

A/ Remarques d'ordre général

1. Erreurs courantes

Dans les copies qui utilisent le langage CaML, les candidats écrivent "if (expression) then, true, else, false)" au lieu de la simple expression. Cela montre une incompréhension profonde de la nature fonctionnelle du langage dont chaque expression a une valeur. De la même manière, certains vont tester dans les conditionnelles (expression = true) ou (expression = false).

Les correcteurs, dans leur ensemble, signalent cette année encore que la majorité des copies contiennent de nombreuses fautes d'orthographe et de grammaire. Il s'agit certes d'une épreuve technologique mais cela n'excuse en rien ces erreurs.

La formalisation, dès qu'elle n'est pas triviale, pose des problèmes importants aux candidats (par exemple, la variable propositionnelle qui doit être introduite en *I.4*, ou l'expression formelle en *II.7*).

Les raisonnements sont, en général, peu rigoureux sous la forme d'un discours vague, redondant, qui reprend le sujet et n'explique pas les éléments qui permettent de conclure et pourquoi ils le permettent. L'exemple le plus flagrant concerne les raisonnements par récurrence ou par induction qui sont en général peu satisfaisants :

- le cas d'initialisation manque,
- la récurrence n'est pas formelle, mais le candidat déroule deux ou trois étapes et conclut par "et ainsi de suite" ou "... etc.",
- la notation n'est pas formelle, mais il s'agit d'un discours vague.

Globalement, les candidats pratiquent trop la paraphrase du sujet, et sont beaucoup plus succincts sur les étapes essentielles de la preuve. Les candidats qui maîtrisent la preuve demandée ont tendance à sauter des étapes essentielles en les jugeant "triviales". Ceux qui n'ont pas trouvé comment l'accomplir tentent d'obtenir quelques points en reprenant le texte du sujet et des questions environnantes. Il serait souhaitable d'obtenir un équilibre entre ces deux approches et de fournir des preuves mathématiques convenables contenant les détails nécessaires, et uniquement les détails nécessaires.

De nombreux traitements par cas en CaML sont incomplets (certains cas ne sont pas prévus).

Les correcteurs signalent qu'en général, les copies rédigées en langage PASCAL sont significativement de moins bonne qualité que celles en langage CaML.

2. Remarques sur le texte, sa compréhension

Le sujet proposé pour la session 2009 de l'épreuve optionnelle d'Informatique du concours commun polytechnique est composé de deux parties. D'une part la modélisation et la résolution en logique des propositions, et d'autre part l'algorithmique et la programmation en langage CaML ou PASCAL. Le sujet ne traite pas des automates et langages, ni des circuits logiques. La partie algorithmique est relativement large comprenant la formalisation et la preuve des algorithmes, le calcul de complexité et la programmation pour des données entières, listes et arbres. Le sujet offre donc une bonne

couverture des points essentiels du programme. Le sujet est sensiblement plus court que les années précédentes, principalement à cause de l'absence de partie sur les automates et les langages.

Le sujet était très détaillé et ne semble pas avoir globalement posé de problème de compréhension.

Les questions étaient de difficultés croissantes. Malgré le nombre plus restreint de questions par rapport aux années précédentes, les candidats se sont généralement arrêtés en milieu de seconde partie ; il est donc difficile de déterminer si la fin de cette partie était accessible ou pas.

3. Réactions (prévues ou non)

Malgré un sujet très classique dans le fond et la forme, certains candidats ne traitent pas correctement la partie logique.

Il est surprenant que de nombreux candidats ne soient pas capables de répondre correctement aux questions *II.3* à *II.6* sur le codage des nombres entiers. Celles-ci sont plus proches de questions de cours que de vraies questions de concours.

Les candidats ont souvent donné des preuves correctes différentes de celles qui étaient suggérées dans le sujet et proposées dans la correction officielle, en particulier pour la question *II.4*.

La difficulté des questions semble souvent dépasser ce que les candidats sont capables de traiter. En fait, les questions étant de difficulté progressive, les candidats sont aptes à traiter les premières questions mais ont des difficultés ensuite. Il n'est pas possible de savoir avec certitude s'ils manquent de temps ou s'ils ne sont pas capables de traiter les questions plus complexes.

4. Bilan

La partie Logique est en général bien traitée. La partie Programmation est en général traitée de manière satisfaisante. Par contre, la partie algorithmique, formalisation, preuve, calcul de complexité n'est en grande majorité pas abordée, et quand c'est le cas, elle est très faible.

