Mines Informatique MP 2005 — Corrigé

Ce corrigé est proposé par Boris Yakobowski (ENS Cachan) ; il a été relu par Samuel Mimram (ENS Lyon) et Vincent Puyhaubert (Professeur en CPGE).

Le sujet est constitué de deux parties totalement indépendantes.

La première partie traite de langages et d'automates. On étudie la transformation de langages (rationnels ou non) par trois fonctions. Parallèlement, étant donné un automate reconnaissant un langage, on en déduit un automate reconnaissant l'image de ce langage par une des fonctions de transformation. Cette étude permet, entre autre, de montrer la non rationalité d'un langage sans utiliser directement le lemme de l'étoile.

Une importante fraction du programme consacré aux automates est abordée : expressions et langages rationnels, automates non déterministes, lemme de l'étoile.

Cette partie est constituée de deux questions d'introduction et de trois sous-parties. Bien que ces dernières paraissent indépendantes, des résultats de la première sous-partie peuvent être utilisés dans la troisième. Les questions les plus difficiles sont les questions 5 et 9.

La seconde partie s'intéresse au problème de la satisfiabilité d'une formule booléenne mise sous la forme d'une conjonction de disjonctions à 2 littéraux (appelée forme NC2). Ce problème, bien connu, peut être résolu en temps polynomial ; le sujet permet la conception d'un algorithme réalisant une telle tâche. Plus précisément, l'algorithme décide si une formule est satisfiable et, dans ce cas, il exhibe une solution.

Le sujet commence par construire un algorithme sur des graphes généraux, avant de s'intéresser à la partie logique proprement dite. Les connaissances demandées sont les résultats du cours sur la logique booléenne, ainsi que les notions d'algorithmique standard : algorithmes récursifs, complexité, preuve de la correction d'une fonction. Quelques connaissances sur les graphes sont également utiles.

Cette partie est longue, mais globalement facile si l'on suit scrupuleusement les indications de l'énoncé. Les questions 15 à 18 proposent des exemples simples, tandis que les questions 19 à 24 introduisent les notations et les fonctions auxiliaires. Ensuite, les questions 25 à 28 préparent le fonctionnement de l'algorithme, qui est explicité dans les questions 29 à 33.

Indications

I. Problème sur les automates

3 Procéder par double inclusion ; pour le sens non trivial, penser à utiliser la division euclidienne.

4 Se rappeler qu'un automate ne sait pas compter.

5 L'automate modifié doit s'arrêter dès qu'il rencontre un b menant à un état final, et accepter les suites de a non suivies par un b. En particulier, toutes les transitions étiquetées par un b deviennent indésirables.

6 Caractériser $\Gamma(\mathrm{L}_3)$ en fonction de la longueur des mots qu'il contient et des lettres qui doivent nécessairement être présentes dans ces mots.

7 Encoder le fait que la lettre b ait déjà été ajoutée ou non au mot avec de nouveaux états.

8 À nouveau, caractériser la longueur des mots de $\Psi(\mathrm{L}_3)$ puis examiner l'inclusion réciproque.

9 Trouver un langage non rationnel sur l'alphabet $\Sigma' = \{a\}$, puis ajouter des b à la fin des mots qu'il contient. Pour prouver la non-rationalité, on essaiera de réutiliser des résultats obtenus sur Φ.

10 Procéder comme à la question 7, en se rappelant si b a été supprimé ou non. Attention, cette fois la suppression ne peut avoir lieu qu'au début.

II. Problème d'algorithmique, logique et programmation

12 Si vous ne connaissez pas de fonction itérant sur les éléments d'une liste, vous pouvez en écrire une récursive.

13 Montrer successivement que tous les sommets marqués sont des descendants (en exhibant un chemin entre un sommet marqué et r), puis que tous les descendants sont marqués (en trouvant un sommet après lequel un descendant est marqué).

14 Étudier l'ensemble des arcs accessibles depuis r qui sont visités.

