Mines Informatique MP 2012 — Corrigé

Ce corrigé est proposé par Gautier Marti (ENS Lyon) ; il a été relu par Benjamin Monmege (ENS Cachan) et Guillaume Batog (Professeur agrégé).

Ce sujet est composé d'un exercice portant sur la théorie des langages et d'un problème consacré aux graphes.

L'exercice invite à montrer que quatre langages donnés sont rationnels. Ils sont définis comme l'ensemble des mots contenant une ou plusieurs occurrences d'un mot fixé. Cette exercice permet de vérifier la maîtrise des différentes techniques de preuve de rationalité d'un langage.

Le problème aborde, quant à lui, des problèmes de couplage sur les graphes bipartis équilibrés. Un couplage, comme son nom l'indique, est un ensemble de couples. Ces couples doivent vérifier certaines propriétés. Considérons par exemple n femmes et n hommes. Chaque femme a ou n'a pas une affinité pour un homme donné. De la même manière, un homme aime ou n'aime pas une femme donnée. En tant qu'agence matrimoniale, l'objectif est de former le maximum de couples en respectant les affinités de chacun.

- La première partie permet de bien s'approprier la définition de couplage en traitant un exemple et en écrivant des fonctions simples qui seront réutilisées par la suite.
- Le sujet aborde ensuite, en deuxième partie, la notion de couplage maximal, intuitivement un couplage qu'il est impossible de faire croître. Un algorithme permettant de le construire est donné dans le sujet. Le but de la partie est de l'implémenter, en commençant par l'appliquer à la main sur deux exemples. Les fonctions à écrire sont courtes et ne présentent pas de difficultés majeures.
- La troisième partie consiste en l'implémentation d'un algorithme naïf de construction d'un couplage de cardinal maximum.
- Enfin la partie 4, la plus difficile, présente l'algorithme hongrois, qui est la version polynomiale de l'algorithme en temps exponentiel de la partie précédente. Les questions sont diverses : des applications de l'algorithme sur des exemples, une question théorique et des programmes à écrire.

Ce sujet est de longueur raisonnable, il était possible de le traiter en majeure partie le jour du concours. L'aspect programmation y est bien représenté. Le travail sur des tableaux incite naturellement à la programmation impérative alors que les parcours de graphe s'effectuent plus aisément avec la récursivité. La principale difficulté du sujet consiste en l'assimilation des nombreuses définitions ; les applications sur les exemples ont pour but d'aider à les intégrer.

Les graphes ne figurent pas explicitement au programme mais apparaissent de manière récurrente dans les sujets des Mines. Bien que les notions utiles à la résolution du problème soient parfaitement définies dans le sujet, une familiarité avec ces dernières est un avantage certain le jour du concours.

Indications

Exercice

1 Écrire une expression rationnelle décrivant L_1.

3 Déterminer la forme de f. Se souvenir qu'un langage fini est rationnel.

4 Utiliser les propriétés de clôture des langages rationnels.

Problème

6 Remarquer que les sommets 1_A et 3_A jouent un rôle particulier.

7 Ne pas oublier de vérifier que les arêtes du couplage sont effectivement des arêtes du graphe.

10 Se rendre compte que le graphe est coupé en deux sous-graphes disjoints bipartis non équilibrés.

11 Créer deux tableaux qui contiennent le degré d'un sommet, un pour A et un pour B. Penser à utiliser 2 boucles `for` imbriquées pour considérer toutes les arêtes possibles du graphe.

15 Pour une arête donnée, déterminer récursivement les meilleurs couplages obtenus en choisissant cette arête ou non puis comparer leurs cardinaux.

17 Construire un couplage C' en considérant les arêtes d'une chaîne alternée augmentante relativement au couplage C qui ne sont pas dans C.

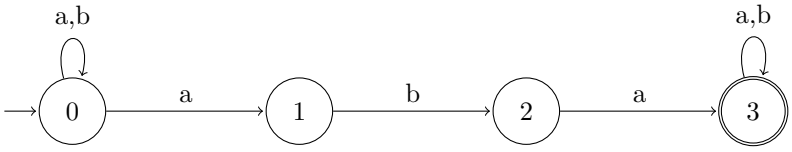
19 Utiliser une fonction récursive qui prend en paramètre un sommet de B, celle-ci saute un sommet sur deux (les sommets de A) de la chaîne alternée augmentante.

Exercice sur les langages rationnels

1 L'expression rationnelle $\Sigma^\star \cdot f \cdot \Sigma^\star$ décrit le langage L_1, donc

L_1 est rationnel.

