Automates pour rechercher

Épreuve pratique d'algorithmique et de programmation
Concours commun des écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/juillet 2007

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.
Lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main.***

1 Introduction

La recherche d'un mot dans un texte est un problème classique. Les editeurs de texte offrent tous une telle fonctionnalité. La difficulté est de pouvoir le faire de façon efficace sur de longs textes. En effet l'ergonomie d'un éditeur de texte tient notament au fait que la recherche d'un mot puisse se faire en un temps à peine perceptible par l'utilisateur. C'est pourquoi il est nécessaire d'employer des algorithmes efficaces, par exemple, en utilisant des automates.

2 Mots pseudo-aléatoires

Considérons la suite d'entiers (u_n) définie pour $n \geq 0$ par :

$$u_n = \begin{cases} \textit{votre } \mathtt{u_0} \textit{ (à reporter sur votre fiche)} & \text{si } n = 0 \\ 15\,091 \times u_{n-1} \mod 64\,007 & \text{si } n \geq 1 \end{cases}$$

Soit m un entier positif non-nul. La suite d'entiers $(v_{m,n})$ est définie pour $n \geq 0$ par :

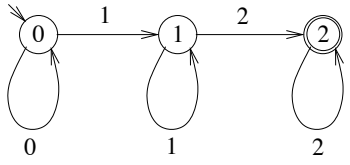
$$v_{m,n} = u_n \mod m$$

Question 1 ***a)*** *Quelle est la valeur de* $v_{7,\,1\,000}$ *?* ***b)*** *Quelle est la valeur de* $v_{10,\,10\,000}$ *?* ***c)*** *Quelle est la valeur de* $v_{3,\,100\,000}$ *?*

On appelle mot toute suite finie de lettres. Le mot de longueur nulle est noté ϵ. On note $x.y = x_1 \dots x_p\, y_1 \dots y_q$ la concatenation de deux mots $x = x_1 \dots x_p$ et $y = y_1 \dots y_q$.
On note $w_{n,k,l}$ le mot constitué des lettres $v_{n,k} \dots v_{n,k+l-1}$.

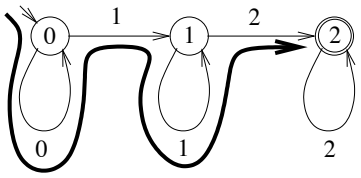
3 Automates

Notons E_n l'ensemble d'entiers naturels $\{0, \dots, n-1\}$. On appelle automate sur l'alphabet E_n, tout quadruplet $\mathcal{A} = (Q, I, F, T)$, où Q est un ensemble fini d'états, $I \subseteq Q$ est un ensemble d'états initiaux, $F \subseteq Q$ est un ensemble d'états finals et $T \subseteq Q \times E_n \times Q$ une relation de transition.
Les automates sont souvent représentés par des graphes dirigés dont les sommets sont les états de l'automate et dont les arcs représentent la relation de transition. Les états initiaux sont désignés par une petite flèche entrante et les états finals sont désignés par un double cercle. La figure ci-dessous donne un exemple d'automate :



Un chemin Γ de l'automate est une suite alternée états-lettres $\Gamma = q_0, e_1, q_1, e_2, q_2, \dots e_l, q_l$ telle que $\forall i \in \{0, \dots l-1\}, (q_i, e_{i+1}, q_{i+1}) \in T$.

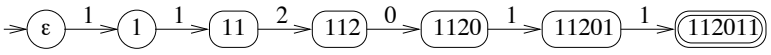
Le mot formé par le chemin Γ est le mot formé des lettres $e_1 e_2 \dots e_n$. Un chemin $\Gamma = q_0, e_1, q_1, e_2, q_2, \dots e_l, q_l$ est dit accepté par l'automate $\mathcal{A}$ si l'état de départ est initial, $q_0 \in I$ et l'état d'arrivée est final, $q_l \in F$. Un mot $w = e_1 \dots e_l$ est reconnu par l'automate $\mathcal{A}$, si et seulement si il existe un chemin $\Gamma = q_0, e_1, q_1, e_2, q_2, \dots e_l, q_l$ accepté par $\mathcal{A}$. Le langage de l'atomate $\mathcal{A}$ est l'ensemble des mots reconnus par $\mathcal{A}$.
Ainsi, l'automate de la figure ci-dessus reconnait le mot 0112, puisque que le chemin ci-dessous va de l'unique état initial vers l'unique état final.



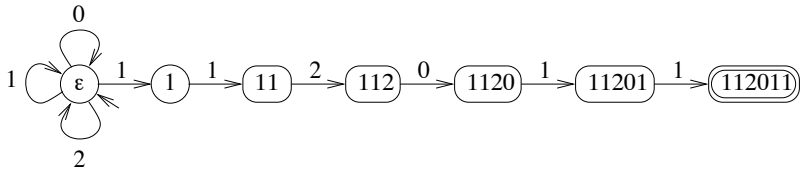
L'automate $\mathcal{A}$ est dit déterministe si et seulement si I est un singleton et T est une fonction de $Q \times E_n$ dans Q, c'est à dire que pour tout état $q \in Q$ et toute lettre $e \in E_n$ il existe au plus un $q' \in Q$ tel que (q, e, q') appartienne à T.