15 Procéder par équivalences logiques successives, ou bien en utilisant une table de vérité.

17 Utiliser la définition de l'implication logique par la disjonction et la négation.

19 La fonction `barre` envoie $[0, p-1]$ sur $[p, 2p-1]$, et vice-versa.

21 Itérer une fonction qui ajoute les arcs correspondant à une unique clause.

22 Détailler les opérations effectuées pour chaque clause.

23 Caractériser exactement quand une clause doit être retirée, et itérer cette fonction sur la liste des clauses.

24 Trouver la complexité des opérations effectuées sur chaque clause.

25 Caractériser ce que représente un arc entre α et β, et utiliser la transitivité de l'implication.

26 Se rappeler que *transformer* ajoute 2 arcs pour chaque clause.

27 Utiliser la question 25 pour voir ce qu'impliquent ces deux chemins.

28 Raisonner par contraposition, en construisant un chemin de α à $\overline{\alpha}$.

29 On peut déduire trivialement une assignation pour F' à partir d'une assignation pour F. Dans l'autre sens il faut assigner tous les littéraux qui ont été retirés de F. Pour tous sauf un, il n'y a qu'une valeur possible.

30 Utiliser une récurrence sur le nombre de littéraux présents dans la formule, ainsi que la question précédente.

31 Distinguer les cas suivants :

- la formule ne comprend plus de littéraux ;
- la formule ne comprend que des littéraux impliquant leur complémentaire ;
- la formule comprend un littéral n'impliquant par son complémentaire.

32 Écrire une fonction auxiliaire retirant récursivement tous les descendants d'un littéral d'une formule, et l'utiliser pour implémenter l'algorithme défini à la question précédente.

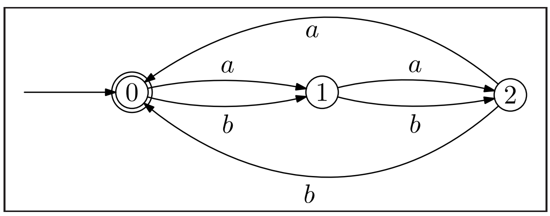
33 Trouver le nombre maximal d'appels récursifs et la complexité du corps de la fonction entre deux appels.

I. Problème sur les automates

Soit u un mot de longueur n appartenant à un langage L ; on note $u[i]$ sa i^{e} lettre (pour $1 \leqslant i \leqslant n$). On s'autorise également l'écriture a^* pour le langage $\{a\}^*$. La longueur du mot u est notée $|u|$.

> L'énoncé définissant l'étoile sur un ensemble de lettres, la notation a^* est a priori impropre. On peut au choix écrire systématiquement $\{a\}^*$ ou prévenir le correcteur de cette petite liberté de notation. Par ailleurs, comme on le rappellera plus bas, l'étoile s'étend également aux langages.

1 L'automate comprend 3 états, chacun encodant le résultat de la congruence de la longueur du mot modulo 3. L'état correspondant à un reste nul est acceptant (en plus d'être l'état initial).



> Il n'y a pas de convention universellement admise sur les états initiaux et acceptants : les premiers sont parfois notés par un cercle épais ou par une flèche entrante, tandis que les états acceptants le sont par un double cercle ou une flèche sortante.

2 Soit L_{3_1} le langage des mots de longueur exactement 3. Une expression régulière de ce langage est

$$\mathrm{L}_{3_1} = (a|b)(a|b)(a|b)$$

> L'alternative entre deux langages se note traditionnellement en utilisant les symboles $|$ ou $+$. Une autre notation pour L_{3_1} est donc
>
> $$\mathrm{L}_{3_1} = (a+b)(a+b)(a+b)$$

Étant donné un entier n, le langage $\mathrm{L}_{3_{k+1}}$ des mots de longueur $3(k+1)$ se déduit de L_{3_k} grâce à l'expression

$$\mathrm{L}_{3_{k+1}} = \mathrm{L}_{3_k}\mathrm{L}_{3_1}$$

Or L_{3_0}, le langage des mots de longueur 0, est exactement $\{\varepsilon\}$. Par définition de l'étoile de Kleene, on obtient donc

$$\boxed{\mathrm{L}_3 = {\mathrm{L}_{3_1}}^* = ((a|b)(a|b)(a|b))^*}$$

> Rappelons que, étant donné un langage X, $\mathrm{X}^0 = \{\varepsilon\}$, $\mathrm{X}^{n+1} = \mathrm{X}^n\mathrm{X} = \mathrm{X}\mathrm{X}^n$, et l'étoile de Kleene, ou fermeture, de ce langage est $\mathrm{X}^* = \bigcup\limits_{n=0}^{\infty} \mathrm{X}^n$.