

## Remarques générales

Le sujet de l'épreuve optionnelle d'Informatique de la session 2008 du Concours Commun Polytechnique traite de la plupart des aspects abordés dans le programme : programmation en CaML ou PASCAL, preuve des algorithmes, calcul de complexité, langages et automates, modélisation et résolution de problèmes en logique des propositions. La forme ainsi que le contenu du sujet sont semblables à ceux proposés les années précédentes au sein du CCP. La difficulté est similaire aux années précédentes. Par contre, le sujet est plus court même s'il faut de nouveau intervenir les trois aspects du programme contrairement aux cinq dernières années qui alternaient logique et automate. Les résultats sont globalement décevants étant donné la grande similitude, d'une part dans la forme des exercices avec ceux des années précédentes, et d'autre part dans le sujet et la forme du problème (arbres équilibrés) par rapport à un sujet récent. Il est très inquiétant, dans un contexte aussi favorable, d'obtenir des résultats aussi décevants. Il semble donc nécessaire de faire des évolutions significatives dans la forme et le fond de la formation des candidats et des épreuves d'évaluation.

**La partie I** traite de la logique des propositions : il s'agit de modéliser puis de résoudre un problème. Cet exercice est extrêmement simple, intégralement conforme au programme et semblable aux exercices posés les années précédentes au sein du CCP. En sus de l'expression formelle en logique des propositions, le sujet exploitait également la notation graphique utilisée en électronique numérique. Cet exercice a été très bien traité par la grande majorité des candidats (plus de 85 % de réussite) sauf la partie exploitant cette notation (25 % de réussite). Cet exercice ne s'est donc globalement pas révélé très discriminant (l'écart type est de l'ordre de 15 % du barème attribué).

**La partie II** traite des automates et des langages : il s'agit d'étudier l'opérateur « conjonction d'automates » qui correspond à l'intersection des langages acceptés par les automates. Cet exercice est très proche d'un exercice de cours. Il est simple, conforme au programme et dans l'esprit de ceux posés dans les sessions précédentes du CCP. Les candidats doivent traiter un exemple de construction d'automates par application de l'opérateur, puis prouver la généralité du résultat constaté sur l'exemple. Il s'agit de preuves par induction sur la structure des mots du langage (ou récurrence sur la longueur des mots). La construction des automates et la preuve par induction sont, d'une part, deux éléments essentiels du programme, et d'autre part, une technique de modélisation et une technique de preuve fondamentale pour l'Informatique et de plus en plus utilisées au niveau des langages de modélisation et des techniques de vérification formelle mises en œuvre pour les systèmes critiques. Les candidats doivent donc absolument être capables de traiter ces deux aspects. Nous avons constaté que les candidats sont toujours plus à l'aise avec la construction des automates (60 % de réussite) qu'avec les preuves (50 % de réussite). La différence est moins nette que les années précédentes, mais ceci est dû à une moins grande réussite dans la construction des automates. La réussite en terme de preuve par induction est au même niveau que les années précédentes. Cet exercice s'est révélé bien discriminant avec un écart type de l'ordre de 30 % du barème affecté.

**La partie III** aborde le domaine de l'algorithmique et la programmation : il s'agit d'étudier la structure d'arbre binaire bicolore qui permet de manipuler des arbres quasi-équilibrés. Cette partie développe trois aspects du programme : la preuve, la complexité et la programmation d'algorithmes. Il s'agit de manipuler des ensembles d'entiers codés, dans une première partie par des listes, et dans

une seconde partie par des arbres binaires, bicolores de recherche. Les deux aspects du programme en algorithmique sont donc considérés.

