MASTER EN SCIENCES INFORMATIQUES

Théorie des sous-ensembles flous et aide au diagnostic médical

Petit, Marie-France

Award date:
1989

Awarding institution:
Universite de Namur

Année académique 1988 - 1989.

PETIT Marie-France.
Promoteur : J. FICHEFET.

THEORIE DES SOUS-ENSEMBLES FLOUS

ET AIDE AU DIAGNOSTIC MEDICAL.

Mémoire de fin d'études en vue de l'obtention du diplôme
de licence et maîtrise en informatique.

REMERCIEMENTS.

Je tiens tout d'abord à remercier Monsieur FICHEFET d'avoir accepté d'être le promoteur de ce mémoire.

Mes remerciements vont également à toutes les autres personnes qui, de façons très diverses, m'ont apporté leur aide ou leur attention durant la réalisation de ce travail.

INTRODUCTION.

A. Sous-ensembles flous et diagnostic. 1
B. Plan du travail..... 2

PARTIE 1 : CONCEPTS GENERAUX.

I.Sous-ensembles et Relations flous..... 4
A. Sous-Ensembles flous. 4
1) Qu'est-ce qu'un SEF ? 4
2) Comment définir un SEF ?..... 4
3) Opérations sur les SEF 7
4) SE vulgaire de niveau alpha..... 8
B. Relations floues..... 8
1) Généralisation aux relations n-aires..... 8
2) Définition..... 8
3) Opérations sur les relations floues. 9
4) Deux autres concepts importants..... 10
II. Théorie des Possibilités..... 12
A. Notion de Restriction Floue..... 12
1) Définition..... 12
2) Exemple..... 12
3) Pourquoi parle-t-on de contrainte élastique ? 13
4) Note..... 13
B. Distribution de possibilité : définition..... 13
C. Exemple..... 14
D. Propriétés..... 14
E. Relations n-aires et cas particulier du produit cartésien..... 15
• Extension de la définition aux relations n-aires. 15
• Cas particulier du produit cartésien..... 15
F.Possibilité versus probabilité..... 15
G. Distributions de possibilité marginale et conditionelle. 16
1) distribution marginale de possibilité. 16
2) distribution de possibilité conditionelle..... 17
III. Variables Linguistiques..... 19
1) Définitions. 19
2) Exemple..... 19
3) Termes primaires et regles sémantiques..... 19
4) Approximation linguistique..... 20

PARTIE 2 : METHODES UTILISANT LES SEF ET RELATIONS FLOUES.

I. Diagnostic par comparaison de matrices..... 21
A. Introduction..... 21
B. Les méthodes..... 22
1) Un début..... 22
2) Un modèle raffiné..... 22
3) Une adaptation moins heureuse..... 26
C. Les résultats..... 27
D. Discussion..... 28
E. Conclusion..... 30
II. Diagnostic par composition de relations..... 31
A. Introduction..... 31
B. Le modèle..... 31
C. Les résultats..... 32
D. Discussion..... 32
E. Conclusion..... 35
III. "Variables linguistiques" et discrimination..... 36
A. Introduction..... 36
B. La méthode..... 36
1. Quantification des variables linguistiques..... 36
2. Sélection des variables les plus discriminantes..... 39
3. Classement des sujets..... 40
C. Les résultats..... 40
D. Discussion..... 41
IV. Discrimination et connectivité floue..... 42
A. Introduction..... 42
B. La méthode..... 42
1. Approche discriminative..... 42
2. Connectivité..... 44
C. Les résultats..... 49
D. Discussion..... 49

PARTIE 3 : RAISONNEMENT APPROXIMATIF ET SYSTEMES EXPERTS.

I. Fuzzy reasoning : éléments de base (objectif 1). 53
A. Introduction. 53
B. Les principes du raisonnement approximatif. 54
1) Les règles de traduction. 54
2) Le mécanisme d'inférence 58
II. Les problèmes du raisonnement approximatif (objectif 2) 65
A. Les fondements de la théorie. 65
B. Discussion des opérateurs logiques. 66
C. Les surprises du raisonnement flou. 69
D. Conclusion. 76
III. Le flou dans les systèmes experts (objectif 3). 77
A. Les espérances. 77
1) Représenter des faits et des règles 77
2) Réaliser des inférences. 78
3) Exemple. 79
4) Raffinements 80
5) Conclusions 82
B. Quelques réalisations. 83
1) MYCIN 83
2) FRIL (ARIES) et les supports logiques. 84
3) LIFTAN 86
4) Analyse d'images echocardiographiques 87
C. Conclusion. 90
Tableau comparatif des possibilités offertes par les SE 91

CONCLUSION.

Un schéma de diagnostic. 92
Comment se comportent les méthodes floues ? 95
A. Méthodes non-floues 95
B. Comparaison matricielle. 95
C. Composition de relations. 96
D. Variables linguistiques. 96
E. Connectivité et discrimination floues. 97
Et la théorie des possibilités ? 98
En conclusion 100

INTRODUCTION

A. SOUS-ENSEMBLES FLOUS ET DIAGNOSTIC.

Quel peut donc bien être le rapport entre la théorie des sous-ensembles flous (SEF) et le diagnostic médical ?

Il y a là, en fait, deux sous-questions :

- 1) quelles sont les idées principales de cette théorie ? et
- 2) en quoi le diagnostic médical est-il un problème flou ?

• 1) • Un SEF se distingue d'un ensemble classique par le fait que sa frontière n'est pas tranchée. Un élément peut donc faire plus qu'appartenir ou non à un ensemble : il peut maintenant lui appartenir à des degrés divers, compris dans l'intervalle [0, 1].

De la même façon, on décrit des relations floues n-aires, où les n éléments peuvent maintenant être en relation entre eux avec un degré compris entre 0 et 1.

La théorie des SEF ne s'arrête pas là et propose bien d'autres choses ; nous aurons l'occasion de parler de distributions de possibilité, logique floue...etc.

Mais l'idée qu'il faut en retenir est qu'elle nous propose de sortir d'une façon binaire de voir les choses en nuançant des faits, des relations, des raisonnements...Elle veut donc nous permettre de manipuler des choses fondamentalement imprécises, entendons par là non-tranchées, non-totalement définies.

Selon cette façon de voir, l'imprécision nous apparaît différente de l'incertitude, cette dernière n'étant pas liée au fait que les choses ne sont pas nettes mais plutôt au fait que nous n'en sommes pas sûrs. Pour prendre un exemple un peu caricatural, on peut éprouver des difficultés à classer un végétal dans la classe des arbustes ou dans celle des arbres pour deux raisons :

- mon objet et la définition des deux classes sont tels que je ne sais pas où je dois le mettre : il s'agit plutôt d'imprécision.
- Je ne suis pas sûr, parce que je le vois mal ou que la description qu'on m'en fait est incertaine et/ou incomplète, de la classe où je dois mettre mon objet.

Bien sûr la distinction entre ces deux aspects n'est pas aussi nette mais on verra que, dans le domaine du diagnostic médical, elle est d'une certaine importance.

• 2) • Ceci nous introduit dans le second point : en quoi un diagnostic médical est-il un problème flou ?

- Tout d'abord dans le recueil des données :

Qu'il s'agisse des éléments de l'anamnèse, des plaintes, des symptômes , des tests de laboratoire ou des examens complémentaires, on ne peut pratiquement rien modéliser de façon binaire et rigoureuse.

Pensons simplement à un antécédent de pathologie gastrique (laquelle? avec quelle gravité? avec quelle fréquence?) ou une plainte de douleur (de quel type? intensité? localisation ? circonstances d'apparition? irradiations?).

- Dans la définition des maladies :

La définition d'une pathologie peut être vue comme une association entre des symptômes et une maladie. Cependant, cette relation est loin d'être simple; non seulement elle possède de nombreuses facettes (intensité des symptômes, fréquence des symptômes...etc) mais elle apparaît également dans une certaine mesure mal définissable : variation des tableaux cliniques et des individus, absence de frontière nette entre le normal et le pathologique, querelles d'écoles, évolution des théories...

- Le processus de diagnostic lui- même nous semble mal défini, imprécis; complètement à l'opposé d'un organigramme de décision.

On peut donc espérer que l'introduction de la théorie des SEF permettra une meilleure modélisation et/ou manipulation des données dans les méthodes d'aide au diagnostic.

B. PLAN DU TRAVAIL.

De multiples modèles ont été proposés depuis maintenant 20 ans, depuis que ZADEH, en 1965, jetait les bases de sa théorie.

On peut schématiquement les classer en deux catégories :

- des méthodes reposant essentiellement sur les notions de SEF et de relations floues, et qui sont tantôt des adaptations de méthodes classiques connues, tantôt des modèles plus "révolutionnaires" dans leur façon d'aborder le problème.
- Des méthodes, beaucoup plus récentes, basées sur la théorie des possibilités et le raisonnement approximatif, et s'inscrivant dans une perspective de systèmes experts.

Nous nous proposons de voir, dans ce travail, quels sont leurs apports dans la modélisation du diagnostic médical.

Nous aurons donc trois grandes parties :

- 1) Concepts généraux :

où nous verrons le résumé de ce qu'il faut savoir pour aborder les deux parties suivantes.

- 2) Méthodes floues :

où nous aborderons les méthodes du premier type, reposant sur les SEF et les relations floues.

- 3) Raisonnement approximatif et systèmes experts :

où nous verrons celles du second type, basées sur la théorie des possibilités.

Enfin, une conclusion nous permettra de résumer,à la lumière d'un schéma de diagnostic, les points forts et les points faibles des différents modèles et, en conséquence, de la théorie des SEF elle-même.

CONCEPTS GENERAUX

I.Sous-ensembles et Relations flous.

Références : [21],[22],[23].

NOTE : Un SEF est un cas particulier de relation floue puisqu'il s'agit d'une relation floue unaire. Il est cependant plus facile d'exposer certaines notions dans le contexte de SEF et de généraliser ensuite aux relations.

A. SOUS-ENSEMBLES FLOUS.

1) QU'EST-CE QU'UN SEF ?

• En mathématiques "classiques",un ensemble possède une frontière bien délimitée. Si E est un tel ensemble (appellé *ensemble vulgaire* ou *ensemble naïf* ou encore *crisp set* en anglais), un élément x soit appartient soit n'appartient pas à E. En termes de fonction d'appartenance, notée $\mu_E(x)$,on a que $\mu_E(x) = 1$ si x appartient à E et $\mu_E(x) = 0$ si x n'appartient pas à E.

• Dans la théorie des SEF, la frontière n'est pas nette et un élément u peut voir sa fonction d'appartenance à un sous-ensemble flou F, notée $\mu_F(u)$, varier de 0 à 1. On appelle *univers du discours*, l'ensemble des valeurs que u peut prendre dans un contexte déterminé : il constitue le *référentiel* à l'intérieur duquel on définit le SEF, en donnant, pour tout u de l'univers du discours, la valeur de $\mu_F(u)$.

2) COMMENT DÉFINIR UN SEF ?

On donne donc pour ce SEF, son univers du discours et sa fonction d'appartenance. On omettra parfois de préciser le premier lorsqu'il est évident.

On peut distinguer plusieurs cas dans l'énoncé de la fonction d'appartenance.

• La fonction d'appartenance est continue :

* On peut alors tout d'abord définir la fonction comme on le désire, pourvu qu'elle donne des valeurs de l'intervalle [0,1] ; par exemple , ∀ u de l'univers du discours U défini comme étant l'intervalle [0,1] :

$$\mu_F(u) = 1 - (u^2 / 3),$$

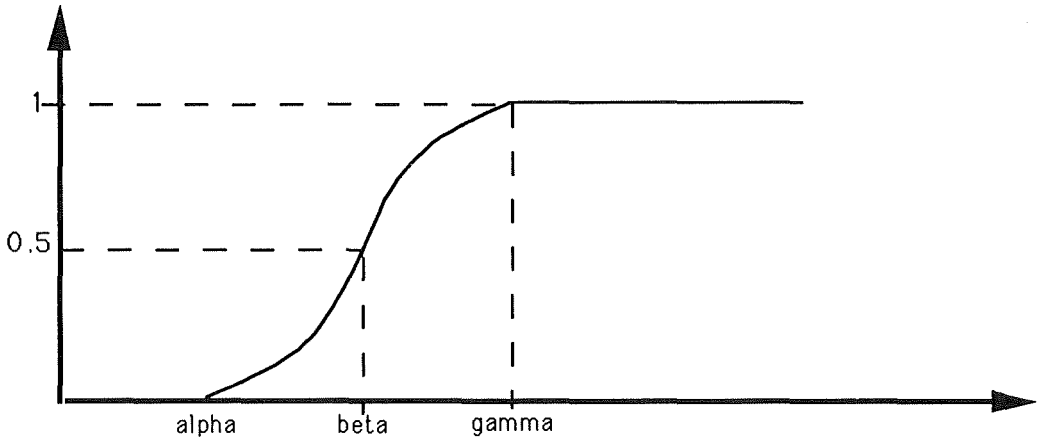
* On peut également utiliser des fonctions "standards", qui varient selon les paramètres choisis.

- La plus courante est S(u; α,β,γ), qui vaut :

$$\begin{cases} 0 & \text{pour } u <= \alpha. \\ 2 * ((u-\alpha) / (\gamma-\alpha))^2 & \text{pour } \alpha <= u <= \beta \\ 1 - 2 * ((u-\gamma) / (\gamma-\alpha))^2 & \text{pour } \beta <= u <= \gamma \\ 1 & \text{pour } u >= \gamma \end{cases}$$

$\beta = (\alpha + \gamma) / 2$ est le cross-over point càd où S(u) = 0.5.

Elle a l'aspect suivant :

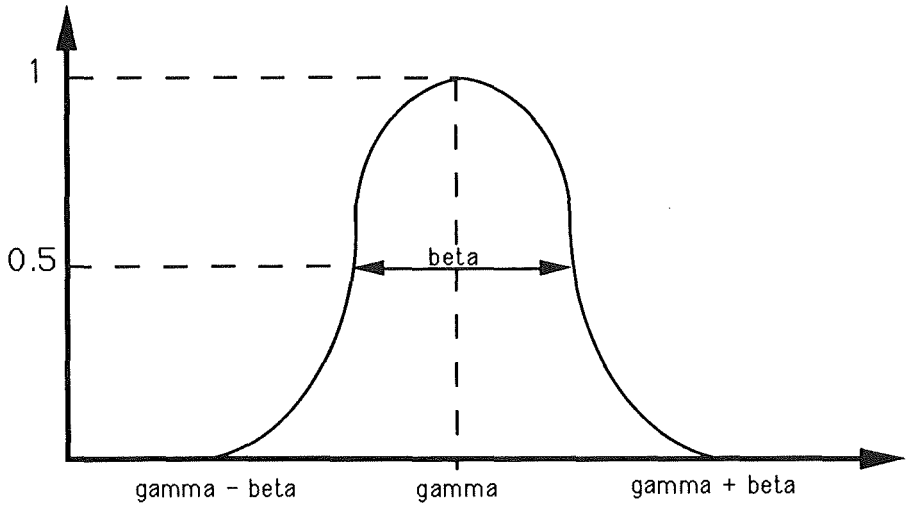


- On a également Π(u;β,γ), qui vaut :

$$S(u; \gamma-\beta, \gamma - \beta/2, \gamma) \qquad \text{pour } u <= \gamma$$

$$1 - S(u; \gamma, \gamma + \beta/2, \gamma+\beta) \qquad \text{pour } u >= \gamma$$

β est cette fois la largeur de bande càd la "distance" entre les 2 points de cross-over et γ est le point où π vaut 1, sur une courbe dont l'aspect est le suivant :



• **La fonction d'appartenance n'est pas continue :**

* On énumère alors la suite des éléments u et de leur valeur pour la fonction d'appartenance $\mu_F(u)$, en sous-entendant ceux pour lesquels $\mu_F(u) = 0$ ou 1 et peut être déduite des valeurs que l'on connaît déjà. On retrouve deux variantes : énumération des singletons $\mu_F(u) / u$ ou alors, des couples $(u, \mu_F(u))$.Dans la définition de F, SEF des "petits entiers", on aurait, par exemple :

- **F** = 1/0 + 1/1 + 0.8/2 + 0.6/3 + 0.3 / 4 + 0.1/5 où + dénote l'union et non la somme arithmétique.
- **F** = { (0,1) , (1,1) , (2,0.8) , (3,0.6) , (4,0.3) , (5,0.1).}.

Poursuivant cette idée,on notera parfois $\mathbf{F} = \int_U \mu_F(u)/u$, pour symboliser qu'il est l'union de tous les singletons $\mu_F(u) / u$ de l'univers du discours U.

On remarquera qu'on n'a pas noté les entiers >= 6, pour lesquels on sous-entend logiquement que $\mu_F(u) = 0$.

* On peut également représenter le SEF sous forme vectorielle, assez facile à visualiser :

0	1	2	3	4	5
1	1	0.8	0.6	0.3	0.1

3) OPÉRATIONS SUR LES SEF .

Soient **F** :

a	b	c
1	0.5	0.6

et **G** :

a	b	c
0.2	0.6	0.3

deux SEF flous définis sur U.

On définit les opérations suivantes :

• UNION = $\mathbf{F} \cup \mathbf{G} = \int_U \max(\mu_F(u), \mu_G(u)) / u$

ce qui nous donne sur l'exemple :

a	b	c
1	0.6	0.6

• INTERSECTION = $\mathbf{F} \cap \mathbf{G} = \int_U \min(\mu_F(u), \mu_G(u)) / u$

ce qui nous donne sur l'exemple :

a	b	c
0.2	0.5	0.3

• SOMME BORNEE = $\mathbf{F} \oplus \mathbf{G} = \int_U \min [1 , (\mu_F(u) + \mu_G(u))] / u$ où + est la somme arithmétique ce qui nous donne sur l'exemple :

a	b	c
1	1	0.9

• DIFFERENCE BORNEE = $\mathbf{F} \ominus \mathbf{G} = \int_U \max [0 , (\mu_F(u) - \mu_G(u))] / u$

ce qui nous donne sur l'exemple :

a	b	c
0.8	0	0.3

• COMPLEMENT DE F = F' = $\int_U (1 - \mu_F(u)) / u$

ce qui nous donne sur l'exemple :

a	b	c
0	0.5	0.4

• PRODUIT CARTESIEN DE F ET G :

Si F et G sont deux SEF définis respectivement sur U et V, alors leur produit cartésien F x G est défini sur U x V et par $\mu_{FxG}(u,v) = \min (\mu_F(u) , \mu_G(v))$.

