ÉPREUVE COMMUNE DE TIPE 2007 - Partie D

TITRE :

La recherche de motifs fréquents : une méthode de fouille de données

Temps de préparation : ……..….2 h 15 minutes
Temps de présentation devant le jury : …….10 minutes
Entretien avec le jury : ……..10 minutes

GUIDE POUR LE CANDIDAT :

Le dossier ci-joint comporte au total : *10* pages

Document principal : 9 pages
Documents complémentaires : 1 page

Travail **suggéré** au candidat :

1. Lire le document tout en repérant les concepts et leurs articulations. Pour cela faire un schéma synoptique mettant en relation les définitions, les propriétés, les problèmes, les algorithmes ...

2. A travers des exemples, illustrer le sens de chaque notion formelle en pratique.

3. Analyser les algorithmes pour n'en retenir que les idées maîtresses. Mettre en rapport ces idées avec les notions et les propriétés qui sont sous-jacentes.

4. Construire une synthèse de l'analyse précédente, qui soit structurée et la plus simple possible, et la présenter au jury.

Attention : s*i le candidat préfère effectuer un autre travail sur le dossier, il lui est* ***expressément recommandé*** *d'en informer le jury avant de commencer l'exposé.*

CONSEILS GENERAUX POUR LA PREPARATION DE L'EPREUVE :

* Lisez le dossier en entier dans un temps raisonnable.
* Réservez du temps pour préparer l'exposé devant le jury.

- Vous pouvez écrire sur le présent dossier, le surligner, le découper … mais tout sera à remettre au jury en fin d'oral.
- En fin de préparation, rassemblez et ordonnez soigneusement TOUS les documents (transparents, *etc.*) dont vous comptez vous servir pendant l'oral, ainsi que le dossier, les transparents et les brouillons utilisés pendant la préparation. En entrant dans la salle d'oral, vous devez être prêts à débuter votre exposé.
- A la fin de l'oral, vous devez remettre au jury le présent dossier, les transparents et les brouillons utilisés pour cette partie de l'oral, ainsi que TOUS les transparents et autres documents présentés pendant votre prestation.

La recherche de motifs fréquents : une méthode de fouille de données.

Les mots suivis d'un astérisque sont définis dans le glossaire de l'annexe A.
La signification des notations mathématiques est précisée en annexe B.

1 Introduction : la fouille de donnée

La *fouille de données* (*data-mining* en anglais) est un domaine de la recherche en informatique répondant à une *problématique** nouvelle : grâce aux progrès technologiques que l'informatique a connus au cours des dernières décennies, il est devenu possible de stocker dans la *mémoire de masse** des ordinateurs d'énormes quantités de données numériques relatives à des applications très variées, qu'elles soient commerciales ou scientifiques (bases de corps célestes en astronomie, de *génomes** en biologie, de composés organiques en chimie, bases géographiques, socio-économiques ...). Ces *bases de données** sont généralement construites dans le but soit d'archiver les informations relatives à un domaine particulier, soit d'apporter une réponse à un problème précis, à l'image des réponses issues de sondages. Dans le second cas, les outils d'analyse statistique permettent d'extraire de ces données des éléments de réponse au problème initial soulevé. Ainsi les transactions commerciales enregistrées par une entreprise permettent d'établir sa comptabilité mais aussi de connaître l'évolution des ventes de telle ou telle gamme de produits. Mais ces éléments de réponse ne représentent souvent qu'une partie infime des connaissances que l'on peut apprendre des données. Dans l'exemple précédent, les transactions renferment des informations sur la *corrélation** entre achats et peuvent même permettre dans certains cas d'identifier différents profils de clients. Cette information est rarement exploitée par l'entreprise alors qu'elle permettrait d'améliorer les services à destination de la clientèle. Le but de la fouille de données est de réaliser une étude systématique des données en multipliant le nombre de questions posées dans l'espoir de déceler de nouveaux éléments de connaissance relatifs au domaine d'étude considéré. Dans la mesure où le nombre de questions envisageables croît extrêmement vite avec la diversité et donc la quantité des données présentes dans la base, le problème n'est plus seulement un problème de statistiques pour mesurer le degré de validité des réponses aux questions, mais aussi un problème d'informatique pour concevoir des algorithmes rapides à même de lutter contre l'explosion combinatoire inhérente au problème. Différentes méthodes de fouille de données ont été mises au point en fonction de la nature des données considérées et des questions envisagées. On distingue classiquement les méthodes de fouille de données numériques traitant de données plongées dans des *espaces continus**, des méthodes de fouille de données symboliques, traitant des données exprimées dans des *espaces discrets**. La présente étude porte sur l'une des méthodes de fouille de données symboliques les plus répandues, dite de *recherche de motifs fréquents*.

