RAPPORT DE L'ÉPREUVE ÉCRITE D'INFORMATIQUE FONDAMENTALE
FILIÈRES MP ET MPI – CONCOURS INFO – SESSION 2023
ÉCOLES CONCERNÉES : ENS DE LYON, ENS DE PARIS, ENS DE PARIS SACLAY, ENS DE RENNES

Coefficients de l'épreuve (en pourcentage du total d'admission) :

École	MP	MPI
LYON	11,3%	11,3%
PARIS	13,3%	13,3%
PARIS SACLAY	13,2%	13,2%
RENNES	8,6%	8,3%

MEMBRES DE JURY : B. SIMON & N. FRANCIS & M. JEANMOUGIN

L'épreuve écrite d'informatique fondamentale concerne les candidates et candidats aux quatre Écoles Normales Supérieures sur le concours INFO, dans les deux filières MP et MPI. Le nombre de candidats ayant composé était de 225 en MP et 254 en MPI pour la session 2023, contre 311 en 2022 [1]. Les notes se sont échelonnées de 0,4 à 20 avec une moyenne de 9,9 et un écart-type de 3,9.

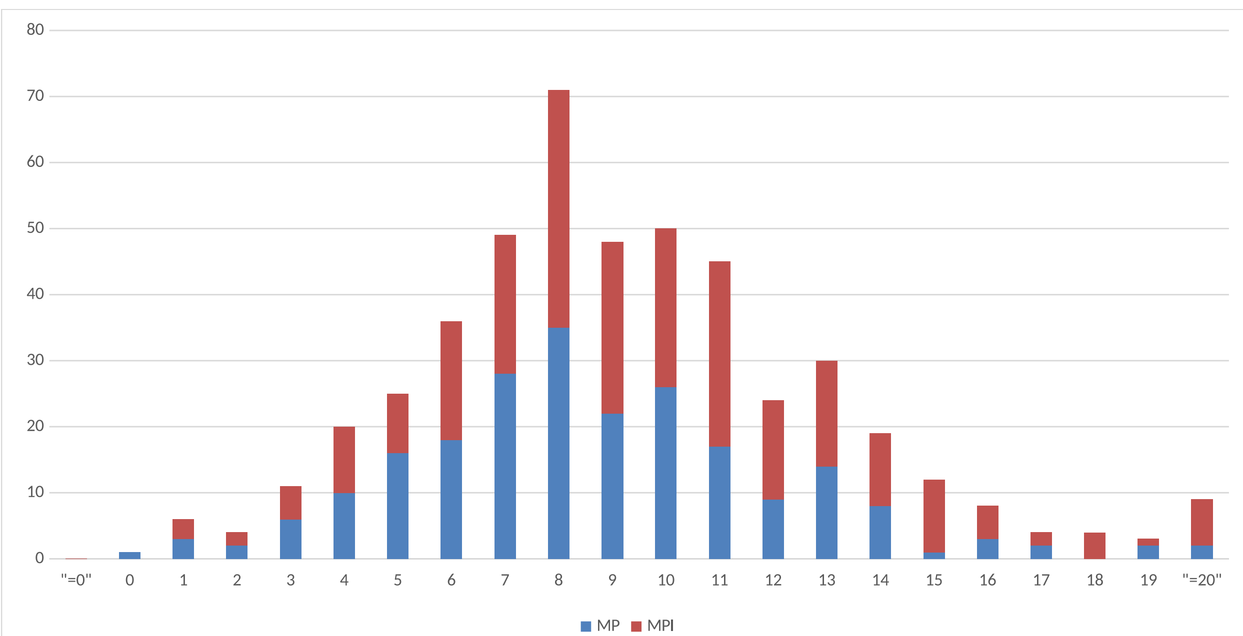

FIGURE 1 – Nombre de copies dans $[n; n+1[$.

1. L'année 2023 introduit la filière MPI pour la première fois, expliquant un important afflux de nouveaux candidats sur cette épreuve. Le chiffre de 2022 concerne l'épreuve « Maths-Info » du concours INFO de la filière MP

1 L'épreuve

Le sujet 2023 portait sur l'étude des automates finis dits *ambigus* et visait à établir des résultats de concision relativement à leurs équivalents déterministes d'une part, et ambigus d'autre part.