On obtient donc une relation floue binaire ; nous parlerons des relations floues dans le point suivant.

4) SE VULGAIRE DE NIVEAU ALPHA.

Soient un SEF F et une valeur α comprise entre 0 et 1, le SE vulgaire de niveau α et noté F_α est défini par :

$F_\alpha = \{ u \in U \text{ tq } \mu_F(u) >= \alpha \}$.

Si nous reprenons notre exemple, $F_{0.6} = \{a,c\}$.

B. RELATIONS FLOUES.

1) GÉNÉRALISATION AUX RELATIONS N-AIRES.

Une relation floue n-aire est un SEF du produit cartésien $U_1 x...x U_n$, et caractérisée par une fonction d'appartenance : $\mu_R(u_1,...,u_n) \in [0,1]$.

2) DÉFINITION.

On définit les relations de la même façon que ce qui a été exposé pour les SEF.

On utilisera entre autres fréquemment la notation sous forme de tableau dans le cas de relations binaires.On peut donner pour exemple le résultat du produit cartésien de F et G, vu au point précédent :

	a	b	c
a	0.2	0.6	0.3
b	0.2	0.5	0.3
c	0.2	0.6	0.3

3) OPÉRATIONS SUR LES RELATIONS FLOUES.

• On retrouve celles qui ont été décrites dans les SEF.

• On y ajoutera **la composition Max-min**, que nous noterons **o**, et définie par :
Soient R_1 définie sur X x Y et R_2 définie sur Y x Z, la composition de R_1 et R_2 détermine une nouvelle relation floue qui est définie par la fonction d'appartenance suivante :

$$\mu_{R1 \text{ o } R2}(x,z) = \text{Max}_{\text{sur tous les y}} (\mu_{R1}(x,y) , \mu_{R2}(y,z)).$$

Ce qui nous donne sur un exemple :
- soit R_1 = :

	y_1	y_2
x_1	0.2	1
x_2	0.4	0.6

- soit R_2 = :

	z_1	z_2
y_1	0.3	0.7
y_2	0.5	0.8

- alors, R_1 o R_2 =

	z_1	z_2
x_1	0.5	0.8
x_2	0.5	0.6

Il est à noter que si R_1est un SEF (soit **A**), la relation R_1 o R_2 sera un SEF également : appelons-le **B**. On dira alors que **B** est un **SEF conditionné** par **A**.

4) DEUX AUTRES CONCEPTS IMPORTANTS.

4.1 : La Projection.

Soit **R** une relation floue n-aire définie sur $U_1 x ... x U_n$ et de fonction d'appartenance $\mu_R(u_1,...,u_n)$.

Soit q une sous-séquence $(i_1,...,i_k)$ de (1,...,n) et soit s = $(j_1,...,j_l)$ sa sous-séquence complémentaire, comme dans l'exemple suivant :

(1,...,n) = (1,2,3,4,5)

q = (1,3)

s = (2,4,5).

La projection de **R** sur $U_{i1} x ... x U_{ik}$

$$= \text{PROJ}_{Ui1\ x...x\ Uik}\ R$$

$$= \int_{Ui1\ x...x\ Uik} \max_{\text{sur tous les uj1,...,ujl}} \mu_R(u_1,...,u_n) / (u_1,...,u_n)$$

$$= \int_{U(q)} \max_{\text{sur tous les u(s)}} \mu_R(u_1,...,u_n) / (u_1,...,u_n)$$

Voyons cela sur un exemple, avec la relation **R** suivante :

	a	b	c
x	0.2	0.7	0.5
y	0.6	0.4	0.8
z	0.1	0.3	1

Dans ce cas, U_1 = {a,b,c} et U_2 = {x,y,z}.

Prenons q = (1) et s = (2) ; alors :

$\text{PROJ}_{U1}\ \mathbf{R} = \text{Max}_{u(s)}\ \mu_R(u_1,u_2)$

$\text{PROJ}_{U1}\ \mathbf{R} = \text{Max}_{u \in U2}\ \mu_R(u_1,u_2)$

$\text{PROJ}_{U1}\ \mathbf{R}$ =

a	b	c
0.6	0.7	1

De même,PROJ $_{U2}$ **R** = Max $_{u \in U1}$ $\mu_R(u_1,u_2)$,càd :

x	0.7
y	0.8
z	1

4.2 L'extension cylindrique.

* Si **R** est une relation floue de Ui1 x...x Uik, alors son extension cylindrique définie sur U_1 x...x U_n (avec $(i_1,...,i_k)$ sous-séquence de (1,...,n)) est une relation, définie par :

$$\mathbf{R^*} = \int_{U1 \times ... \times Un} \mu_R(u_{i1},...,u_{ik}) / (u_1,...,u_n).$$

C'est-à-dire que si **R** est une relation unaire sur U_1 :

a	b	c
0.6	0.7	1

alors, **R***, définie sur U_1 x U_2, est égale à :

	a	b	c
x	0.6	0.7	1
y	0.6	0.7	1
z	0.6	0.7	1

* Une autre façon de présenter les choses est de dire que :

$$\mathbf{R^*} = \mathbf{R} \times U_{j1} \times ..... \times U_{jl}$$
$$\text{avec } \mu_{R^*}(u_1,...,u_n) = \mu_R(u_{i1},...,u_{ik})$$

Il en résulte l'importante conséquence suivante : si **F** et **G** sont relations floues, alors leur produit cartésien **F x G** est égal à **F*** $\cap$ **G***. On peut le vérifier aisément.

II. Théorie des Possibilités.

Références : [23],[1],[3],[7]

A. NOTION DE RESTRICTION FLOUE.

1) DÉFINITION.

* **Intuitivement** : Une restriction floue est une contrainte élastique sur les valeurs que peut prendre une variable floue.

* **Formellement**, on la définit come suit :

• Soit **p** une proposition de la forme **p** = X is F où F est un SEF.

En raison du caractère flou de **F**, on ne peut pas interpréter la proposition comme l'affirmation de l'appartenance de X à **F** mais bien comme une restriction sur les valeurs que peut prendre X (ou un attribut A(X) de X).

• On dit que la signification de **p** = X is F est exprimée par **R**(A(X)) = F, où :

- F est un SEF
- **R** est la restriction floue
- X est la variable floue
- A(X) est un attribut explicite ou implicite de X : dans ce dernier cas, il est déterminé par **F** et X.

$$\mathbf{p} = X \text{ is } \mathbf{F} \Rightarrow \mathbf{R}(A(X)) = \mathbf{F}$$

• Cette définition **R**(A(X)) = **F** est souvent reprise sous le nom de *Equation d'assignation relationelle*(*Relational Assignment Equation* en anglais), ce qui signifie qu'on assigne la relation **F** à la restriction floue sur X.

2) EXEMPLE.

Soit la proposition **p** = John is Young, où :

- X = John
- **F** = Young
- A(X) = Age(John), où Age est un attribut implicite de John déterminé par John et Young

alors, **p** est exprimée par **R**(Age(John)) = Young.

3) POURQUOI PARLE-T-ON DE CONTRAINTE ÉLASTIQUE ?

On a un SEF, défini par sa fonction d'appartenance : considérons toujours l'exemple "Young", et on affirme que "John is Young".

Etant donné cette affirmation, on comprend aisément que l'âge de John ne peut plus être quelconque : il y a une contrainte sur les valeurs qu'il peut prendre.

D'autre part, vu le caractère flou de "Young", cette contrainte n'est pas absolue.

Si on avait envisagé le cas de la proposition "X est un entier de l'intervalle [1,3]",où "intervalle [1,3]" n'est pas un concept flou,la contrainte aurait été stricte : X peut prendre les valeurs 1,2 ou 3 et rien d'autre.

Dans notre cas, les choses ne sont pas aussi tranchées et il y a toute une gradation de possibilités dans les valeurs. C'est pourquoi on parle de contrainte élastique.

Ainsi, si la définition de Young est telle que μ_{young} (28) = 0.7, dire que **R**(Age(John)) = Young implique que le degré de compatibilité de 28 avec la restriction floue **R** est égal à μ_{young} (28), est égal à 0.7.

4) NOTE.

On a présenté la définition dans le cas où **F** est un SEF mais elle est valable pour toute relation floue n-aire.

Exemple binaire : **R**($A_1(X)$, $A_2(X)$) = **R**.

B. DISTRIBUTION DE POSSIBILITÉ : DÉFINITION.

• **Soit** U l'univers du discours dans lequel une variable X prend ses valeurs, **et soit F** un SEF de cet univers du discours, caractérisé par une fonction d'appartenance $\mu_F(u)$, et agissant comme restriction floue ,**R**(X) sur X

Alors, la proposition **p** = X is **F** (qui est traduite par **R**(X) = **F**, selon ce que nous avons vu dans les restrictions floues) induit une distribution de possibilité $\prod_X$, qu'on pose égale à **R**(X).

$$\mathbf{p} = X \text{ is } \mathbf{F} \Rightarrow \prod_X = \mathbf{R}(X)$$

• **De la même façon,** la fonction de distribution de possibilité est notée π_X et est définie comme numériquement égale à la fonction d'appartenance μ_F de **F**. Et donc,

$$\pi_X(u) = \mu_F(u).$$

• Avec l'équation d'assignation relationelle **R(X) = F**, on peut donc écrire ce que l'on nomme l'*équation d'assignation de possibilité* (*possibility assignment equation*) càd $\Pi_X = \mathbf{F}$, et donc :

$$\mathbf{p} = \mathbf{X \text{ is } F} \Rightarrow \Pi_X = \mathbf{F}$$

C. EXEMPLE.

Il faut comprendre la signification de ces définitions comme suit :

* Soit **F** le SEF des "petits entiers", défini par :
F = { (1,1) , (2,1) , (3,0.8) , (4,0.6) , (5,0.4) , (6,0.2) }
Si nous prenons par exemple le terme (3,0.8), il signifie que le degré de compatibilité de 3 avec l'étiquette "petit entier" est 0.8.

* Soit maintenant la proposition **p** = X is "petit entier"; nous savons qu'elle induit $\Pi_X = \mathbf{F}$ = { (1,1) , (2,1) , (3,0.8) , (4,0.6) , (5,0.4) , (6,0.2) }.
Mais cette fois, le terme (3,0.8) est à comprendre comme l'expression du fait que la possibilité que X prenne la valeur 3, compte tenu de la proposition **p**, est égale à 0.8.

D.PROPRIÉTÉS.

• Nous avons donc, avec ce que nous avons vu, le moyen de traduire des propositions simples contenant un concept flou, en une distribution de possibilité. Nous verrons que ce sera, en fait, le plus souvent, la première étape de ce que l'on appelle le *raisonnement approximatif*

• Il est possible de complexifier les propositions (d'y introduire des quantifications, des modifications, des compositions...etc) et de modifier en conséquence les distributions. Nous verrons tout cela dans la partie réservée à la *logique floue* et au *raisonnement approximatif*, où nous pourrons mieux situer ces opérations dans leur contexte et ainsi, mieux en voir l'utilité.

E.RELATIONS N-AIRES ET CAS PARTICULIER DU PRODUIT CARTÉSIEN.

• EXTENSION DE LA DÉFINITION AUX RELATIONS N-AIRES.

La distributionde possibilité prend alors la forme $\prod_{(A1(X),...,An(X))} = F$ et $\pi_{(A1(X),...,An(X))}(u_1,...,u_n) = \mu_F(u_1,...,u_n)$.

• CAS PARTICULIER DU PRODUIT CARTÉSIEN.

Si F est le produit cartésien de n relations unaires $F_1,...,F_n$, alors

$$\prod_{(A1(X),...,An(X))} = \prod_{(A1(X))} x...x \prod_{(An(X))}$$

et donc

$$\pi_{(A1(X),...,An(X))}(u_1,...,u_n) = \min [\ \pi_{(A1(X))}(u_1),...,\pi_{(An(X))}(u_n)\ ].$$

F.POSSIBILITÉ VERSUS PROBABILITÉ.

Pour illustrer la différence entre possibilité et probabilité, considérons l'exemple classique où **p** = "Hans mange X oeufs au déjeuner", avec X prenant ses valeurs dans l'univers U = {1,2,3,...}.

Cette proposition induit donc $\prod_X$ et $\pi_X(u)$ exprime le degré de possibilité que Hans mange u oeufs pour déjeuner.

On pourrait d'autre part relever la distribution de probabilité en observant ce qu'il mange chaque jour.

- Intuitivement, la possibilité exprime le degré de facilité, d'aisance tandis que la probabilité exprime le degré de fréquence.

- D'autre part, le concept de possibilité n'implique aucunement l'idée de répétition d'une expérience. Il est donc non statistique, par nature.

• On peut également s'apercevoir que les distributions peuvent avoir des aspects très différents. Ce qui est possible n'est pas nécessairement probable et ce qui est improbable n'est pas nécessairement impossible.Toutefois ce qui est impossible est improbable : c'est ce qu'on nomme parfois le principe de cohérence possibilité / probabilité.

• Enfin, les règles qui gouvernent les possibilités sont différentes de celles des probabilités.
Ainsi, soit A un SEF de U et soit $\prod_X = F$ une distribution de possibilité : la mesure de possibilité de A est définie par $\prod_{(A)} = Max_{u \in U} [\min (\mu_F(u), \mu_A(u))]$.
Dans le cas particulier où A est non-flou, on a que $\prod_{(A)} = Max_{u \in U} (\mu_F(u))$.

- On ne retrouve donc pas la propriété des probabilités qui est que $P(A) = \sum_{u \in A} P(u)$.

- D'autre part, $\prod_{(A \cup B)} = Max [\prod_{(A)} , \prod_{(B)}]$. On n'a donc pas la propriété d'additivité des probabilités en cas d'ensembles disjoints : $P(A \cup B) = P(A) + P(B)$.

- Enfin, $\sum_{u \in U} \pi(u) \neq 1$, au contraire des probabilités.

G. DISTRIBUTIONS DE POSSIBILITÉ MARGINALE ET CONDITIONELLE.

1) DISTRIBUTION MARGINALE DE POSSIBILITÉ.

Soient :
- $X = (X_1,...,X_n)$ une variable floue n-aire
- $U = U_1 x....x U_n$ l'univers du discours.
- $\prod_{(X)}$ la distribution de possibilité associée à X
- q une sous-séquence $(i_1,...,i_k)$ de $(1,...,n)$
- q' la sous-séquence $(j_1,...,j_m)$ complémentaire de q
- $U(q)= U_{i1} x....x U_{ik}$
- $U(q')= U_{j1} x....x U_{jm}$
- X(q) la variable floue q-aire $(X_{i1},...,X_{ik})$

alors, la distribution marginale de possibilité $\Pi(X_q)$ est induite par $\Pi(X)$, par projection sur U(q) telle que :

$$\Pi(X_q) = \text{PROJ}_{U(q)}\ \Pi(X)$$
$$\text{avec } \pi_{(X_q)}(u_{(q)}) = \text{Max}_{U(q')}\ \pi_{(X)}(u).$$

2) DISTRIBUTION DE POSSIBILITÉ CONDITIONELLE.

• **Définition :**

Soient :

- $X = (X_1,...,X_n)$ une variable floue n-aire
- $\Pi(X)$ la distribution de possibilité associée à X
- q une sous-séquence $(i_1,...,i_k)$ de (1,...,n)
- q' la sous-séquence $(j_1,...,j_m)$ complémentaire de q
- X(q) la variable floue q-aire $(X_{i1},...,X_{ik})$
- $\Pi_X [X_{j1} = aj_1 ;....;X_{jm} = aj_m]$ une distribution de possibilité où le $j_1{}^e$ élément est aj_1 ;...; le $j_m{}^e$ élément est aj_m.

Alors,

$$\Pi_{X(q)} [X_{j1} = aj_1 ;....;X_{jm} = aj_m] = \text{PROJ}_{U(q)}\ \Pi_X [X_{j1} = aj_1 ;....;X_{jm} = aj_m]$$

• **Pour exemple :**

Donnons-nous une relation ternaire où l'univers du discours de chaque variable X_1,X_2,X_3 ne comprend que 2 éléments : a et b. On n'énumère que les triplets pour lesquels la valeur de la fonction de possibilité est ≠ de 0 :

- $\pi_X (a,a,a) = 0.8$
- $\pi_X (a,a,b) = 1$
- $\pi_X (b,a,a) = 0.6$
- $\pi_X (b,a,b) = 0.2$
- $\pi_X (b,b,b) = 0.5.$

Ainsi, si je prends $\Pi_{X(q)} [X_{j1} = aj_1 ;....;X_{jm} = aj_m] = \Pi_{(X2,X3)} [X_1 = a]$:

* j'obtiens pour $\Pi_{(X)} [X_1 = a]$:
  - $\pi_X (a,a,a) = 0.8$
  - $\pi_X (a,a,b) = 1$

* $\text{PROJ}_{U2 \times U3}\ \Pi_{(X)} [X_1 = a] = \Pi_{(X2,X3)} [X_1 = a]$:
  - $\pi_{(X2,X3)} (a,a) = 0.8$
  - $\pi_{(X2,X3)} (a,b) = 1$

• **Plus généralement :**

Plutôt que de conditionner $\prod X$ à des valeurs $X_{j1},\ldots,X_{jm}$, on peut le faire en donnant une distribution de possibilité (par exemple : soit **G** une distribution de possibilité m-aire) : on dira que $\prod X$ est *particularisé* par la spécification de $\prod X(S) = \mathbf{G}$ et on a que :

$$\prod X(q)\,[\prod X(s) = \mathbf{G}] = \text{PROJ}\ U(q)\ \prod X \cap \mathbf{G}^*$$

Ainsi, sur notre exemple, si l'on prend **G**, définie sur X_1,X_2 et telle que :

- $\pi(X1,X2)$ (a,a) = 0.4
- $\pi(X1,X2)$ (b,a) = 0.8
- $\pi(X1,X2)$ (b,b) = 1

On obtient $\mathbf{G}^*$ selon la définition donnée de l'extension cylindrique et on en déduit aisément que $\prod X \cap \mathbf{G}^*$ est tel que :

- πX (a,a,a) = 0.4
- πX (a,a,b) = 0.4
- πX (b,a,a) = 0.6
- πX (b,a,b) = 0.2
- πX (b,b,b) = 0.5.