2 La recherche de motifs fréquents

La recherche des motifs fréquents fait l'hypothèse d'une base de données décrivant un ensemble d'objets $\mathcal{O} = \{o_1, \ldots, o_m\}$ par un ensemble d'attributs $\mathcal{A} = \{a_1, \ldots, a_n\}$. Un objet $o \in \mathcal{O}$ est décrit ou non par un attribut $a \in \mathcal{A}$ de sorte que la base de données est assimilable à une relation binaire $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$. On note ainsi $o\,\mathcal{D}\,a$ si l'objet $o \in \mathcal{O}$ présente l'attribut $a \in \mathcal{A}$. Une base de données peut donc être modélisée par une matrice booléenne où les lignes et les colonnes correspondent respectivement aux objets et aux attributs. L'exemple de la figure 1 représente le résultat d'un sondage fictif réalisé auprès de touristes étrangers en visite à Paris. Les objets correspondent à des touristes anonymes (T_i) et les attributs représentent les lieux visités : les bateaux mouches (BM), le centre Pompidou (CP), le musée du Louvre (LO), le musée d'Orsay (MO), la cathédrale Notre Dame (ND) et la tour Eiffel (TE). Dans la suite le triplet $(\mathcal{O}, \mathcal{A}, \mathcal{D})$ est supposé fixé. La

$\mathcal{D}$	BM	CP	LO	MO	ND	TE
T_1	×		×	×	×	
T_2	×		×	×	×	
T_3	×		×	×	×	×
T_4	×				×	×
T_5	×		×	×	×	×
T_6		×	×		×	
T_7					×	
T_8			×		×	
T_9			×		×	
T_{10}	×	×	×	×	×	

FIG. 1 – Exemple de base de données

méthode de recherche de motifs fréquents s'appuie sur la notion formelle de motif :

Définition 2.1. Un *motif* est un sous-ensemble de $\mathcal{A}$. Un motif M *décrit* un object o quand $\forall a \in M$, $o\,\mathcal{D}\,a$ et on note $o\,\underline{\mathcal{D}}\,M$. La *description* d'un objet $o \in \mathcal{O}$ est le motif $d(o) = \{a \in \mathcal{A} | o\,\mathcal{D}\,a\}$.

L'ensemble des motifs, noté $\mathcal{M} = \mathcal{P}(\mathcal{A})$, est muni de la relation d'ordre d'inclusion notée $\subseteq$. Nous réserverons la notation $\subset$ pour désigner l'inclusion stricte : $M_1 \subset M_2$ quand $M_1 \subseteq M_2$ et $M_1 \neq M_2$.

Propriété 2.2. *Un objet $o \in \mathcal{O}$ est décrit par un motif $M \in \mathcal{M}$ si et seulement si M est inclus dans la description $d(o)$: $o\,\underline{\mathcal{D}}\,M \Leftrightarrow M \subseteq d(o)$.*

Définition 2.3. Le *support* d'un motif M est l'ensemble des objets décrits par M : $\text{support}(M) = \{o \in \mathcal{O} | o\,\underline{\mathcal{D}}\,M\}$. La *fréquence* de M est le cardinal de son support : $\text{freq}(M) = |\,\text{support}(M)|$. Un motif est *fréquent* relativement à un entier naturel $f_{min} \in \mathbb{N}$ donné si sa fréquence est supérieure ou égale à f_{min}.

La fréquence d'un motif M divisée par la constante $|\mathcal{O}|$ donne la proportion d'objets de $\mathcal{O}$ décrits par M. Cette proportion est plus expressive que la fréquence équivalente mais les méthodes présentées ci-après préfèrent utiliser directement la fréquence pour éviter toute manipulation de nombres rationnels et tout calcul de division.

