C39212

Ecole Normale Supérieure de Cachan
61 avenue du président Wilson
94230 CACHAN

Concours d'admission en **3ème année**
Informatique

Session 2009

INFORMATIQUE 2

Durée : 5 **heures**

« Aucun document n'est autorisé »

« Aucun dictionnaire n'est autorisé »

« L'usage de toute calculatrice est interdit »

Épreuve d'Informatique II

Dans toute l'épreuve, Σ désigne un alphabet fini non-vide. On considère également quatre symboles distincts $\triangleright$, $\triangleleft$, $\triangledown$ et $\vartriangle$ n'appartenant pas à Σ, appelés *délimiteurs*.

Les suites (finies ou infinies) sont notées $r = (r_i)_{0 \leq i < n}$, où $n \in \mathbb{N} \cup \{+\infty\}$ est la longueur de la suite r (notée $|r|$). Pour toute suite non-vide r, on définit $\mathsf{first}(r) = r_0$ et, si r est finie, $\mathsf{last}(r) = r_{|r|-1}$. Étant donnés une suite r et un entier $p < |r|$, on définit $r[p] = (r_i)_{0 \leq i < p}$ le p-ième préfixe de r.

1 Automates finis bi-directionnels

Un *mot fini* w sur l'alphabet Σ est une suite finie de lettres de Σ. L'ensemble des mots de longueur n sur Σ est noté Σ^n, et celui des mots finis sur Σ est noté Σ^*. Étant donnés deux mots finis u et v, la *concaténation* de u et v, notée $u \cdot v$, est le mot $w = (w_i)_{0 \leq i < |u|+|v|}$ tel que $w_i = u_i$ si $i < |u|$ et $w_i = v_{i-|u|}$ si $|u| \leq i < |u| + |v|$.

Étant donné un mot fini w sur l'alphabet Σ, on note $\triangleright w \triangleleft$ le mot sur $\Sigma \cup \{\triangleright, \triangleleft\}$ de taille $|w| + 2$ obtenu par concaténation des mots $\triangleright$, w et $\triangleleft$. On appelle *mot délimité* un tel mot.

Un *automate fini bi-directionnel* (que l'on appellera 2NFA dans la suite) sur l'alphabet Σ est un 4-uplet $\mathcal{A} = \langle Q, Q_0, Q_f, \delta \rangle$ où Q est un ensemble fini d'états, $Q_0 \subseteq Q$ est l'ensemble des états initiaux, $Q_f \subseteq Q$ est l'ensemble des états finals, et $\delta \colon Q \times (\Sigma \cup \{\triangleright, \triangleleft\}) \to \mathcal{P}(Q \times \{-1, 0, +1\})$ est la fonction de transition de l'automate, satisfaisant la condition suivante :

$$\forall q \in Q. \quad \delta(q, \triangleright) \subseteq Q \times \{+1\} \qquad \text{et} \qquad \delta(q, \triangleleft) \subseteq Q \times \{-1\}.$$

Cette condition assure que les lettres $\triangleright$ et $\triangleleft$ jouent bien leur rôle de délimiteurs du mot d'entrée. La *taille* de $\mathcal{A}$ est le nombre d'états de $\mathcal{A}$.

Un 2NFA peut être vu comme une machine de Turing « en lecture seule ». Formellement, étant donnés un mot w et un 2NFA $\mathcal{A}$, une *configuration de $\mathcal{A}$ sur w* est un couple (q, p) où $q \in Q$ et $0 \leq p < |w|$. Une *exécution de $\mathcal{A}$ sur w* est une suite (finie ou infinie) $r = ((q_i, p_i))_{0 \leq i < n}$ de configurations de $\mathcal{A}$ sur w telle que pour tout $i < |r| - 1$,

$$\exists (q, m) \in \delta(q_i, w_{p_i}) \text{ tel que } q_{i+1} = q \text{ et } p_{i+1} = p_i + m.$$

Une exécution r de $\mathcal{A}$ sur w est *initiale* si sa première configuration (q_0, p_0) est telle que $q_0 \in Q_0$ et $p_0 = 0$. Une exécution initiale r de $\mathcal{A}$ sur w est *acceptante* si elle contient une configuration (q, p) telle que $q \in Q_f$. Dans toute la suite de cette section, on ne considère que des exécutions sur des mots délimités. On note $\mathcal{L}(\mathcal{A})$ l'ensemble des mots $w \in \Sigma^*$ tels que $\triangleright w \triangleleft$ admet une exécution acceptante sur $\mathcal{A}$.