Une partie des éléments manipulés (ensemble sous la forme de listes et d'arbres binaires de recherche) est traitée de manière classique et peut être considérée comme des questions de cours. Le traitement des listes a été bien réussi par la grande majorité des candidats (75 % de réussite) et s'est révélée peu discriminante (l'écart type est de l'ordre de 20 % du barème affecté). Le traitement des arbres binaires a été moyennement réussi (50 % de réussite) tout en étant moins discriminant. Cela est dû au fait qu'il ne se limitait pas à de la programmation et les candidats ont globalement mal traités les aspects formalisation des algorithmes, que ce soit la complexité ou la correction. Le traitement des arbres bicolores a été relativement médiocre (35 % de réussite) mais beaucoup plus discriminant (écart type de 30 % du barème affecté). Ceci est dû, d'une part à la raison évoquée précédemment concernant la nature des questions, et d'autre part au nombre de candidats plus faible qui a traité cette partie. Le traitement des arbres de recherche bicolore est très faible (10 % de réussite) avec un écart type de 20 % du barème affecté. Ceci est dû aux deux raisons précédentes. La grande majorité des candidats s'est arrêtée à la question III.17.

Il faut noter que les candidats sont globalement plus à l'aise avec les aspects concrets tels la programmation ou le développement d'exemples, que par les aspects plus abstraits tels la preuve de correction ou le calcul de complexité.

Cette partie du sujet est significativement plus courte que les années précédentes. Cette réduction est en partie compensée par la présence des deux exercices (logique et automate), ce qui n'était pas le cas des dernières années. Cela permet quand même d'obtenir une moyenne supérieure de deux points à l'année précédente.

Globalement, la réduction du sujet permet d'obtenir de meilleurs résultats mais la tendance notée au sein de l'équipe en charge des corrections se maintient. A part les deux ou trois premières années qui correspondaient à la découverte du type d'épreuve et au rodage des options, les candidats traitent chaque année une quantité plus restreinte de questions. L'augmentation de la durée de l'épreuve avait permis de compenser ce phénomène, mais maintenant, nous constatons que les candidats font moins de choses en trois heures qu'ils n'en faisaient en deux heures il y a cinq ou six ans.

Avant de donner des informations détaillées sur chaque partie, il convient de faire quelques remarques globales sur le comportement des candidats, et d'évoquer les différences par rapport aux années précédentes :

- la proportion de candidats utilisant le langage PASCAL par rapport aux candidats utilisant le langage CaML est similaire aux années précédentes. La différence entre les notes moyennes des uns et des autres est également similaire aux années précédentes malgré les efforts de rédaction du sujet pour compenser les différences en terme d'expressivité des deux langages. Il est donc nécessaire de rappeler une fois de plus que le langage CaML est bien plus adapté que le langage PASCAL au traitement de l'algorithmique récursive qui constitue le cœur du programme.
- la qualité de rédaction des copies n'évolue pas de manière significative. Celle-ci reste extrêmement faible. Ce point est pourtant un aspect essentiel de leur futur métier d'ingénieur qui se retrouve maintenant au cœur de toutes les normes de qualité qui reposent principalement sur la construction de documents clairs, précis, complets et sans ambiguïté. Malheureusement, c'est très loin d'être le cas de la grande majorité des copies (certaines étant tout simplement illisibles). Une réponse correcte mal rédigée sera généralement considérée comme incomplète dans le meilleur

des cas, et comme fausse dans le pire. Il est nécessaire que les candidats soient mieux préparés sur ce point.

- le rapport entre le traitement des éléments théoriques (preuves, complexité) et les éléments pratiques (programmes, exemples) est en nette régression. Plus précisément, les candidats consacrent de moins en moins de temps aux questions théoriques qui sont pourtant essentielles par rapport à l'évolution actuelle de l'Informatique et à la construction d'applications de plus en plus complexes et dont le rôle est de plus en plus critique. La qualité de la programmation et du traitement des exemples est maintenant globalement satisfaisante. Le traitement des aspects théoriques manque, par contre, de plus en plus de rigueur. Les raisonnements sur des ... remplacent l'induction sur la structure des mots (ou la récurrence sur leur longueur) pour exprimer que la même chose se répète autant de fois que nécessaire. L'induction est pourtant la technique fondamentale de raisonnement en Informatique et les candidats doivent absolument la maîtriser. Il est d'ailleurs étonnant que les candidats ne soient pas mieux préparés à répondre à ce type de question qui se retrouve dans tous les sujets depuis la création de l'épreuve optionnelle d'Informatique.