On retire $\prod X3\,[\prod(X1,X2) = \mathbf{G}]$ par projection de la distribution précédente sur X_3 ; elle est telle que :

- $\pi X3$ (a) = 0.6
- $\pi X3$ (b) = 0.5

III. Variables Linguistiques.

Références : [3],[23].

1) DÉFINITIONS.

De façon informelle, une *variable linguistique* X est une variable dont les valeurs sont des mots ou des phrases d'un language naturel ou "artificiel".Ces valeurs sont appelées *valeurs linguistiques.*

L'ensemble des valeurs que peut prendre une telle variable est appelé *ensemble des termes* ou *term-set.*. Chaque terme est donc une valeur linguistique.

D'autre part, chaque terme est le label d'un SEF de l'univers du discours et on appelle *signification d'un terme* le SEF qui lui est associé.

2) EXEMPLE.

Soit la variable linguistique "Age", définie sur un univers du discours U = [0,100].

Elle peut prendre les valeurs "jeune", "vieux", "plus ou moins jeune", "très vieux"..

Chaque valeur est l'étiquette d'un SEF de U ; par exemple, "jeune" peut dénommer le SEF défini par (1 - S(20,30,40)).

3) TERMES PRIMAIRES ET REGLES SÉMANTIQUES.

On distingue les termes primaires, dont la signification doit être définie a priori et les termes non-primaires, dont la signification est calculée à partir des premiers et des règles sémantiques.

Dans notre exemple, les termes primaires seraient "jeune" et "vieux". Les autres peuvent être déduits : ainsi, $\mu_{\text{très jeune}} = (\mu_{\text{jeune}})^2$, déduit de "jeune" et de la règle sémantique "très".

Les règles sémantiques les plus classiquement adoptées (mais elles peuvent être redéfinies) sont, si on note par St la signification du terme t :

- S(t1 et t2) = St1 $\cap$ St2
- S(not t) = St'
- S(très t) = St^2
- S(plus ou moins t) = $\sqrt{St}$
- S(t1 ou t2) = St1 $\cup$ St2

4) APPROXIMATION LINGUISTIQUE.

On appelle approximation linguistique le fait de trouver une valeur linguistique càd un terme, un label pour un SEF donné.

Ceci se fait lorsqu'on a déduit un SEF de diverses manipulations ou inférences logiques (cfr ultra). Il faut alors lui trouver une étiquette la plus proche possible de sa signification, en accord avec les termes primaires et les règles sémantiques.

METHODES BASEES SUR LES SEF ET RELATIONS FLOUES

I. Diagnostic par comparaison de matrices.

A. INTRODUCTION.

Le principe de ce type de méthode n'est pas totalement nouveau [14] mais ce qui est différent et qui risque de changer la valeur de ces modèles, c'est le soin qui est apporté dans la modélisation des données et qui est rendu possible par la référence aux SEF.

La méthode de diagnostic par comparaison matricielle procède en deux étapes :

1) recueillir un ensemble de données se rapportant au patient et les modéliser sous forme de matrice : appelons la Mp.
2) On dispose d'autre part d'une "base de données", caractérisant chaque maladie en compétition pour le diagnostic.

Parfois, il s'agira d'une collection de cas qui se sont déjà présentés avec, pour chacun, les symptômes qui étaient présents et le diagnostic final. L'établissement du diagnostic actuel consistera à chercher les k plus proches cas dans la base de données et à choisir le diagnostic qui a été le plus fréquemment posé dans ceux-ci.[26].

Le plus souvent, la "base de données" est la collection des matrices-types des maladies en compétition et qui ont alors été construites sur base des connaissances des experts. Appelons ces matrices Mm. Poser un diagnostic reviendra alors à rechercher la ou les plus proches.

Finalement, ces deux approches s'équivalent si on accepte l'hypothèse que les avis des experts ne sont pas tout à fait arbitraires et ne changent pas trop souvent dans le temps...ce que nous espérons assez raisonnable.

La variabilité des approches se situe en fait dans :

- la représentation des données (binaires ou floues, quantitatives ou qualitatives, finesse de la représentation,...)
- et dans l'attribution éventuelle d'un poids à chaque variable d'une matrice, modélisant ainsi le fait que tous les attributs n'ont pas la même importance pour le diagnostic d'une maladie donnée.

Meilleurs seront ces deux points et meilleur sera le modèle.

B. LES MÉTHODES.

1) UN DÉBUT.

Les travaux de FORDON et BEZDEK [26] constituent une approche tout à fait basale puisqu'on admet que des données binaires ou des valeurs quantitatives et qu'il n'y a aucune pondération. Ils se servent d'une "base de données" d'anciens cas et posent le diagnostic en prenant le plus fréquent des k plus proches cas.

2) UN MODELE RAFFINÉ.

Le modèle le plus complet dans cette catégorie est certainement celui d'ESOGBUE et ELDER [27],[28],[29]. Un effort tout particulier a été apporté à la modélisation des données, rendant même l'approche un peu lourde. On y a également introduit une pondération. Voyons les principales étapes de ce modèle.

2.1 : Modéliser l'information du patient.

On recueille les informations concernant le patient dans quatre matrices H, A, S et Z, qui contiennent respectivement les antécédents, les symptômes, les signes cliniques et les résultats des examens complémentaires. On réunira ces différentes matrices pour former la matrice caractéristique du patient : Mp.

2.1.1 : Construction d'une matrice binaire H.

On construit par l'anamnèse un vecteur H caractéristique de l'historique du patient, où les h_i :

- sont tous des éléments importants pour le diagnostic d'au moins une maladie de N où N est l'ensemble des maladies en compétition pour le diagnostic
- et valent 0 ou 1 selon que l'élément ne se retrouve pas ou se retrouve dans l'histoire du patient.

Exemple : telle maladie infantile, travail en milieu toxique...etc.

2.1.2 : Construction d'une matrice floue A.

Il est important d'assigner un degré de sévérité au symptôme et de ne pas se limiter au simple fait de la présence ou de l'absence.

Ce degré n'est pas attribué directement par le médecin mais on procède en deux étapes :

1) on relève la description que fait le patient de ses problèmes
2) on réalise un mapping vers un degré de sévérité d'un symptôme.

La fonction de mapping peut prendre des formes et des degrés de complexité divers. Le plus souvent, il s'agira d'une correspondance 1-1 avec le problème signalé par le patient. Mais parfois, il faudra faire intervenir plusieurs caractéristiques du relevé de ces problèmes.

Par exemple, pour parler de *dyspnée paroxystique nocturne,* il faudra faire intervenir la plainte de *courtesse d'haleine* avec en plus des caractéristiques comme *apparaît brusquement* et *apparaît la nuit.*

2.1.3 : Construction d'une matrice floue S.

On adopte la même démarche en deux temps que pour les symptômes :

1) recherche des facteurs observables
2) mapping vers un ensemble de signes cliniques avec attribution d'un degré de sévérité.

Exemple : la sévérité d'un souffle à l'auscultation cardiaque (qui est un signe clinique) dépend à la fois de l'intensité et du type de bruit, qui sont tous deux des facteurs observables.

On obtient donc une matrice floue S des signes cliniques du patient.

2.1.4 : Construction d'une matrice floue Z.

Pour chacun des tests, on définit un SEF dont la fonction d'appartenance reflète le degré d'anormalité du test.

Par exemple, on pourrait avoir, pour un taux de cholestérol normalement inférieur à 260 :

- μ(taux) = 0 ssi le taux est inférieur à 260
- μ(taux) = (taux / 340) - (26 / 34) ssi 260 <= taux <= 600
- μ(taux) = 1 ssi taux > 600.

On réunit les résultats des différents tests dans la matrice floue Z.

2.1.5 : Rassemblement des matrices H, A, S et Z en une matrice Mp.

2.2 : Les matrices caractéristiques des maladies.

Ces matrices,appelées Mm, auront la même structure que Mp et contiendront les profils types des différentes maladies.

Pour affiner le processus, on considère qu'une maladie i peut présenter plusieurs stades. Afin de simplifier la notation, on ne parlera plus dorénavant que de maladies-stades càd une maladie à un stade précis et nous les indicerons par j.

On retrouvera donc les "sous-matrices" suivantes, dans toute matrice Mm :

2.2.1 : H(i).

Matrice binaire des antécédents typiques à cette maladie. Elle est la même pour toutes les maladies-stades d'une même maladie.

2.2.2 : A(j).

On aura cette fois une matrice donnant les bornes inférieure et supérieure de l'intervalle de degré de sévérité du symptôme typique de ce stade, càd

A(j) =

alb(j,1),.....,alb(j,t)
aub(j,1),.....,aub(j,t)

où :

- alb(j,k) est la borne inférieure du k^e symptôme de la j^e maladie
- aub(j,k) est la borne supérieure de ce même symptôme

Si le symptôme est binaire, alors alb(j,k) = aub(j,k) et appartiennent à {0,1}.

Sinon, les bornes sont différentes et aub >= alb.

2.2.3 : S(j).

Elle se réfère aux signes cliniques et est construite selon les mêmes principes que A(j); elle apparaît donc comme :

slb(j,1),.....,slb(j,f)
sub(j,1),.....,sub(j,f)

2.2.4 : Z(j).

Elle est expansible et dépendra du nombre k de tests effectués :

zlb(j,1),.....,zlb(j,k)
zub(j,1),.....,zub(j,k)

Si, pour un stade donné, le résultat du test doit être dans les limites de la normale, alors zlb = zub = 0.

Sinon, zub >= zlb et zub ≠ 0.

2.3 : Poser un diagnostic.

2.3.1 : Le principe et ses variantes.

Le processus de diagnostic consiste à comparer la matrice caractéristique du patient à celles caractéristiques des différentes maladies-stades et à voir de laquelle elle se rapproche le plus : on a ainsi la maladie à laquelle on conclut.

On peut développer quelques variantes :

- obtenir le groupe des k maladies les plus proches, k étant posé par l'utilisateur
- obtenir le groupe des maladies qui sont au-dessus d'un seuil défini par l'utilisateur.

2.3.2 : Mesure de la distance.

Pour donner une idée de la similarité entre matrices, on fait appel à la notion de mesure de distance.

Soient les deux matrices X(j) et X(n) respectivement matrice caractéristique de la maladie-stade X(j) et du patient, la distance est donnée par :

$$D^P(X(j),X(n)) = [\sum_{\text{sur k de 1 à m}} | x(j,k) - x(n,k) |^p]^{1/p}$$

avec :

- m est le nombre total d'attributs à considérer
- p >= 1; on prendra le plus souvent p = 2, ce qui nous ramène à la distance dite euclidienne.

Vu les cas différents qui peuvent se présenter pour les valeurs de x(j,k) et x(n,k), on définit les règles suivantes :

- lorsque x(j,k) et x(n,k) sont binaires, la formule est valable telle quelle. Appelons ce groupe de valeurs Cj.
- sinon, on distingue les situations suivantes :
  - xlb(j,k) <= x(n,k) <= xub(j,k) : la valeur trouvée chez le patient est dans l'intervalle caractéristique de la maladie et la distance comptabilisée doit être nulle
  - x(n,k) < xlb(j,k) : la distance devient (xlb(j,k) - x(n,k)). Soit ce groupe Fj.
  - xub(j,k) < x(n,k) : la distance est donnée par (x(n,k) - xub(j,k)). Soit ce groupe Gj.

D'autre part, on définit pour chaque attribut k d'une maladie j la pertinence w(j,k) appartenant à [0,1] de l'attribut , pour le diagnostic de cette maladie. C'est une mesure de l'importance du signe, symptôme, test ou élément du passé dans le diagnostic. Ceci nous permet d'introduire une pondération des distances.

On peut donc écrire la forme finale de la mesure de distance, en prenant p = 2 :

$$D^2(X(j),X(n)) =$$

$$[\ \sum_{\text{sur k de Cj}} \left| w(j,k) * (x(j,k) - x(n,k)) \right|^2 +$$

$$\sum_{\text{sur k de Fj}} \left| w(j,k) * (xlb(j,k) - x(n,k)) \right|^2 +$$

$$\sum_{\text{sur k de Gj}} \left| w(j,k) * (x(n,k) - xub(j,k)) \right|^2]^{1/2}.$$

3) UNE ADAPTATION MOINS HEUREUSE.

Le modèle de JUNIAN, YAOZENG et CHENGHAN [30] est un peu particulier, dans le sens où l'accent est mis sur l'incidence des symptômes dans les maladies considérées tandis que les symptômes eux-mêmes sont considérés comme binaires.

Il est développé dans le cadre du diagnostic d'un syndrome abdominal aigu.

Parcourons ce modèle dans ses grandes lignes.

• Toute maladie i est caractérisée par un ensemble SMi de symptômes (a_{ij}/x_j) où :

- x_j est le j^e symptôme de la maladie
- a_{ij} est son incidence, sa fréquence dans la maladie i.

Par exemple :

occlusion intestinale = $0.9/x_{26} + 0.9/x_{68} + ...+ 0.1/x_{55}$.

où x_{26} = constipation, x_{55} = augmentation des bruits intestinaux et x_{68} = signes iléaux à la Rx...etc.

Ces descriptions constituent les matrices-types des maladies et on associe à chacune d'elles une mesure $d_i = \sum_{sur\ j} a_{ij}$.

• Pour un patient donné, on relève l'ensemble SP des signes présents.

• Pour établir un diagnostic :

- a) pour chaque maladie possible :
  - on calcule

$$d_i' = \sum_{sur\ j} \alpha_{ij}\, a'_{ij}$$

où : - $\alpha_{ij} = 1$ si le signe est présent chez le patient et est égal à 0 sinon
- $a'_{ij} = a_{ij}$ si $\alpha_{ij} = 1$et est égal à 0 sinon.

  - on fait le rapport d'_i / d_i.

  Ceci revient donc à rendre compte de la proportion des symptômes présents, en pénalisant plus fortement l'absence des symptômes les plus fréquents donc les plus caractéristiques de la maladie.

- b) On prend le maximum des rapports (d'_i / d_i) et on sélectionne comme diagnostic la maladie i correspondante.

C. LES RÉSULTATS.

1) Premier modèle.

Comme on pouvait s'y attendre, il ne donne de résultats valables que lorsqu'on l'applique à des cas simplistes.

2) Deuxième modèle.

Le domaine d'application choisi est celui de la pathologie valvulaire cardiaque, avec des résultats corrects dans 50 % des cas.

3) Troisième modèle.

Les auteurs nous disent que sur 100 cas, tous les diagnostics ont été correctement posés par le système.

D. DISCUSSION.

1) Représentation et pondération des données.

Il apparaît essentiel, si l'on veut disposer d'un bon modèle, d'introduire des pondérations pour les différents attributs et de développer assez finement la représentation des données, ce qui est le cas dans ESOGBUE [27] [28] [29].

Sans pondération, en effet, tous les éléments sont gratifiés de la même importance, ce qui n'est manifestement pas réel.

De même, lorsque les données sont binaires [26] [30], on est forcé de passer à côté des nuances. Non seulement une erreur d'estimation dans la présence d'un symptôme aura des conséquences plus importantes mais de plus, pour des symptômes comme la fièvre, l'ictère...etc, l'intensité constitue déjà une orientation et on se demande comment réduire à un simple "oui" ou "non" des choses aussi nuancées qu'un affaiblissement des bruits abdominaux, une contracture musculaire...etc. De toute évidence, cela doit biaiser le diagnostic.

Les cas présentés dans la méthode sont caricaturaux : on peut peut-être trouver là l'explication des 100 % de réussite.

Le corollaire de la constatation qu'il est important de bien modéliser les données, c'est qu'il faut s'attendre à ce que ce soit lourd et complexe. Il suffit, pour s'en convaincre, de parcourir la méthode d'ESOGBUE, dont nous n'avons encore donné ici qu'un résumé! On peut à ce propos faire remarquer que le fait de procéder en deux étapes (relevé des données et mapping vers un degré de sévérité) alourdit le modèle et pourrait être avantageusement remplacé par une estimation du clinicien mais tout cela illustre bien qu'il n'est pas simple de gérer la nuance et l'imprécision.

2)L'information manquante.

Un symptôme ou un signe absent n'a pas la même signification qu'un symptôme ou un signe qu'on n'a pas recherché ou qu'on n'a pas à sa disposition. C'est tout le problème de l'information manquante lors de l'établissement d'un diagnostic. Passé une certaine mesure, cela handicape la certitude du diagnostic mais dans des proportions raisonnables, il faut pouvoir y faire face parce qu'il est quotidien en médecine.

On peut observer, dans cette catégorie de modèles, deux réactions différentes face à ce problème :

1) on n'imagine pas qu'il puisse manquer quelque chose : on doit disposer de toutes les données sinon le système ne pourra pas fonctionner parce qu'on compare les matrices pour chacun de leurs éléments. C'est l'approche d'ESOGBUE et on se rend bien compte que c'est inapplicable en réalité.

2) On admet qu'on peut ne pas disposer de tout et qu'il ne faut pas nécessairement tout investiguer : c'est le choix de JUNIAN et coll.[30]. Ici, le système peut fonctionner sans tout connaître mais il considère les symptômes non-recherchés comme des symptômes absents, ce qui n'est pas exact. Si on ajoute à ce modèle le choix d'une proportion (d_i' / d_i) pour le diagnostic, on peut en arriver à des cas limites problématiques comme par exemple le cas où un ou bien quelques attributs, en eux-mêmes suffisants pour affirmer une maladie, ne constituent en fait pas une proportion suffisante pour que la pathologie soit sélectionnée.

Ces deux approches sont l'une comme l'autre insatisfaisantes parce que la première impose une "standardisation" du recueil des données inapplicable en pratique (il faut de plus se rendre compte que ces problèmes vont croître presque exponentiellement avec le nombre de maladies considérées parce que chacune amènera ses propres attributs "intéressants" pour son diagnostic) et parce que la deuxième risque fort, étant donné sa stratégie, de déboucher sur des diagnostics erronés.

3) Le choix de la "fonction" de distance.

L'établissement du diagnostic selon la proportion des symptômes présents, dans la troisième méthode, pose certainement deux problèmes :

- on ne comptabilise dans la distance que l'absence de symptômes : les symptômes présents "en trop" et qui devraient signifier un éloignement du tableau-type ne sont pas pris en compte.
- De plus, une absence dans une petite entité clinique (càd peu de symptômes) risque d'avoir des conséquences plus lourdes dans la mesure ou elle risque plus de faire pencher la balance. Or il faut se rappeler dans ce contexte que l'évaluation des signes présents est binaire.

