e3a

CONCOURS ARTS ET MÉTIERS ParisTech - ESTP - POLYTECH

Épreuve d'Informatique MP

Durée 3 h

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, d'une part il le signale au chef de salle, d'autre part il le signale sur sa copie et poursuit sa composition en indiquant les raisons des initiatives qu'il est amené à prendre.

082

L'usage de calculatrices est interdit.

AVERTISSEMENT

La **présentation**, la lisibilité, l'orthographe, la qualité de la **rédaction, la clarté et la précision** des raisonnements entreront pour une **part importante** dans **l'appréciation des copies**. En particulier, les résultats non justifiés ne seront pas pris en compte. Les candidats sont invités à encadrer les résultats de leurs calculs.

Tournez la page S.V.P.

Indiquer en tête de copie le langage de programmation, Caml ou Pascal, choisi pour l'ensemble du sujet.

- L'épreuve est composée de 5 exercices, totalement indépendants.
- Dans toute la suite, on supposera que les éléments d'un tableau de longueur n sont indexés de 0 à $n-1$ et ce, **quel que soit le langage choisi** en début de sujet. De plus, on supposera disposer d'une fonction *longueur* qui donne le nombre d'éléments d'un tableau.
- On rappelle que pour $k \geqslant 2$ chaque entier naturel n se décompose de manière unique en base k sous la forme :

$$n = \sum_{i \geqslant 1} n_i k^{i-1} \qquad \text{avec } \forall i \in \mathbb{N}^\star, n_i \in [\![0; k-1]\!] = \{0, \ldots, k-1\}.$$

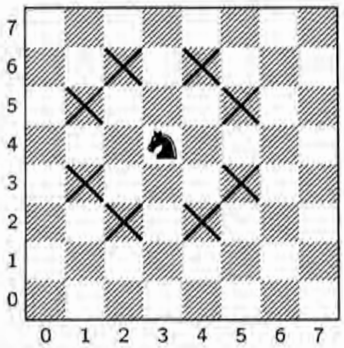
n_i est appelé $i^{\text{ième}}$ chiffre de n en base k. On dit que n a au plus d chiffres en base k si $n_i = 0$ pour tout $i > d$.

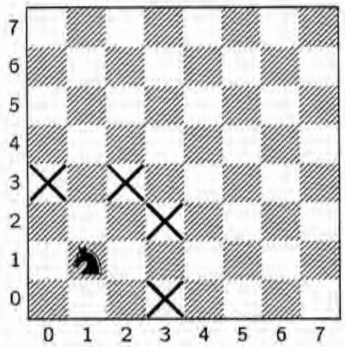
Ainsi, $6 = 0 + 1 \times 2 + 1 \times 2^2$, son premier chiffre en base 2 est 0, le second est 1 et le troisième est 1.

Exercice 1

Un échiquier est un plateau avec 8 lignes et 8 colonnes. Ces lignes et ces colonnes seront, dans cet exercice, numérotées de 0 à 7. Une position sur l'échiquier est un couple (i, j) d'entiers compris entre 0 et 7 inclus, avec i le numéro de ligne et j le numéro de colonne.

Un cavalier placé sur l'échiquier se déplace en bougeant de 2 cases dans une direction (verticale ou horizontale) et de 1 case perpendiculairement. Le dessin ci-dessous à gauche illustre les 8 possibilités de déplacement d'un cavalier situé loin des bords de l'échiquier. Comme le cavalier ne peut pas sortir du plateau, lorsqu'il est près des bords, il a moins de possibilités de se déplacer, comme l'illustre le dessin ci-dessous à droite.





1. Écrire une fonction **Valide** prenant en argument deux entiers relatifs i et j et vérifiant que le couple (i, j) est bien une position de l'échiquier. **Valide** renvoie un booléen.

2. Écrire une fonction **CoupSuivant** prenant en argument une position (i, j) et renvoyant la liste des positions que peut atteindre un cavalier placé en (i, j) en un seul coup.

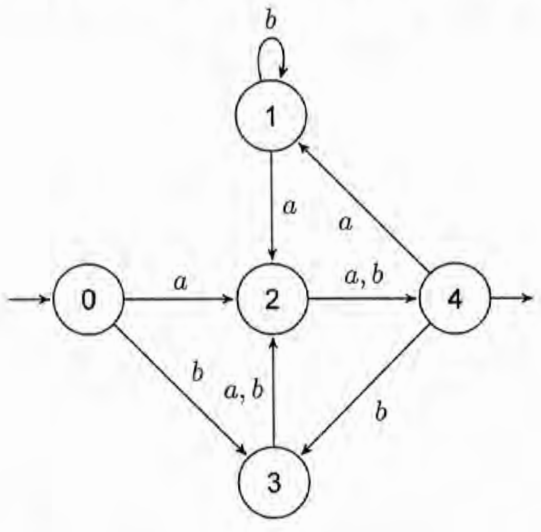
3. Écrire une fonction **Cavalier** prenant en argument une position (i_0, j_0) et renvoyant une matrice M de taille 8×8 telle que $M[i, j]$ est le nombre minimum de coups nécessaires à un cavalier situé en (i_0, j_0) pour arriver à la position (i, j).