— La partie 1 permettait aux candidats de se familiariser avec les deux notions clefs du sujet : déterminisme et ambiguïté, et de constater en particulier qu'un automate non déterministe n'est pas forcément ambigu.

— La partie 2 s'intéressait à tester algorithmiquement le caractère ambigu ou non d'un automate. On commençait par établir les propriétés d'un automate auxiliaire, dont le vide du langage correspondant est équivalent à la non-ambiguïté de l'automate donné en entrée. Cette partie demandait en particulier aux candidats d'écrire un algorithme correct en pseudo-code ou dans le langage de leur choix.

— La partie 3 visait à prouver qu'il existe des familles de langages qui peuvent être reconnus par des automates non-ambigus exponentiellement plus concis que les automates minimaux déterministes équivalents. Autrement dit, il existe des cas dans lesquels remplacer le déterminisme (qui est une propriété plus forte) par la non-ambiguïté permet de réduire exponentiellement la taille des automates.

— À l'inverse, la partie 4 montrait qu'il existe des familles de langages dont tous les automates non-ambigus les reconnaissant sont au moins exponentiellement plus grands que les automates ambigus équivalents. Autrement dit, il existe des cas dans lequels la non-ambiguïté ne permet pas de gagner en concision par rapport au déterminisme : pour désambiguïser certains automates, on ne pourra pas espérer mieux que de les déterminiser. Cette partie était de loin la plus difficile de l'épreuve, et exploitait des résultats d'algèbre linéaire pour la démonstration finale.

2 Commentaires généraux

Le sujet de cette année était relativement court par rapport aux sujets des années précédentes, et une bonne partie des candidats sont parvenus à aborder l'ensemble du sujet. De ce fait, le jury a accordé une importance toute particulière à la rigueur et à la qualité des démonstrations. Le jury rappelle que toute affirmation doit être soigneusement justifiée, soit parce qu'elle provient d'une hypothèse de l'énoncé, soit parce qu'elle est issue d'un raisonnement formel utilisant éventuellement l'un des théorèmes du cours.

Le sujet demandait une certaine familiarité avec les automates et le jury a été surpris de constater que beaucoup de candidats n'étaient pas à l'aise avec les notions de base. Par exemple, les questions 3.1 et 4.1 demandant de construire "à la main" un petit automate à partir d'un langage donné explicitement ont été réussies par moins de la moitié des candidats. De même, le jury a relevé les mêmes affirmations fausses dans bon nombre de copies, qui témoignent d'un important manque de recul par rapport au contenu du cours. Par exemple, de nombreux candidats affirment que, si $q \xrightarrow{w} p$ et $q \xrightarrow{w'} p$ dans un automate déterministe, alors $w = w'$, ce qui impliquerait immédiatement qu'un automate déterministe ne peut pas reconnaître un langage infini !

L'épreuve demandait aux candidats de s'approprier la notion d'*ambiguïté* d'un automate, qui leur était nouvelle. Bon nombre de candidats n'ont pas réussi à comprendre précisément la différence entre *ambiguïté* et *non déterminisme*, ce qui les a lourdement pénalisés tout au long de l'épreuve [2]. Ainsi, le jury rappelle l'importance de faire soigneusement les questions introductives, qui visent généralement à rassurer le candidat sur sa bonne compréhension des définitions du sujet.

Passée la première partie, chacune des parties suivantes contenait au moins une question difficile, demandant au candidat de démontrer sa compréhension fine des notions et des enjeux du sujet. Comme à l'accoutumée, ces questions, très discriminantes, ont été bien plus valorisées que les questions d'application et d'exemple. Ainsi, les stratégies visant à éviter les questions difficiles pour se limiter aux questions introductives de chaque partie n'ont pas été très rentables.

3 Commentaires détaillés

Afin de comparer la réussite à des questions indépendamment de leurs poids respectifs, les moyennes et écart-type par question sont données sur 1.