E. CONCLUSION.

La meilleure version dans ce que nous avons vu est sans conteste celle d'ESOGBUE : on a vu qu'elle tenait compte des nuances dans les données et des pondérations des attributs. Ses résultats moyens viennent du fait que les auteurs se sont attaqués à forte partie dans leur exemple et qu'ils ont en plus essayé de ne poser le diagnostic que sur base des antécédents et des plaintes du patient, dans le but de ne pas avoir à développer de trop grands vecteurs de données. Cela signe quand même d'où vient la faiblesse de cette méthode : la standardisation des inputs.

L'idéal serait d'y ajouter la possibilité qu'il manque de l'information et, ce qui va de pair, ramener la mesure de la distance à une distance relative par rapport au nombre d'attributs intervenus pour poser le diagnostic.

En bref, disons que les difficultés pratiques abondent mais que, toutes ces précautions étant prises, le principe en lui-même est bon et n'est en tous cas, pas facile à prendre en défaut sur un exemple, ce qui ne sera pas le cas pour toutes les méthodes.

II. Diagnostic par compositon de relations.

A. INTRODUCTION.

• SANCHEZ [31] nous propose de voir le problème du diagnostic en termes de composition de relations :

Si **F** est le SEF des symptômes du patient et **R** est une relation floue exprimant l'association entre symptômes et maladies, alors, **G**, le SEF des maladies du patient peut être trouvé par **G = F o R** (composition Max-min).

• Bien que s'exprimant en termes de degrés de possibilité, VILA et DELGADO [32] appliquent la même démarche. L'équivalence d'expression vient en effet de ce qu'on définit la distribution de possibilité induite par une proposition à partir du SEF contenu dans cette proposition (voir théorie des possibilités dans les concepts généraux) :

$$p = X \text{ is } R \Rightarrow \prod_X = R \text{ avec } \pi_X(u) = \mu_R(u).$$

B. LE MODELE.

Décrivons le problème en termes de relations :

- Soient S l'univers des symptômes et M l'univers des maladies.

- On dispose d'une relation binaire floue **R** sur S x M, qu'on pourrait exprimer par "X est un symptôme de I" où :
  - X prend ses valeurs dans S
  - I prend ses valeurs dans M
  - sa fonction d'appartenance $\mu_R(x,i)$ représente le degré de compatibilité de (x,i) avec la relation **R**.

  Elle a été élaborée sur base, par exemple, de l'avis des experts.

- Soit F le SEF, défini sur S, des symptômes du patient où $\mu_F(x)$ exprime le degré de compatibilité du symptôme x avec le label "symptôme du patient". Ce SEF est construit lors de l'observation du patient : le clinicien attribue, selon son estimation, un degré de sévérité au symptôme.

- On en déduit G, le SEF defini sur M et tel que $\mu_G(i)$ indique le degré de compatibilité de la maladie i avec l'étiquette "maladie du patient". Il est conditionné par F et on l'obtient par composition Max-min : $G = F \circ R$.

C.LES RÉSULTATS.

Seul DELGADO nous présente un exemple, relevant du domaine neurologique. Le résultat semble bon puisque, dans tous les cas, le diagnostic posé par un clinicien figure parmi les maladies présentant le degré maximal de vraisemblance.

D. DISCUSSION.

1. UN EXEMPLE POUSSÉ À L'EXTREME.

Il faut tout d'abord se rendre compte que les maladies en compétition ont des tableaux cliniques fort différents les uns des autres.

On peut d'ailleurs le constater en examinant les tables exprimant les relations : les colonnes ont des "profils" fort différents et on trouve presque toujours un plusieurs symptômes fortement représentatifs : si ce symptôme est "suffisament" présent,on ne pourra manquer d'évoquer le diagnostic.

2. ET POURTANT UN MANQUE D'EFFICACITÉ.

D'autre part, si le diagnostic réel figure à chaque fois parmi le groupe de maladies sélectionnées comme les plus vraisemblables, il faut préciser que ce groupe comporte le plus souvent 4, 5 ou même 6 maladies possédant ce même degré de possibilité...sur un total de 10 maladies à différencier!

3. LE PRINCIPE DE BASE APPARAIT DISCUTABLE.

La faiblesse majeure de ce moyen de diagnostic semble bien résider dans la façon dont on calcule le degré de compatibilité de chaque maladie m_i : une composition Max-min.

On prend, en effet, pour chaque symptôme, le minimum de deux choses :

- le degré de liaison du symptôme à la maladie
- le degré de présence du symptôme chez le patient.

Puis, on prend le maximum sur tous les symptômes qu'on envisage.

D'un sens, cela paraît logique.Tout d'abord il est normal de prendre le minimum car rien ne sert d'avoir un symptôme évident s'il n'a rien à voir avec la maladie et inversémént : peu importe qu'un symptôme soit très fortement associé à une maladie s'il n'est pas présent. Ensuite, prendre le maximum est assez compréhensible également : si deux signes sont présents et évoquent la maladie, on conclut à la maladie avec un degré de possibilité relevant du signe le plus "fort".

D'un autre côté, si ce procédé peut marcher pour des "tableaux typiques", des maladies qu'on diagnostique sur un critère spécifique et toujours présent de façon marquée, il défavorise par contre les diagnostics qu'on est amené à poser sur base d'un ensemble de symptômes peut-être peu marqués mais dont le nombre et la "convergence" vers une maladie emportent la conviction.

Pour prendre un exemple caricatural, supposons :

- la relation suivante entre symptômes et maladies:

	M1	M2
S1	0.5	1
S2	0.5	0
S3	0.5	0

- avec un degré de présence des symptômes chez un patient :

S1	S2	S3
0.4	0.4	0.4

Les deux maladies seront placées à égalité de vraisemblance puisque :

$$\text{Max } [\min(1, 0.4), \min(0, 0.4), \min(0, 0.4)] = 0.4$$

et

$$\text{Max } [\min(0.5, 0.4), \min(0.5, 0.4), \min(0.5, 0.4)] = 0.4$$

Pourtant, le tableau symptomatologique évoque manifestement plus la première.

De plus, on n'atteint dans ce cas qu'un degré de possibilité égal à 0.4, ce qui est trop peu pour M1.

Ces problèmes seraient évités dans la méthode de comparaison matricielle utilisant un critère de distance.

4. DES PROBLEMES DE GÉNÉRALISATION.

Enfin, on est resté ici dans un domaine très restreint de la pathologie. Si on veut passer à un système d'aide au diagnostic couvrant un nombre plus grand de maladies, on risque, devant l'acroissement parallèle du nombre de symptômes à considérer, d'atteindre des tailles de matrices assez élevées.

D'autant plus que, comme dans le cas de la méthode précédente, on va se trouver confronté au problème de l'information manquante :

- ou bien on accepte de considérer comme absentes des informations dont on ne dispose pas et on biaise le diagnostic
- ou bien on ne l'accepte pas et il faut à tout prix disposer de toute l'information pour que le système tourne.

E. CONCLUSION.

On retrouve tous les problèmes d'application pratique de la méthode précédente avec en plus, un principe fondamental qui nous apparaît beaucoup plus mauvais et qui nous amène rapidement, même dans des cas simplistes, à des diagnostics peu discriminatifs voire erronés.

Il semble donc difficile d'envisager son application en réalité.

Il semble cependant que le système CADIAG [33] repose sur ce principe, bien que compliquant le raisonnement par la prise en compte de multiples relations (de symptôme à symptôme, de groupe de symptômes à maladie, de maladie à maladie,...etc). Ceci l'amènerait à diagnostiquer certaines maladies rhumatismales et pancréatiques avec succès: la question reste ouverte.

III. "Variables linguistiques" et discrimination.

A. INTRODUCTION.

Les travaux de SAITTA et TORASSO [34], sur la maladie coronarienne et de KRUSINSKA et LIEBHART [35], sur les pathologies respiratoires, veulent répondre au même problème, qui est le suivant.

Etant donné une pathologie où interviennent plusieurs variables qualitatives, comment utiliser celles-ci de façon à :

1) mettre en évidence les variables ou les combinaisons de variables qui différencient le mieux les malades des sujets sains. Ceci constitue déjà un enseignement en soi.

Ensuite, utiliser les résultats du point 1) pour :

2) classer un sujet dans l'une des deux catégories.

B. LA MÉTHODE.

1. QUANTIFICATION DES VARIABLES LINGUISTIQUES.[1]

On se reportera, pour la nomenclature, au schéma de la page suivante.

• Soient L_i, les variables linguistiques (i = 1,...,r), chacune étant décrite par son ensemble $Q^{(i)}$ de questions. On note :

- M_i le nombre de questions de $Q^{(i)}$
- $q_j^{(i)}$ la j^e question de $Q^{(i)}$, avec j = 1,...M_i.

Pour toute question $q_j^{(i)}$, il existe un ensemble de réponses possibles. On note :

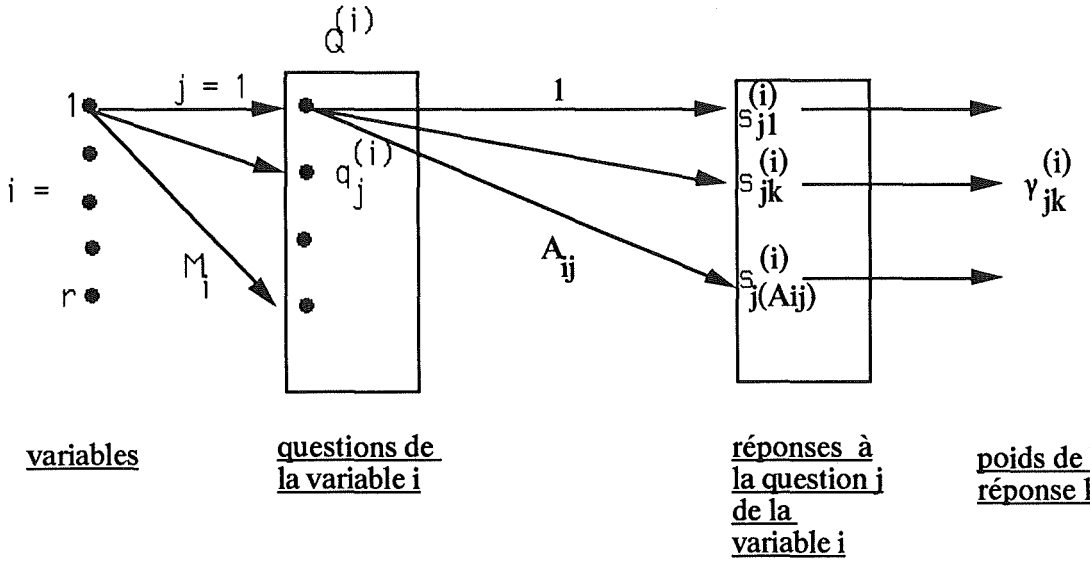
- A_{ij} le nombre de réponses possibles
- $s_{jk}^{(i)}$ la k^e réponse possible de la question j de la variable linguistique i.

---

[1]On notera qu'il ne s'agit pas de variables linguistiques au sens strict du terme, comme nous l'avons défini dans la partie réservée aux concepts généraux.

Nous les avions décrites comme des variables dont les valeurs sont des SEF étiquettés par un mot ou une phrase en langage naturel.

Il ne s'agit ici que de variables non quantitatives, d'où leur nom de "linguistique", tel que par exemple : un caractère obsessionel.



On associe à toute réponse possible $s_{jk}^{(i)}$ un poids $\gamma_{jk}^{(i)}$ appartenant à [-1, 1].

- Un poids de -1 signifie que la réponse ne va pas dans le sens de la variable linguistique.
- Un poids de +1 signifie le contraire.
- Un poids de 0 signifie qu'aucune information n'est apportée par la réponse; ce sera également le cas lorsqu'une réponse est manquante.
- Tous les intermédiaires sont bien sûr possibles.

Tous ces poids sont fixés par les experts.

• Soit maintenant le vecteur :

$$\alpha^{(i)} = (\alpha_1^{(i)},.....,\alpha_{Mi}^{(i)}),$$

vecteur des poids $\alpha_j^{(i)}$, se rapportant aux différentes questions de la variable linguistique i.

On peut le construire de différentes façons :

- parfois, on fera appel aux experts
- ici, on a décidé d'attribuer aux questions un poids proportionnel au pouvoir qu'elles ont de différencier les sujets du groupe contrôle, des malades (test de Student).

• Le poids total d'une réponse $s_{jk}^{(i)}$ est donc :

$$w_{jk}^{(i)} = \alpha_j^{(i)} * \gamma_{jk}^{(i)}.$$

• On définit pour chaque sujet l (l = 1,..,n où n est le nombre total de patients de tous les groupes rassemblés), les variables m_{il} correspondant à chaque variable linguistique :

$$m_{il} = \sum_{\text{sur j de 1 à Mi}} w_{jk*}^{(i)}$$

où $w_{jk*}^{(i)}$ est le poids total de la réponse donnée par le patient à la question j de la variable linguistique i.

Cette variable m_{il} est donc le poids calculé à partir des réponses données pour la variable linguistique i, par le patient l.

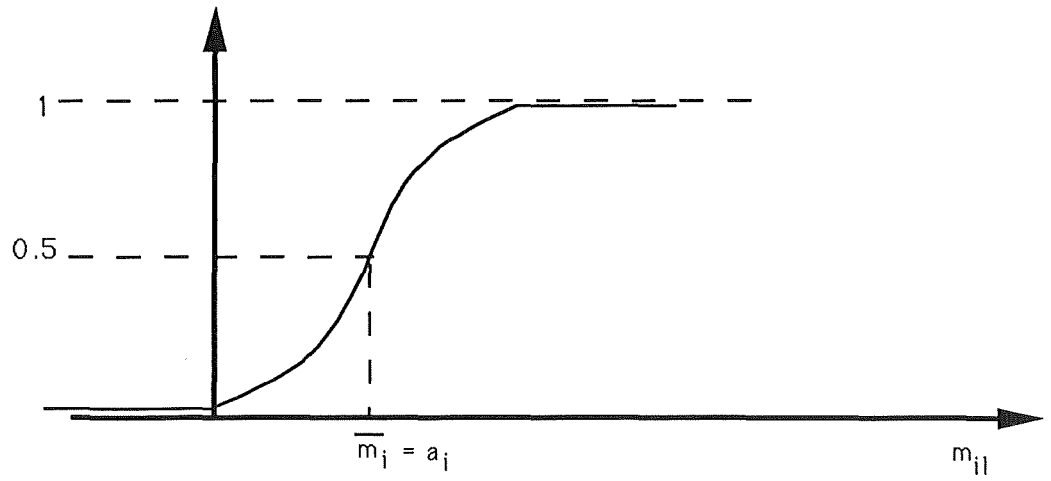
• De façon à mieux "visualiser" la signification de cette variable m_{il}, on définit sa fonction d'appartenance, qu'on peut considérer comme donnant le degré de compatibilité du patient l avec la variable linguistique i.

$$\mu_{il}(m_{il}) = 1/2 + 1/\pi * \text{arctg}((m_{il} - a_i) / b_i) \text{ avec } i = 1,...,r \text{ et } l = 1,...,n$$

où:

- $a_i = 1/n \, (\sum_{\text{sur l de 1 à n}} m_{il})$ càd la moyenne sur tous les patients des m_{il}
- b_i est un paramètre choisi de façon à ce que la fonction d'appartenance soit proche de 0 quand m_{il} approche sa valeur minimale possible ($\sum$ min $w_{jk}^{(i)}$) et proche de 1 quand m_{il} approche sa valeur maximale possible ($\sum$ max $w_{jk}^{(i)}$).

Ce qui nous donne une forme proche de :



2. SÉLECTION DES VARIABLES LES PLUS DISCRIMINANTES.

2.1 : Maladies respiratoires.

KRUSINSKA et LIEBHART arrêtent là leur référence à la théorie des SEF et appliquent les techniques classiques de l'analyse discriminante sur les valeurs des variables m_{il} elles-mêmes.

2.2 : Maladie coronarienne.

• Pour les autres, on procède de la façon suivante :

un questionnaire R sera étiquetté "C" (pour coronaire) si et seulement si :

$$\text{Poss}\{ R \text{ is } C \} > \text{Poss}\{ R \text{ is } S \}$$

et étiquetté "S" dans le cas contraire.

• Poss{ R is C } est défini comme :

$$\text{Max}_{\text{sur } u \in B^*} [\min (\prod_C(u), \mu_R(u, c))]$$

où :

- u est un string composé de m variables linguistiques avec 1 <= m <= nombre total de variables linguistiques
- B* est l'ensemble de tous les string possibles
- $\mu_R(u,c)$ donne le degré de compatibilité du string u avec l'étiquette C
- $\prod_C(u) = \min_{\text{sur les var. ling. de u}} \mu'_{Vi}(R)$ avec Vi est une variable linguistique et en définissant $\mu'_{Vi}(R)$ par :
  - $\mu'_{Vi}(R) = \mu_{Vi}(R)$ ssi de fortes valeurs de Vi vont dans le sens de C
  - $\mu'_{Vi}(R) = 1 - \mu_{Vi}(R)$ sinon.

Avec $\mu_{Vi}(R) = \mu_{il}(m_{il})$ en considérant que le questionnaire R est celui du patient l.

On peut donc considérer que $\prod_C(u)$ donne le degré de compatibilité du questionnaire R avec le string u dans le contexte C.

• Trouver les variables ou les combinaisons de variables les plus significatives revient donc à trouver les $\mu^*_R(u,c)$ et les $\mu^*_R(u,s)$ telles qu'on minimise le nombre d'erreurs de classement.

Ce problème n'est pas abordé ici.

3. CLASSEMENT DES SUJETS.

Il se fera selon la fonction exposée plus haut pour ce qui est de la maladie coronarienne.

Pour les pathologies respiratoires, on peut utiliser différentes fonctions discriminantes (linéaire, quadratique, canonique).

C. LES RÉSULTATS.

• Dans SAITTA et TORASSO, l'application d'une telle procédure permet de réduire le pourcentage de faux positifs et négatifs dans le classement, par rapport à d'autres méthodes plus classiques. On avoisine les 8 % de faux positifs et les 18 % de faux négatifs.