Exercice 2

Dans cet exercice, on considère l'alphabet $\Sigma = \{a, b\}$. On note ε le mot vide, $|w|$ la longueur du mot w. Étant donnés un automate A, et deux états q et q' de A, on note $q \xrightarrow{w} q'$ le fait qu'il existe un chemin dans A, qui part de l'état q et arrive dans l'état q', étiqueté par le mot w. De plus, étant donné un mot y sur l'alphabet Σ, on définit le langage y^* par $y^* = \{y^n \mid n \in \mathbb{N}\}$. On remarque que $\varepsilon = y^0 \in y^*$.

Enfin, on dit qu'un langage L satisfait la propriété de l'étoile si et seulement si il existe $N \in \mathbb{N}$ tel que pour tout mot $w \in L$ tel que $|w| \geqslant N$, il existe trois mots x, y et z tels que $w = xyz$, et $y \neq \varepsilon$, et $|xy| \leqslant N$, et $xy^\star z \subset L$.

1. On considère l'automate suivant, et on appelle L_0 le langage qu'il reconnaît :



   Soit $w_0 = aabbbabbaa$.

   (a) Justifier que l'automate est déterministe complet.

   (b) Indiquer le chemin parcouru dans l'automate pour reconnaître w_0.

   (c) Démontrer qu'il existe 3 mots x, y et z tels que $|xy| \leqslant 5$, $y \neq \varepsilon$ et $w_0 = xyz$ et $xy^\star z \subset L_0$. Vous donnerez une valeur de (x, y, z) qui convient.

2. On considère dans cette question un langage L reconnu par un automate fini déterministe complet A ayant N états et d'état initial q_0. Soit un mot w dans L tel que $|w| \geqslant N$; démontrez qu'il existe un état q_1 et 3 mots x, y et z tels que $xyz = w$ et $y \neq \varepsilon$ et $q_0 \xrightarrow{x} q_1 \xrightarrow{y} q_1$ et $|xy| \leqslant N$.

3. Démontrer que tout langage rationnel satisfait la propriété de l'étoile.

4. Le langage $X = \{a^n b^n \mid n \in \mathbb{N}^\star\}$ est-il rationnel ?

5. Citer un langage X_2 qui n'est pas rationnel, mais tel que le langage $a^\star X_2 = \{a^k w \mid k \in \mathbb{N}$ et $w \in X_2\}$ est rationnel.

6. On considère le langage $Y = a^\star X = \{a^k a^n b^n \mid (k, n) \in \mathbb{N} \times \mathbb{N}^\star\}$.

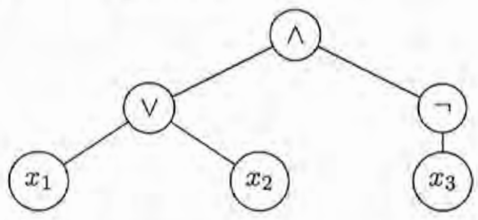
   (a) Démontrer que Y satisfait la propriété de l'étoile.

   (b) Y est-il rationnel ? Si oui, dessiner un automate qui le reconnaît.

Exercice 3

On considère un ensemble de variables $X = \{x_0, x_1, x_2, \ldots, x_n, \ldots\} = \{x_k \mid k \in \mathbb{N}\}$, et on note $X_n = \{x_0, x_1, \ldots, x_n\} = \{x_k \mid k \in [\![0; n]\!]\}$.

Dans cet exercice, on considère les formules logiques exprimées avec les variables de X et les connecteurs logiques OR, AND et NOT qui sont notés, respectivement, $\vee$, $\wedge$ et $\neg$. Une formule peut être représentée par un arbre, par exemple la formule $(x_1 \vee x_2) \wedge (\neg x_3)$ est représentée par l'arbre suivant.



Enfin, une interprétation des variables de X_n sera représentée par un tableau de booléens de longueur $n+1$. Le $i^{\text{ème}}$ élément du tableau étant la valeur de vérité de x_i.

En Caml, les formules seront représentées par un type `formule` défini comme suit :

```
type formule = OR of formule * formule | AND of formule * formule | NOT of
formule | Var of int;;
```

En Pascal, on supposera défini un type `FORMULE`, ainsi que les 3 fonctions suivantes :

- `FUNCTION symbole(f:FORMULE):INTEGER` qui étant donné une formule renvoie respectivement -1, -2 ou -3 si `f` est de la forme `g OR h` ou `g AND h` ou `NOT g`. Si `f` est de la forme x_n, alors `symbole` renvoie l'entier n.
- `FUNCTION fils_gauche(f:FORMULE):FORMULE` qui étant donné une formule `f` renvoie la formule `g` si `f` est de la forme `g OR h` ou `g AND h` ou `NOT g`.
- `FUNCTION fils_droit(f:FORMULE):FORMULE` qui étant donné une formule `f` renvoie la formule `h` si `f` est de la forme `g OR h` ou `g AND h` ou `NOT h`.