Propriété 2.4. *Le support et la fréquence sont des applications décroissantes de* $(\mathcal{M}, \subseteq)$ *vers respectivement* $(\mathcal{P}(\mathcal{O}), \subseteq)$ *et l'ensemble des entiers naturels* $(\mathbb{N}, \leq)$.

Propriété 2.5. *Pour tous motifs* M_1 *et* M_2 *de* $\mathcal{M}$ *:*

$$\text{support}(M_1 \cup M_2) = \text{support}(M_1) \cap \text{support}(M_2)$$

Définition 2.6. Le problème de *recherche des motifs fréquents* associé aux données $(\mathcal{O}, \mathcal{A}, \mathcal{D}, f_{min})$ consiste à déterminer le sous-ensemble $\mathcal{M}_f \subseteq \mathcal{M}$ des motifs fréquents ainsi que la fréquence de chaque motif fréquent.

Les algorithmes de recherche des motifs fréquents doivent parcourir la totalité de la base de données chaque fois qu'ils doivent déterminer la fréquence d'un ou de plusieurs motifs. La base de données $\mathcal{D}$ peut être grande au point de ne pouvoir être stockée en *mémoire vive**. La base réside alors sur un disque dur et sa lecture se traduit par des accès au disque beaucoup plus lents que des accès à la mémoire. La fonction `fréquence` (cf. le pseudo-code 1 de la page 3) permet d'accéder efficacement au disque pour déterminer toutes les fréquences d'un ensemble de motifs $\mathcal{C} \subseteq \mathcal{M}$: une seule passe est en effet nécessaire, selon une lecture séquentielle beaucoup plus rapide qu'une lecture aléatoire. La fonction place les fréquences calculées dans une *table d'association* $F : \mathcal{M} \to \mathbb{N}$ stockant certaines associations entre un motif M et sa fréquence $F[M]$ (voir l'annexe B pour une description plus précise des tables d'association). Pour être efficaces, les algorithmes de recherche de motifs fréquents doivent alors concilier deux objectifs antagonistes : d'une part le nombre d'accès à la base doit être réduit autant que possible compte tenu des accès particulièrement lents au disque. D'autre part, seuls les motifs susceptibles d'être fréquents doivent être générés et évalués car le nombre de motifs possibles croît de manière exponentielle avec le nombre n d'attributs ($|\mathcal{M}| = 2^n$). La solution consistant à calculer la fréquence de tous les motifs possibles en un seul accès à la base n'est donc pas envisageable.

Algorithme 1 fréquence($\mathcal{O}$, $\mathcal{D}$, $\mathcal{C}$, F) $\to F$

```
1: pour chaque M ∈ C faire
2:    F[M] ← 0
3: fin pour
4: pour chaque o ∈ O faire
5:    Lecture de d(o) dans la base D
6:    pour chaque M ∈ C faire
7:       si M ⊆ d(o) alors
8:          F[M] ← F[M] + 1
9:       fin si
10:   fin pour
11: fin pour
12: retourner F
```

L'algorithme `apriori` (cf. le pseudo-code 2 de la page 5) présente un bon compromis entre le nombre de motifs générés et le nombre d'accès à la base en tirant partie de la propriété 2.4 de décroissance de la fréquence. La fonction `apriori` accepte en argument

d'entrée le quadruplet $(\mathcal{O}, \mathcal{A}, \mathcal{D}, f_{min})$ du problème de recherche de motifs fréquents étudié. Les résultats en sortie sont l'ensemble des motifs fréquents $\mathcal{M}_f$ et la *table d'association* F : $\mathcal{M} \to \mathbb{N}$ associant à chaque motif fréquent M sa fréquence $F[M]$. Le principe d'`apriori` repose sur un *parcours par niveau* de l'ensemble des motifs : le *niveau* de longueur l de $\mathcal{M}$ correspond à l'ensemble des motifs de cardinal égal à l. Un parcours par niveau cherche les motifs fréquents niveau par niveau, dans l'ordre croissant de leur longueur. A chaque niveau l, n'est considéré que le sous-ensemble $\mathcal{C}_l$ des motifs *candidats* de cardinal l : un *motif candidat* est un motif M dont tous les motifs qui lui sont inclus strictement sont prouvés être fréquents (i.e $\forall M', M' \subset M \Rightarrow \text{freq}(M') \geq f_{min}$). L'ensemble des motifs fréquents du niveau l noté $\mathcal{F}_l$, est nécessairement inclus dans $\mathcal{C}_l$. On peut enfin remarquer que tout motif candidat M ne peut se construire qu'à partir des attributs fréquents (i.e $\forall a_i \in M$, $\{a_i\} \in \mathcal{F}_1$).

