Rapport sur l'épreuve écrite d'informatique 2004 groupes I et MPI, Écoles Normales Supérieures

Hubert Comon-Lundh

26 juillet 2004

1 Présentation du sujet

Le sujet portait sur le problème d'accessibilité dans les graphes ET/OU, problème pour lequel il existe un algorithme linéaire (question 7) et qui est par ailleurs l'un des prototypes de problèmes complets pour PTIME. L'algorithme est souvent re-découvert dans différents contextes, par exemple en logique propositionnelle ou en théorie des automates. La deuxième partie du problème portait justement sur (une version simple des) automates alternants, avec d'une part des questions algorithmiques et d'autre part des question d'expressivité. En particulier, il est possible à l'aide des automates alternants, d'obtenir une représentation exponentiellement plus succincte des langages réguliers. La technique de preuve est semblable à celle qui est suggérée dans la question 13.

Cette épreuve d'une durée de 4 heures avait pour but de tester l'aptitude des candidats à :

1. concevoir des algorithmes simples (questions 4,5,6,11),
2. étudier un algorithme particulier (question 7)
3. raisonner sur un modèle de calcul simple (questions 8 à 13)

Les questions 1 à 3 étaient superflues et n'avaient pour but que de se familiariser avec les notions introduites. Malheureusement, les questions 1 et 2 étaient fausses. Nous expliquons dans le paragraphe 3 comment les correcteurs en ont tenu compte. Nous commençons par plusieurs remarques générales dans le paragraphe 2 et concluons par des commentaires plus détaillés question par question dans le paragraphe 4.

Le sujet était faisable en 4 heures (si l'on excepte la question 13.2) par des candidats exceptionnels. Quelques-uns ont d'ailleurs abordé toutes les questions, sauf 13.2.

2 Remarques générales

Comme les années précédentes, nous avons choisi de privilégier la conception d'algorithmes. Un candidat qui aurait traité complètement les questions 4 à 7 aurait obtenu la note maximale.

Lors de la description des algorithmes, nous n'avons pas tenu compte des détails de syntaxe ni du style de programmation, dans la mesure où les constructions (structures de contrôle, instructions élémentaires) avaient une signification claire, ce qui était presque toujours le cas. Le langage algorithmique de l'annexe n'était donné qu'à titre d'exemple. Par contre, les candidats qui déclarent décrire leurs algorithmes en CAML puis qui commettent des erreurs de type ou des erreurs de référence/déréférence ont été sanctionnés. En effet, beaucoup de candidats utilisent les pointeurs de CAML et commettent des erreurs dues à une mauvaise compréhension de leur signification.

L'immense majorité des candidats est capable d'écrire un algorithme correct pour l'union ou l'intersection d'ensembles, quelle qu'en soit la représentation. Par contre, l'évaluation des algorithmes s'avère très souvent incorrecte dès qu'il y a plusieurs boucles imbriquées (dont les bornes ne sont pas connues à l'avance).

Les candidats n'ont pas été sanctionnés lorsqu'ils ont donné un algorithme de complexité plus élevée que celle du corrigé, dès l'instant que l'évaluation de complexité est correcte et qu'on ne passe pas d'un algorithme polynômial à un algorithme exponentiel, voire pire.

3 Comment il a été tenu compte des erreurs d'énoncé

Les questions 1 et 2 de l'énoncé étaient fausses. L'erreur de la question 1 a été remarquée par la moitié des candidats environ. Cette erreur ne semble avoir gêné personne. Par contre, l'erreur de la question 2 n'a été relevée que dans très peu de copies, alors que 40% environ des candidats n'ont pas pu la résoudre (à juste titre). Dans ce dernier cas, le problème est qu'il n'est pas possible d'évaluer le temps perdu par chaque candidat. Voici ce que les correcteurs ont décidé afin de ne pénaliser le moins possible ceux qui ont perdu du temps sur cette question :