4 Recherche d'un mot dans un texte

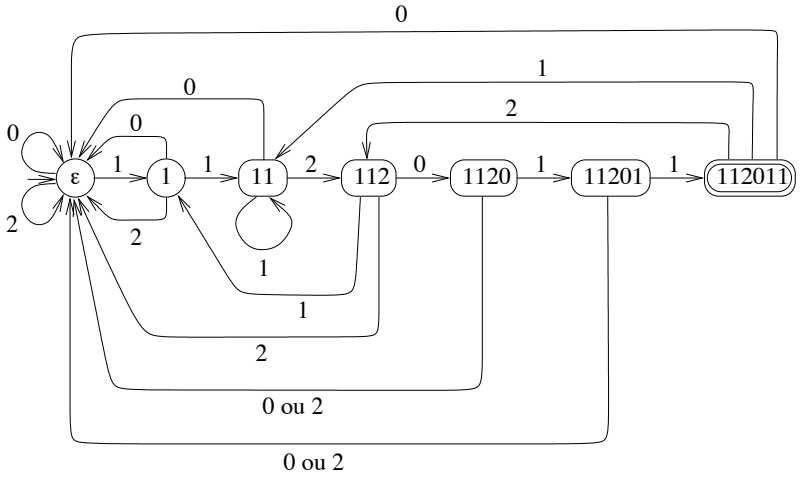
Nous souhaitons rechercher toutes les occurrences d'un mot $y = f_1 \dots f_k$ formé de lettres appartenant à E_n dans un mot $z = e_1 \dots e_l$ lui aussi formé de lettres appartenant à E_n. Il s'agit donc d'énumérer les mots $t = e_1 \dots e_m$, $m \leq l$, préfixes de z et terminant par le mot y. Cette recherche peut se faire efficacement sur des mots de grande taille, à l'aide d'un automate.
Prenons un exemple. Pour rechercher rapidement le mot $y = 112011$, on peut considérer l'automate $\mathcal{A}_y$ donné ci-dessous, dont le langage est le singleton $\{y\}$.



Considerons maintenant l'automate $\mathcal{A}'_y$ donné ci-dessous.



Il s'agit simplement de l'automate $\mathcal{A}_y$ auquel on a ajouté, pour chaque lettre e de l'alphabet E_n, une transition (ϵ, e, ϵ) bouclant sur l'état initial. Ce nouvel automate reconnait le langage des mots terminant par y. Malheureusement, il est non-déterministe, et ne peut être utilisé directement pour résoudre le problème de recherche. Il faut le transformer en automate déterministe.
La particularité de l'automate $\mathcal{A}'_y$ est qu'il est très facile de construire un automate déterministe reconnaissant le même langage. Pour cela, il suffit de prendre l'automate $\mathcal{A}_y$, et de le compléter, par ajout de transitions, en l'automate $\mathcal{A}''_z$, donné ci-dessous :



Les automates $\mathcal{A}'_y$ et $\mathcal{A}''_y$ reconnaissent le même langage, et $\mathcal{A}''_y$ est déterministe. C'est ce dernier qui peut être utilisé pour rechercher les occurrences de y.

Question 2 *Calculer le nombre d'occurrences du mot $y = 112011$ dans les mots $w_{3,0,l}$, pour les valeurs suivantes de l :* ***a)*** $l = 1\,000$ ***b)*** $l = 10\,000$ ***c)*** $l = 100\,000$

Nous allons maintenant nous intéresser à la construction de la relation de transition de l'automate $\mathcal{A}''_y$, pour un mot y quelconque**.** Pour cela, il est utile de définir le bord d'un mot z comme étant le plus grand mot $\delta(z)$ qui est à la fois préfixe et suffixe de z, mais aussi préfixe de y. Ainsi, le bord de $y = 112011$ est le mot $\delta(y) = 11$.
La relation de transition T de $\mathcal{A}''_y$ est la plus petite relation telle que :

$(\epsilon, e, e) \in T$, si e est la première lettre de y
$(\epsilon, e, \epsilon) \in T$, pour toute lettre e différente de la première lettre de y

Pour tout mot z, préfixe de longueur non nulle du mot y :

$(z, e, z.e) \in T$, si $z.e$ est préfixe du mot y
$(z, e, \delta(z.e)) \in T$, si $z.e$ n'est pas préfixe du mot y

Question à développer pendant l'oral : Détailler les structures de données employées pour construire l'automate $\mathcal{A}''_y$**.** Quelle est la complexité en temps et en mémoire de cette construction, en fonction du cardinal de l'alphabet et de la longueur du mot y ?

Question 3 ***a)*** *Quel est le nombre d'occurrences du mot $w_{5,0,10}$ dans le mot $w_{5,10,1\,000}$?* ***b)*** *Quel est le nombre d'occurrences du mot $w_{7,0,30}$ dans le mot $w_{5,30,10\,000}$?* ***c)*** *Quel est le nombre d'occurrences du mot $w_{10,0,100}$ dans le mot $w_{5,100,100\,000}$?*

On modifiera l'algorithme pour calculer le plus grand prefixe du mot y ayant au moins une occurrence dans un texte donné**.**

Question 4 ***a)*** *Quelle est la longueur du plus grand préfixe du mot $w_{5,0,100}$ ayant au moins une occurrence dans le mot $w_{5,100,10\,000}$?* ***b)*** *Quelle est la longueur du plus grand préfixe du mot $w_{7,0,300}$ ayant au moins une occurrence dans le mot $w_{7,300,30\,000}$?* ***c)*** *Quelle est la longueur du plus grand préfixe du mot $w_{10,0,1\,000}$ ayant au moins une occurrence dans le mot $w_{10,1\,000,100\,000}$?*