1. Dessinez l'arbre correspondant à la formule $x_1 \wedge (x_2 \wedge ((\neg x_3) \wedge (x_4 \wedge x_0)))$.
2. Écrire une fonction récursive **eval** prenant comme argument un entier naturel n, une formule φ ne faisant intervenir que les variables de X_n, et une interprétation des variables de X_n et renvoyant la valeur de vérité de φ.
3. Écrire une fonction **maxVar** qui, étant donné une formule φ, renvoie le plus grand n tel que x_n apparaît dans φ.

Tournez la page S.V.P.

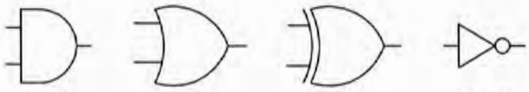
4. Écrire une fonction **satisfiable** prenant en argument une formule φ et renvoyant un booléen valant *vrai* si et seulement si la formule φ est satisfiable.

   Vous pouvez introduire une fonction intermédiaire qui servira pour cette question et la suivante.

5. Écrire une fonction **tautologie** prenant en argument une formule φ et renvoyant un booléen valant *vrai* si et seulement si la formule φ est une tautologie.

Exercice 4

On rappelle les représentations des portes AND, OR, XOR et NOT.



AND OR XOR NOT

Dans cet exercice, nous n'utiliserons que ces 4 portes et aucune autre.
Implémentez avec le minimum de portes, les fonctions booléennes suivantes :

1. $f(x,y) = (x \Rightarrow y) \wedge \neg(y \Rightarrow x)$.

2. $g(x,y) = x \Leftrightarrow y$

Exercice 5

Dans les trois algorithmes suivants, A et T sont des tableaux, k est un entier naturel, et clef est une fonction qui à chaque élément de A associe un entier compris entre 0 et k inclus. Enfin, on appelle clef d'un élément a du tableau A l'entier clef(a).

```
Algo1(A,k,clef)
   Crée un tableau T de longueur (k + 1) rempli de zéros
   Pour i allant de 0 à (longueur(A) − 1) faire
      j prend la valeur clef(A[i])
      T[j] prend la valeur T[j] + 1
   fin faire
Fin : Renvoie T.
```

1. Exécuter Algo1 avec $k = 3$ et $A = [1, 2, 3, 1, 2, 3]$ et clef valant la fonction identique.

2. Démontrez, à l'aide d'un invariant de boucle, qu'à l'issue de l'exécution de Algo1, pour tout i, $T[i]$ est le nombre d'éléments du tableau A ayant pour clef i.

```
Algo2(T)
    Pour i allant de 1 à (longueur(T) − 1) faire
        T[i] prend la valeur T[i] + T[i − 1]
    fin faire
Fin : Ne renvoie rien.
```

3. Que fait l'algorithme `Algo2` ?

```
Algo3(A,k,clef)
    Crée un tableau B de même longueur que A
    T prend la valeur Algo2(Algo1(A,k,clef))
    Pour i allant de 1 à longueur(A) faire
        j prend la valeur longueur(A) − i
        p prend la valeur clef(A[j])
        T[p] prend la valeur T[p] − 1
        B[T[p]] prend la valeur A[j]
    fin faire
Fin : Renvoie B.
```

4. Démontrer que `Algo3` trie les éléments du tableau A par clef croissante.

5. Démontrez que si a et b sont deux éléments du tableau A ayant la même clef, alors a et b sont placés dans le même ordre dans le tableau A et le tableau B.

6. Déterminer la complexité d'`Algo3` en fonction de la longueur de A et de k.

On considère à présent l'algorithme suivant, qui prend en entrée un tableau d'entiers naturels A ayant chacun au plus d chiffres en base k. Ici chiffre$_{k,i}$ est la fonction qui a n associe son $i^{\text{ième}}$ chiffre en base k.

```
Algo4(A,k,d)
    B prend la valeur A
    Pour i allant de 1 à d faire
        B prend la valeur Algo3(B,k − 1,chiffre_{k,i})
    fin faire
Fin : Renvoie B
```

7. Que fait l'algorithme Algo4 ? Justifier votre réponse avec un invariant de boucle.

8. Démontrer que la complexité de Algo4 est $O(d(n+k))$ où n est la longueur de A.

9. On suppose dans cette question que A contient n entiers naturels d'au plus b chiffres en base 2.

   (a) Soit r un entier naturel inférieur ou égal à b. Comment choisir k et d en fonction de r pour que `Algo4` termine en temps $O\left(\frac{b}{r}(n+2^r)\right)$?

(b) En supposant de plus que $b = \lambda \ln(n)$ avec $\lambda \in \mathbb{R}_+^*$. Comment choisir k et d en fonction de n pour qu'`Algo4` termine en temps $O(n)$.

10. Citer d'autres algorithmes pouvant faire la même chose qu'Algo4. Comparer la complexité d'Algo4 avec ces algorithmes.