La table 2 donne les valeurs successives des ensembles $(\mathcal{C}_l)_{l\geq 1}$ calculées par l'algorithme `apriori` à partir de l'exemple ($\mathcal{O}$, $\mathcal{A}$, $\mathcal{D}$) de la figure 1 et d'une fréquence minimale f_{min} égale à 3. Les éléments de $\mathcal{C}_l$ écrits en gras sont également éléments de $\mathcal{F}_l$. Le nombre entre parenthèses figurant à la suite de chaque motif correspond à la fréquence du motif telle qu'elle est calculée par la fonction `fréquence`. L'exemple a ainsi nécessité 22 calculs de fréquence effectués en 4 parcours de la base.

Niveau l	Motifs éléments de $\mathcal{C}_l$
1	**{BM}**(6), {CP}(2), **{LO}**(8), **{MO}**(5), **{ND}**(10), **{TE}**(3)
2	**{BM,LO}**(5), **{BM,MO}**(5), **{BM,ND}**(6), **{BM,TE}**(3), **{LO,MO}**(5) **{LO,ND}**(8), {LO,TE}(2), **{MO,ND}**(5), {MO,TE}(2), **{ND,TE}**(3)
3	**{BM,LO,MO}**(5), **{BM,LO,ND}**(5), **{BM,MO,ND}**(5) , **{BM,ND,TE}**(3), **{LO,MO,ND}**(5)
4	**{BM,LO,MO,ND}**(5)
5	

FIG. 2 – Résultats de l'algorithme 2

3 Optimisation

L'algorithme 2 peut être rendu plus efficace en considérant la notion de motif générateur développée ci-après :

Définition 3.1. Deux motifs M_1 et M_2 sont *équivalents*, et on note $M_1 \simeq M_2$, s'ils ont même ensemble support.

Propriété 3.2. *$\simeq$ est une relation d'équivalence sur $\mathcal{M}$. La classe d'équivalence du motif M est notée $\overline{M}$.*

Définition 3.3. Un *motif générateur* est un motif M minimal dans sa classe d'équivalence : $\forall M' \in \overline{M}, M' \subseteq M \Rightarrow M' = M$.

Ainsi dans l'exemple des sites parisiens les plus visités, le motif $\{BM, LO\}$ est générateur puisque sa fréquence égale à 5 est strictement inférieure à celles de $\{BM\}$ et

Algorithme 2 apriori($\mathcal{O}$, $\mathcal{A}$, $\mathcal{D}$, f_{min}) $\rightarrow (\mathcal{M}_f, F)$