Un *automate fini mono-directionnel* (appelé NFA dans la suite) est un 2NFA dont la fonction de transitions est à valeurs dans $\mathcal{P}(Q \times \{+1\})$. Un 2NFA *déterministe* (appelé 2DFA dans la suite) est un 2NFA tel que $|Q_0| \leq 1$ et $|\delta(q, \sigma)| \leq 1$ pour tout $q \in Q$ et tout $\sigma \in \Sigma \cup \{\triangleright, \triangleleft\}$. Un DFA est un NFA déterministe.

Question 1.

Montrer que l'on peut, sans perte de généralité et sans modifier la taille et la nature (déterministe ou non-déterministe, mono-directionnel ou bi-directionnel) de l'automate, supposer que :
- Q_f contient un et un seul état, que l'on notera q_f ;
- un mot délimité $\triangleright w \triangleleft$ n'est accepté que s'il admet une exécution finie se terminant dans la configuration $(q_f, |w| + 1)$ (c'est-à-dire sur la dernière lettre $\triangleleft$ de $\triangleright w \triangleleft$).

Dans toute la suite de cette section, on suppose que les automates ont un et un seul état acceptant q_f, et que, pour tout automate $\mathcal{A} = \langle Q, Q_0, \{q_f\}, \delta \rangle$ et tout mot w, une exécution initiale de $\mathcal{A}$ sur $\triangleright w \triangleleft$ est acceptante si, et seulement si, elle est finie et se termine dans la configuration $(q_f, |w| + 1)$.

Question 2.

Soit $\mathcal{A}$ un 2DFA.

2.1 Montrer que l'ensemble des exécutions finies (pas nécessairement initiales) se terminant dans l'unique configuration acceptante peut se représenter sous la forme d'un arbre fini dont la racine est la configuration acceptante.

2.2 Montrer qu'il existe un 2DFA $\mathcal{A}'$ acceptant le même langage que $\mathcal{A}$ et ne possédant aucune exécution infinie. Donner une approximation de la taille de $\mathcal{A}'$ en fonction de celle de $\mathcal{A}$.

2.3 Montrer que pour tous 2DFA $\mathcal{A}_1$ et $\mathcal{A}_2$, il existe trois 2DFA $\mathcal{C}$, $\mathcal{U}$ et $\mathcal{I}$, de taille polynomiale en $|\mathcal{A}_1|$ et $|\mathcal{A}_2|$, tels que :

$$\mathcal{L}(\mathcal{C}) = \Sigma^* \setminus \mathcal{L}(\mathcal{A}_1), \qquad \mathcal{L}(\mathcal{U}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2), \qquad \mathcal{L}(\mathcal{I}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2).$$

Question 3.

3.1 Montrer que pour tous 2NFA $\mathcal{A}_1$ et $\mathcal{A}_2$, il existe deux 2NFA $\mathcal{U}$ et $\mathcal{I}$, de taille polynomiale en $|\mathcal{A}_1|$ et $|\mathcal{A}_2|$, tels que :

$$\mathcal{L}(\mathcal{U}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2), \qquad \mathcal{L}(\mathcal{I}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2).$$

3.2 Soient $\mathcal{A} = \langle Q, Q_0, Q_f, \delta \rangle$ un 2NFA sur Σ et $w \in \Sigma^*$. Montrer que $\mathcal{A}$ ***n'***accepte ***pas*** le mot $x = \triangleright w \triangleleft$ si, et seulement si, il existe une suite $(T_i)_{0 \leq i < |w|+2}$ d'ensembles d'états de $\mathcal{A}$ telle que
- $Q_0 \subseteq T_0$;
- $T_{|w|+1} \cap Q_f = \varnothing$;

– pour tout $0 \leq i < |w| + 2$, pour tout $q \in T_i$, pour tout $(q', k) \in \delta(q, x_i)$, on a $q' \in T_{i+k}$.