- Les candidats ayant remarqué l'erreur de la question 1 ont reçu une gratification
- Les candidats n'ayant pas pu résoudre la question 2 ont aussi reçu une gratification, celle-ci étant plus importante s'ils ont de plus remarqué que la question était fausse.
- Pour réduire les inégalités créees par le temps passé le cas échéant à tenter de résoudre la question 2, il a été procédé comme suit. Soit P_1 la population des candidats n'ayant pas buté sur cette question et P_2 le reste des candidats. Une transformation f a été appliquée aux notes de sorte à réduire l'écart type de P_2. De plus f vérifie :
  - si $x, y \in P_1$ et $x > y$, alors $f(x) > f(y)$,
  - si $x, y \in P_2$ et $x > y$, alors $f(x) > f(y)$,
  - si $x \in P_2, y \in P_1$ et $x > y$, alors $f(x) > f(y)$.
  - si m est la moyenne de P_2, $x \in P_1, y \in P_2$, $x > m > y$, alors $f(x) > m > f(y)$.

On peut remarquer par ailleurs que, même sans ces transformations, la moyenne des notes de P_1 était significativement plus basse que celle de P_2. Enfin, les notes finales de cette épreuve sont assez bien corrélées avec les autres notes (écart type des différences à 2).

4 Commentaires sur chaque question

Le taux de réussite est ici la moyenne des notes obtenues à la question sur la note maximale (on ne tient pas compte d'éventuelles gratifications pour les questions 1 et 2).

question 1. Taux de réussite : 98%.

question 2. Taux de réussite : 97%.

question 3. Taux de réussite : 24%.

question 4.1. Taux de réussite : 84%.

question 4.2. Taux de réussite : 69%. Pour une question aussi facile, on espérait que, en cas de choix de la représentation par tableaux de Booléens, l'algorithme évite de parcourir entièrement les deux tableaux.

question 4.3. Taux de réussite : 80%. Ceux qui ont choisi la représentation sous forme de listes sans répétition ont souvent donné un algorithme en $O(|E_1| \times (|E_1| + |E_2|))$. Dans de nombreuses copies l'évaluation de la complexité est, de plus, inexacte dans ce cas.

question 4.4. Taux de réussite : 52% Dans le cas des ensembles représentés par des listes sans répétition, on accepte aussi la majoration par $O(|V|^3)$.

question 4.5. Taux de réussite : 20%. Parmi les erreurs les plus courantes :
- Utilisation de boucles imbriquées, puis erreurs d'indices
- Calcul de complexité absent ou incorrect
- Pour les tableaux de Booléens : pas de création des tableaux représentant un ensemble singleton
- Utilisation de "=" (égalité de listes) au lieu de l'égalité d'ensembles
- Pas de définition de inter ou ginter, ou mauvaise initialisation de ginter
- Utilisation de la concaténation au lieu de l'union

question 5.1. Taux de réussite : 48%.

Certains candidats (environ un sur vingt) ont considéré que les arêtes étaient bi-directionnelles, bien qu'elles soient clairement orientées sur la figure (dans le cas des sommets 6 et 9, il y a une arête dans chaque sens) et enfin que E n'ait jamais été supposé symétrique. Comme le calcul était beaucoup plus simple dans ce cas, les candidats n'ont obtenu que 25% des points dans ce cas de figure.

Il fallait un peu de patience pour cette question qui était notée en conséquence.

question 5.2. Taux de réussite : 28%.

10% environ des candidats montrent seulement la convergence de $A_n(s)$ pour chaque s. Ils affirment ainsi que l'on peut choisir $M = N$. Ces candidats n'ont pas obtenu les points de la question.

50% des candidats montrent l'existence de M mais ne précisent pas sa valeur. Ces candidats obtiennent les deux tiers des points.

20% des candidats donnent une valeur exponentielle pour M. Ces candidats obtiennent un tiers des points.

question 5.3. Taux de réussite : 20%.

Tous les candidats ont vu qu'il suffisait de montrer que A est compatible et que, pour toute application compatible g et tout sommet s, $A(s) \subseteq g(s)$, mais beaucoup n'y parviennent pas. La justification "A est compatible par construction" sans plus d'arguments était insuffisante.

question 5.4. Taux de réussite : 8.6%. Un algorithme simple consistait à itérer le calcul de A_n jusqu'à $n = N \times (N-1)$. Cependant, ceux qui appliquent cette solution peu élaborée n'ont obtenu que les deux tiers des points, d'une part parce qu'un tel algorithme est particulièrement inefficace, d'autre part parce qu'il évite la difficulté du calcul des points fixes.