• Pour les pathologies respiratoires, on met en évidence que :

- les variables linguistiques, non-quantitatives, apparaissent comme 5 des 10 variables les plus discriminatives, dont 4 en tête. Ceci va d'ailleurs dans le sens de l'avis des experts et correspond à leur classement intuitif.

- Un diagnostic ne peut pas être efficacement posé si on ne tient compte que des variables quantitatives : 86 % seulement de classification correcte. L'amélioration est notoire quand on fait intervenir les variables linguistiques : on passe de 86 à 97 % .

- On ne perd pas beaucoup à ne tenir compte que de la sélection des 10 variables les plus discriminantes au lieu de l'ensemble des 45 : 94 au lieu de 97 %.

D. DISCUSSION.

• Le processus de classification, réalisé d'abord sur des groupes de patients connus pour "étalonner" la méthode, peut ensuite s'appliquer, à titre diagnostic sur des patients inconnus.

Cependant, cette méthode ne peut jamais qu'effectuer un classement binaire pathologique - non pathologique (et éventuellement [35] s'il y a pathologie : à quel degré). On est donc très limité dans l'ampleur des conclusions.

• En revanche, elle peut être très utile dans le but de trouver les variables les plus discriminatives et de mettre ainsi en évidence les facteurs étiologiques ou symptomatiques les plus importants dans la caractérisation d'une maladie.

• Selon le domaine auquel on l'appliquera et la manière dont on le fera, on pourra prétendre affirmer l'existence ou l'absence d'une pathologie (KRUSINSKI et LIEBHART) càd réaliser un diagnostic ou, plus modestement, décider d'un point plus précis du problème comme par exemple dans SAITTA et TORASSO : un profil socio-psychologique prédisposant ou non à une maladie.

• Pour ce qui est de la référence à la théorie des SEF, on remarquera qu'il est tout à fait marginal dans le cas des pathologies respiratoires puisqu'elle n'intervient que dans la transformation de la variablr m_{il} en un réel de l'intervalle [0, 1] par le biais d'une fonction d'appartenance et ce, de façon à mieux visualiser la signification de la valeur de m_{il}.

Dans l'exemple de la maladie coronarienne, par contre, on travaille directement sur les valeurs de la fonction d'appartenance $\mu(m_{il})$, en introduisant des notions de relation floue et de possibilité.

Il semble d'ailleurs,à première vue, que l'on obtienne de meilleurs résultats avec des techniques classiques (à peu près 95 % de bons classements) qu'avec la fonction de possibilité (24 % de faux négatifs et positifs).

Les domaines d'application sont certes très différents, rendant difficile l'affirmation d'une conclusion définitive mais cette constatation va dans le sens de ce que nous avions déjà noté : l'usage d'un calcul type Max-min est rarement heureux. Etant donné l'effort apporté dans la première partie de la méthode pour la quantification des variables linguistiques (càd initialement qualitatives), on peut se demander pourquoi réintroduire une manipulation floue beaucoup moins précise que ne peut l'être l'usage d'une fonction discriminante.

IV. Discrimination et connectivité floue.

A. INTRODUCTION.

NORRIS, PILSWORTH et BALDWIN [36] nous proposent, dans le cadre du syndrome abdominal aigu, un modèle de diagnostic combinant deux approches : une approche discriminative, approche individuelle des symptômes et une mesure de connectivité entre symptômes, mettant en évidence des "patterns" caractéristiques d'une pathologie.

Ils s'appuyent sur l'étude de cas d'archives, permettant de savoir quels symptômes, et dans quel degré, présentaient les patients chez qui on a pu prouver l'existence d'une maladie bien précise. Cette façon de procéder pourrait constituer une alternative à l'acquisition, auprès des experts, des règles de diagnostic. Cette étape est en effet souvent délicate à mener et il n'est pas rare que les heuristiques de diagnostic qui en découlent soient relativement incomplètes et / ou incohérentes, rendant leur modélisation et leur utilisation difficile.

B. LA MÉTHODE.

1. APPROCHE DISCRIMINATIVE.

1.1 : Identification des symptômes discriminatifs.

- On dispose donc, après étude de cas d'archives, d'une table $\{f_{ij}\}$ où f_{ij} est la proportion (et pas le degré d'intensité!) dans laquelle le symptôme i est apparu dans la maladie j.

- On en retire les tables $\{p_{ij}\}$ et $\{n_{ij}\}$, de discriminations positive et négative respectivement. On les calcule par :

$$- p_{ij} = \sum_{\text{sur les } k \in D \text{ avec } k \neq j} \{X_{\text{large ratio}}(f_{ij}/f_{ik})\} / (C_D - 1)$$

$$- n_{ij} = \sum_{\text{sur les } k \in D \text{ avec } k \neq j} \{X_{\text{large ratio}}(f_{ik}/f_{ij})\} / (C_D - 1)$$

où :

- p_{ij}, n_{ij} appartiennent à [0, 1]
- C_D est le cardinal de l'ensemble des maladies D
- $X_{\text{Large ratio}}$ est la fonction d'appartenance du SEF Large Ratio, défini sur les réels positifs.

Elle peut être définie à l'aide de différents paramètres qui semblent avoir peu d'influence sur le résultat. On peut par exemple prendre a = 1 et b = 3 dans :

$$X_{\text{Large Ratio}} = 0 \text{ ssi } x <= a$$

$$X_{\text{Large Ratio}} = 1 \text{ ssi } x >= b$$

$$X_{\text{Large Ratio}} = 1/2 * (x - 1) \text{ sinon.}$$

- Intuitivement :
  - p_{ij} représente le degré de confiance dans le fait que le symptôme i est plus significatif de la maladie j que de toutes les autres maladies
  - n_{ij} représente le degré de confiance dans le fait que le symptôme i est plus significatif de ne pas avoir la maladie j que de ne pas avoir toute autre maladie.

1.2 : Diagnostic.

- Soit un patient présentant un ensemble $S = \{S_1,..,S_i,..,S_n\}$ de symptômes.

On sélectionne dans les tables p_{ij} et n_{ij} uniquement les rangées correspondant aux S_i de S, ce qui nous donne deux nouvelles tables $\{p_{ij}'\}$ et $\{n_{ij}'\}$.

- Un diagnostic peut être défini comme une distribution $\{\delta_j\}$ sur les maladies comme suit :

$$\delta_j = 1/2 * \{X_{\text{Large}}(\pi_j) + X_{\text{Small}}(\nu_j)\}$$

où :

- $j \in D$
- $\pi_j = \{\sum_{\text{sur i}} p_{ij}'\} / C_S$ avec C_S = cardinal de S
- $\nu_j = \{\sum_{\text{sur i}} n_{ij}'\} / C_S$.

π_j et ν_j représentent donc les moyennes des mesures de discriminations positive et négative respectivement, de la maladie j.

Les SEF Large et Small sont définis sur [0, 1] et ont pour fonction d'appartenance :

$$\mu_{Small}(x) = \mu_{Large}(x) = x.$$

- La maladie j possédant le plus grand δ_j peut être interprétée comme le diagnostic le plus crédible.

On peut y ajouter une mesure de l'étalement des δ_j pour estimer le degré d'incertitude du diagnostic.

1.3 : Une approche plus parallèle.

On obtient déjà ainsi de bons résultats (comparativement au diagnostic posé par des cliniciens expérimentés) et ce, avec une base de données de 90 patients environ, mais on voudrait les améliorer par une approche plus "parallèle" des symptômes, en identifiant des ensembles de symptômes étroitement connectés au sein d'une maladie, constituant ainsi des patterns caractéristiques. C'est le but du point suivant : la connectivité.

2. CONNECTIVITÉ.

2.1 : Le principe.

2.1.1 : La théorie de Atkin.

Supposons une matrice d'incidence $R = \{r_{ij}\}$ où $r_{ij} = 1$ s'il existe une relation entre l'élément de la i^e rangée et de la j^e colonne; $r_{ij} = 0$ sinon.

On peut donner une idée de la "ressemblance", de la connectivité entre les rangées ou les colonnes par :

- pour les colonnes :

$$C_C = R^T R - \Omega$$

- pour les rangées :

$$C_R = R R^T - \Omega$$

où Ω est la matrice unité.

Ces matrices C_C et C_R sont symétriques et peuvent donc être représentées sous forme triangulaire.

Exemple :

- avec la matrice d'incidence suivante :

	P1	P2	P3	P4
S1	1	1	1	1
S2	0	0	1	1
S3	1	1	1	1

- nous aurions la matrice de connectivité entre les rangées, ici C_S :

S1	S2	S3	
3	1	3	S1
	1	1	S2
		3	S3

On peut ainsi définir des chaînes de connectivité d'un niveau donné, chaînes qui peuvent être directement déduites des matrices C_R et C_C. Dans notre exemple, nous aurions, pour les rangées toujours :

q-connectivité	chaînes C_R
3	{S1,S3}
2	{S1,S3}
1	{S1,S2,S3}
0	{S1,S2,S3}

La valeur de q indique le niveau de connectivité tandis qu'on donne, entre accolades, les rangées ou colonnes concernées.

Dans le domaine qui nous intéresse, si on représente les symptômes par des rangées et les patients qui ont souffert d'une même maladie par des colonnes, alors :

- les chaînes C_R décrivent des sous-groupes particuliers de symptômes qui se présentent plus ou moins fréquemment (selon le niveau q de connectivité) ensemble; càd, elles identifient des patterns plus ou moins fortement indicateurs de la maladie
- les chaînes C_C, dont nous ne parlerons pas, indiqueraient des sous-groupes de patients càd des tableaux différents d'une maladie.

2.1.2 : Connectivité normalisée.

Plutôt que de donner le nombre absolu de 1 partagés par les rangées ou colonnes, on donne la proportion par rapport au total, respectivement, de colonnes et de rangées. cela nous donnerait sur l'exemple :

S1	S2	S3	
4/4	2/4	4/4	S1
	2/4	2/4	S2
		4/4	S3

Une connectivité égale à 1 indique une totale équivalence tandis qu'une connectivité égale à 0 indique une disparité complète.

2.1.3 : Connectivités positive et négative.

On peut résumer en disant que la connectivité positive examine le nombre de 1 que les rangées ou colonnes ont en commun, tandis que la connectivité négative mesure le nombre de 0.

Dans notre problème, la première établirait les groupes de symptômes indicateurs d'une maladie, tandis que la seconde établirait les groupes de symptômes indicateurs de l'absence de la maladie.

2.1.4 : Données floues.

On peut appliquer la même méthode lorsque les données ne sont pas binaires mais lorsqu'on veut, par exemple, donner une indication du degré de sévérité des symptômes.

On calcule alors la connectivité positive sur la matrice d'incidence et la connectivité négative sur son complément, selon les formules suivantes.

2.1.5 : Les algèbres de connectivité.

Le choix de la formule pour le calcul de connectivité peut changer assez largement sa signification.

•Dans la théorie de Atkin (point 2.1.1), on a pris :

$$c = \sum_{\text{sur } i} (a_i * b_i) - 1$$

où :

- a et b sont deux colonnes ou deux rangées
- i = 1,...,m avec m = le nombre de rangées si on mesure une connectivité entre colonnes et vice-versa

• On peut prendre une autre formule de calcul où, selon les mêmes conventions, on aurait :

$$c = \{\sum_{\text{sur } i} \min(a_i, b_i)\} / \{\sum_{\text{sur } i} \text{Max}(a_i, b_i)\}$$

On note que cette formule est applicable à la fois aux données floues et non-floues.

• La différence assez fondamentale est que, dans la première solution, on pénalise à la fois le cas où l'un des deux est égal à 0 et le cas où les deux sont égaux à 0.

Par contre, avec le deuxième calcul, on ne pénalisera que la différence càd le cas où l'un des deux seulement est égal à 0.

Replacé dans notre problème, cela veut dire que si on calcule la connectivité selon Atkin (1er cas), une forte valeur de connectivité entre deux symptômes indiquera à la fois que ces symptômes sont très liés et que d'autre part, ils apparaissent fréquemment dans la maladie. Par contre, si on prend la deuxième formule, ce qui est le choix des auteurs, une forte valeur de connectivité indiquera uniquement le fait que les symptômes sont très étroitement liés mais ne donne aucune indication sur leur fréquence d'apparition dans la maladie.

2.2 : Le diagnostic.

2.2.1 : Le calcul des patterns représentatifs.

• A partir des cas d'archives, on peut établir la matrice d'incidence, floue et définie sur S x P.(symptômes x patients).

• On en déduit les matrices de connectivités positive et négative normalisées, selon la seconde formule que nous avons vue.

• Ensuite, on peut mettre en évidence, pour différentes valeurs q choisies dans l'intervalle [0, 1] (ex: 0, 0.2, 0.4, 0.6, 0.8, 1), les chaînes de connectivité q pour les symptômes.

- Comme la taille de connectivité peut être grande, on ne prend en ligne de compte, pour limiter les calculs, que les symptômes suffisament discriminatifs. Typiquement, on peut prendre un seuil de 0.4.

- Tous les symptômes sont individuellement connectés à eux-mêmes au niveau 1 tandis que l'ensemble des symptômes est connectée au niveau 0. Entre ces deux extrêmes, On voit se dessiner des groupes, de plus en plus petits à mesure que le niveau devient plus haut.

• A chaque niveau, on retient un groupe candidat pour constituer le pattern caractéristique de la maladie à ce niveau. On choisit le groupe de symptômes qui est le plus grand sous-ensemble du groupe sélectionné dans le niveau précédent et qui contient au moins deux éléments.

Par exemple, on pourrait avoir :

Groupe	Niveau
{4, 6, 9}	0.8
{4, 6, 9}	0.6
{4, 6, 9}	0.4
{4, 5, 6, 9}	0.2
{1, 2, 3, 4, 5, 6, 7, 8, 9}	0.0

- Ayant effectué cela sur les tables de connectivités négative et positive, on obtient un ensemble de paires ordonnées :

$$P = \{ (Q_i, c_i) \} \text{ pour la connectivité positive}$$

$$N = \{ (R_i, d_i) \} \text{ pour la connectivité négative}$$

où :

- Q_i, R_i sont les ensembles de symptômes sélectionnés aux différents niveaux i
- c_i, d_i sont les connectivités du i^e niveau.

• On réalise cela pour chaque maladie, ce qui nous donne les distributions $(P_1,...,P_n)$ et $(N_1,...,N_n)$.

2.2.2 : Heuristique de diagnostic.

Soit donc, un patient présentant l'ensemble de symptômes S_j, on dérive la distribution $D = (D_1,..,D_j,..,D_n)$ sur les maladies en pondérant par les valeurs de connectivité, la proportion des symptômes présents chez le patient par rapport au groupe sélectionné pour ce niveau de connectivité càd formellement :

$$D_j = [\, X_{Large}(\{\sum_{sur\ i}\ c_i * p(Q_i/S_j)\} / \{\sum_{sur\ i}\ c_i\})$$
$$+ X_{Small}(\{\sum_{sur\ i}\ d_i * p(Q_i/S_j)\} / \{\sum_{sur\ i}\ d_i\}) \,]/2$$

où :

- $p(Q_i/S_j)$ est la proportion des symptômes de Q_i présents dans S_j
- X_{Small} et X_{Large} sont des SEF définis sur [0, 1] et tels que $\mu\ (x) = x$.

C. LES RÉSULTATS.

Sans avancer de résultats précis, les auteurs assurent qu'une telle approche permet de concurrencer les performances de cliniciens expérimentés dans le diagnostic d'abdomen aigu.

L'heuristique proposée pourrait être modifiée, par exemple pour tenir compte, lorsqu'on calcule la proportion de symptômes présents,du fait que certains d'entre eux ne peuvent prendre qu'une seule valeur (ex: catégorie d'âge) tandis que d'autres (ex: facteurs aggravants) peuvent en prendre plusieurs.

D. DISCUSSION.

- Difficile de se faire une idée puisqu'on ne dispose pas d'un minimum d'indications concrètes, comme par exemple les symptômes considérés et les maladies en compétition.

- On remarquera que l'intensité des symptômes est prise en compte dans la matrice d'incidence issue des cas d'archives, mais que lors du diagnostic, on ne précise pas le degré de sévérité du symptôme que l'on note chez le patient ; on ne note que sa présence.

- Comme nous l'avons noté plus haut, la formule choisie pour le calcul de connectivité fait qu'on "récompense" la simultanéité d'apparition des symptômes mais qu'on ne tient pas compte de la fréquence avec laquelle ces symptômes apparaissent dans la maladie. Deux symptômes pourront dès lors de bénéficier d'une forte connectivité s'ils apparaissent toujours ensemble, même s'ils n'apparaissent que très rarement dans la maladie! Cela peut paraître étonnant.

Sans doute le fait d'exclure les symptômes non-suffisament discriminatifs corrige-t-il un peu ce problème. De même si l'on tient compte à la fois des résultats de la connectivité et de la discrimination.

- D'autre part, cela pose également la question du choix des symptômes que l'on prend en considération et de l'identification du pattern représentatif. En effet, avec cette méthode,deux symptômes résultant du même mécanisme physiopathologique (en d'autres termes qui se produiront toujours ensemble) vont bénéficier d'une forte connectivité, un peu comme si on citait deux fois le même symptôme.

RAISONNEMENT APPROXIMATIF ET SYSTEMES EXPERTS

Une des voies majeures d'investigation de la théorie des SEF est sans doute le domaine du *raisonnement approximatif* (*Fuzzy Reasoning*) et de la logique sur laquelle il repose : la *logique floue* (*Fuzzy Logic*).

•I• Partant de la constatation que l'homme est capable, dans ses raisonnements quotidiens, de faire face à l'imprécision et de la gérer, et s'appuyant sur l'idée que l'imprécision linguistique est de nature possibiliste (càd qu'elle vient d'une absence de définition précise et claire des mots et des concepts employés), plusieurs auteurs, en exploitant ainsi largement la théorie des SEF, se sont attachés à formaliser les propositions émises en langage naturel et modéliser les lois qui régissent les processus de déduction dans un tel contexte d'imprécision. L'objectif étant, ainsi, d'approcher le raisonnement humain.

Il faut citer, bien sûr, en tout premier lieu ZADEH [1] [2] [3] [4] (...et bien d'autres) mais aussi YAGER [5], GOGUEN [4], MAMDANI, BANDLER et KOHOUT [4]...etc.