3.3 Montrer que pour tout 2NFA $\mathcal{A}$, il existe un NFA $\mathcal{C}$ de taille exponentielle en la taille de $\mathcal{A}$ tel que $\mathcal{L}(\mathcal{C}) = \Sigma^* \setminus \mathcal{L}(\mathcal{A})$. On pourra considérer un automate dont les états sont des paires d'ensembles d'états $(T, U) \in (2^Q)^2$.

Question 4.

Nous montrons maintenant que pour tout 2NFA $\mathcal{A}$, il existe un DFA $\mathcal{A}'$ de taille exponentielle acceptant le même langage. L'idée est de mémoriser, au cours de l'exécution, les informations nécessaires sur les chemins « en arrière ». Plus précisément, après avoir lu les p premières lettre d'un mot délimité x, on mémorise les couples d'états (s, s') tels qu'il existe une exécution de la configuration (s, p) à la configuration (s', p) ne parcourrant que le préfixe $x[p]$.

Soit $\mathcal{A}$ un 2NFA. Pour tout mot délimité $\triangleright w \triangleleft = x = (x_i)_{0 \leq i < |w|+2}$, on définit les suites $l = (l_i)_{0 \leq i < |x|}$ et $m = (m_i)_{0 \leq i < |x|}$ d'ensemble de paires d'états de façon récursive, comme suit :

$$
\begin{array}{lcl}
(s, s') \in l_i & \Leftrightarrow & (s', 0) \in \delta(s, x_0) \text{ ou } i > 0 \text{ et il existe deux états } t \text{ et } t' \text{ tels que} \\
 & & (t, t') \in m_{i-1} \text{ et } (t, -1) \in \delta(s, x_i) \text{ et } (t, +1) \in \delta(t', x_{i-1}) \text{ ;} \\
(s, s') \in m_i & \Leftrightarrow & \exists (s_j)_{0 \leq j < k} \text{ telle que, pour tout } j < k-1, (s_j, s_{j+1}) \in l_i \text{ et} \\
 & & s = s_0,\ s' = s_{k-1}.
\end{array}
$$

Le *mot étiqueté* correspondant au mot délimité x est le mot $(\langle x_i, m_i \rangle)_{0 \leq i < |x|}$. On note $\mathcal{L}^e(\mathcal{A})$ le *langage étiqueté* de $\mathcal{A}$, défini par

$$\mathcal{L}^e(\mathcal{A}) = \{(\langle x_i, m_i \rangle)_{1 \leq i < |x|-1} \mid \exists w \in \mathcal{L}(\mathcal{A}).\ x = \triangleright w \triangleleft\}.$$

C'est un langage sur l'alphabet $\Sigma \times Q^2$.

4.1 Montrer qu'il existe un DFA $\mathcal{B}$ sur l'alphabet $\Sigma \times Q^2$ tel que
– $|\mathcal{B}| = 2^{|\mathcal{A}|}$,
– $\mathcal{L}(\mathcal{B}) = \mathcal{L}^e(\mathcal{A})$.

On pourra adapter la construction usuelle permettant de déterminiser un NFA en prenant en compte l'information supplémentaire contenue dans le mot étiqueté.

4.2 Montrer qu'il existe un DFA $\mathcal{D}$ sur l'alphabet Σ tel que
– $|\mathcal{D}| = 2^{|\mathcal{A}|} \cdot 2^{2|\mathcal{A}|^2}$,
– $\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{A})$.

4.3 En déduire que pour tous 2NFA $\mathcal{A}_1$ et $\mathcal{A}_2$, il existe un DFA $\mathcal{C}$ de taille exponentielle en $|\mathcal{A}_1|$ et $|\mathcal{A}_2|$ reconnaissant la concaténation des langages de $\mathcal{A}_1$ et $\mathcal{A}_2$:

$$\mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2) \stackrel{\text{def}}{=} \{w_1 \cdot w_2 \mid w_1 \in \mathcal{L}(\mathcal{A}_1),\ w_2 \in \mathcal{L}(\mathcal{A}_2)\}.$$

Question 5.

Soit n un entier strictement positif. On note $L_n = \{u \cdot \Sigma^* \cdot u \mid u \in \Sigma^n\}$.