$n \leftarrow |\mathcal{A}|$; $m \leftarrow |\mathcal{O}|$; $F[\emptyset] \leftarrow m$
si $f_{min} > m$ **alors**
  retourner $(\emptyset, F)$
fin si
$\mathcal{C}_1 \leftarrow \{\{a_1\} \dots \{a_n\}\}$; $l \leftarrow 1$; $\mathcal{M}_f \leftarrow \{\emptyset\}$
tant que $\mathcal{C}_l \neq \emptyset$ **faire**
  $F \leftarrow$ `fréquence`($\mathcal{O}$, $\mathcal{D}$, C_l, F)
  $\mathcal{F}_l \leftarrow \emptyset$
  pour chaque $M \in \mathcal{C}_l$ **faire**
    si $F[M] \geq f_{min}$ **alors**
      $\mathcal{F}_l \leftarrow \mathcal{F}_l \cup \{M\}$
    fin si
  fin pour
  $\mathcal{M}_f \leftarrow \mathcal{M}_f \cup \mathcal{F}_l$
  {On pose $\mathcal{F}_1 = \{\{a_{k_1}\}, \dots, \{a_{k_q}\}\}$ avec $1 \leq k_1 < \dots < k_q \leq n$}
  $\mathcal{C}' \leftarrow \emptyset$
  pour chaque $M \in \mathcal{F}_l$ **faire**
    {On pose $M = \{a_{k_{i_1}}, \dots, a_{k_{i_l}}\}$ avec $1 \leq i_1 < i_2 < \dots < i_l \leq q$}
    pour $i = i_l + 1$ à q **faire**
      $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{M \cup \{a_{k_i}\}\}$
    fin pour
  fin pour
  $l \leftarrow l + 1$
  $\mathcal{C}_l \leftarrow \emptyset$
  pour chaque $M \in \mathcal{C}'$ **faire**
    {On pose $M = \{a_{k_{i_1}}, \dots, a_{k_{i_l}}\}$ avec $1 \leq i_1 < i_2 < \dots < i_l \leq q$}
    $j \leftarrow 1$; *candidat* $\leftarrow$ **vrai**
    tant que $j < l$ et *candidat* = **vrai faire**
      si $M \setminus \{a_{k_{i_j}}\} \notin \mathcal{F}_{l-1}$ **alors**
        candidat $\leftarrow$ **faux**
      sinon
        $j \leftarrow j + 1$
      fin si
    fin tant que
    si *candidat* = **vrai alors**
      $\mathcal{C}_l \leftarrow \mathcal{C}_l \cup \{M\}$
    fin si
  fin pour
fin tant que
retourner $(\mathcal{M}_f, F)$

de $\{LO\}$ respectivement égales à 6 et 8. Une classe d'équivalence C peut avoir plusieurs motifs générateurs, dont l'ensemble est noté gen(C). Ainsi la classe d'équivalence $\overline{\{BM, LO, MO, ND\}}$ dont le support est $\{T_1, T_2, T_3, T_5, T_{10}\}$, présente deux générateurs $\{MO\}$ et $\{BM, LO\}$. Le support support(C) et la fréquence freq(C) d'une classe d'équivalence C est le support et la fréquence d'un de ces éléments. L'équivalence entre motifs amène à plusieurs propriétés intéressantes.

Propriété 3.4. *Quels que soient les ensembles d'attributs A, B et C de $\mathcal{M}$, si $A \simeq B$ alors $A \cup C \simeq B \cup C$.*

Démonstration. Si $A \simeq B$ alors support$(A) \cap$ support$(C) =$ support$(B) \cap$ support(C). La propriété 2.5 implique que support$(A \cup C) =$ support$(B \cup C)$ et donc que $A \cup C \simeq B \cup C$. □

Propriété 3.5. *Les motifs contenus dans un motif générateur sont tous générateurs.*

Démonstration. Soient deux motifs M_1 et M_2 vérifiant $M_1 \subset M_2$. Il existe un ensemble d'attributs C non vide et disjoint de M_1 tel que $M_2 = M_1 \cup C$. Si M_1 est supposé être non générateur alors M_1 admet un sous-ensemble propre A qui lui est équivalent : $A \subset M_1$ et $A \simeq M_1$. La propriété 3.4 entraîne que $A \cup C \simeq M_1 \cup C$. De plus $M_1 \cap C = \emptyset$ donc $A \cup C \subset M_1 \cup C$. M_2 étant équivalent à un sous-ensemble propre $A \cup C$, il est donc non générateur. La *contraposée** donne la propriété 3.5. □

Propriété 3.6. *La fréquence d'un motif non générateur M est égale au minimum des fréquences des motifs inclus strictement dans celui ci :*

$$\text{freq}(M) = \min_{M' \subset M} (\text{freq}(M'))$$

Démonstration. Soit l'ensemble $\mathcal{M}^-(M)$ des motifs inclus strictement dans M. Soit M_m un motif de $\mathcal{M}^-(M)$ qui minimise la fréquence dans cet ensemble. Du fait de la décroissance de la fréquence, $M_m \subset M$ entraîne freq$(M_m) \geq$ freq(M). Par ailleurs M est non générateur : il existe donc un motif $M' \in \mathcal{M}^-(M)$ tel que freq(M') = freq(M). Or freq(M_m) est minimale dans $\mathcal{M}^-(M)$ donc freq$(M_m) \leq$ freq(M'). Finalement freq$(M_m) =$ freq(M). □