> Rappelons que tout langage rationnel est décrit par une infinité d'expressions rationnelles équivalentes. L'expression rationnelle ci-dessus est la réponse la plus naturelle.
>
> On pourrait répondre aux questions de l'exercice, grâce au théorème de Kleene, par un automate fini A. Il faudrait alors prouver que le langage reconnu par l'automate fini est bien le langage L de la question en montrant la double inclusion $L(A) \subseteq L$ et $L \subseteq L(A)$. C'est une perte de temps le jour du concours. Par exemple, pour $f = aba$, le langage reconnu par l'automate suivant est L_1.



2 L'expression rationnelle $\Sigma^\star \cdot f \cdot \Sigma^\star \cdot f \cdot \Sigma^\star$ décrit le langage L_2. Par conséquent,

L_2 est rationnel.

3 Soit $C(f)$ l'ensemble des mots de préfixe et suffixe f non disjoints. Formellement,

$$C(f) = \{uvw \in \Sigma^\star \mid f = uv = vw \text{ avec } u, v, w \in \Sigma^\star \text{ et } v \neq \varepsilon\}$$

Un mot w appartient à L_3 si et seulement si w possède un facteur dans $C(f)$. Comme f est fini, $C(f)$ est fini. Par conséquent, $C(f)$ est rationnel puisque tout langage fini est rationnel car c'est une union finie des singletons des mots qui la composent. Le langage $\Sigma^\star$ est aussi rationnel. Puisque la famille des langages rationnels est fermée par concaténation, il vient que

$L_3 = \{x \cdot c \cdot y \mid x \in \Sigma^\star, y \in \Sigma^\star, c \in C(f)\}$ est rationnel.

4 Les mots qui contiennent exactement une occurrence du facteur f sont les mots qui contiennent au moins une occurrence de f mais qui ne contiennent pas au moins deux occurrences disjointes ou non de f, par conséquent $L_4 = L_1 \cap {}^{c}(L_2 \cup L_3)$. D'après les questions 1, 2 et 3 les langages L_1, L_2 et L_3 sont rationnels. Puisque la famille des langages rationnels est fermée par union, intersection et complémentaire,

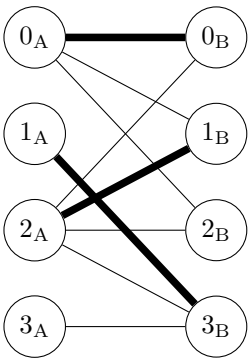
L_4 est un langage rationnel.

> La clôture par intersection et complémentaire des langages reconnaissables par automate fini se montre facilement et le théorème de Kleene permet de conclure pour les langages rationnels.

Problème d'algorithmique et programmation

I. Généralités

5 Le couplage $\{(0_A, 0_B), (1_A, 3_B), (2_A, 1_B)\}$ convient.



6 Il n'existe pas de couplage de cardinal 4. En effet, supposons par l'absurde qu'un tel couplage existe. Chaque sommet de A est alors couplé avec un unique sommet de B. Considérons le sommet 1_A. Le degré de ce sommet est 1, il est donc couplé avec 3_B son unique voisin. L'arête $(1_A, 3_B)$ appartient au couplage. Considérons à présent le sommet 3_A, son degré est 1, il est nécessairement couplé à son unique voisin 3_B. L'arête $(3_A, 3_B)$ appartient donc au couplage. Or, $(1_A, 3_B)$ et $(3_A, 3_B)$ sont incidentes ce qui contredit la définition de couplage.

Il n'existe pas de couplage de cardinal 4 dans G_0.

7 Pour vérifier que le tableau C représente un couplage dans G, il suffit de vérifier qu'une arête du couplage C est effectivement une arête du graphe G et que chaque sommet de B apparaît au plus une fois dans le couplage C à l'aide d'un tableau de booléens indicé par les sommets de B.

```
let verifie G C =
  let n = vect_length G in
  let resultat = ref true in
  let couple = make_vect n false in
  let sommet = ref 0 in
  while !sommet < n && !resultat do
    if C.(!sommet) <> -1 then
    begin
      if not G.(!sommet).(C.(!sommet)) || couple.(!sommet)
        then resultat := false;
      couple.(!sommet) <- true
    end;
    incr sommet
  done;
  !resultat;;
```

La fonction `verifie` est constituée d'une boucle exécutée au plus n fois et le corps de cette boucle contient un test et des instructions réalisés en temps constant.

La fonction `verifie` a une complexité en $O(n)$.