La qualité générale des réponses aux questions semble en légère baisse. La qualité globale des copies poursuit sa baisse continue depuis la création des épreuves optionnelles. Cela est principalement dû à la baisse du nombre de questions traitées alors que le nombre de questions baisse également.

B/ Rapport détaillé (*erreurs rencontrées, rédaction de certains candidats, questions sans problème*)

J'indique, pour chaque question, le % de réussite, c'est-à-dire la note moyenne obtenue par rapport au barème attribué. Cette information ne permet pas de distinguer les réponses fausses de l'absence de réponse.

Logique et calcul des propositions

Cette partie est globalement bien traitée (66% de réussite) et un peu mieux réussie que les années précédentes. Étant donné son caractère systématique, il est encore surprenant de ne pas trouver une réussite à 100 % pour questions *I.1* à *I.3* qui ne contiennent aucune innovation. Les candidats

maîtrisent très bien le formalisme de la logique des propositions ainsi que les techniques de résolution syntaxique (formule de De Morgan) et sémantique (table de vérité).

- Modélisation sous la forme d'une formule du calcul des propositions (questions *I.1* - 88%, *I.2* - 89% et *I.4* - 62%).

Les candidats n'ont pas rencontrés de difficultés avec les questions classiques. Par contre, la plupart n'ont pas introduit de variable dans la question *I.4* pour représenter l'information manquante. Ils ont en général remplacé cette information par la disjonction des propositions possibles. Certains candidats ont utilisé une formulation sans variable correcte, quoique plus complexe quand ils dupliquent la formule, mais toutes ne permettent pas de conclure ensuite lors de l'interprétation de la résolution (B or (R xor V)).

Certains candidats ont proposé des formules erronées pour la question *I.1* (i.e. $A1 \text{ xor } A2 \text{ xor } A3$, $A1 \text{ or } A2 \text{ or } A3$, $A1 \Rightarrow \wedge A2 \vee \wedge A3$, ...).

L'expression logique pour l'affirmation *A1* est souvent fautive dans la question *I.4*.

- Résolution avec les formules de De Morgan et interprétation (question *I.3* - 69%).

La résolution et l'interprétation sont correctes dans la grande majorité des cas quand la modélisation est correcte.

Certains candidats remplacent $A \Rightarrow B$ par $\wedge A \vee (A \wedge B)$ ce qui est correct mais plus complexe que nécessaire.

- Résolution avec les tables de vérité et interprétation (question *I.5* - 45%).

Par contre, les étapes intermédiaires dans la table de vérité sont rarement bien choisies et certaines étapes essentielles sont masquées (i.e. la distinction explicite entre les deux valeurs possibles pour l'information incomplète).

Certains candidats utilisent De Morgan en lieu et place de la table de vérité. Ceci dénote une lecture trop rapide et peu approfondie du sujet. Ceci est souvent visible dans d'autres questions pour les mêmes candidats.

L'interprétation n'est pas commentée, en général, alors qu'il fallait donner quelques éléments sur l'information incomplète, qu'elle soit représentée par une variable ou une autre forme de codage.

Algorithmique et programmation en CaML/PASCAL

Le problème proposé consiste à étudier une technique d'allocation de mémoire exploitant un arbre binaire de blocs. Il permet aux candidats d'écrire des programmes manipulant des entiers, des listes et des arbres et couvre ainsi l'ensemble du programme en algorithmique et programmation.

La première partie sur les entiers et listes est globalement mieux traitée (59%) que la partie sur les arbres (16%) mais ce chiffre n'est pas très représentatif car cela correspond également à la fin du sujet. Le chiffre monte à 20% si l'on considère la fin à la question *II.17* qui correspond globalement à la fin des copies signalées par les correcteurs.

Le nombre de candidats qui utilisent les constructions impératives hors-programme interdites par le sujet pour plus d'équité, semble en nette régression cette année par rapport aux années précédentes.

Par contre, le langage PASCAL reste présent de manière significative (i.e. entre 15 et 20 % des copies).

Programmation avec entier et structure liste (questions *II.1* - 86%, *II.2* - 91%, *II.5* - 61% et *II.6* - 48%) :

- *Q. II.1* et *II.2* : ne posent aucun problème aux candidats sauf ceux qui ont considéré que l'adjacence devait être symétrique.
- *Q. II.5* : Le calcul est parfois fait en utilisant la formule proposée pour la question *II.3* sinon avec un algorithme récursif (terminal ou non-terminal). Certains candidats oublient qu'il faut un bit et un seul pour coder les nombres 0 (et pas 0 bits) et 1 (et pas 2 bits).
- *Q. II.6* : Certains candidats oublient qu'il faut un bit pour coder le nombre 0 et renvoie alors une liste vide de bits. Certains candidats construisent la liste en sens inverse.