5.1 Montrer qu'il existe un 2DFA $\mathcal{A}$ de taille polynomiale en n et $|\Sigma|$, tel que $\mathcal{L}(\mathcal{A}) = L_n$. Donner une approximation de sa taille en fonction de n et $|\Sigma|$.

5.2 Montrer qu'il existe un NFA $\mathcal{B}$ tel que $\mathcal{L}(\mathcal{B}) = L_n$.

5.3 Montrer que tout NFA $\mathcal{B}$ tel que $\mathcal{L}(\mathcal{B}) = L_n$ a au moins $|\Sigma|^n$ états.

2 Automates finis quadri-directionnels

Une *image sur l'alphabet* Σ est une matrice d'élements de Σ. Le but de cette section est d'étudier les langages d'images reconnus par des automates quadri-directionnels.

Formellement, une *image finie de dimension* $l \times h$ est une matrice $m = (m_{i,j})_{0 \leq i < l, 0 \leq j < h}$ d'élements de Σ. On note Σ^{**} l'ensemble des images finies sur Σ. On définit les deux opérations de concaténation sur ces objets :

– la *concaténation verticale* : étant données deux images m et m' de dimensions respectives $l \times h$ et $l' \times h'$, la concaténation verticale de m et m' est définie si, et seulement si, $l = l'$; dans ce cas, elle est notée $m \ominus m'$, est de dimension $l \times (h + h')$, et est définie par

$$(m \ominus m')_{i,j} = \begin{cases} m_{i,j} & \text{si } j < h \\ m'_{i,j-h} & \text{si } j \geq h \end{cases}$$

– la *concaténation horizontale*, notée $\oplus$, est définie de manière similaire.

Comme dans le cas des mots uni-dimensionnels, on étend l'alphabet Σ avec quatre lettres $\triangleright$, $\triangleleft$, $\triangledown$ et $\vartriangle$, permettant à l'automate de détecter les limites de l'image. Étant donnée une image m de dimension $l \times h$, l'*image délimitée* correspondante, que l'on note $\boxed{m}$, est l'image $t = (t_{i,j})_{0 \leq i < l+2, 0 \leq j < h+2}$ telle que

– $t_{0,j} = \triangleright$ pour tout $0 \leq j < h + 2$;
– $t_{l+1,j} = \triangleleft$ pour tout $0 \leq j < h + 2$;
– $t_{i,0} = \triangledown$ pour tout $1 \leq i < l + 1$;
– $t_{i,h+1} = \vartriangle$ pour tout $1 \leq i < l + 1$;
– $t_{i,j} = m_{i-1,j-1}$ pour tous $1 \leq i < l + 1$ et $1 \leq j < h + 1$.

Un *automate d'image fini quadri-directionnel* (que l'on appellera 4NFA dans la suite) sur l'alphabet Σ est un 4-uplet $\mathcal{A} = \langle Q, Q_0, Q_f, \delta \rangle$ où Q est un ensemble fini d'états, Q_0 et Q_f sont les ensembles d'états initiaux et finals, respectivement, et

$$\delta \colon Q \times (\Sigma \cup \{\triangleright, \triangleleft, \triangledown, \vartriangle\}) \to \mathcal{P}(Q \times \{(-1, 0), (+1, 0), (0, 0), (0, -1), (0, +1)\})$$

est la fonction de transition de l'automate, satisfaisant les conditions suivantes :

$$\delta(q, \triangleright) \subseteq Q \times \{(+1, 0)\} \qquad \delta(q, \triangleleft) \subseteq Q \times \{(-1, 0)\}$$
$$\delta(q, \triangledown) \subseteq Q \times \{(0, +1)\} \qquad \delta(q, \vartriangle) \subseteq Q \times \{(0, -1)\}.$$

La *taille* de $\mathcal{A}$ est le nombre d'états de $\mathcal{A}$.

