C30212

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

_____

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

Session 2010

_____

INFORMATIQUE 2

_____

Durée : 5 **heures**

_____

« Aucun document n'est autorisé »

« Aucun dictionnaire n'est autorisé »

« L'usage de toute calculatrice est interdit »

Informatique II

La notation tiendra compte de la rigueur des raisonnements et de la clarté des explications.
Chaque affirmation devra être soigneusement justifiée.
Chaque question peut être traitée en admettant le résultat des questions précédentes.

Les algorithmes devront être écrits en pseudo-langage, en utilisant les structures de contrôle usuelles (Si... alors... sinon... FinSi, TantQue... FinTantQue, etc.).

Notations et terminologie

Relations. Dans tout le problème, X désigne un ensemble. Une *relation binaire* R sur X est un sous-ensemble de $X \times X$. Toutes les relations considérées dans le problème étant binaires, on écrira *relation* au lieu de relation binaire. On écrira $x\ R\ y$ au lieu de $(x, y) \in R$. On rappelle qu'une relation R sur X est *réflexive* si pour tout $x \in X$, on a $x\ R\ x$; *transitive* si pour tous $x, y, z \in X$ tels que $x\ R\ y$ et $y\ R\ z$, on a $x\ R\ z$; *antisymétrique* si pour tous $x, y \in X$ tels que $x\ R\ y$ et $y\ R\ x$, on a $x = y$; *symétrique* si pour tous $x, y \in X$ tels que $x\ R\ y$, on a $y\ R\ x$. Une relation d'*ordre* (ou simplement un *ordre*) est une relation réflexive, transitive et antisymétrique. Un ordre R est *total* si pour tous $x, y \in X$, on a $x\ R\ y$ ou $y\ R\ x$. Une relation d'*équivalence* est une relation réflexive, transitive et symétrique.

Mots, sous-mots. Soit A un alphabet fini. On désigne par A^* l'ensemble des mots finis sur A. On note $|x|$ la longueur du mot x, c'est-à-dire son nombre de lettres, et on note ε le mot vide, de longueur nulle.

On dit qu'un mot x est *sous-mot* d'un mot y si soit $x = \varepsilon$, soit on peut écrire $x = a_1 \cdots a_m$, avec $m \geqslant 1$, $a_i \in A$, et $y = y_0 a_1 y_1 \cdots a_m y_m$, avec $y_0, \ldots, y_m \in A^*$. Par exemple, *abab* est sous-mot de *cca*$\underline{a}$*cabb*$\underline{d}$*a*$\underline{b}$. On note $\sqsubseteq$ la relation sur A^* définie par $x \sqsubseteq y$ si x est sous-mot de y.

I Mots et sous-mots

Pour l'écriture des algorithmes, on suppose les mots mémorisés dans des tableaux. Le mot $a_1 \cdots a_m$ ($a_i \in A$) sera ainsi représenté par un tableau `a` et une variable `m` indiquant son nombre de lettres : `a[1]` contient la lettre a_1, `a[2]` la lettre a_2, etc. (pour simplifier, on peut indicer les tableaux à partir de 1).

I.1 Écrire un algorithme prenant en entrée deux mots x et y et leurs longueurs respectives m et n, qui retourne Vrai si x est un sous-mot de y et Faux sinon. On demande un algorithme de complexité $O(n)$, évaluée en nombre de comparaisons de lettres. Justifier la correction et la complexité de l'algorithme.

Soient $x, y \in A^*$, avec $x = a_1 \cdots a_m$, où $a_i \in A$ pour $i = 1, \ldots, m$ (par convention, $m = 0$ si $x = \varepsilon$). On note $C(x, y)$ le nombre d'éléments de l'ensemble $\Big\{(y_0, \ldots, y_m) \in (A^*)^{m+1} \mid y = y_0 a_1 y_1 \cdots a_m y_m\Big\}$. Par exemple, si $x = ab$ et $y = ababc$, cet ensemble est $\big\{(\varepsilon, \varepsilon, abc), (\varepsilon, ba, c), (ab, \varepsilon, c)\big\}$, donc $C(x, y) = 3$.

I.2 Pour une lettre $a \in A$ et deux entiers naturels m, n, calculer $C(a^m, a^n)$.

I.3 Pour $y \in A^*$, calculer $C(\varepsilon, y)$. Pour $x, y \in A^*$ avec $|x| \geqslant |y|$, calculer $C(x, y)$.