Ces deux dernières questions n'auraient dû poser aucun problème car il s'agit quasiment de questions de cours. Cela n'a pas été le cas.

Programmation avec structure arbre (questions *II.12* - 57%, *II.13* - 23%, *II.15* - 8% et *II.18* - 2%).

- *Q. II.12* : Cette question est traitée correctement par la plupart des candidats.
- *Q. II.13* : Certains candidats oublient de vérifier les contraintes sur la taille. D'autres calculent la taille par une fonction auxiliaire et parcourent donc plusieurs fois l'arbre. Cette contrainte de parcours unique est en général mal prise en compte.
- *Q. II.15* et *II.18* : Ces questions sont souvent traitées de manière similaire par les candidats. Certains essaient de raffiner en plusieurs fonctions intermédiaires pour traiter les différentes contraintes et les différents cas.

Traitement d'exemples (questions *II.14* - 24% et *II.17* - 11%).

Ces questions sont souvent traitées correctement. Par contre, les candidats oublient en général certaines étapes intermédiaires.

Calcul de complexité (questions *II.16* - 4%, *II.19* - 1% et *II.20* - 0%).

Ces questions sont très peu souvent abordées par les candidats. La réponse à la question *II.16* est en général juste. La réponse à la question *II.19* est quasiment toujours erronée.

Spécification et preuve formelle (questions *II.3* - 34%, *II.4* - 51%, *II.7* - 31%, *II.8* - 15%, *II.9* - 25%, *II.10* - 19% et *II.11* - 6%) :

- *Q. II.3* : Certains candidats ont mal compris la question (i.e. $\max \{i \mid v_i = 1\}$) ou proposent une démonstration erronée (i.e. $E(\log_2(n))$, ...).
- *Q. II.4* : De nombreuses réponses différentes ont été proposées par les candidats autres que le raisonnement sur la forme binaire suggéré par le sujet (i.e. décomposition en facteurs premiers, récurrence, ...).
- *Q. II.7* : La réponse est souvent donnée en langage naturel au lieu d'une formule logique du premier ordre, ce qui était le résultat suggéré et attendu. La plupart des candidats oublient la propriété sur les fils gauche et droit d'un nœud $AA(G(a))$ et $AA(D(a))$. Cette récurrence sur la profondeur (ou induction) est parfois remplacée à tort par une quantification universelle.
- *Q. II.8* : De nombreux candidats se contentent de citer C3 sans traiter le cas des feuilles adjacentes qui ne sont pas filles d'un même nœud. Il y a eu de nombreuses mauvaises interprétations de la question qui n'était pourtant pas ambiguë.
- *Q. II.9* : Cette question est la mieux traitée par les candidats parmi celles de cette catégorie (souvent par une récurrence immédiate).

- *Q. II.10 et II.11* : Très peu de candidats ont proposé des réponses satisfaisantes pour ces questions.

Pour ces quatre questions, les correcteurs font l'hypothèse que les candidats n'arrivent pas à s'approprier en profondeur le sujet et font des réponses basées sur une compréhension trop superficielle qui masque certaines parties des théorèmes qui doivent être prouvés.

Les candidats se sont très majoritairement arrêtés à la question *II.17*.

Les candidats maîtrisent mal la spécification des algorithmes sous une forme logique et ne savent pas comment traiter la question *II.7*. Cet aspect devient fondamental en Informatique pour la rédaction de spécification d'exigence. Il est essentiel que les candidats acquièrent cette maîtrise pour les programmes, de la même manière qu'ils savent bien traduire des phrases en langage naturel en une formule de la logique des propositions.

La plupart des candidats qui abordent les questions concernant les arbres n'utilisent pas formellement une récurrence sur la profondeur (ou une induction structurale) et tentent d'explicitier les choses informellement par le discours.

C/ Conclusion

La moyenne de l'épreuve est de **8,74**. L'écart type est de **2,74**.

Ces valeurs reflètent bien l'avis des correcteurs par rapport à l'épreuve. Il y a très peu de très mauvaises ou de très bonnes notes. Le sujet contient des questions simples, proches de questions de cours qui permettent à tous les candidats de répondre de manière satisfaisante (*I.1 à I.3*, et *II.1 à II.6*, *II.12* et *II.13*). Par contre, le temps passé sur ces questions semble trop long et ne permet pas ensuite aux candidats de bien se distinguer sur les autres questions dont la difficulté devrait permettre une meilleure discrimination.