Le comportement d'un tel automate $\mathcal{A}$ sur une image d'entrée est similaire au comportement des automates finis sur les mots finis : une *configuration de* $\mathcal{A}$ *sur une image finie* m *de dimensions* $l \times h$ est un couple (q, p) où $q \in Q$ et $p \in [\![0, l-1]\!] \times [\![0, h-1]\!]$. Une *exécution de* $\mathcal{A}$ *sur* m est une suite (finie ou infinie) $r = ((q_i, p_i))_{0 \leq i < |r|}$ de configurations de $\mathcal{A}$ sur m telle que pour tout $i < |r| - 1$,

$$\exists (q, d) \in \delta(q_i, m_{p_i}) \text{ tel que } q_{i+1} = q \text{ et } p_{i+1} = p_i + d.$$

Une exécution r de $\mathcal{A}$ sur m est *initiale* si sa première configuration (q_0, p_0) est telle que $q_0 \in Q_0$ et $p_0 = (0, 0)$. Une exécution initiale r de $\mathcal{A}$ sur une image finie m est *acceptante* si elle contient une configuration de la forme (q_f, p) avec $q_f \in Q_f$. Comme dans le cas mono-dimensionnel, on ne considèrera ici que des exécutions sur des images finies délimitées. On note $\mathcal{L}(\mathcal{A})$ l'ensemble des images $m \in \Sigma^{**}$ telles que $\boxed{m}$ admet une exécution acceptante sur $\mathcal{A}$.

Un 4NFA *déterministe* (appelé 4DFA dans la suite) est un 4NFA tel que $|Q_0| \leq 1$ et $|\delta(q, \sigma)| \leq 1$ pour tout $q \in Q$ et tout $\sigma \in \Sigma \cup \{\triangleright, \triangleleft, \triangledown, \vartriangle\}$.

Question 6.

6.1 Montrer qu'il existe un 4DFA $\mathcal{A}$ tel que $\mathcal{L}(\mathcal{A})$ est l'ensemble des images carrées (c'est-à-dire des images de dimension $l \times l$, pour tout $l \geq 0$).

6.2 Montrer qu'il existe un 4DFA $\mathcal{B}$ tel que $\mathcal{L}(\mathcal{B})$ est l'ensemble des images dont le bord supérieur est identique au bord inférieur (c'est-à-dire $m_{i,0} = m_{i,h-1}$ pour tout $0 \leq i < l$).

Question 7.

Montrer que l'ensemble des langages reconnus par un 4DFA est clos par union, intersection et complémentaire.

Question 8.

Pour les mêmes raisons qu'à la question 3.1, l'ensemble des langages reconnus par un 4NFA est clos par union et intersection. Nous montrons ici qu'il n'est pas clos par complémentaire.

8.1 On se place, jusqu'à la fin de cette question, dans le cas où $\Sigma = \{1\}$. Montrer que, si un 2NFA de taille p reconnait un langage fini non-vide de Σ^*, alors le mot le plus long de ce langage est de longueur au plus p.

8.2 Soit $\mathcal{A}$ un 4NFA de taille k, et $S = \mathcal{L}(\mathcal{A}) \subseteq \Sigma^{**}$ le langage qu'il reconnaît. Soit h un entier strictement positif. Montrer qu'il existe un 2NFA $\mathcal{B}_h$ de taille $k \cdot h$ tel que

$$\mathcal{L}(\mathcal{B}_h) = \{1^l \mid S \text{ contient un mot de dimension } l \times h\} \subseteq \Sigma^*.$$

8.3 Soit T le langage de Σ^{**} suivant :

$$T = \{(1)_{0 \leq i < l, 0 \leq j < h} \mid h > 0 \text{ et il existe deux entiers } m \text{ et } n \text{ tels que } 0 \leq m \leq n \text{ et } l = n \cdot h + m\}.$$

Montrer que T est reconnu par un 4NFA.

8.4 Montrer que $\Sigma^{**} \smallsetminus T$ n'est reconnu par aucun 4NFA, et que T n'est reconnu par aucun 4DFA.

Question 9.