Sont reportés ci-dessous, pour chaque question :
- Le nombre de points associés ;
- le nombre de copies ayant reçu des points pour cette question (c'est à dire le nombre de copies où la question a été traitée et non nulle) ;
- la moyenne **sur 1** de toutes les copies ayant traité la question ;
- l'écart-type **sur 1** de toutes les copies ayant traité la question.

Pour chaque partie, on donne le nombre total de points qui pouvaient être obtenus, puis la moyenne et l'écart type des points obtenus.

Partie 1 (5.1 pts | moy 3.9 stdev 0.8)

Cette partie introductive permettait aux candidats et candidates de se familiariser avec la notion (nouvelle) d'*ambiguïté*.

▷ **Question 1.1. [0.2 pt | 440 / 0.9 / 0.3]** Toute description simple et intuitive permettant d'avoir les points, quasiment tous les candidats ont correctement traité la question.

▷ **Question 1.2. [0.4 pt | 472 / 0.8 / 0.3]** La définition (donnée) de déterminisme possède deux conditions, l'oubli de parler de $|I|$ pour A_3 a donc été sanctionné.

▷ **Question 1.3a. [0.4 pt | 386 / 0.7 / 0.4]**

▷ **Question 1.3b. [0.4 pt | 402 / 0.7 / 0.4]**

2. En particulier, il ne suffit pas d'exhiber deux calculs différents sur un même mot pour prouver qu'un automate est ambigu ; encore faut-il que ces calculs soient tous les deux *acceptants*.

▷ **Question 1.3c. [0.4 pt | 467 / 0.8 / 0.3]** Cette question permet d'introduire un automate non-déterministe, mais non-ambigu, et n'a pas posé de problèmes spécifiques. La moitié des points était attribuée au résultat, l'autre aux justifications.

▷ **Question 1.4. [0.8 pt | 461 / 0.8 / 0.3]** Cette question est la première question d'une série de 4 demandant de formaliser une preuve, et les affirmations non justifiées y ont été pénalisées. Il convient, particulièrement en début d'épreuve, de faire preuve de rigueur pour mettre en confiance le correcteur, et donc de suivre les définitions données. En l'occurrence, il était attendu une mention non seulement des transitions mais également des états initiaux et finaux.

▷ **Question 1.5. [0.4 pt | 467 / 0.8 / 0.2]** L'involutivité de $A \to \tilde{A}$ a été admise pour les copies y faisant référence, mais les candidats se contentant de montrer une implication ont été pénalisés.

▷ **Question 1.6. [0.8 pt | 407 / 0.7 / 0.3]** Le jury déplore de nombreuses copies se contentant de paraphraser l'affirmation à prouver, sans apporter d'argument. Des points d'attention sur cette question concernent le fait de considérer un mot dans un langage sans vérifier le cas du langage vide, et les récursions (fausses) se basant sur les longueurs de mot dans un langage donné.

De manière générale, les erreurs mineures entrainent une petite pénalité sur la question, alors qu'une preuve structurellement fausse n'apporte aucun point.

▷ **Question 1.7. [0.4 pt | 471 / 1.0 / 0.1]** Question simple, mais peu discriminante, la quasi-totalité des candidats l'ayant traité.

▷ **Question 1.8.i. [0.4 pt | 371 / 0.8 / 0.4]**

▷ **Question 1.8.ii. [0.4 pt | 341 / 0.7 / 0.4]**

▷ **Question 1.8.iii. [0.4 pt | 417 / 0.9 / 0.3]** Ces questions d'exemple permettaient de s'assurer de la bonne maîtrise des notions introduites (en particulier avec la justification), et ont été plutôt bien traitées dans l'ensemble.

Partie 2 (10.0 pts | moy 3.3 stdev 1.6)

Cette partie introduit un algorithme de test d'ambiguïté, se basant sur un automate construit comme un produit de l'automate de départ avec lui-même et un booléen indiquant l'égalité des chemins suivis (ainsi, un chemin parcouru par un mot dans un tel automate correspond à une paire de chemins suivis par ce mot dans l'automate de départ, et arriver à une paire d'états finaux en ayant à un moment « activé » le booléen indiquant que les chemins diffèrent, permet de conclure que l'automate est ambigu pour ce mot). L'automate étant posé formellement sans explications, les deux premières parties servaient à introduire (et démontrer) cette interprétation, avant de la généraliser dans la troisième partie.