I.4 Montrer que $C(x, y'y'') = \displaystyle\sum_{\substack{x', x'' \in A^* \\ x = x'x''}} C(x', y')C(x'', y'')$.

I.5 Pour $x, y \in A^*$ et $a, b \in A$, exprimer $C(xa, yb)$ en fonction de $C(x, y)$ et $C(xa, y)$, en distinguant les cas $a = b$ et $a \neq b$.

I.6 En utilisant la question précédente, écrire un algorithme calculant $C(x, y)$, étant donnés les mots x et y ainsi que leurs longueurs respectives m et n. On demande que sa complexité, évaluée en nombre d'opérations arithmétiques effectuées, soit $O(mn)$.

I.7 Pour $L \subseteq A^*$, on définit $\mathrm{Sous}(L) = \{x \in A^* \mid \exists y \in L,\ x \sqsubseteq y\}$. Montrer que si L est un langage rationnel, alors $\mathrm{Sous}(L)$ est également rationnel. Décrire comment construire une expression rationnelle pour $\mathrm{Sous}(L)$ à partir d'une expression rationnelle pour L. Décrire également comment construire un automate fini reconnaissant $\mathrm{Sous}(L)$ à partir d'un automate fini reconnaissant L.

II Beaux ordres

Étant donnée une relation $\preccurlyeq$, on définit la relation $\sim$ par $x \sim y$ si et seulement si $x \preccurlyeq y$ et $y \preccurlyeq x$. Une relation $\preccurlyeq$ est un *quasi-ordre* si elle est réflexive et transitive.

II.1 Montrer que si $\preccurlyeq$ est un quasi-ordre, la relation $\sim$ est une relation d'équivalence. On note $[x]_\sim = \{y \in X \mid x \sim y\}$ la classe d'équivalence d'un élément x pour $\sim$. Montrer que l'on peut définir une relation $\leqslant$ sur les classes d'équivalence de $\sim$ par :

$$[x]_\sim \leqslant [y]_\sim \text{ si et seulement si } x \preccurlyeq y,$$

et montrer que $\leqslant$ est une relation d'ordre.

Pour un quasi-ordre $\preccurlyeq$ sur X, on note $\not\preccurlyeq$ la relation telle que $x \not\preccurlyeq y$ si et seulement si on n'a pas $x \preccurlyeq y$. On écrit $x \prec y$ si $x \preccurlyeq y$ et $y \not\preccurlyeq x$. Pour $P \subseteq X$, un *élément minimal* de P est un élément $p \in P$ tel que l'ensemble $\{x \in P \mid x \prec p\}$ est vide.

II.2 Montrer que si $\preccurlyeq$ est un ordre, p est un élément minimal de P si et seulement si pour tout $x \in P$ tel que $x \preccurlyeq p$, on a $x = p$. Est-ce toujours vrai si l'on suppose seulement que $\preccurlyeq$ est un quasi-ordre ?

À partir de maintenant et jusqu'à la fin du problème, $\preccurlyeq$ désigne une relation d'ordre. Un ordre $\preccurlyeq$ sur X est un *bel* ordre si pour toute suite infinie $(x_n)_{n\in\mathbb{N}}$ d'éléments de X :

$$\text{Il existe } i, j \in \mathbb{N} \text{ avec } i < j \text{ tels que } x_i \preccurlyeq x_j. \tag{BO}$$

Pour $x \in X$, on note $\uparrow x = \{y \in X \mid x \preccurlyeq y\}$. Pour $P \subseteq X$, on note $\uparrow P = \bigcup_{p\in P} \uparrow p$. On dit que P est *fermé par le haut* si $P = \uparrow P$, et que $B \subseteq X$ est une *base* de P si $P = \uparrow B$.

II.3 Montrer que si $\preccurlyeq$ est un bel ordre sur X, alors tout sous-ensemble de X a un nombre fini d'éléments minimaux. En déduire que tout sous-ensemble de X fermé par le haut admet une base finie.

Une suite $(x_n)_{n\in\mathbb{N}}$ est *croissante* si on a $x_i \preccurlyeq x_j$ pour tous i, j tels que $i < j$. Une suite $(x_n)_{n\in\mathbb{N}}$ est *strictement décroissante* si on a $x_j \prec x_i$ pour tous i, j tels que $i < j$.

II.4 Soit $\preccurlyeq$ un bel ordre sur X et $(x_n)_{n\in\mathbb{N}}$ une suite infinie d'éléments de X. Montrer que l'ensemble $\{i \in \mathbb{N} \mid \forall j > i,\ x_i \not\preccurlyeq x_j\}$ est fini. En déduire qu'un ordre est un bel ordre si et seulement si on peut extraire de toute suite infinie une sous-suite infinie croissante.