Il serait beaucoup trop complexe, tant les voies d'abord sont diverses, les nouveaux concepts nombreux et tant la discussion est encore fort animée et controversée, de tenter une comparaison un tant soit peu complète. Nous nous limiterons à présenter le type de raisonnement et les principales étapes de la démarche. Pour cela, nous suivrons surtout ZADEH. Il a, le premier, jeté les bases d'une telle méthode et ses concepts, même s'ils ne sont pas tous repris ni acceptés comme tels, se retrouvent chez beaucoup d'autres auteurs. Ce sera donc notre premier objectif.

Il faut cependant se rendre compte que l'unanimité est loin d'être faite et que des problèmes non négligeables se posent dans cette nouvelle logique : paradoxes, résultats surprenants et contraires à l'intuition, multiplicité des définitions des opérateurs logiques (principalement l'implication) menant à des divergences dans les résultats. Certains auteurs (SCHEFE [4], GILES [4] [6]) proposent également une redéfinition de la logique floue sur une base de probabilités.

Mettre en évidence les problèmes et les critiques soulevées sera notre deuxième objectif.

•II• Une autre voie classiquement suivie pour tâcher de "concurrencer" le raisonnement humain est le développement de systèmes experts (SE). Ils sont basés, du moins pour une partie d'entre eux, sur un ensemble de faits et de règles (celles-ci étant le plus souvent construites à partir des connaissances d'experts) qui seront exploitées de façon à essayer d'atteindre un objectif.

On peut donc comprendre que le raisonnement approximatif et la théorie des SEF "fassent irruption" dans le domaine des systèmes experts, dans le but d'affiner la représentation des données et le processus de déduction et de le rendre plus semblable encore au raisonnement humain.

De plus, l'insertion de connaissances dans les SE est actuellement un long et fastidieux travail d'échange entre experts et informaticiens. Disposer d'un système de lois et de règles pour formaliser "directement" des propositions émises en langage naturel nous faciliterait évidemment grandement la tâche.

Il est donc intéressant de faire le point sur ce sujet : dans l'état actuel des choses, les SE intègrent-ils le raisonnement approximatif et la théorie des SEF et si oui, dans quelle mesure et avec quels bénéfices. Voilà notre troisième objectif.

Tout cela peut apparaître très général et éloigné du domaine médical mais si une issue se dégageait dans une telle approche, nul doute que la médecine serait un terrain idéal d'application et qu'elle en tirerait de gros bénéfices. Ceci justifie donc qu'on s'y attarde.

I. Fuzzy reasoning : éléments de base (objectif 1).

Références : [1], [2], [3], [4].

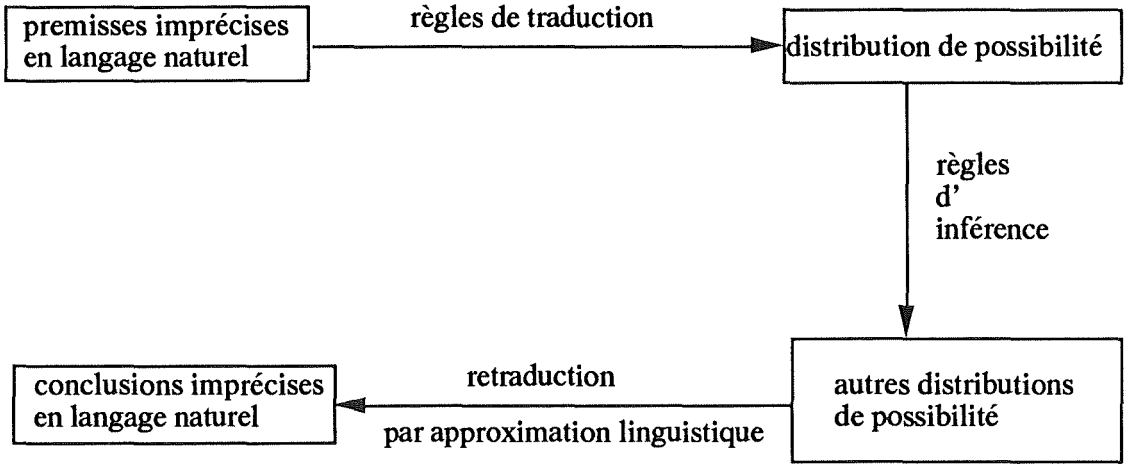
A. INTRODUCTION.

• On peut définir le raisonnement approximatif comme le processus qui consiste à déduire une conclusion possible imprécise à partir de prémisses elles-mêmes imprécises. C'est donc un processus essentiellement qualitatif.

Le but d'une telle démarche est, comme on l'a déjà dit, d'approcher le raisonnement humain.

Pour cela, le raisonnement approximatif va s'appuyer sur la logique floue, tout comme pour le raisonnement précis,on se servait de la logique classique binaire.

• Suivant l'approche de ZADEH, on peut résumer la démarche par le schéma suivant :



• Nous verrons donc successivement :

- les principales idées des règles de traduction, pour illustrer comment on peut obtenir des distributions de possibilité caractéristiques des propositions en langage naturel
- de façon un peu plus approfondie, les règles d'inférence qu'on applique aux distributions afin de tirer des conclusions.

B. LES PRINCIPES DU RAISONNEMENT APPROXIMATIF.

1) LES REGLES DE TRADUCTION.

1.0. Nous avons vu, dans l'exposé des principaux concepts de la théorie des SEF, comment une proposition **p** = X is **F** induit une distribution de possibilité $\Pi_X = \mathbf{F}$. Nous avons également vu les principales notations et propriétés qui vont intervenir dans les points suivants.

1.1 : Les règles de modification.

- Soient :
  X une variable prenant ses valeurs dans U = {u}
  F un SEF de U
  et **p** une proposition de la forme **p** = X is **F**.

- Si la traduction de **p** peut être exprimée par :

$$X \text{ is } \mathbf{F} \Rightarrow \Pi_X = \mathbf{F}.$$

alors, la traduction de **p** = X is m**F** où m est un "modifieur" tel que "not", "very", "more or less",...est donnée par :

$$\boxed{X \text{ is } \mathbf{F} \Rightarrow \Pi_X = \mathbf{F}^+}$$

où $\mathbf{F}^+$ est une modification de **F** induite par m.

- Nous nous reportons, pour la définition de ces modifications, au point concernant les variables linguistiques que nous avons vues dans l'introduction aux concepts généraux de la théorie des SEF.

Rappelons, par exemple, que :
si m = "not", alors $\mathbf{F}^+ = \mathbf{F}'$
si m = "très", alors $\mathbf{F}^+ = \mathbf{F}^2$
...etc.

Rappelons également que ces modifications classiques peuvent être redéfinies et qu'elles s'appliquent à toute relation n-aire et pas seulement aux SEF.

1.2 : Les règles de composition.

Elles permettent de traduire une proposition composée de propositions "simples" qu'on peut traduire par 1.0 et 1.1.

- Soient :
  X une variable prenant ses valeurs dans l'univers du discours U
  Y une variable prenant ses valeurs dans l'univers du discours V
  F un SEF de U
  G un SEF de V

- Si (X is F $\Rightarrow$ $\prod_X$ = F) et (Y is G $\Rightarrow$ $\prod_Y$ = G), alors :

(X is F) and (Y is G) $\Rightarrow$ $\prod_{(X,Y)} = F^* \cap G^* = F \times G$.

(X is F) or (Y is G) $\Rightarrow$ $\prod_{(X,Y)} = F^* \cup G^*$.

If (X is F) then (Y is G) $\Rightarrow$ $\prod_{(X,Y)} = F^{*\prime} \oplus G^*$ ou, autre définition : If (X is F) then (Y is G) $\Rightarrow$ $\prod_{(X,Y)} = (F \times G) + (F' \times V)$

1.3 : Les règles de quantification.

• Elles s'appliquent aux propositions de la forme p = QX are F où Q est un quantifieur flou comme "la plupart", "quelques-uns",...etc.

• Un quantifieur flou est :
  - en général, un SEF des réels
  - ou, quand il exprime une proportion, comme "la plupart", un SEF de l'intervalle unitaire, dont la fonction d'appartenance est, par exemple, μ_{most} = S(0.5,0.7,0.9).

• Il faut ici définir deux indicateurs de la cardinalité d'un SEF.

- Si F est un SEF discret, la puissance $|F|$ est définie par :

$$|F| = \sum_{\text{sur i de 1 à n}} \mu_F(u_i),$$

avec F est un SEF défini sur l'univers $U = \{u_1,...,u_n\}$ et où $\sum$ est la somme arithmétique.

Exemple :

Si F = 1/0 = 1/1 + 0.6/2 + 0.3/3 + 0.1/4

Alors, $|F| = 1 + 1 + 0.6 + 0.3 + 0.1 = 3.$

- Lorsque n devient plus grand et que U devient un continuum, la puissance laisse place au concept de mesure de F.

Ainsi, si F est un SEF de U et ρ est une fonction de densité définie sur U, alors :

$$||F|| = \int_U \rho(u)\, \mu_F(u)\, du.$$

Exemple :

Si F = "grand" et que $\rho(u)du$ donne la proportion des hommes dont la taille est dans l'intervalle [u, u + du]

Alors, $||\text{grand}|| = \int_{\text{de 0 à } \infty} \rho(u)\, \mu_{\text{grand}}(u)\, du$, et donne la proportion des hommes qui sont grands.

• En conséquence, si **p** = X is F $\Rightarrow \Pi_X = F$, une proposition **p** = QX are F donne lieu à la traduction suivante :

- si F est discret :

$$\boxed{\text{QX are F} \Rightarrow \Pi_{|F|} = Q}$$

- et si U est continu :

$$\boxed{\text{QX are F} \Rightarrow \Pi_{||F||} = \mu_Q [\int_U \rho(u)\, \mu_F(u)\, du\,]}$$

càd que la proposition induit une distribution de possibilité de la fonction de densité exprimée par $\pi_{(\rho)}$.

• Prenons l'exemple de "grand" et de "la plupart" :

- si $\mu_{\text{la plupart}} = S(0.5,0.7,0.9)$ et si on exprime la taille en centimètres (posons de 155 à 185),

- alors, la proposition **p** = "la plupart des hommes sont grands" induit une distribution de possibilité sur la fonction de densité ρ de la taille où :

$$\pi(\rho) = \mu_{\text{la plupart}} [\int_{\text{de 155 à 185}} \rho(u)\, \mu_{\text{grand}}(u)\, du]$$

Prenons deux exemples de fonction de densité pour illustrer qu'à chacune d'elle correspond un degré de possibilité différent, en considérant, pour simplifier, des intervalles "du" assez grands.

du	155 - 165	165 - 175	175-185
μ_{grand}	0	0.5	1
ρ_1	0.2	0.6	0.2
ρ_2	0.2	0.6	0.2

On voit ainsi que $\pi(\rho_1) = \mu_{\text{la plupart}} ((0 * 0.2) + (0.6 * 0.5) + (0.2 * 1))$

$= \mu_{\text{la plupart}} (0.5)$

$= 0$

Tandis que $\pi(\rho_2) = \mu_{\text{la plupart}} ((0 * 0.2) + (0.2 * 0.5) + (0.6 * 1))$

$= \mu_{\text{la plupart}} (0.7)$

$= 0.5$

Ce qui nous montre bien que toutes les fonctions de densité n'ont pas le même degré de possibilité, étant donné une proposition **p** càd, en d'autres termes que cette proposition **p** induit une distribution de possibilité sur les fonctions de densité.

1.4 : Les règles de qualification.

• Nous avons défini les variables linguistiques dans la partie réservée aux concepts généraux. En deux mots, rappelons qu'il s'agit d'une variable dont les valeurs sont des mots étiquettant un SEF de l'univers du discours.

Appliquons cette définition à la valeur de vérité d'une proposition qui appartient à l'intervalle [0,1]. Ceci veut dire que les valeurs de vérité deviennent des SEF de l'intervalle [0,1], étiquettés par des labels comme "vrai", "faux", "très vrai"...etc.

De nouveau, il existe des définitions classiques de ces valeurs :

- $\mu_{vrai}(\upsilon) = \upsilon \qquad \forall\, \upsilon \in [0,1]$
- $\mu_{faux}(\upsilon) = \mu_{vrai}(1 - \upsilon)$
- $\mu_{très\ vrai}(\upsilon) = (\mu_{vrai}(\upsilon))^2 = \upsilon^2$

mais il est bien sûr possible de les redéfinir.

• Donc,

- si τ est un SEF de l'intervalle [0,1] (ce qui est bien le cas des valeurs de vérité linguistiques)
- et si $\mathbf{p} = X \text{ is } F \Rightarrow \prod X = F$
- alors :

$$\boxed{\mathbf{p} = X \text{ is } F \text{ is } \tau \Rightarrow \prod X = F^+}$$

où $\mu_{F^+}(u) = \mu_\tau(\mu_F(u))$.

• Exemple : **p** = (Lucie est jeune) est très vrai , avec jeune = 1 - S(25,35,45) induit :
$\prod_{Age(Lucie)} = F^+$ avec $\pi_{Age(Lucie)}(u) = \mu_{très\ vrai}(\mu_{jeune}(u))$.
Si u = 35 alors, $\mu_{jeune}(u) = 0.5$ et, prenant les définitions par défaut : $\mu_{très\ vrai}(0.5) = 0.25$.

2) LE MÉCANISME D'INFÉRENCE.

2.1 : Les principales règles d'inférence.

2.1.1 : Le principe de projection.

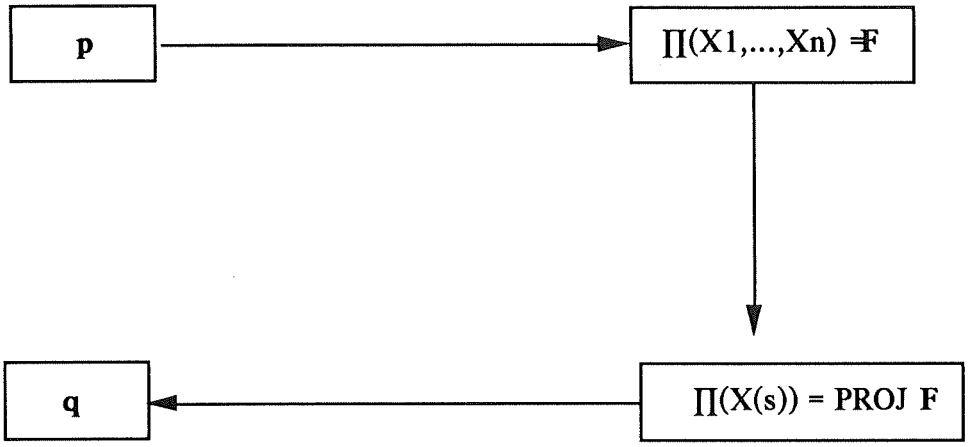
• Soient :

- **p** une proposition traduite par $\prod_{(X1,\ldots,Xn)} = F$
- X(s) une "sous-variable" $(X_{i1},\ldots,X_{ik})$ de $X = (X_1,\ldots,X_n)$ qui est définie sur U(n)
  où s est une sous-séquence $(i_1,\ldots,i_k)$ de $(1,\ldots,n)$ et
  où X(s) est définie sur l'univers U(s).

• On sait que l'on peut projeter (voir concepts généraux) F sur U(s) pour obtenir une autre distribution de possibilité, caractérisée par $\pi_{X(s)}(u_{i1},\ldots,u_{ik}) = Max_{uj1,\ldots,ujm}\, \mu_F(u_1,\ldots,u_n)$.

• Soit **q** la retraduction possible de $\prod_{X(s)} = PROJ_{U(s)}$ **F** :

alors, il est possible d'inférer **q** à partir de **p** par le mécanisme de projection, ce qui se résume par le schéma suivant :



• Cas particulier : si **p** $\Rightarrow \prod_{(X,Y)} = \mathbf{G} \times \mathbf{H}$ càd le produit cartésien de **G** et **H**, alors on peut inférer :

- **q** $\Leftarrow \prod_{(X)} = \mathbf{G}$ et
- **r** $\Leftarrow \prod_{(Y)} = \mathbf{H}$.

Par exemple : **p** = "le tapis est long et large" permet d'inférer :

- **q** = "le tapis est long"
- **r** = "le tapis est large"

2.1.2 : Le principe de particularisation / conjonction.

Le principe de particularisation est en fait un cas particulier du principe de conjonction. Toutefois, comme il est fréquemment utilisé, on l'expose séparément du second.

*** Particularisation.**

• Soit une proposition **p** traduite par $\prod_{(X1,...,Xn)} = \mathbf{F}$ où **F** est défini sur $U_1 x ... x U_n$.

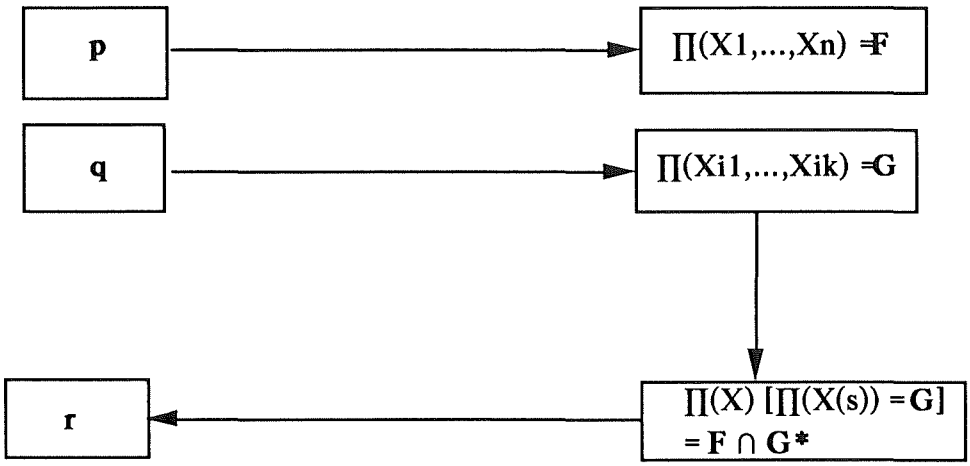
• Nous avons vu (toujours dans les principes généraux) que nous pouvions *particulariser* une distribution par une autre. Appelons cette autre distribution **G**, de sorte que :

$$\prod_{(X1,\cdots,Xn)} [\prod_{X(s)} = \mathbf{G}] = \mathbf{F} \cap \mathbf{G}^*$$

où X(s) est une "sous-variable" $(X_{i1},...,X_{ik})$ de X et

où **G*** est l'extension cylindrique de **G**.