## Remarques particulières

### Partie I - Logique et calcul des propositions

L'exercice proposé au candidat était très classique. Ceux-ci ont, en grande majorité, proposé des réponses satisfaisantes. Seule la dernière question a été mal traitée car les candidats ont négligé un aspect important qui les a conduit à fournir des réponses incorrectes.

*Question I.1* : Bien traitée par tous les candidats (94 % de réussite).

*Questions I.2* : Bien traitées par la grande majorité des candidats (85 % de réussite).

*Question I.3* : Bien traitée par la grande majorité des candidats (83 % de réussite).

*Question I.4* : Traitée par la majorité des candidats mais de manière incorrecte (30 % de réussite). Cette question comportait deux aspects, d'une part, l'utilisation de la notation graphique pour les circuits en logique booléenne, d'autre part, l'adaptation des résultats précédents pour prendre en compte le fait que la salle ne devait pas être détruite lorsque les interrupteurs étaient au repos. Les candidats ont, en général, négligé ce second aspect tout en traitant correctement le premier, ceci explique la faible qualité des résultats.

### Partie II - Automates et langages

La forme de l'exercice sur les automates, les langages et les expressions régulières est très classique et celui-ci peut être considéré comme une simple question de cours. Les candidats maîtrisent bien la construction des automates et des expressions régulières représentant le langage accepté par un automate, même s'il faut noter une régression difficile à comprendre sur ce sujet bien maîtrisé les années précédentes. Par contre, les preuves sur les langages (induction sur la structure des mots, récurrence sur la longueur des mots) sont très mal réalisées, alors qu'elles sont quasiment identiques à celles réalisées durant les enseignements et requises dans les sujets des années précédentes. Dans de nombreux cas, les candidats font un dépliage des définitions et raisonnent sur un mot de la forme "C1 ... Cn" (composé de n caractères) pour indiquer que le dépliage doit être fait  $n$  fois. Il s'agit bien sûr d'une induction ou une récurrence informelle. Mais les candidats doivent absolument maîtriser l'induction ou la récurrence. Et leur réponse doit donc être notée avec une grande exigence sur ce

sujet. Ils obtiennent donc en général une note faible voire nulle. Les correcteurs tiennent à rappeler que l'induction structurelle et la récurrence sont les mécanismes de preuve fondamentaux en Informatique et qu'elles doivent être maîtrisées sans la moindre hésitation, en particulier dans le cadre d'exercices très proches du cours et similaires aux exercices des années précédentes. Les correcteurs notent cette année une légère régression par rapport aux années précédentes en ce qui concerne ce traitement formel.

*Question II.2* (construction des automates) La majorité des candidats ont traité cette question et les réponses proposées sont généralement correctes. Le taux de réussite est de 75 % avec un écart type de 35 % du barème. La plupart des erreurs constatées concernent les états initiaux ou terminaux.

*Questions II.1 et II.3* (description et construction des langages) La notation des expressions régulières utilisée pour représenter le langage accepté par un automate, et la construction d'une expression régulière pour un automate simple, est un domaine que les candidats connaissaient bien les années précédentes. Les correcteurs ont noté avec surprise une baisse significative dans les résultats. Le taux de réussite est de 65 % avec un écart type de 35 % par rapport au barème. Ces chiffres sont très nettement différents des années précédentes.

*Question II.4* Cette question est abordée par la plupart des candidats mais les réponses fournies sont en général incomplètes. Les candidats oublient en général un des éléments nécessaires (automate fini, déterministe (état initial unique, pas de transition arbitraire, une seule cible par symbole) et complet). Le taux de réussite est de 50 % avec un écart type de 30 % selon le barème. Ce faible taux de réussite reste similaire aux années précédentes et toujours aussi étonnant car la question est simple et classique.