Deux éléments x et y sont *incomparables* si $x \not\preccurlyeq y$ et $y \not\preccurlyeq x$. Une suite (finie ou infinie) $(x_n)_n$ d'éléments de X est une *antichaîne* si x_i et x_j sont incomparables pour tous indices $i \neq j$.

II.5 Montrer qu'un ordre est un bel ordre si et seulement s'il n'a aucune antichaîne infinie et aucune suite strictement décroissante infinie.

II.6 Montrer qu'un ordre est un bel ordre si et seulement si pour toute suite $\uparrow P_0 \subseteq \uparrow P_1 \subseteq \uparrow P_2 \subseteq \cdots \subseteq X$, il existe $N \in \mathbb{N}$ tel que $\uparrow P_i = \uparrow P_j$ pour tous $i, j \geqslant N$.

II.7 Soit $\preccurlyeq$ un bel ordre sur X. Pour $d \geqslant 1$, on munit l'ensemble X^d de la relation $\preccurlyeq^d$ définie par : $(x_1, \ldots, x_d) \preccurlyeq^d (y_1, \ldots, y_d)$ si et seulement si pour tout $i = 1, \ldots, d$, on a $x_i \preccurlyeq y_i$. Montrer que $\preccurlyeq^d$ est un bel ordre sur X^d. On pourra raisonner par récurrence sur d.

Jusqu'à la fin de cette partie, on se place sur $X = A^*$. Soient $x, y \in A^*$. On dit que x est un *préfixe* de y s'il existe $z \in A^*$ tel que $y = xz$. On suppose donné un ordre total $\leqslant$ sur l'alphabet A. On définit les relations suivantes sur A^* :

– L'ordre *préfixe*, défini par $x \leqslant_p y$ si x est un préfixe de y.
– L'ordre *lexicographique*, défini par $x \leqslant_\ell y$ si soit x est un préfixe de y, soit $x = zax'$, $y = zby'$ avec $z, x', y' \in A^*$, $a, b \in A$ et $a < b$ (c'est-à-dire $a \leqslant b$ et $a \neq b$).
– L'ordre *hiérarchique*, défini par $x \leqslant_h y$ si soit $|x| < |y|$, soit $|x| = |y|$ et $x \leqslant_\ell y$.

II.8 Vérifier que les relations $\sqsubseteq$, $\leqslant_p$, $\leqslant_\ell$ et $\leqslant_h$ sont bien des ordres. Préciser lesquels sont totaux. Parmi les trois ordres $\leqslant_p$, $\leqslant_\ell$ et $\leqslant_h$, préciser ceux qui sont de beaux ordres.

On veut maintenant montrer que $\sqsubseteq$ est un bel ordre sur A^*. On appelle *mauvaise suite* une suite infinie d'éléments de A^* ne satisfaisant pas (BO) pour $\sqsubseteq$, et on suppose par l'absurde qu'il existe une mauvaise suite. On construit la suite $(x_n)_{n\in\mathbb{N}}$ de la façon suivante : on choisit pour x_0 un mot de taille minimale tel qu'il existe une mauvaise suite dont le premier élément est x_0. Puis, en supposant $x_0, \ldots, x_{k-1}$ choisis, on choisit pour x_k un mot de taille minimale tel qu'il existe une mauvaise suite dont les $(k+1)$ premiers éléments sont $x_0, \ldots, x_k$.

II.9 Justifier l'existence de la suite $(x_n)_{n\in\mathbb{N}}$ et montrer que c'est une mauvaise suite.

II.10 Vérifier que $x_n \neq \varepsilon$ pour tout n. Soit $x_n = a_n y_n$, où a_n est la première lettre de x_n. Vérifier qu'il existe une suite d'entiers $(n_k)_{k\in\mathbb{N}}$ strictement croissante telle que $(a_{n_k})_{k\in\mathbb{N}}$ soit constante. En considérant la suite $x_0, x_1, x_2, \ldots, x_{n_0-1}, y_{n_0}, y_{n_1}, y_{n_2}, y_{n_3}, \ldots$ conclure que $\sqsubseteq$ est un bel ordre sur A^*.

II.11 Déduire des questions précédentes que si $L \subseteq A^*$, alors $\mathrm{Sous}(L) = \{x \in A^* \mid \exists y \in L,\ x \sqsubseteq y\}$ est rationnel (contrairement à la question I.7, on ne suppose plus ici que L est rationnel).