▷ **Question 2.1. [0.6 pt | 323 / 0.5 / 0.4]** Le jury a été surpris par la faible proportion de candidats construisant correctement l'automate avec la description donnée (en particulier, l'état (2,1,1) ou la boucle sur l'état final ont été souvent ignorés).

De nombreux candidats ayant un automate faux ont toutefois donné le bon langage, probablement à partir de la lecture de la Q2.3 et de la Q1.3, et obtenu une partie des points.

▷ **Question 2.2.i. [0.4 pt | 433 / 0.8 / 0.3]** Question simple, bien traitée dans l'ensemble.

▷ **Question 2.2.ii. [0.6 pt | 412 / 0.7 / 0.3]** De rares candidats ont été récompensés pour traiter le cas où $w = \varepsilon$ souvent oublié par les raisonnements se basant sur T.

▷ **Question 2.2.iii. [0.4 pt | 420 / 0.8 / 0.3]** Invoquer (i) et la définition de F suffit.

▷ **Question 2.2.iv. [0.4 pt | 369 / 0.7 / 0.3]** L'hypothèse de la partie 2 $L(A) \neq \emptyset$ est nécessaire ici. Les points ont été donnés si la copie montre que suivre « en double » un chemin acceptant de A nous amène sur ce cas.

▷ **Question 2.3. [0.8 pt | 371 / 0.6 / 0.3]** Une démonstration formelle était attendue sur cette question de conclusion, où il convient d'être précis sur les hypothèses des questions utilisées et sur la manière de construire le calcul ρ dans le sens réciproque.

▷ **Question 2.4. [0.2 pt | 401 / 0.8 / 0.3]** Le jury attend ici simplement des bornes quadratiques en fonction des tailles des ensembles correspondants de A.

▷ **Question 2.5. [1.3 pt | 200 / 0.4 / 0.4]** Cette question a introduit des propositions extrêmement variées de la part des candidats, allant de la paraphrase de la définition de l'automate à un code ocaml détaillé. Le jury s'attendait en général à un pseudocode faisant explicitement apparaître les boucles et les tests.

▷ **Question 2.6. [0.6 pt | 268 / 0.5 / 0.4]** Le jury attend un parcours de graphe *à partir de chacun des états initiaux*, qui peut se réaliser avec un parcours en largeur ou en profondeur. La complexité d'un parcours de graphe (O(|sommets|+|arêtes|)), bien qu'au programme, n'est souvent pas utilisée, ou fausse.

La réinvention d'algorithmes peu efficaces dont le but est de parcourir un graphe a en général été pénalisée.

▷ **Question 2.7.i. [0.4 pt | 319 / 0.8 / 0.4]** Le jury a été généreux avec les erreurs de longueurs du mot (Q^k au lieu de Q^{k+1}), le concept pertinent ici étant « exponentiel ».

▷ **Question 2.7.ii. [0.4 pt | 277 / 0.7 / 0.4]** Le jury s'attend ici à une réponse démontrant la compréhension du cas général, un automate à zéro ou un état n'ayant pas d'intérêt pour la généralisation de la sous-partie.

▷ **Question 2.8. [3.8 pt | 42 / 0.1 / 0.3]** Cette question, l'une des plus difficiles du sujet, demandait de bien comprendre la construction de $\hat{A}$ et comment la généraliser à la détection de k chemins *deux à deux distincts*. En particulier,

