G. HUNAULT Angers, 2000

Mini-Projet en Algorithmique et Programmation : Variables Qualitatives et Taux de Contravis (Licence MASS)

1. Position du problème

Soit X une variable logique, c'est à dire une colonne indicée de chiffres 0 et 1 (on pourra interpréter 0 comme *oui* et 1 comme *non*). On dit qu'on effectue l'analyse qualitative de X lorsqu'on fournit les renseignements suivants : nombre de valeurs en tout, nombre de fois o on a le chiffre 0 (avec le pourcentage correspondant), nombre de fois o on a le chiffre 1 (avec le pourcentage correspondant), le tout présenté par ordre décroissant de pourcentage (voir un exemple plus bas).

Si M est une matrice logique, c'est à dire une matrice de chiffres 0 et 1, on dit qu'on effectue l'analyse qualitative de M si on effectue l'analyse des colonnes de M triée par pourcentage décroissant (voir un exemple plus bas).

Soient maintenant X et Y deux variables logiques, c'est à dire deux colonnes indicées de chiffres 0 et 1. On dit que X et Y sont en contravis à l'indice i si la i-ème valeur de X, notée X_i est différente de la i-ème valeur de Y, notée Y_i.

On appelle *taux de contravis* de X et de Y que l'on note $t(X,Y)$ le rapport du nombre de contravis sur le nombre total de valeurs. On appelle *matrice de contravis* des variables logiques V_i la matrice symétrique m telle que

$$m_{i,j} = t(V_i, V_j)$$

Soient par exemple les variables $V_1 = (1,1,0,0,0)$, $V_2 = (1,0,0,0,0)$ et $V_3 = (0,0,1,1,1)$ et M la matrice (V_1, V_2, V_3). La matrice M est donc

```
1  1  0
1  0  0
0  0  1
0  0  1
0  0  1
```

L'analyse de M produit le résultat trié suivant

```
Variable V2 :  4 oui (80 %)  1 non (20 %)
Variable V1 :  3 oui (60 %)  2 non (40 %)
Variable V3 :  3 non (60 %)  2 oui (40 %)
```

et la matrice de contravis donnée sous forme triangulaire inférieure, vaut,

```
0
0.2  0
1.0  0.8  0
```

Le but du miniprojet est de programmer en *Rexx* et en *Apl* l'analyse qualitative et le taux de contravis d'une matrice logique.

On suppose désormais qu'il y a p variables de chacune n valeurs mises dans un fichier-texte, à raison de p valeurs par ligne, la ligne i du fichier contenant les valeurs à l'indice i des variables. Par exemple, on pourrait utiliser le fichier suivant de variables logiques :

```
1    0    0    0    0
1    0    1    1    1
1    0    1    1    1
```

2. Travail demandé

2.a) Ecrire un algorithme complet (en-tête, paramètres, variables locales, variables globales et corps de l'algorithme) nommé analyseQualitative qui fournit l'analyse qualitative d'une matrice logique M.

2.b) Ecrire un algorithme complet (en-tête, paramètres, variables locales, variables globales et corps de l'algorithme) nommé tauxContr qui fournit la taux de contravis de deux variables logiques X et Y de n valeurs.

2.c) Si on suppose connues les fonctions algorithmiques mots et mot dont vous rappelerez la syntaxe, donner un algorithme "principal" qui fournit l'analyse qualitative et la matrice de contravis des variables, la matrice étant lue à partir d'un fichier de variables logiques. (on utilisera les fonctions analyseQualitative et tauxContr définies par les algorithmes précédents).

2.d) Si on décide d'utiliser des variables qualitatives plutot que des variables logiques (c'est à dire des variables V_i dont les valeurs vont de 0 à $maxmod_i$), donner, si besoin est, le corps du nouvel algorithme de calcul de la fonction tauxContr. Indiquez comment (et où) on pourrait lire les noms associés aux valeurs 0, 1, 2...

2.e) Traduire en *Rexx* les algorithmes précédents en un programme complet. On fera de l'algorithme de tauxContr un sous-programme qui renvoie la valeur du taux de contravis. Le programme utilisera un et un seul paramètre, à savoir celui du nom du fichier qui contient les valeurs des variables (logiques ou qualitatives). Le programme testera la présence du fichier ; sans paramètre, le programme rappelera sa syntaxe.