*Question II.5 et II.6* (preuve par induction) L'opérateur de composition traité dans cet exercice correspond à l'intersection des langages acceptés par les automates. La preuve d'égalité du langage accepté par l'automate composé et de l'intersection des langages acceptés par les opérandes est relativement simple par rapport aux preuves effectuées dans les sujets des années précédentes. Le sujet proposait de décomposer cette preuve de manière très classique en deux parties.

La première est l'extension de la définition de l'opérateur de composition à un mot de longueur quelconque. Celle-ci a été traitée par un peu plus de la moitié des candidats. Le taux de succès est de 39 % avec un écart type de 45 % du barème. Deux éléments justifient ce résultat, d'une part le faible nombre d'étudiants qui ont traités cette équation, et d'autre part les tentatives pour ne pas faire une preuve correcte par induction ou récurrence. Certains candidats traitent correctement la progression de l'induction ou de la récurrence et oublient le cas d'arrêt. D'autres considèrent comme cas d'arrêt le mot contenant un seul symbole (mot de taille 1) au lieu du mot vide (de taille 0).

La seconde est l'exploitation de ce résultat pour prouver qu'un mot accepté par la composition de deux automates est également accepté par les deux automates. Cette forme de preuve se retrouve dans chaque exercice d'étude d'un opérateur et est donc très classique dans les sujets de l'épreuve d'Informatique pour le CCP. Cette question a été traitée par un nombre de candidats plus important que la précédente. Le taux de réussite est de 56 % avec un écart type de 40 % par rapport au barème. Ceci s'explique, d'une part, par le nombre encore faible de candidats ayant traités cette question, et d'autre part, par le fait qu'il n'est pas nécessaire de faire une preuve par induction ou récurrence ce qui pose un problème à un grand nombre de candidats.

*Question II.7* Cette question est généralement bien traitée par les candidats qui ont abordé le reste de l'exercice (taux de réussite de 73 % avec un écart type de 25 % par rapport au barème). Certains candidats donnent une réponse intuitive, sans avoir abordé les questions précédentes, qui n'a bien sûr aucune valeur.

### Partie III - Algorithmique et programmation

Le problème que les candidats devaient traiter est l'application de la structure d'arbre bicolore pour implanter des ensembles d'entiers. Cette structure correspond à des arbres semi-équilibrés. Le sujet était composé de deux parties, la première concerne le codage des ensembles à base de liste, la seconde traite des arbres bicolores, en plusieurs étapes : arbres binaires, arbres bicolores, arbres de recherche, préservation de la structure bicolore, insertion équilibrée. Cette partie était significativement plus courte que les années précédentes. Elle était de difficulté progressive et permettait de distinguer les candidats. Ceux-ci se sont majoritairement arrêtés à la fin de l'étape arbres bicolores (2.2), certains ont poursuivi jusqu'à la fin de l'étape préservation de la structure bicolore (2.4). Ce problème s'est donc révélé plus adapté que ceux des années précédentes aux candidats qui ont pu en traiter une plus grande partie. Le sujet global reste toutefois trop long pour la grande majorité des candidats.

Les correcteurs ont remarqué que les candidats sont plus performants dans le traitement des exemples et de la programmation que dans les parties plus formelles correspondant à la preuve de correction des algorithmes et au calcul de la complexité. Ceci est en opposition avec l'évolution des méthodes et outils mis en œuvre par les ingénieurs en Informatique. Pour traiter des applications de plus en plus complexes et critiques, il est nécessaire de s'appuyer sur les fondements Mathématiques de l'Informatique et de développer des techniques plus rigoureuses que les solutions mises en œuvre précédemment. L'Informatique devient une discipline industrielle comme les autres qui s'appuie sur des modèles Mathématiques et des solutions d'ingénierie rigoureuse. Les candidats doivent donc faire des progrès significatifs dans ces sujets pour se rapprocher des exigences de leur futur métier.