- toutes les constructions itérant ˆ sont immédiatement fausses et ne rapportent aucun point, puisque $L(\hat{\hat{A}}) = L(\hat{A})$. En effet, tout chemin acceptant de $\hat{A}$ correspondant à un (μ, μ') de A^2 a un chemin miroir correspondant à (μ', μ), et est donc ambigu.
- les constructions avec Q^2 et un compteur ne peuvent a priori pas suivre plus de deux chemins parallèles par construction, et sont donc de même immédiatement fausses.
- les constructions impliquant Q^k et un compteur dans $[0,k]$ qui s'incrémente quand deux chemins divergent ont obtenu une partie des points, mais ne fonctionnent que sur les arbres : deux chemins arrivant à un même état, puis se reséparant en deux, produisent 4 chemins possibles, et non trois, et sont indistinguables (en terme d'états) de cas qui ne produisent que trois chemins.
- La construction attendue demandait de remarquer que pour détecter k chemins *deux à deux distincts*, il est nécessaire de conserver au moins $\binom{k}{2}$ bits d'information pour garder trace de si le chemin i et le chemin j ont divergé.
  Les candidats ayant proposé un ensemble d'états permettant de s'en assurer (que ce soit $Q^k \times [0 \ldots 2^{\binom{k}{2}}]$, $Q^k \times \{0,1\}^{\binom{k}{2}}$ ou $Q^k \times M_k(\{0,1\})$) et des définitions compatibles pour les états initiaux, finaux, et les transitions, ont été fortement valorisés.

Le jury n'attend évidemment pas une preuve formelle, aussi détaillée que le reste de la partie 2, qu'une telle construction permet effectivement de calculer le langage demandé, mais proposer le bon automate donne le poids nécessaire à l'ellipse « De même qu'aux questions 2.2 et 2.3, ... » pour conclure.

▷ **Question 2.9. [0.6 pt | 153 / 0.7 / 0.4]** L'argument attendu sur cette question est qu'une différence de langages réguliers est un langage régulier.

Partie 3 (4.9 pts | moy 1.6 stdev 1.3)

▷ **Question 3.1. [0.8 pt | 413 / 0.6 / 0.3]** Lorsque des candidats décident de déterminiser un automate avec l'automate des parties, le jury recommande d'annoter les états du déterminisé avec des noms montrant la compréhension de l'état obtenu (soit $\{0,4\}$, soit *baa*, par exemple) pour éviter des erreurs ou des oublis (et pour faciliter la correction).

▷ **Question 3.2. [0.4 pt | 394 / 0.8 / 0.3]** Si la question demande bien de généraliser la précédente, le jury s'attend à une justification sur le côté non-ambigu de l'automate (par exemple, en remarquant qu'il est co-déterministe)

▷ **Question 3.3. [0.9 pt | 163 / 0.4 / 0.4]** La simple déterminisation de l'automate de la question précédente ne donne pas le bon nombre d'états sans arguments sur les états accessibles.

▷ **Question 3.4. [0.6 pt | 313 / 0.6 / 0.4]** Induction simple, traitée correctement dans l'ensemble.

▷ **Question 3.5. [1.9 pt | 82 / 0.2 / 0.4]** Si la réciproque ne pose pas de problème, le sens direct demande une preuve délicate (en général par l'absurde), qui fait intervenir l'égalité des longueurs des mots. Certaines copies ont supposé que l'automate est co-déterministe, et ont été fortement pénalisées.

▷ **Question 3.6. [0.4 pt | 165 / 0.7 / 0.4]** On vient de démontrer l'injectivité d'une fonction de $\{a,b\}^n \to Q$, il ne reste ici qu'à conclure.

Partie 4 (12.0 pts | moy 1.0 stdev 1.6)

▷ **Question 4.1.i. [0.6 pt | 261 / 0.7 / 0.4]** Cette question ne posait pas de difficultés particulières et a été plutôt bien réussie. L'erreur la plus fréquente a été d'imposer que la première lettre du mot est aussi celle qui apparaît deux fois.

▷ **Question 4.1.ii. [0.9 pt | 88 / 0.4 / 0.5]** Le sujet teste encore une fois la bonne compréhension de la notion d'*ambiguïté*. De nombreux candidats ont ici proposé un automate *ambigu*, qui ne répondait donc pas à la question, et n'ont donc marqué aucun point. L'automate erroné le plus souvent proposé est constitué de trois chemins, reconnaissant respectivement les mots contenant deux a, deux b et deux c. Malheureusement, si un mot satisfait simultanément plusieurs de ces conditions, il peut alors être reconnu par plusieurs calculs différents, et est donc ambigu.