Les candidats qui veulent calculer A_n jusqu'à ce que $A_n = A_{n+1}$ commettent souvent l'erreur de ne pas recopier le tableau à chaque itération ; ils utilisent un algorithme de la forme “pour $s \in V$ faire (si $f(s) = \vee$ alors $A(s)$:= union (A(s), gunion(G(s),A))”

qui, en fait, ne calcule pas A_{n+1} à partir de A_n ; il faudrait remplacer A par un autre tableau B en membre gauche, puis recopier après chaque itération. (Noter cependant que l'algorithme esquissé ci-dessus calcule bien la relation d'accessibilité, et même de manière plus efficace, mais les candidats ayant suivi cette voie ont cru calculer A_n et ont donc été sanctionnés).

question 6. Taux de réussite : 26% Cette question était facile. Nous avons sanctionné l'absence d'étude de coût ainsi que la description trop informelle d'un algorithme.

question 7.1. Taux de réussite :6.3%.

La plupart des candidats n'ont pas vu qu'il faut d'abord montrer que chaque sommet ne peut entrer qu'au plus une fois dans S, ce qui est essentiel pour le calcul de la complexité.

question 7.2 Taux de réussite :0.6%.

C'était la question la mieux payée de cette partie, et aussi la plus délicate. Très peu de candidats l'ont abordée, mais la plupart de ceux qui l'ont fait ont visiblement compris, même si les solutions proposées manquaient parfois de rigueur. Nous avons donné les points aux candidats qui ont montré l'indication, vu comment l'utiliser et vu également qu'il fallait faire une récurrence sur n pour la réciproque.

question 7.3. Taux de réussite : 4%.

Cette question a été abordée par beaucoup de candidats (pour la plupart après avoir sauté les questions 7.1 et 7.2). Pour obtenir le maximum de points, nous attendions une justification du fait que prec(s) est sans répétition et une analyse de coût.

Ceux qui avaient choisi une représentation des ensembles sous forme de tableaux de Booléens et ont donné une complexité en $O(|V|^2)$ n'ont pas été sanctionnés.

question 7.4. Taux de réussite : 7.5%

question 8. Taux de réussite : 31%.

question 9. Taux de réussite : 6%.

La question a été bien réussie dans l'ensemble, mais les réponses étaient souvent trop informelles.

qustion 10. Taux de réussite : 0.6%.

Beaucoup de candidats ayant abordé cette question ont tenté un coup de bluff. Sans, au minimum, l'énoncé d'une récurrence, nous n'avons pas mis de point. Une erreur assez courante aussi consiste à montrer que

Si w est accepté par $\mathcal{A}$, alors w n'est pas accepté par $\mathcal{A}^c$

puis à prétendre que la réciproque s'obtient en remplaçant $\mathcal{A}$ par $\mathcal{A}^c$ et en remarquant que $(\mathcal{A}^c)^c = \mathcal{A}$. (En d'autres termes, de nombreux candidats ont confondu réciproque et contraposée).

Il était néanmoins possible de raisonner par récurrence et par équivalence, ce que certains candidats ont fait correctement.

question 11. Taux de réussite : 0.9%.

Cette question a été abordée dans une trentaine de copies. Dans une bonne moitié d'entre elles, l'algorithme proposé est l'algorithme naïf,... exponentiel.

Aucun candidat n'a donné la solution proposée par le corrigé (qui consistait à utiliser la question 7), mais plusieurs solutions correctes ont été proposées, consistant à calculer, pour chaque suffixe w' de w, l'ensemble des états à partir desquels w' est accepté. Il fallait néanmoins faire attention aux structures de données pour obtenir un algorithme linéaire.

question 12. Taux de réussite : 0.3%.

La question n'a été abordée que dans peu de copies. De façon surprenante, dans ces copies l'idée était presque toujours la bonne. Cependant, aucun candidat n'a eu le temps de développer une solution rigoureuse.

question 13.1. Taux de réussite : 18%.

Cette question a été abordée par beaucoup de candidats qui ont sauté les questions qui précèdent. Mais elle n'était pas bien payée (environ 0.25 points sur 20).

question 13.2. Taux de réussite : 0%. La question n'a été significativement abordée dans aucune copie.