2.f) Traduire les algorithmes en fonctions *Apl*. On fera de l'algorithme de tauxContr une fonction "à résultat explicite" qui renvoie la valeur du taux de contravis. La fonction principale utilisera deux paramètres, à savoir le nombre de décimales (pour les % et pour les taux de contravis) et celui du nom de la matrice qui contient les valeurs des variables (logiques ou qualitatives). Par exemple, si la fonction principale est AQL alors on tapera 2 AQL M pour exécuter la fonction.

2.g) Résoudre le problème du calcul de la matrice de contravis en APL "pur et dur" c'est à dire sans boucles. Indication : on pourra utiliser ∘.F avec une fonction F judicieuse sur des indices de colonne avec une copie de la matrice M en variable globale avant de gérer l'affichage en matrice triangulaire inférieure par une autre fonction.

Ce miniprojet est à rédiger seul(e). Vous avez certainement à effectuer une recherche bibliographique simple (indiquez vos sources). Vous m'enverrez en attaché par mail un fichier .zip ou un fichier .tar qui contient tout votre travail, à savoir : une zone de travail Apl (fichier .ws = "*Workspace*" ou .apl), un fichier .rex, texte du programme Rexx, ainsi qu'un document .doc au sens de *Word* pour *Windows* (éventuellement au format RTF) ou un fichier .tex pour LateX qui contiendra les algorithmes, la documentation et vos commentaires.

Ces fichiers auront pour nom vos initiales : Jomes Band fournira donc JB.WS, JB.REX et JB.DOC le tout dans JB.ZIP ou dans JB.TAR ; par contre Nona Higane rendra NH.WS, NH.REX et NH.DOC, le tout dans NH.ZIP ou dans NH.TAR. Lors de l'archivage, vous ne mettrez aucun sous-répertoire.

Ce travail de recherche, programmation et rédaction est à rendre pour le mardi 28 novembre, 9h (aucun retard ne sera toléré).

3. Remarques concernant la rédaction

Il vaut mieux rendre un travail incomplet que de ne rien rendre. La qualité des algorithmes, en particulier leur simplicité et leur lisibilité sera appréciée. Indiquez clairement vos choix : si un terme vous parait ambigu, précisez sa définition pour vous. Justifiez vos modes de programmation ; récursivité ou linéarité, appel d'opérateurs, de fonctions prédéfinies ou écriture de boucles explicites. Nommez distinctement vos variables (en particulier, vous utiliserez systématiquement au moins 4 lettres par identificateur). N'hésitez pas à commenter sans dépasser toutefois un commentaire toutes les 4 lignes en moyenne (sauf pour l'en-tête).

Vous n'êtes pas obligé(e) de tout tester dans tous les sens. Effectuez donc des tests élémentaires. L'affichage des résultats à l'écran devra être très propre, très soigné. Ainsi en cas d'affichage numérique multi-lignes, on mettra bien les unités sous les unités, les dizaines sous les dizaines etc. Soignez aussi la rédaction, l'orthographe (utilisez des correcteurs orthographiques et grammaticaux) ...

La note sera principalement fonction de la qualité de l'analyse du travail à effectuer, de l'exposé de la méthode et de la clarté de la rédaction. La valeur attribuée au programme ne représentera que 25 % de la note ; aucune interface n'est demandée.

Aucun langage autre que cité ne devra être utilisé, aucun traitement de textes autre que cité ne devra être utilisé. Tout fichier qui ne correspondra pas aux initiales de l'élève ne sera pas lu. En particulier, les fichiers nommés minip, algo, stat,projet, doc ne seront pas lus.

Un programme même incomplet mais commenté et bien indenté sera au moins autant apprécié qu'un programme qui "marche" mais sans commentaire et illisible quand aux structures de flot (boucles, tests etc.).

4. Références

Le logiciel *APL* pour *Dos/Windows* version spéciale Université d'Angers peut être récupéré à l'adresse

`http://www.info.univ-angers.fr/pub/gh/tuteurs/tutapl.htm`

on y trouvera aussi comment récupérer un *APL* gratuit pour *Linux*.

Le logiciel *REGINA (REXX)* pour *Dos/Windows* et *Linux* peut être récupéré à l'adresse

`http://www.info.univ-angers.fr/pub/gh/wals/getrex.htm`

Plus généralement, tous les documents pour le cours Algorithmique, Langages et Styles de Programmation dont un manuel élémentaire d'*Algorithmique raisonnée*, les consignes pour le projet et ce texte sont disponibles à l'adresse

`http://www.info.univ-angers.fr/pub/gh/wals/alsgen.htm`