Le jury est également surpris des copies proposant des automates contenant strictement moins de 9 états, alors que le but annoncé de cette partie est précisément de montrer qu'il n'en existe pas.

▷ **Question 4.2. [0.6 pt | 206 / 0.5 / 0.4]** Il s'agissait ici de généraliser la construction de la question 4.1.i. Pour marquer tous les points, le jury attendait au minimum une phrase d'explication justifiant la taille de l'automate et le langage reconnu. Un dessin, contenant forcément des parties ellipsées puisque n n'est pas donné explicitement, est certes utile à la compréhension, mais ne constitue pas une preuve.

▷ **Question 4.3. [0.9 pt | 78 / 0.5 / 0.4]** Comme à la question précédente, il s'agissait ici de généraliser l'exemple de la question 4.1.ii, et le même barème a été utilisé.

Certains candidats ont essayé d'argumenter qu'il suffit de déterminiser l'automate de la question 4.2. Cette approche n'aboutit malheureusement pas au nombre d'états demandés, et n'a donc marqué aucun point.

▷ **Question 4.4.i. [0.8 pt | 105 / 0.5 / 0.5]** *(voir question suivante)*

▷ **Question 4.4.ii. [0.4 pt | 69 / 0.5 / 0.5]** La question demandait de construire un calcul *acceptant* de $\mathcal{A}$ sur $u' \cdot v$, à partir des calculs sur $u \cdot v$ et $u' \cdot v'$ (i), puis de constater que l'état atteint après avoir lu u' (appelé, par définition, $q_{u',v}$) était bien $q_{u,v}$ (ii).

Le jury a sanctionné les candidats ayant manqué une partie du raisonnement, soit en n'argumentant pas que la séquence d'états construite est bien un calcul acceptant, soit en n'expliquant pas l'égalité entre $q_{u',v}$ et $q_{u,v}$. En effet, il ne suffit pas que le calcul construit "passe par $q_{u,v}$", comme de nombreuses copies l'ont affirmé, il faut que ce soit précisément l'état atteint après avoir lu u'.

De plus, le jury n'a donné aucun point aux candidats qui confondent *non-ambiguïté* et *déterminisme* dans leur raisonnement. Par exemple, de nombreuses copies affirment qu'il n'y a qu'un seul état accessible en lisant v' depuis $q_{u,v}$ "par non-ambiguïté de $\mathcal{A}$", ou encore que "le calcul de $\mathcal{A}$ sur u est unique, donc $q_{u,v} = q_{u,v'}$".

▷ **Question 4.5. [0.8 pt | 107 / 0.6 / 0.4]** La question permettait de vérifier la bonne compréhension de la définition de M_n, et ne présentait pas de difficultés particulières.

La question étant plutôt simple, le jury a été particulièrement exigeant sur la rigueur et la précision de la rédaction. Ainsi, les candidats qui se sont contentés d'affirmer que comparer un élément de s^{n+1} à un élément de s^n revient à comparer deux éléments de s^n, ce qui justifie la forme de la matrice, sans donner de preuve ou sans préciser les correspondances entre les éléments des deux suites, ont été assez fortement pénalisés.

Idéalement, il était attendu que le candidat quantifie sur un coefficient quelconque de M_n, déconstruise formellement la définition de la matrice et des suites, et conclue avec la bonne valeur du coefficient choisi.

▷ **Question 4.6. [0.8 pt | 82 / 0.6 / 0.5]** Le sens direct s'obtient immédiatement par définition de M_n et de K_n, mais il était attendu que les candidats fassent au moins la remarque.

Pour le sens réciproque, il s'agit de montrer que les éléments de s contiennent au plus une fois chaque lettre.

Il était possible de raisonner directement par équivalences, mais attention dans ce cas à ce que chaque affirmation maintienne bien l'équivalence. Certains candidats se sont contentés d'écrire la preuve (immédiate) du sens direct en affirmant que chaque étape du raisonnement est une équivalence et en omettant l'argument nécessaire à la preuve du sens réciproque.