Propriété 3.7. *Un motif M est générateur si et seulement si :*

$$\text{freq}(M) < \min_{M' \subset M} (\text{freq}(M'))$$

Démonstration. Posons $f = \min_{M' \subset M} (\text{freq}(M'))$. Si M est générateur alors la définition 3.3 entraîne :

$$\forall M' \in \mathcal{M}, M' \subset M \Rightarrow \text{support}(M) \subset \text{support}(M') \text{ donc } \text{freq}(M) < \text{freq}(M')$$

Le passage au minimum sur l'ensemble fini des minorants M' de M donne freq$(M) < f$. Réciproquement, si freq$(M) < f$ alors freq$(M) \neq f$. La contraposée de la propriété 3.6 implique que M est générateur. □

Définition 3.8. Un *motif non fréquent minimal* est un motif M non fréquent dont tous les sous-motifs sont fréquents : $\forall M' \subset M, \text{freq}(M') \geq f_{min}$.

A partir des fréquences de l'ensemble des motifs générateurs fréquents et de l'ensemble des motifs non fréquents minimaux, il est possible, en vertu de la propriété 3.6, de calculer les fréquences de tous les motifs fréquents sans jamais consulter la base de données :

Propriété 3.9. *Un motif non fréquent minimal est un motif générateur.*

Démonstration. Tous les motifs inclus strictement dans un motif non fréquent minimal M sont fréquents et ne peuvent donc pas être équivalents à M. M est donc minimal dans sa classe d'équivalence et donc générateur. □

L'algorithme `Pascal` (cf. le pseudo-code 3 de la page 8) est une adaptation de l'algorithme `apriori` exploitant les propriétés précédentes. Il calcule l'ensemble des motifs générateurs fréquents $\mathcal{G}_f$, l'ensemble des motifs non fréquents minimaux $\mathcal{G}_{nfm}$ et une table d'association $F : \mathcal{G}_f \rightarrow \mathbb{N}$ associant à tout motif générateur fréquent sa fréquence.

La table 3 donne les valeurs successives des ensembles $(\mathcal{C}_i)_{i\geq 1}$ calculées par l'algorithme `Pascal` à partir de l'exemple $(\mathcal{O}, \mathcal{A}, \mathcal{D})$ de la figure 1 et d'une fréquence minimale $f_{min} = 3$. Les motifs de $\mathcal{C}_i$ figurant respectivement en gras et en italique sont également éléments des ensembles $\mathcal{G}_i$ et de $\mathcal{G}_{nfm}$.

Niveau l	Motifs éléments de $\mathcal{C}_l$
1	**{BM}**(6), {*CP*}*(2)*, **{LO}**(8), **{MO}**(5), {ND}(10), **{TE}**(3)
2	**{BM,LO}**(5), {BM,MO}(5), {BM,TE}(3), {LO,MO}(5), {*LO,TE*}(2), {*MO,TE*}(2)
3	

FIG. 3 – Résultats de l'algorithme 3

L'avantage de l'algorithme `Pascal` est de réduire le nombre d'accès à la base de données $\mathcal{D}$ sans pour autant perdre d'information. La donnée des fréquences de tous les générateurs fréquents ainsi que la donnée de l'ensemble des motifs non fréquents minimaux permettent en effet de déduire la fréquence de tous les motifs fréquents selon la propriété 3.6. L'algorithme `complétion` (cf. le pseudo-code 4) reconstitue ainsi l'ensemble $\mathcal{M}_f$ à partir des résultats $(\mathcal{G}_f, \mathcal{G}_{nfm}, F)$ retournés par `Pascal`. Il étend pour cela l'ensemble de définition de la table d'association F de l'ensemble des générateurs fréquents $\mathcal{G}_f$ à l'ensemble des motifs fréquents $\mathcal{M}_f$.

L'économie du nombre d'accès à la base est particulièrement efficace en présence de données de forte densité.

Définition 3.10. La *densité* d'une base de données $(\mathcal{O}, \mathcal{A}, \mathcal{D})$ est le quotient du nombre de motifs fréquents divisés par le nombre de motifs générateurs fréquents.