Nous montrons maintenant que les langages reconnus par 4NFA ne sont pas clos par concaténation. Étant donnée une image m de dimension $l \times h$, une *sous-image* de m est une image n de dimension $l' \times h'$, avec $l' \leq l$ et $h' \leq h$, pour laquelle il existe $l_0 \leq l - l'$ et $h_0 \leq h - h'$ tels que

$$n_{i,j} = m_{i+l_0,j+h_0} \qquad \text{pour tout } 0 \leq i < l' \text{ et } 0 \leq j < h'.$$

Soit $\mathcal{A}$ un 4NFA, et m une image de dimension $l \times h$. On note $\mathsf{lignes}(m)$ l'ensemble des lignes de m :

$$\mathsf{lignes}(m) = \{(w_i)_{0 \leq i < l} \mid \exists 0 \leq j < h.\ \forall 0 \leq i < l.\ w_i = m_{i,j}\}$$

On note $\mathsf{bord}(m)$ les deux lignes extrémales de m, correspondant à $j = 0$ et $j = h - 1$. Soit n une sous-image de $\boxed{m}$ de dimension $(l + 2) \times h'$. On note $B_{\mathcal{A},n} = \{(q, p) \mid q \in Q,\ p \in [\![0, l + 2]\!] \times \{0, h' - 1\}\}$ l'ensemble des configurations de $\mathcal{A}$ sur $\mathsf{bord}(n)$, et on définit la fonction $\mathsf{succ}\colon B \to 2^B$ par

$$(q', p') \in \mathsf{succ}(q, p) \quad \Leftrightarrow \quad \text{il existe une exécution de } \mathcal{A} \text{ sur } n \text{ de } (q, p) \text{ à } (q', p').$$

9.1 On se place sur l'alphabet $\Sigma = \{0, 1\}$. Montrer que, pour tout 4NFA $\mathcal{A}$, il existe deux images n et n' de même dimension et telles que
- ces deux images définissent les mêmes fonctions succ ;
- $\mathsf{lignes}(n) \neq \mathsf{lignes}(n')$.

9.2 En utilisant par exemple le langage de la question 6.2, montrer que les 4NFA ne sont pas clos par concaténation verticale.

3 Reconnaissance par pavages

Un *pavé* sur un alphabet Γ est une image de dimension 2×2 sur Γ. Étant donnée une image m, on note $\mathcal{T}(m)$ l'ensemble des sous-images de dimension 2×2 de m. Étant donné un ensemble $\mathcal{T}$ de pavés, une image m est *engendrée* par $\mathcal{T}$ si $\mathcal{T}(m) \subseteq \mathcal{T}$. Un *système de pavage* sur l'alphabet Σ est un 3-uplet $\mathcal{P} = \langle \Gamma, \pi, \mathcal{T} \rangle$ tel que Γ est un alphabet fini n'intersectant pas $\{\triangleright, \triangleleft, \triangledown, \vartriangle\}$, la fonction π projette chaque lettre de Γ sur une lettre de Σ, et $\mathcal{T}$ est un ensemble de pavés sur $\Gamma \cup \{\triangleright, \triangleleft, \triangledown, \vartriangle\}$. On définit

$$\mathcal{L}(\mathcal{P}) = \{\pi(m) \in \Sigma^{**} \mid m \in \Gamma^{**} \text{ et } \mathcal{T}(\boxed{m}) \subseteq \mathcal{T}\}$$

où $\pi(m)$ est l'image $(\pi(m_{i,j}))_{i,j}$. Un langage L est *reconnu par pavage* s'il existe un système de pavage $\mathcal{P}$ tel que $L = \mathcal{L}(\mathcal{P})$.

Question 10.

On se restreint dans un premier temps au cas particulier des images de hauteur 1 (que l'on voit comme des mots). Montrer qu'un langage de Σ^* est reconnu par un NFA si, et seulement si, ce langage (vu comme un langage d'images de hauteur 1) est reconnu par pavage.

Question 11.

Montrer que l'ensemble des langages reconnus par pavages est clos par concaténation (verticale et horizontale), union et intersection.

Question 12.

Soit Σ un alphabet fini. On considère le langage

$$L = \{m \in \Sigma^{**} \mid \text{il existe une image carrée } s \text{ sur } \Sigma \text{ telle que } m = (s \ominus s)\}.$$

12.1 Montrer que, si $\Sigma = \{1\}$, alors L est reconnaissable par pavage.

12.2 Jusqu'à la fin de cette question, on prend $\Sigma = \{0, 1\}$. Montrer que L n'est pas reconnaissable par pavage.

12.3 Montrer que le complémentaire de L dans Σ^{**} est reconnaissable par pavage.

– Fin de l'épreuve –