*Questions III.1, III.3, III.9, III.10, III.12, III.13, III.20, III.21* (programmation) Ces questions ont été globalement bien traitées par les candidats (taux de réussite 86 %, 87 %, 77 %, 24 %, 20 %, 11 %, 5 %) quand ceux-ci ont eu le temps de les aborder. Il s'agit des questions les mieux traitées de l'épreuve. Les contraintes d'utilisation des langages, que ce soit CaML ou PASCAL, ont été bien respectées par les candidats.

Les correcteurs ont noté, une fois encore, que les réponses en langage PASCAL sont significativement plus longues que les réponses en langage CaML.

Certains candidats expliquent systématiquement les fonctions qu'ils programment alors que cela n'est pas explicitement demandé. Cela leur demande beaucoup plus de temps et n'apporte en général pas de points supplémentaires. Les explications ne sont demandées que pour certaines fonctions particulièrement complexes (question III.22) et ne doivent pas être fournies dans les autres cas. Si le candidat respecte les règles usuellement présentées au cours de sa formation pour la décomposition d'une fonction en sous fonctions, pour l'utilisation de variables intermédiaires, et pour le choix des noms des identificateurs, il ne devrait pas être nécessaire de donner des commentaires en plus des programmes (sauf si cela est explicitement demandé).

D'autres candidats fournissent des fonctions particulièrement longues, sans décomposition en sous fonctions, sans utilisation de variables intermédiaires et souvent sans la moindre explication de ces fonctions particulièrement complexes. La forme d'un programme est prise en compte dans l'évaluation de sa qualité, que ce soit dans le barème de l'épreuve, ou plus tard dans le cadre d'une utilisation industrielle de l'Informatique. Nous insistons donc, une fois encore, sur l'importance de la forme d'un programme comme outil de maîtrise de sa complexité. Ceci doit absolument être enseigné aux candidats.

Au niveau des fonctions de manipulation des arbres, de nombreux candidats oublient de traiter le cas du milieu de l'arbre et ne considèrent que les frontières externes.

Certains candidats oublient, ou n'assimilent pas, la contrainte de parcours uniquement des arbres.

*Questions III.5, III.6, III.7, III.8, III.15, III.16, III.17, III.18, III.19* (preuve algorithme) Les correcteurs souhaitent insister sur le fait qu'une dégradation est notable dans les réponses à cette catégorie de questions. Les candidats ne répondent pas avec la rigueur nécessaire pour réaliser une preuve Mathématique de correction des algorithmes considérés. Les réponses s'appuient souvent sur une paraphrase du sujet, et sur des arguments d'évidence ou de trivialité. La validation et la vérification des programmes par rapport à leurs spécifications sont des activités qui prennent une importance croissante pour contrer l'évolution de la complexité et de la criticité des systèmes à logiciels prépondérants. La meilleure technique dans ce cadre reste la preuve Mathématique de correction de l'algorithme. Cette technique commence à prendre sa place dans le cadre industriel. Elle correspond tout à fait à une activité d'ingénieur qui s'appuie sur ses connaissances en Mathématiques pour résoudre des problèmes complexes. Il est donc essentiel que la formation des candidats leur permette d'aborder correctement ce type d'exigences.

Le taux de réussite à ces questions est respectivement de 87 %, 40 %, 45 %, 27 %, 55 %, 23 %, 38 %, 6 %, 7 %. Seule la première question est traitée correctement par la majorité des candidats. Les deux dernières ont été très peu abordées faute de temps. Les écarts type sont de l'ordre de 15 % à 25 % par rapport au barème.

*Question III.22* (explication des programmes) Cette question, placée à la fin du sujet, n'a été traitée que par un très petit nombre de candidats malgré sa simplicité. Etant donné l'importance de la rédaction formelle dans le processus de développement du logiciel, il serait souhaitable que ce type de question soit positionné plus tôt dans les problèmes d'algorithmique et programmation les années suivantes.