La densité de la base exemple est ainsi de 20/6 alors que le minimum théorique de la densité est de 1. Le fait que la base soit relativement dense explique que `Pascal` ait effectué seulement 12 calculs de fréquence et 2 passes sur la base, là où `apriori` nécessitait 22 calculs de fréquences et 4 passes sur la base.

Algorithme 3 Pascal($\mathcal{O}$, $\mathcal{A}$, $\mathcal{D}$, f_{min}) $\rightarrow$ $(\mathcal{G}_f, \mathcal{G}_{nfm}, F)$

$n \leftarrow |\mathcal{A}|$; $m \leftarrow |\mathcal{O}|$; $F[\emptyset] \leftarrow m$
si $f_{min} > m$ **alors**
  retourner $(\emptyset, \{\emptyset\}, F)$
fin si
$\mathcal{C}' \leftarrow \{\{a_1\} \ldots \{a_n\}\}$; $l \leftarrow 1$; $\mathcal{G}_0 \leftarrow \{\emptyset\}$; $\mathcal{G}_f \leftarrow \mathcal{G}_0$; $\mathcal{G}_{nfm} \leftarrow \emptyset$
tant que $\mathcal{C}' \neq \emptyset$ **faire**
  $\mathcal{C}_l \leftarrow \emptyset$
  pour chaque $M \in \mathcal{C}'$ **faire**
    {On pose $M = \{a_{i_1}, \ldots, a_{i_l}\}$ avec $1 \leq i_1 < i_2 < \ldots < i_l \leq n$}
    $j \leftarrow 1$; *candidat* $\leftarrow$ **vrai** ; $F_{min}[M] \leftarrow m$
    tant que $j < l$ et *candidat* = **vrai faire**
      $M' \leftarrow M \setminus \{a_{i_j}\}$
      si $M' \notin \mathcal{G}_{l-1}$ **alors**
        candidat $\leftarrow$ **faux**
      sinon
        $F_{min}[M] \leftarrow \min(F_{min}[M], F[M'])$; $j \leftarrow j + 1$
      fin si
    fin tant que
    si *candidat* = **vrai alors**
      $\mathcal{C}_l \leftarrow \mathcal{C}_l \cup \{M\}$
    fin si
  fin pour
  $F \leftarrow$ `fréquence`($\mathcal{O}$, $\mathcal{D}$, C_l, F)
  $\mathcal{G}_l \leftarrow \emptyset$
  pour chaque $M \in \mathcal{C}_l$ **faire**
    si $F[M] \geq f_{min}$ **alors**
      si $F[M] < F_{min}[M]$ **alors**
        $\mathcal{G}_l \leftarrow \mathcal{G}_l \cup \{M\}$;
      fin si
    sinon
      $\mathcal{G}_{nfm} \leftarrow \mathcal{G}_{nfm} \cup \{M\}$
    fin si
  fin pour
  $\mathcal{G}_f \leftarrow \mathcal{G}_f \cup \mathcal{G}_l$
  {On pose $\mathcal{G}_1 = \{\{a_{k_1}\}, \ldots, \{a_{k_q}\}\}$ avec $1 \leq k_1 < \ldots < k_q \leq n$}
  $\mathcal{C}' \leftarrow \emptyset$
  pour chaque $M \in \mathcal{G}_l$ **faire**
    {On pose $M = \{a_{k_{i_1}}, \ldots, a_{k_{i_l}}\}$ avec $1 \leq i_1 < i_2 < \ldots < i_l \leq q$}
    pour $i = i_l + 1$ à q **faire**
      $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{M \cup \{a_{k_i}\}\}$
    fin pour
  fin pour
  $l \leftarrow l + 1$
fin tant que
retourner $(\mathcal{G}_f, \mathcal{G}_{nfm}, F)$

Algorithme 4 complétion($\mathcal{A}$, $\mathcal{G}_f$, $\mathcal{G}_{nfm}$, $F : \mathcal{G}_f \to \mathbb{N}$) $\to (\mathcal{M}_f, F)$