• Si on peut retraduire **r** à partir de $\prod_{(X1,\ldots,Xn)} [\prod_{X(s)} = \mathbf{G}]$,
alors, il est possible d'inférer **r** à partir de **p** par particularisation de telle sorte que :



• Exemple :

- **p** = X et Y sont à peu près égaux ⇒ $\prod_{(X,Y)} = \mathbf{F}$. Soit :

	a	b	c	d
w	1	0.5	0	0
x	0.5	1	0.5	0
y	0	0.5	1	0.5
z	0	0	0.5	1

- **q** = X est petit ⇒ $\prod_{(X)} = \mathbf{G}$.Soit :

a	b	c	d
1	0.6	0.2	0

-Et nous obtenons pour $\prod_{(X,Y)} [\prod_X = \mathbf{G}] = \mathbf{F} \cap \mathbf{G}^*$:

	a	b	c	d
w	1	0.5	0	0
x	0.5	0.6	0.2	0
y	0	0.5	0.2	0
z	0	0	0.2	0

- Sans effectuer l'approximation linguistique, le résultat pourrait s'exprimer :

$$\mathbf{r} = X \text{ et } Y \text{ sont (à peu près égaux} \cap \text{(petit x V))}$$

*** Conjonction.**

• Principe :

- Soient :

$$\mathbf{p} \Rightarrow \Pi_{(Y1,\ldots,Yk,Xk+1,\ldots,Xn)} = \mathbf{F}$$

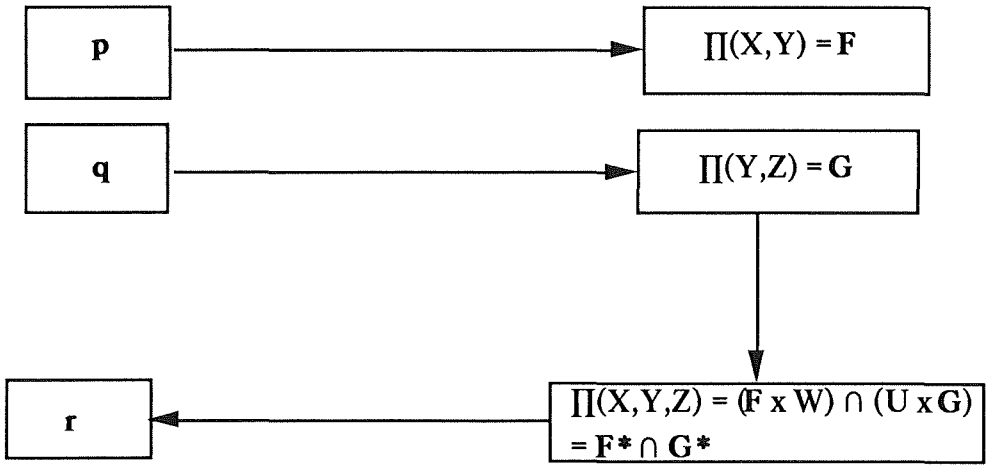
$$\mathbf{q} \Rightarrow \Pi_{(Y1,\ldots,Yk,Zk+1,\ldots,Zm)} = \mathbf{G}$$

avec la sous-variable $(Y_1,\ldots,Y_k)$ apparaissant dans les deux distributions.

- On peut en inférer :

$$\mathbf{r} \Leftarrow \Pi_{(Xk+1,\ldots,Xn,Y1,\ldots,Yk,Zk+1,\ldots,Zm)} = \mathbf{F^*} \cap \mathbf{G^*}$$

• Un cas particulier fréquent :

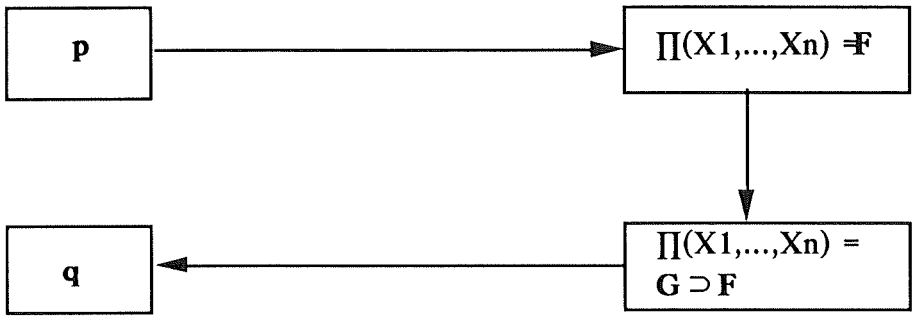


où W et U sont les univers du discours respectivement de Z et X.

2.1.3 : Principe de substitution.

• Ce principe affirme qu'on peut inférer une proposition **q** à partir d'une proposition **p** à condition que la distribution de possibilité induite par **p** soit incluse dans celle induite par **q**.

• Schématiquement, on a :



• Exemple :

De **p** = "X est très petit", on peut inférer que **q** = "X est petit".

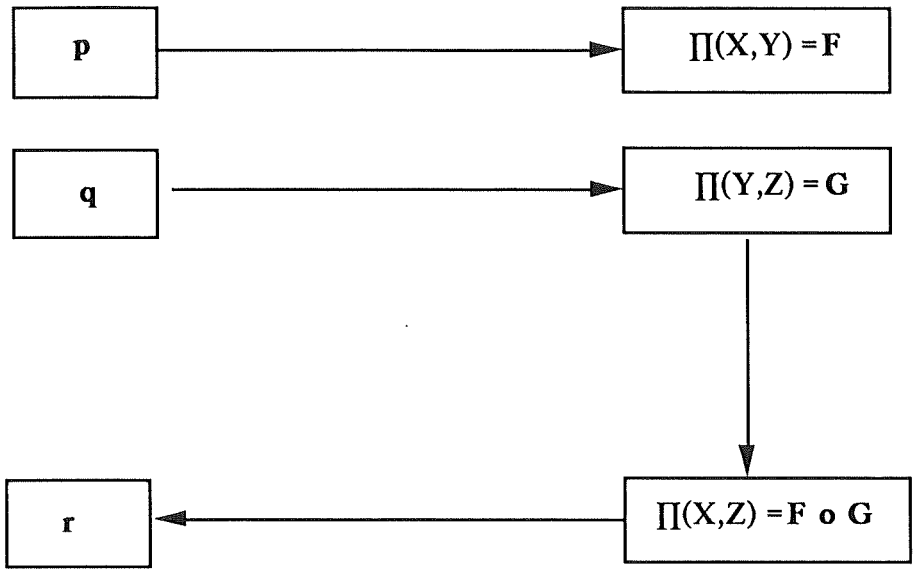
En effet, $\mu_{\text{très petit}}(u) = (\mu_{\text{petit}}(u))^2$ et donc, $\prod_{\text{très petit}}$ est inclus dans $\prod_{\text{petit}}$.

2.2 : L'inférence.

En toute généralité, on se sert de ces trois principes,éventuellement en combinaison, pour inférer des propositions.

• Une combinaison particulièrement employée est l'application du principe de particularisation/conjonction, suivi d'une projection. On l'appelle *règle compositionelle d'inférence* dont un cas particulier est le *modus ponens généralisé* ou *modus ponens compositionel*:

- Schématiquement, elle peut se traduire par :

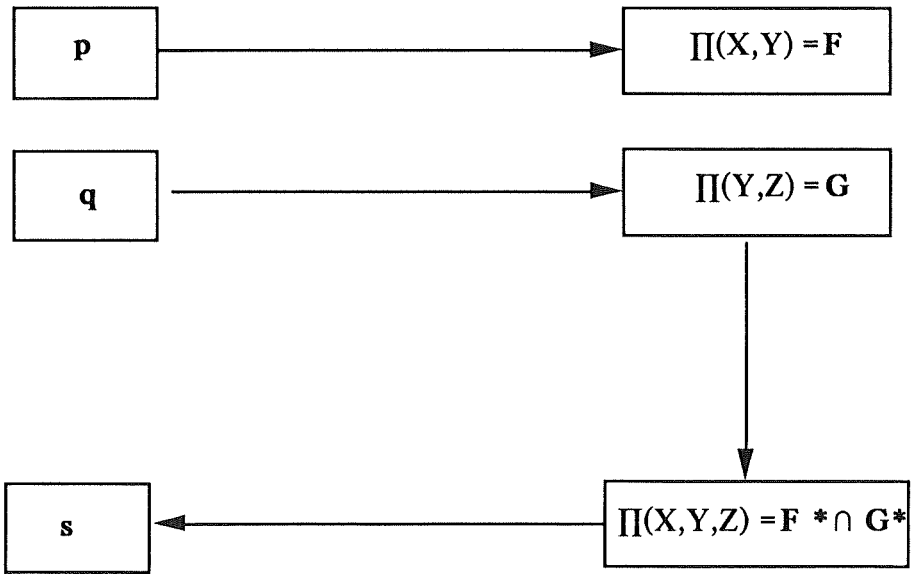


où :

- X,Y et Z prennent leurs valeurs dans U,V et W respectivement
- F et G sont des SEF de U x V et V x W respectivement
- F o G est la composition max-min vue dans les concepts généraux.

• C'est bien une particularisation /conjonction suivie d'une projection. En effet :

1) Particularisation / conjonction :



où $\pi_{(X,Y,Z)}(u,v,w) = \min [\ \mu_F(u,v)\ ,\ \mu_G(v,w)\ ]$.

2) La projection nous donne :

$PROJ_{U \times W}\ \Pi_{(X,Y,Z)} = \text{Max}_{\text{sur tous les } v}(\ \min [\ \mu_F(u,v)\ ,\ \mu_G(v,w)\ ]\ )$

$= F \circ G$

d'où on obtient **r** par retraduction.

• Cas particulier du modus ponens compositionel :

$$p = X \text{ is } F \longrightarrow \prod(X) = F$$

$$q = \text{If } X \text{ is } G \text{ then } Y \text{ is } H \longrightarrow \prod(Y,Z) = G^{*\prime} \oplus H^{*}$$

$$r \longleftarrow \prod(Y) = F \circ (G^{*\prime} \oplus H^{*})$$

• Dans le cas particulier où F, G, H sont non-flous et où $F = G$, on retrouve le modus ponens classique :

$$\frac{\begin{array}{c} X \text{ is } F \\ \text{If } X \text{ is } F \text{ then } Y \text{ is } H \end{array}}{Y \text{ is } H}$$

II. Les problèmes du raisonnement approximatif (objectif 2).

A. LES FONDEMENTS DE LA THÉORIE.

Nous avons vu dans la partie présentant les concepts généraux ce qui avait poussé ZADEH à considérer qu'on ne se trouvait pas dans un contexte de probabilités.

Sans entrer dans la discussion, signalons que tous les auteurs ne partagent pas cette opinion et que certains voudraient replacer la théorie dans un contexte probabiliste.

Pour NATVIG [7], il est possible, dans certains cas, d'interpréter une possibilité comme une famille de probabilités. A titre d'illustration de son raisonnement, en face de l'exemple classique "Hans mange x oeufs au déjeuner", il propose de redéfinir cette possibilité comme la probabilité que Hans soit capable de manger x oeufs au moins une fois dans sa vie.

GILES [4,p.117] [6] propose toute une logique "pragmatique" basée sur la réaction d'une personne à l'écoute d'une proposition. Dans cette perspective, celui qui émet une proposition qu'il considère comme vraie prend un risque : celui que son "opposant" ne soit pas d'accord. Ainsi, une proposition sera considérée comme vraie par moi si et seulement si je suis sûr que l'auditeur dira "oui" et fausse si et seulement si je suis sûr qu'il dira "non". Une proposition dite *dispersive* càd à laquelle les réactions varient sera considérée comme ni vraie ni fausse. Dans le cas où on admet de telles propositions, on retrouve la notion de SEF de ZADEH mais on s'est placé dans un contexte d'expérience, aléatoire et statistique. Cette approche inclut également tout un ensemble de règles de composition à partir des propositions dites *premières*, comme par exemple, si P et Q sont deux phrases arbitraires :

- Celui qui affirme (P ou Q) accepte d'affirmer soit P soit Q selon son choix.
- Celui qui affirme (P et Q) accepte d'affirmer P ou Q au choix de son opposant.
- Celui qui affirme $(P \Rightarrow Q)$ accepte d'affirmer Q si son opposant affirme P.

....etc.

SCHEFE [4,p.189], qui se range aux côtés de GILES, pense aussi que le degré d'accord avec une proposition est de nature probabiliste. Il reconnaît les difficultés d'interprétation dans certains cas (réponse = "je ne sais pas") et admet la nécessité de prendre en compte la gradation souvent mise dans les réponses : "je suis d'accord à 80 %", "ce n'est pas très précis"...etc. Mais de toute évidence, pour lui, non seulement les distinctions possibilité - probabilité émises par ZADEH ne tiennent pas mais surtout, une approche axiomatique lui paraît beaucoup plus adéquate pour construire une base théorique, de façon à rencontrer les trois exigences suivantes :

- être cohérent sur le plan formel
- respecter les lois de la logique classique
- ne pas conduire à des résultats contre l'intuition.

Donner une base probabiliste à la théorie des possibilités aurait l'avantage d'élargir l'horizon des deux théories, les faisant profiter des résultats l'une de l'autre et surtout, amènerait plus de cohérence à la plus jeune théorie des possibilités en la reliant à une théorie beaucoup mieux connue.

B. DISCUSSION DES OPÉRATEURS LOGIQUES.

Si l'unanimité est faite sur le choix des connecteurs "et" et "ou", en revanche, l'opérateur d'implication dans les propositions du type "si A alors B" suscite bien des discussions. De nombreuses définitions ont été proposées : nous en avons vu deux mais il en existe bien d'autres.

A titre d'illustration des effets divers qu'elles peuvent avoir, on peut trouver à la page suivante les tableaux de comparaison de quelques-unes, tirés de BANDLER et KOHOUT [4,p.219] où $A \Rightarrow B$ est donné par :

TABLE 2

Values of the implication operators on the decile set V_{11}

$\rightarrow_1$ $S^{\#}$

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	1	1	1	1	1	1	1	1	1	1	1
0·2	1	1	1	1	1	1	1	1	1	1	1
0·3	1	1	1	1	1	1	1	1	1	1	1
0·4	1	1	1	1	1	1	1	1	1	1	1
0·5	1	1	1	1	1	1	1	1	1	1	1
0·6	1	1	1	1	1	1	1	1	1	1	1
0·7	1	1	1	1	1	1	1	1	1	1	1
0·8	1	1	1	1	1	1	1	1	1	1	1
0·9	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	1

$\rightarrow_2$ S

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0	1	1	1	1	1	1	1	1	1	1
0·2	0	0	1	1	1	1	1	1	1	1	1
0·3	0	0	0	1	1	1	1	1	1	1	1
0·4	0	0	0	0	1	1	1	1	1	1	1
0·5	0	0	0	0	0	1	1	1	1	1	1
0·6	0	0	0	0	0	0	1	1	1	1	1
0·7	0	0	0	0	0	0	0	1	1	1	1
0·8	0	0	0	0	0	0	0	0	1	1	1
0·9	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	1

$\rightarrow_3$ S^*

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0	1	1	1	1	1	1	1	1	1	1
0·2	0	0·1	1	1	1	1	1	1	1	1	1
0·3	0	0·1	0·2	1	1	1	1	1	1	1	1
0·4	0	0·1	0·2	0·3	1	1	1	1	1	1	1
0·5	0	0·1	0·2	0·3	0·4	1	1	1	1	1	1
0·6	0	0·1	0·2	0·3	0·4	0·5	1	1	1	1	1
0·7	0	0·1	0·2	0·3	0·4	0·5	0·6	1	1	1	1
0·8	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	1	1	1
0·9	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	1	1
1	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1

$\rightarrow_4$ $G43$

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0	1	1	1	1	1	1	1	1	1	1
0·2	0	0·5	1	1	1	1	1	1	1	1	1
0·3	0	0·33	0·67	1	1	1	1	1	1	1	1
0·4	0	0·25	0·5	0·75	1	1	1	1	1	1	1
0·5	0	0·2	0·4	0·6	0·8	1	1	1	1	1	1
0·6	0	0·17	0·33	0·5	0·67	0·83	1	1	1	1	1
0·7	0	0·14	0·29	0·43	0·57	0·71	0·86	1	1	1	1
0·8	0	0·13	0·25	0·38	0·5	0·63	0·75	0·88	1	1	1
0·9	0	0·11	0·22	0·33	0·44	0·56	0·67	0·78	0·89	1	1
1	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1

$\rightarrow_{4'}$ $G43'$

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0	1	1	1	1	1	1	1	1	1	1
0·2	0	0·5	1	1	1	1	1	1	1	1	1
0·3	0	0·33	0·67	1	1	1	1	1	1	1	1
0·4	0	0·25	0·5	0·75	1	1	1	1	1	1	1
0·5	0	0·2	0·4	0·6	0·8	1	1	1	1	1	1
0·6	0	0·17	0·33	0·5	0·67	0·8	1	1	1	1	1
0·7	0	0·14	0·29	0·43	0·5	0·6	0·75	1	1	1	1
0·8	0	0·13	0·25	0·29	0·33	0·4	0·5	0·67	1	1	1
0·9	0	0·11	0·13	0·14	0·17	0·2	0·25	0·33	0·5	1	1
1	0	0	0	0	0	0	0	0	0	0	1

$\rightarrow_5$ t

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0·9	1	1	1	1	1	1	1	1	1	1
0·2	0·8	0·9	1	1	1	1	1	1	1	1	1
0·3	0·7	0·8	0·9	1	1	1	1	1	1	1	1
0·4	0·6	0·7	0·8	0·9	1	1	1	1	1	1	1
0·5	0·5	0·6	0·7	0·8	0·9	1	1	1	1	1	1
0·6	0·4	0·5	0·6	0·7	0·8	0·9	1	1	1	1	1
0·7	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1	1	1	1
0·8	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1	1	1
0·9	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1	1
1	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1

$\rightarrow_6$ KD

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9	1
0·2	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·9	1
0·3	0·7	0·7	0·7	0·7	0·7	0·7	0·7	0·7	0·8	0·9	1
0·4	0·6	0·6	0·6	0·6	0·6	0·6	0·6	0·7	0·8	0·9	1
0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·6	0·7	0·8	0·9	1
0·6	0·4	0·4	0·4	0·4	0·4	0·5	0·6	0·7	0·8	0·9	1
0·7	0·3	0·3	0·3	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0·8	0·2	0·2	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0·9	0·1	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
1	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1