*Questions III.2, III.4, III.11, III.14* (complexité) Les correcteurs ont noté une baisse dans la qualité du traitement de ces questions par rapport aux années précédentes, y compris pour les questions extrêmement simples sur les listes.

La première partie concernait les listes. Son taux de réussite est de 60 % avec un écart type de 20 % par rapport au barème attribué. Les questions étaient extrêmement simples. Les candidats qui les ont traitées ont fournis en général des réponses satisfaisantes. Par contre, une part non négligeable des candidats n'a pas abordé ces questions.

La seconde partie concernait les arbres. Son taux de réussite est de 18 % avec un écart type de 30 % par rapport au barème attribué. Ce résultat est en partie imputable à la position des questions dans la seconde partie du problème qui n'a pas été abordée par une majorité des candidats faute de temps. La qualité des réponses n'était, de plus, pas satisfaisante.

## Conclusion

Le sujet soumis aux candidats, composé de trois parties indépendantes, a permis d'évaluer leurs compétences sur l'ensemble du programme. Par rapport aux années précédentes, la qualité des réponses est constante au niveau du traitement des exemples dans les différentes parties ainsi que la programmation. Par contre, celle-ci est en régression en ce qui concerne les éléments formels, tels les preuves d'algorithmes, ou d'équivalence entre automates, et le calcul de complexité. Ces éléments formels constituent la différence majeure entre un ingénieur et un technicien. Le

technicien applique des solutions sans avoir besoin de les comprendre, l'ingénieur doit comprendre les solutions pour en concevoir de nouvelles ou être capable de les appliquer dans les contextes les plus défavorables. L'ensemble des correcteurs insiste donc sur le fait que les futurs ingénieurs doivent absolument maîtriser ces aspects formels et qu'il est nécessaire, pendant leur préparation, d'insister sur ceux-ci.

Les candidats proposent quasiment tous des réponses satisfaisantes aux questions simples situées en début de chaque partie. Par contre, ceux-ci rencontrent rapidement des difficultés, même lorsqu'il s'agit de questions de cours ou de questions similaires à des éléments traités en cours. Le sujet très détaillé et progressif devrait permettre de mieux distinguer les candidats en fonction de leur connaissance, leur rapidité et leur compréhension du sujet et de la matière. Ce n'est malheureusement pas suffisamment le cas. L'impression globale des correcteurs est que les candidats rencontrent une difficulté certaine pour appréhender un problème plus large (telle la partie algorithmique et programmation) que de simples exercices indépendants (telles les parties logique et automate).

Les correcteurs sont surpris, comme chaque année malheureusement, par le manque de rigueur déployé par les candidats dans le traitement des questions faisant appel à des arguments formels. Il semble que les candidats ne considèrent toujours pas l'Informatique comme une discipline proche des Mathématiques, et la programmation comme une activité aussi rigoureuse que la construction d'une preuve. Les arguments développés sont souvent des paraphrases du sujet, des déclamations d'évidence ou de trivialité, voire des absences complètes de justification. Il semble nécessaire de rappeler que l'absence de justification est généralement considérée comme une réponse partielle dans le meilleur des cas, et fautive dans le cas général.

La moyenne est de 10,11 et l'écart type de 2,90. Nous pouvons constater un net progrès par rapport à l'année précédente que nous pouvons attribuer à la réduction de la taille de l'épreuve. En prenant en compte cet aspect, nous pouvons considérer que les résultats sont similaires à l'année précédente.

Pour conclure, il est important d'insister, une fois de plus, sur le fait que les recommandations données dans les rapports depuis l'instauration de l'épreuve optionnelle d'Informatique, ne semblent toujours pas être prise en compte de manière effective. La qualité de la rédaction et du traitement des exercices et des questions récurrentes, n'est plus en progrès depuis les troisièmes et quatrièmes années, alors qu'une marge importante de progrès est encore disponible. Une légère régression est même notable. Pour faciliter l'intégration des remarques, les épreuves du concours commun polytechnique suivent chaque année la même structure et les mêmes attentes dans la rédaction en accord avec le programme et son esprit.