plupart des étapes en particulier quand celles-ci ne sont pas évidentes.

Algorithmique et programmation

L'objectif était l'étude d'un algorithme de construction d'un automate ne comportant aucune transition arbitraire à partir d'un automate quelconque. La programmation de l'algorithme se faisait de façon conjointe avec la preuve de sa correction et l'étude de sa complexité.

L'ensemble des correcteurs souhaite signaler que les candidats sont beaucoup plus à l'aise avec la programmation qu'avec le traitement formel de l'algorithmique ou l'explication de leurs fonctions. Ceci est une dérive usuelle de l'Informatique qu'il est important de corriger dès le début de l'apprentissage.

Une première sous-partie consistait à programmer une structure d'ensemble en utilisant une structure de liste.

La difficulté des programmes était croissante mais aurait dû être plus progressive pour que l'épreuve soit plus discriminante en permettant à chaque candidat, dans le temps imparti, d'atteindre son niveau de compétence maximum.

L'ensemble des correcteurs souhaite insister sur la faiblesse du traitement des questions concernant l'explication d'un programme. Cet élément qui est essentiel dans le métier actuel d'ingénieur en Informatique, n'est pas simple car il est difficile de choisir le bon niveau de détail sans une certaine pratique. Mais, il est essentiel que les candidats pratiquent le plus tôt possible au fur et à mesure de l'apprentissage de la programmation. Le préluce ajouté au sujet depuis deux ans précise les éléments que les candidats doivent fournir pour les explications d'une fonction. Ces éléments sont inspirés des normes en vigueur dans les groupes industriels en Informatique (en général dans le cadre de norme de qualité ISO 9001). Les candidats restent perplexes car certains semblent juger que le texte du programme contient déjà les informations nécessaires. Il est cependant important d'être capable d'expliquer les choix effectués dans un langage non-informatique. L'ensemble des correcteurs espérait un progrès significatif depuis la mise en place de ce préluce. Cela n'a pas été le cas. Seules quelques réponses ont été convenables et donc très discriminantes alors que l'objectif est d'obtenir un niveau moyen que tous les candidats peuvent atteindre sans difficulté s'ils pratiquent ce type d'exercice avant l'épreuve du concours.

Remarques générales

Questions II.2, II.4, II.6, II.7, II.10, II.11, II.12, II.15, II.17, II.18, II.22, II.26, II.30
(programmes)

Les candidats ont traité la plus grande partie de ces questions. Les fonctions proposées sont satisfaisantes et respectent les contraintes indiquées dans le sujet sur les restrictions concernant le langage (ni références, ni boucles).

Notons que certains candidats ne réutilisent pas systématiquement les fonctions des questions précédentes et perdent un temps précieux à proposer une seconde fois les fonctions.

Notons que les candidats font en général l'hypothèse que les listes utilisées pour représenter les ensembles sont triées et ne contiennent pas de doublons. Ceci est en général vérifié sur l'ensemble des fonctions qu'ils ont écrites mais ils oublient de le prouver ou, tout du moins, de le signaler.

L'ensemble des correcteurs constate que l'utilisation de CaML avantage nettement les candidats par rapport à PASCAL comme cela a été indiqué dans chaque rapport depuis l'introduction de l'option Informatique.

Questions II.8, II.13, II.16, II.23 (explications des programmes)

Une part non négligeable des candidats persiste à ne pas suivre les indications fournies dans le sujet. En particulier, ceux-ci fournissent des explications pour tous les programmes et pas seulement lorsque le sujet le demande. Cela cause une importante perte de temps. D'autres n'ont pas lu le préambule et ne fournissent pas les éléments attendus. Globalement, un effort important est à faire pour expliquer convenablement en langage naturel l'algorithme exprimé dans un langage de programme.

Questions II.3, II.5, II.9, II.20, II.24, II.28 (complexité)

Le traitement de la complexité a été décomposé dans le sujet en de nombreuses questions intermédiaires extrêmement simples. Celles-ci ont été traitées correctement par la majorité des candidats qui les ont abordées.

Conclusion

Cette épreuve composée de deux parties distinctes a permis de valider les compétences des candidats sur une grande partie du programme. Elle semble confirmer que la quantité de candidats proposant un traitement formel satisfaisant en logique et automate a atteint son maximum. Vu la faible difficulté des questions dans ces domaines, il est regrettable que ce maximum ne corresponde pas à la totalité des candidats. Les programmes sont de qualité satisfaisante. La preuve des algorithmes doit encore être développée. Un travail important reste à faire dans le cadre de l'explication des programmes pour se rapprocher des standards industriels.

La moyenne et l'écart-type de l'épreuve sont de 10,34 et 3.79. Ces valeurs sont nettement plus satisfaisantes que l'année précédente mais il serait souhaitable de rendre l'épreuve plus pertinente et discriminante en obtenant des valeurs plus élevées à la fois pour la moyenne et l'écart-type. Dans ce but, il semble encore nécessaire de réduire la longueur du sujet tout en développant les différences de difficulté entre les questions. L'intégration de l'exercice sur les automates dans le problème d'algorithmique a été un apport significatif pour réduire la longueur de l'épreuve. Il semble nécessaire maintenant de mieux organiser les différentes questions concernant la preuve des algorithmes pour obtenir une meilleure progressivité dans les difficultés et permettre de mieux distinguer les candidats sans se baser exclusivement sur l'algorithmique et la programmation.