▷ **Question 4.7. [3.4 pt | 11 / 0.2 / 0.4]** Cette question était de loin la question la plus difficile de l'épreuve. Peu de candidats l'ont tentée, et très peu de candidats ont marqué des points.

Il *suffisait* ici de prouver l'égalité proposée par le sujet, typiquement par deux inégalités. Les deux inégalités étaient difficiles, et la preuve nécessitait en particulier d'invoquer précisément le résultat de la question 4.4. Les candidats ayant mené à bien la question ont démontré leur compréhension fine des objets manipulés par le sujet et de la stratégie de preuve générale, et ont donc été fortement récompensés.

Parmi les tentatives infructueuses, une bonne part des candidats ayant essayé la question ont tenté de prouver que pour chaque i, il existe un unique q tel que $v_q[i] = 1$. Bien que suffisante, cette propriété n'est pas nécessaire, et impliquerait en fait que l'automate est *déterministe*. Toute la difficulté de la question reposait justement sur la bon traitement du cas où deux tels vecteurs v_q et $v_{q'}$ ont un coefficient non nul à la même position, en montant qu'ils ne peuvent alors pas tous deux contribuer simultanément au même coefficient de M_n.

▷ **Question 4.8. [1.3 pt | 10 / 0.1 / 0.2]** Cette question, assez technique, était en fait indépendante du reste du sujet et s'appuyait uniquement sur le résultat de la question 4.5. De nombreux candidats, vraisemblablement bloqués sur les autres parties du sujet, ont tenté d'y répondre, mais peu d'entre eux ont marqué des points, ce qui explique la très faible moyenne de la question.

Le jury a été surpris de l'importante quantité d'arguments *évidemment faux* donnés à cette question, peut-être expliqués par la fatigue en fin d'épreuve et la difficulté de changer de contexte, pour aller des langages formels vers l'algèbre linéaire. Par exemple, de nombreux candidats se permettent de sommer les rangs des blocs de la matrice sans aucune précaution ou bien affirment que $\mathbf{1}_n$ est de rang plein.

▷ **Question 4.9. [0.8 pt | 38 / 0.5 / 0.4]** La question demandait d'assembler les résultats des questions 4.7 et 4.8, puis de faire un petit raisonnement d'algèbre linéaire pour conclure : il fallait typiquement remarquer que, comme $\{v_q\}$ engendre M_n, alors $rg(\{v_q\}) \geq rg(M_n)$, et donc que $\{v_q\}$ contient une famille libre de taille au moins 2^{n-1}, dont tous les vecteurs sont donc nécessairement non nuls.

Le jury n'a pas donné de point aux candidats qui ont affirmé que la conclusion était une conséquence immédiate de 4.7 et 4.8, sans plus d'arguments.

▷ **Question 4.10. [0.8 pt | 21 / 0.5 / 0.5]** La question ne présentait pas de difficultés particulières : il s'agissait de montrer, par contraposée ou par l'absurde, que si q est un état initial ou final, alors $v_q = 0$, en utilisant encore une fois le fait que les éléments de s ne contiennent pas deux occurrences de la même lettre.

▷ **Question 4.11. [0.6 pt | 19 / 0.3 / 0.4]** Finalement, cette question conclusive demandait d'assembler les résultats des questions 4.9 et 4.10 pour obtenir 2^{n-1} états non-initiaux et non-finals, puis d'argumenter que $\mathcal{A}$ contient un état initial et un état final (sans quoi le langage reconnu serait vide), forcément distincts (sans quoi le mot vide serait accepté).

En fin d'épreuve, de nombreux candidats, vraisemblablement bloqués, essaient de marquer des points en essayant de deviner quelles questions précédentes invoquer pour résoudre la question posée. Leurs démonstrations se résument en général à dire "d'après les questions précédentes, le résultat est immédiat." Ces tentatives ne marquent généralement aucun point : il est en effet très rare qu'une question, même d'assemblage, ne requière pas au moins un argument supplémentaire.

Ainsi, les candidats ayant affirmé que le résultat découlait immédiatement des questions 4.9 et 4.10 n'ont pas marqué de point.