$n \leftarrow |\mathcal{A}|$; $l \leftarrow 0$; $m \leftarrow F(\emptyset)$
$\mathcal{C}_0 \leftarrow \{\emptyset\}$; $\mathcal{M}_f \leftarrow \emptyset$
tant que $\mathcal{C}_l \neq \emptyset$ **faire**
  $\mathcal{C}_{l+1} \leftarrow \emptyset$
  pour chaque $M \in \mathcal{C}_l$ **faire**
    si $M \in \mathcal{G}_f$ **alors**
      $frequent \leftarrow$ **vrai**
    sinon si $M \in \mathcal{G}_{nfm}$ **alors**
      $frequent \leftarrow$ **faux**
    sinon
      $j \leftarrow 1$; $frequent \leftarrow$ **vrai** ; $F[M] \leftarrow m$
      {On pose $i_0 = 0$ si $M = \emptyset$ ou sinon $M = \{a_{i_1}, ..., a_{i_l}\}$ avec $1 \leq i_1 < i_2 < \ldots < i_l \leq n$}
      tant que $j < l$ et $frequent =$ **vrai faire**
        $M' \leftarrow M \setminus \{a_{i_j}\}$
        si $M' \notin \mathcal{M}_f$ **alors**
          $frequent \leftarrow$ **faux**
        sinon
          $F[M] \leftarrow \min(F[M], F[M'])$
        fin si
      fin tant que
    fin si
    si $frequent =$ **vrai alors**
      $\mathcal{M}_f \leftarrow \mathcal{M}_f \cup \{M\}$
      pour $i = i_l + 1$ à n **faire**
        $\mathcal{C}_{l+1} \leftarrow \mathcal{C}_{l+1} \cup \{M \cup \{a_i\}\}$
      fin pour
    fin si
  fin pour
  $l \leftarrow l + 1$
fin tant que
retourner $(\mathcal{M}_f, F)$

ANNEXES

A Glossaire

Base de données Ensemble de données logiquement organisé pour être exploité au moyen d'un logiciel appelé système de gestion de base de données (SGBD).

Contraposée La contraposée d'une implication $A \Rightarrow B$ est l'implication $non(B) \Rightarrow non(A)$ équivalente à $A \Rightarrow B$ dans la logique des propositions.

Corrélation Dépendance réciproque de deux phénomènes qui varient simultanément, qui sont fonction l'un de l'autre, qui évoquent ou manifestent un lien de cause à effet.

Espace continu Un espace continu est synonyme ici d'espace Euclidien : les objets sont des éléments de $\mathbb{R}^n$ caractérisables par les notions de distance, de produit scalaire, d'angle ...

Espace discret Un espace discret est par opposition à un espace continu, un espace métrique dans lequel l'ensemble des distances entre paires d'éléments de l'espace admet un minorant non nul.

Génome Ensemble de gènes portés par les chromosomes et caractéristique d'une espèce vivante.

Mémoire de masse Mémoire externe de grande capacité d'un ordinateur, réalisée à l'aide de périphériques tels que disques durs, bandes magnétiques, CD/DVD ...

Mémoire vive Mémoire volatile d'un ordinateur à base de transistors, dont les temps d'accès aléatoires sont beaucoup plus rapides que ceux des mémoires de masse.

Problématique Ensemble de questions qu'une science ou une philosophie se pose relativement à un domaine particulier.

B Notations mathématiques et informatiques

- L'ensemble des parties d'un ensemble E est noté $\mathcal{P}(E)$.
- Les relations d'inclusion et d'inclusion stricte entre ensembles sont notées respectivement $\subseteq$ et $\subset$.
- La différence entre ensembles est notée $E_1 \setminus E_2 = \{e \in E_1 \text{ et } e \notin E_2\}$.
- Le cardinal d'un ensemble E est noté $|E|$.
- $\leftarrow$ est le symbole d'affectation : $V \leftarrow v$ copie la valeur v dans la variable V.
- Une table d'association A d'un ensemble E vers un ensemble F est une structure de données abstraite capable de stocker des associations d'un élément e de E vers un élément f de F. Ainsi l'écriture $A[e] \leftarrow f$ associe à e la valeur f en remplaçant l'ancienne association reliant e à sa valeur ou sinon en créant une nouvelle association.
- Chaque algorithme commence par un en-tête de déclaration du type :
  `nom-de-l'algorithme(liste des arguments en entrée)` $\rightarrow$ `(liste des arguments en sortie)`