III Problème de couverture et systèmes bien structurés

Un *système de transitions ordonné* est un triplet $\langle S, \to, \preccurlyeq\rangle$ où S est un ensemble (pas nécessairement fini), $\preccurlyeq$ est un ordre sur S, et $\to$ est une relation binaire sur S. On note $s \xrightarrow{*} t$ s'il existe $s_0, \ldots, s_n$ tels que $s = s_0 \to s_1 \to \cdots \to s_n = t$ (on observera qu'on peut avoir $n = 0$, auquel cas $s = t$).

On s'intéresse au problème de *couverture* :

Données : Un système de transitions ordonné $\langle S, \to, \preccurlyeq\rangle$ et $s, t \in S$.
Question : Existe-t-il $u \in S$ tel que $s \xrightarrow{*} u$ et $t \preccurlyeq u$?

III.1 À une machine de Turing M, on associe le système $\langle S_M, \to_M, =\rangle$ où S_M est l'ensemble des configurations de M (notées c, c',...), et $c \to_M c'$ s'il existe une transition de c à c' dans M (on rappelle qu'une configuration d'une machine de Turing M à une bande infinie à droite peut s'écrire uqv, où uv est le mot écrit sur la bande et q l'état courant de M, dont la tête se trouve sur la $(|u|+1)$-ème case). Montrer qu'étant donnée une machine de Turing M, le problème de couverture est indécidable pour le système de transitions ordonné $\langle S_M, \to_M, =\rangle$, même si M est supposée déterministe.

Un système de transitions ordonné $\langle S, \to, \preccurlyeq\rangle$ est *bien structuré* si l'on a les deux propriétés suivantes :

(*i*) $\preccurlyeq$ est un bel ordre sur S, et
(*ii*) pour tous $s, t, s' \in S$ tels que $s \to t$ et $s \preccurlyeq s'$, il existe t' tel que $s' \xrightarrow{*} t'$ et $t \preccurlyeq t'$.

III.2 Pour une machine de Turing M déterministe et une configuration c de M, soit $\|c\| \in \mathbb{N} \cup \{\infty\}$ la longueur du calcul le plus long de M à partir de c, évaluée en nombre de transitions effectuées. Si M ne s'arrête pas depuis la configuration c, on convient que $\|c\| = \infty$. Soit $\langle S_M, \to_M, \preccurlyeq_M\rangle$ le système de transitions ordonné où S_M et $\to_M$ sont décrits en question III.1, et $c \preccurlyeq_M c'$ si $\|c\| \leqslant \|c'\|$ (avec $n \leqslant \infty$ pour tout $n \in \mathbb{N} \cup \{\infty\}$). Montrer que $\langle S_M, \to_M, \preccurlyeq_M\rangle$ est bien structuré.

Pour $s \in S$, on pose $\mathsf{Pre}(s) = \{t \in S \mid t \to s\}$ et $\mathsf{Pre}^*(s) = \{t \in S \mid t \xrightarrow{*} s\}$. Pour $P \subseteq S$, on pose $\mathsf{Pre}(P) = \bigcup_{p\in P} \mathsf{Pre}(p)$ et $\mathsf{Pre}^*(P) = \bigcup_{p\in P} \mathsf{Pre}^*(p)$.

III.3 Soit $\langle S, \to, \preccurlyeq \rangle$ un système de transitions ordonné bien structuré. Montrer que si $P \subseteq S$ est fermé par le haut, alors $\mathsf{Pre}^*(P)$ l'est aussi.

On suppose maintenant la propriété additionnelle ($\mathscr{B}$) suivante :

Il existe un algorithme qui, étant donné $s \in S$, calcule une base finie $b(s)$ de $\uparrow\mathsf{Pre}(\uparrow s)$. ($\mathscr{B}$)

On étend la définition de b aux ensembles finis $P \subseteq S$ par $b(P) = \bigcup_{p \in P} b(p)$.

III.4 Soit $s \in S$. On définit une suite d'ensembles finis $P_i \subseteq S$ par $P_0 = \{s\}$, et $P_{i+1} = P_i \cup b(P_i)$ pour $i \geqslant 0$. Montrer qu'il existe n tel que $\uparrow P_n = \mathsf{Pre}^*(\uparrow s)$.

III.5 En déduire que le problème de couverture est décidable pour les systèmes bien structurés qui satisfont ($\mathscr{B}$) et tels qu'on peut décider, étant donnés $s, t \in S$, si $s \preccurlyeq t$.

III.6 Interpréter le résultat de la question III.2 à la lumière de celui de la question III.5.

III.7 Déduire des questions précédentes que le problème suivant est décidable, pour $d, n > 0$:

Données : Deux vecteurs $\vec{c}_0, \vec{c}_1 \in \mathbb{N}^d$ et n vecteurs $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{Z}^d$.
Question : Existe-t-il un entier K et une suite d'indices $n_1, n_2, \ldots, n_K$ tels que

(i) $\vec{c}_1 \leqslant^d \vec{c}_0 + \sum_{i=1}^{K} \vec{v}_{n_i}$, pour l'ordre $\leqslant^d$ sur $\mathbb{N}^d$ défini dans la question II.7, et
(ii) pour tout $1 \leqslant \ell < K$, toutes les coordonnées de $\vec{c}_0 + \sum_{i=1}^{\ell} \vec{v}_{n_i}$ sont positives ou nulles.

IV Automates à une file

Un *automate à une file* sur l'alphabet A est un triplet $\mathscr{A} = (A, Q, \delta)$ où Q est son ensemble fini d'*états*, et $\delta \subseteq Q \times (\{!, ?\} \times A) \times Q$ est sa *relation de transition*. Un tel automate manipule une file FIFO de messages. De façon informelle, ! s'interprète comme « envoyer » et ? comme « recevoir » : lorsque $\mathscr{A}$ effectue une transition étiquetée par $(!, a)$, le message a est ajouté en queue de file ; une transition étiquetée $(?, a)$ n'est franchissable que si le message le plus ancien dans la file (se trouvant en tête de file) est un a, auquel cas son franchissement retire ce message de la file.

Formellement, soit $S = \{(q, x) \mid q \in Q,\ x \in A^*\}$ l'ensemble des *configurations* de $\mathscr{A}$. Posons $B = (\{!, ?\} \times A)$. Pour $b \in B$, on définit la relation $\xrightarrow{b}$ sur S par $(q, x) \xrightarrow{b} (q', x')$ si $(q, b, q') \in \delta$, et
– si $b = (!, a)$, alors $x' = xa$ (ajout du message a dans la file) ;
– si $b = (?, a)$, alors $x = ax'$ (retrait du message a de la file).
Pour $z \in B^*$, on écrit $(q, x) \xrightarrow{z} (q', x')$ si soit $z = \varepsilon$ et $(q, x) = (q', x')$, soit $z = b_1 \cdots b_n$ avec $b_i \in B$ et il existe des états $q_0 = q, q_1, \ldots, q_n = q'$ et des mots $x_0 = x, x_1, \ldots, x_n = x'$ tels que $(q_{i-1}, x_{i-1}) \xrightarrow{b_i} (q_i, x_i)$ pour $i = 1, \ldots, n$. On définit enfin $\to$ sur S : $(q, x) \to (q', x')$ s'il existe $b \in B$ tel que $(q, x) \xrightarrow{b} (q', x')$. Ainsi, $\mathscr{A}$ définit un système de transitions ordonné $\langle S, \to, = \rangle$.

IV.1 Soit l'automate à une file $\mathscr{A} = \big(\{a\}, \{q\}, \{(q, (!, a), q), (q, (?, a), q)\}\big)$. Décrire la relation $\to$ du système $\langle S, \to, = \rangle$ associé, ainsi que le langage $\{z \in B^* \mid (q, \varepsilon) \xrightarrow{z} (q, \varepsilon)\}$.

IV.2 Montrer que le problème de couverture est indécidable pour les systèmes $\langle S, \to, = \rangle$ associés à des automates à une file. On pourra donner le principe de la simulation d'une machine de Turing M, en utilisant la file pour y écrire des configurations de M, et en choisissant convenablement l'alphabet A.

IV.3 On considère maintenant que les files peuvent perdre des messages : formellement, on augmente la relation $\to$ en y ajoutant $(q, y) \to (q, x)$ pour tout $q \in Q$ et tous mots $x, y \in A^*$ tels que $x \sqsubseteq y$. On définit la relation $\preccurlyeq$ sur S par $(p, x) \preccurlyeq (q, y)$ si $p = q$ et $x \sqsubseteq y$. Vérifier que $\preccurlyeq$ est un bel ordre sur S et montrer que le problème de couverture est décidable pour les systèmes de transitions ordonnés $\langle S, \to, \preccurlyeq \rangle$ associés à des automates à une file avec pertes.