$\rightarrow_7$ EZ

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	1	1	1	1	1	1	1	1	1	1
0·1	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9	0·9
0·2	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·8	0·8
0·3	0·7	0·7	0·7	0·7	0·7	0·7	0·7	0·7	0·7	0·7	0·7
0·4	0·6	0·6	0·6	0·6	0·6	0·6	0·6	0·6	0·6	0·6	0·6
0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5
0·6	0·4	0·4	0·4	0·4	0·4	0·5	0·6	0·6	0·6	0·6	0·6
0·7	0·3	0·3	0·3	0·3	0·4	0·5	0·6	0·7	0·7	0·7	0·7
0·8	0·2	0·2	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·8	0·8
0·9	0·1	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	0·9
1	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1

$\rightarrow_8$ W

	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1
0	1	0·9	0·8	0·7	0·6	0·5	0·6	0·7	0·8	0·9	1
0·1	0·9	0·9	0·8	0·7	0·6	0·5	0·6	0·7	0·8	0·9	0·9
0·2	0·8	0·8	0·8	0·7	0·6	0·5	0·6	0·7	0·8	0·8	0·8
0·3	0·7	0·7	0·7	0·7	0·6	0·5	0·6	0·7	0·7	0·7	0·7
0·4	0·6	0·6	0·6	0·6	0·6	0·5	0·6	0·6	0·6	0·6	0·6
0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5	0·5
0·6	0·4	0·4	0·4	0·4	0·4	0·5	0·6	0·6	0·6	0·6	0·6
0·7	0·3	0·3	0·3	0·3	0·4	0·5	0·6	0·7	0·7	0·7	0·7
0·8	0·2	0·2	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·8	0·8
0·9	0·1	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	0·9
1	0	0·1	0·2	0·3	0·4	0·5	0·6	0·7	0·8	0·9	1

- 1) Standard Sharp =

$$1 \text{ ssi } a \neq 1 \text{ ou } b = 1$$

$$0 \text{ sinon.}$$

- 2) Standard Strict =

$$1 \text{ ssi } a <= b$$

$$0 \text{ sinon}$$

- 3) Standard Star =

$$1 \text{ ssi } a <= b$$

$$b \text{ sinon}$$

- 4) Gaines =

$$\min (1, b/a)$$

- 4') Modified Gaines =

$$\min (1, b/a, (1-a)/(1-b))$$

- 5) Lukasiewicz =

$$\min (1, 1-a+b)$$

- 6) Kleene - Dienes =

$$\text{Max } [(1-a), b]$$

- 7) Early Zadeh =

$$\text{Max } [\min(a,b), (1-a)]$$

$$\min [\text{ Max } [(1-a), b], \kappa a]$$

- 8) Willmott =

$$\min [\kappa a, \kappa b, \text{Max } [(1-a), b]]$$

où :

- κa est une mesure du caractère *crisp* de $a \in [0,1]$: $\kappa a = \text{Max } [a, (1-a)]$ et donc, varie de 0.5 (pour a = 0.5) à 1 (pour a = 0 ou 1).
- on retrouve les deux définitions énoncées plus haut : 7 (ou Early Zadeh) et 5 (de Lukasiewicz). En effet, pour une proposition "If X is F then Y is G ", prendre (F x G) + (F' x V) revient à appliquer 7 et prendre ($F^{*\prime} \oplus G^{*}$) revient à 5.

Les opérateurs 1 et 2 ne sont pas flous et illustrent ce qui peut se passer si on tente de tirer des conclusions binaires à partir de prémisses floues.

On observe que jusque 5, la diagonale est composée de 1, de façon à faire de la proposition "a ⇒ a" une tautologie forte ; à partir de 6, elle n'est plus qu'une tautologie modérée càd dont la valeur est de 0.5 à 1.

A partir de 6 également, on note ce que l'on appelle *la conservation du caractère crisp* : celui-ci, pour une formule bien formée, reste entre le caractère crisp du plus flou et celui du plus "crisp" des atomes de l'expression.

exemple : le caractère crisp de "a ⇒ b" reste, pour toutes valeurs de a et b, entre le caractère crisp du plus flou des atomes (a ou b) et celui du plus crisp.

Ceci correspond assez bien à l'intuition.

La formule 6 souligne l'équivalence classique de "si a alors b" et de "non-a ou b" tandis que la définition 7 met en évidence l'égalité de "si a alors b" avec ((a et b) ou (non-a et n'importe quoi)).

MIZUMOTO et ZIMMERMAN [8] ont également proposé une étude des opérateurs d'implication, dans la perspective de leur utilisation dans les modus ponens et tollens généralisés. Nous en reparlerons dans le point suivant, en étudiant les problèmes des règles d'inférence.

On peut toutefois d'ores et déjà se rendre compte que chaque définition a ses avantages et ses inconvénients, sans qu'il soit possible d'en trouver une parfaite.

C. LES SURPRISES DU RAISONNEMENT FLOU.

Nous nous proposons, dans ce point, de mettre en évidence un certain nombre de résultats contraires à notre intuition, auxquels nous conduisent les mécanismes d'inférence et les définitions des opérateurs logiques.

1) Tautologies.

Tout d'abord, certaines tautologies en logique classique nous apparaissent avoir maintenant un degré de vérité <= 1.

- "a ou non-a" = Max (a, 1-a) n'aura comme valeur 1 que dans les cas particuliers où a vaut 0 ou 1 càd quand on revient à une logique binaire. Dans les autres cas, ce sera inférieur à 1.

- De même, nous avons vu qu'avec certains opérateurs d'implication, "a ⇒ a" n'a pas toujours un degré de vérité égal à 1.

2) Définitions.

Dans le même ordre d'idées, certaines définitions que nous employons dans notre langage quotidien prennent un sens différent lorsque décrites par des concepts flous. Ainsi, si l'on définit "d'âge moyen" comme "ni-vieux, ni-jeune" càd "non-jeune et non-vieux", "d'âge moyen" = min(jeune', vieux'). Sa distribution dépendra des définitions de "jeune" et "vieux" mais sera rarement normalisée (càd dont les valeurs extrêmes sont 0 et 1) et donc, il risque de ne pas exister de personne typiquement d'âge moyen dont le degré d'appartenance serait de 1.

3) Opérateurs d'implication.

Nous avons déjà évoqué la multiplicité des opérateurs d'implication. Il faut se rendre compte qu'à chaque définition différente va correspondre un effet différent lors de leur utilisation dans les règles d'inférence et entre autres dans les modus ponens et tollens généralisés.

Rappelons que le modus ponens généralisé est de la forme :

Ant1 : If X is **F** then Y is **P**.

Ant2 : X is **G**.

Cons : Y is **Q**,

qui se ramène au modus ponens classique lorsque **F** = **G** et (**F** et **P**) sont non-flous.

Le modus tollens généralisé s'exprime, lui, par :

Ant1 : If X is **F** then Y is **P**.

Ant2 :Y is **Q**.

Cons : X is **G**,

qui revient au modus tollens classique dans le cas où **Q** = non-**P** et (**P** et **F**) sont non-flous.

*** Regardons tout d'abord le modus ponens.**

• Dans le cas où **F** = **G** mais **F** est flou, on n'obtient pas, avec un certain nombre d'implicateurs dont ceux décrits par ZADEH lui-même, que Y is **P**. Illustrons ceci par un exemple :

- X a de la température
- Si X a de la température, alors X frissone

où, pour les besoins de la cause, nous définirons :

- avoir de la température par :

37	38	39	40
0	0.5	1	1

- et frissoner (en nombres de couvertures nécessaires...!) par :

1	2	3	4
0	0.5	0.8	1

Dans ce cas, le résultat donne X is **P** o (**Q*'** ⊕ **S***) càd :

0	0.5	1	1

o

1	1	1	1
0.5	1	1	1
0	0.5	0.8	1
0	0.5	0.8	1

dont le calcul selon les définitions vues plus haut donne :

0.5	0.5	0.8	1

- On se serait attendu intuitivement à retrouver la même distribution que dans "X frissone". Mais il n'en est rien.

- On observe une tendance à l'uniformisation des possibilités, qui complique les conclusions que l'on voudra tirer. Ce sera particulièrement le cas lorsqu'on voudra gérer des données binaires, où les distributions de possibilité n'auront que deux valeurs.

- Et il devient maintenant possible que X n'ait qu'une couverture tout en frissonant, alors qu'on l'avait déclaré impossible.

• Tout ceci choque quelque peu notre intuition.

-En fait, on peut voir cela comme la conséquence du fait que, dans la théorie des SEF, une proposition peut être à la fois vraie et à la fois fausse.

En logique classique, en effet, "P ⇒ Q" est vrai dans le cas où P est faux ou bien dans le cas où P et Q sont vrais. Si l'on intègre cela dans le modus ponens où l'autre condition est d'avoir P vrai, on ne "sélectionne" donc que le deuxième cas.

Par contre, ici, lorsque P est faux et donc rend l'implication vraie, P peut aussi être vrai et la composition Max-min ne donnera pas la valeur nulle.

Exemple :

P	Q	P ⇒ Q	P et (P ⇒ Q)
0	1	1	0
0	0	1	0
1	1	1	1
1	0	0	0

où on a bien effectivement Q dans ce cas.

Tandis que :

P	Q	P ⇒ Q	P et (P ⇒ Q)
0	1	1	0
0.3	0	0.7	0.3
0.7	0	0.3	0.3
1	0	0	0

qui illustre que dans les deux cas où (P et (P ⇒ Q)) est vrai à 0.3, on avait pourtant Q faux.

- On peut également dire que c'est la conséquence du fait qu'en logique, P ⇒ Q est vrai dans le cas où P est faux et que ceci est assez contraire à notre "logique quotidienne".

- Quoi qu'il en soit, ceci montre que le processus de raisonnement approximatif ne reflète pas de façon "automatique" et immédiate la façon dont l'homme raisonne en langage naturel.

- L'étude systématique de MIZUMOTO et ZIMMERMAN [8] (dont on peut trouver, à titre indicatif, quelques données numériques à la page suivante) envisage différentes valeurs pour G dans l'antécédent 2 du modus ponens (G = très F, G = F...) et regarde, avec les différents opérateurs d'implication, la valeur du conséquent. Ainsi, dans le cas où G = F, on retrouve nos constatations puisque la fonction d'appartenance du conséquent, lorsqu'on emploie les opérateurs décrits par ZADEH (opérateurs a et b de la page suivante), est Max(0.5, μ_P) pour ($F^{*'} \oplus G^*$) et ((1 + μ_P) / 2) pour l'autre ; càd qu'on n'a pas μ_P. Certains opérateurs s'avèrent effectivement meilleurs mais parfois au détriment du caractère flou càd qu'on se déplace vers le haut de la liste des opérateurs décrits dans le point B (opérateurs c et d page suivante).

*** De même, dans le modus tollens généralisé :**

Si on considère le cas où Q est non-P, on ne retrouve (1 - μ_F) càd non-F que pour certains opérateurs qui ne sont pas non plus ceux de ZADEH (qui donnent Max(0.5 , (1-μ_F)) et (1-(μ_F/2)) respectivement).

On notera encore de cette étude que certains opérateurs d'implication (opérateurs e, f, g, h) permettent de tenir compte du fait que nous évoquions plus haut, à savoir que dans la logique de tous les jours, lorsqu'on dit "si A alors B", on sous-entend le plus souvent que si non-A alors non-B.

Enfin, on retiendra surtout que, bien que certains apparaissent meilleurs que d'autres, l'opérateur parfait n'a pas encore été défini. Rien n'empêche, bien sûr, d'utiliser tantôt l'un tantôt l'autre selon les problèmes mais cela complique singulièrement les choses, en particulier lors de chaînages d'opérateurs d'implication dont les définitions seraient différentes.

4) Chaînage et contraposée.

Dans la foulée du point précédent, on peut souligner également que :

• Soient :

P1 : Si X is **A** alors Y is **B**.

P2 : Si Y is **B** alors Z is **C**.

P3 : Si X is **A** alors Z is **C**.

Appliquer P1 puis P2 ne conduit pas toujours (selon les opérateurs d'implication) au même résultat qu'appliquer P3.

Table 1. Inference results by each method (case of generalized modus ponens)

	A	*very* A	*more or less* A	*not* A
correspondance p.67				
7 = a) R_m	$0.5 \vee \mu_B$	$\frac{3-\sqrt{5}}{2} \vee \mu_B$	$\frac{\sqrt{5}-1}{2} \vee \mu_B$	1
5 = b) R_a	$\frac{1+\mu_B}{2}$	$\frac{3+2\mu_B-\sqrt{5+4\mu_B}}{2}$	$\frac{\sqrt{5+4\mu_B}-1}{2}$	1
2 = c) R_s	μ_B	μ_B^2	$\sqrt{\mu_B}$	1
3 = d) R_g	μ_B	μ_B	$\sqrt{\mu_B}$	1
e) R_{sg}	μ_B	μ_B^2	$\sqrt{\mu_B}$	$1-\mu_B$
f) R_{gg}	μ_B	μ_B	$\sqrt{\mu_B}$	$1-\mu_B$
g) R_{gs}	μ_B	μ_B	$\sqrt{\mu_B}$	$1-\mu_B$
h) R_{ss}	μ_B	μ_B^2	$\sqrt{\mu_B}$	$1-\mu_B$

Table 2. Inference results by each method (case of generalized modus tollens)

	not B	*not very* B	*not more or less* B	B
R_m	$0.5 \vee (1-\mu_A)$	$(1-\mu_A) \vee \left(\frac{\sqrt{5}-1}{2} \wedge \mu_A\right)$	$\frac{3-\sqrt{5}}{2} \vee (1-\mu_A)$	$\mu_A \vee (1-\mu_A)$
R_a	$1-\frac{\mu_A}{2}$	$\frac{1-2\mu_A+\sqrt{1+4\mu_A}}{2}$	$\frac{3-\sqrt{1+4\mu_A}}{2}$	1
R_s	$1-\mu_A$	$1-\mu_A^2$	$1-\sqrt{\mu_A}$	1
R_g	$0.5 \vee (1-\mu_A)$	$\frac{\sqrt{5}-1}{2} \vee (1-\mu_A^2)$	$\frac{3-\sqrt{5}}{2} \vee (1-\sqrt{\mu_A})$	1
R_{sg}	$1-\mu_A$	$1-\mu_A^2$	$1-\sqrt{\mu_A}$	$0.5 \vee \mu_A$
R_{gg}	$0.5 \vee (1-\mu_A)$	$\frac{\sqrt{5}-1}{2} \vee (1-\mu_A^2)$	$\frac{3-\sqrt{5}}{2} \vee (1-\sqrt{\mu_A})$	$0.5 \vee \mu_A$
R_{gs}	$0.5 \vee (1-\mu_A)$	$\frac{\sqrt{5}-1}{2} \vee (1-\mu_A^2)$	$\frac{3-\sqrt{5}}{2} \vee (1-\sqrt{\mu_A})$	μ_A
R_{ss}	$1-\mu_A$	$1-\mu_A^2$	$1-\sqrt{\mu_A}$	μ_A

Les opérateurs e) à h) sont définis par :

$$R_{gs} = (A \times V \underset{g}{\Rightarrow} U \times B) \cap (\neg A \times V \underset{s}{\Rightarrow} U \times \neg B)$$
$$= \int_{U \times V} [\mu_A(u) \underset{g}{\rightarrow} \mu_B(v)] \wedge [1-\mu_A(u) \underset{s}{\rightarrow} 1-\mu_B(v)]/(u, v).$$

$$R_{ss} = (A \times V \underset{s}{\Rightarrow} U \times B) \cap (\neg A \times V \underset{s}{\Rightarrow} U \times \neg B)$$
$$= \int_{U \times V} [\mu_A(u) \underset{s}{\rightarrow} \mu_B(v)] \wedge [1-\mu_A(u) \underset{s}{\rightarrow} 1-\mu_B(v)]/(u, v).$$

$$R_{sg} = (A \times V \underset{s}{\Rightarrow} U \times B) \cap (\neg A \times V \underset{g}{\Rightarrow} U \times \neg B)$$
$$= \int_{U \times V} [\mu_A(u) \underset{s}{\Rightarrow} \mu_B(v)] \wedge [1-\mu_A(u) \underset{g}{\rightarrow} 1-\mu_B(v)]/(u, v).$$

$$R_{gg} = (A \times V \underset{g}{\Rightarrow} U \times B) \cap (\neg A \times V \underset{g}{\Rightarrow} U \times \neg B)$$
$$= \int_{U \times V} [\mu_A(u) \underset{g}{\rightarrow} \mu_B(v)] \wedge [1-\mu_A(u) \underset{g}{\rightarrow} 1-\mu_B(v)]/(u, v).$$

• D'autre part, on n'a plus toujours l'équivalence entre une proposition "Si A alors B" et sa contraposée "Si non-A alors non-B".

5) Projection.

Venons-en enfin au mécanisme de projection et illustrons d'emblée le problème par un exemple.

• Supposons :

- une proposition P1 "La pièce est longue", donnant la distribution de possibilité:

2	4	5(m)
0.1	0.6	1

- une proposition P2 "La pièce est large":

2	4	5(m)
0.2	0.7	1

- P3 "La pièce est haute":

2	3	4(m)
0	0.5	1

• Définissons F comme P1 ∩ P2 et G comme P2 ∩ P3.

• On peut construire (par conjonction), une distribution caractéristique du volume de la pièce : $\prod_{(L,l,H)} = F^* \cap G^*$ où F* et G* sont les extensions cylindriques de F et G.

Elle aura les valeurs suivantes :

- $\pi_{(2,2,2)} = 0$
- $\pi_{(2,2,3)} = 0.1$
- $\pi_{(2,2,4)} = 0.1$

...

- $\pi_{(5,5,2)} = 0$
- $\pi_{(5,5,3)} = 0.5$
- $\pi_{(5,5,4)} = 1$

• On voudrait maintenant inférer la distribution de possibilité caractéristique de la superficie de la pièce, par projection. On aura que $\prod_{(L,l)} = PROJ_{H} \prod_{(L,l,H)}$. Et entre autres :

- $\pi_{(2,2)} = 0.1$

...

- $\pi_{(5,5)} = 1$

• Jusqu'ici tout va bien. Mais si on change la distribution de possibilité de P3,qui porte sur la hauteur et qu'on la définit par :

2	3
0	0.5

on aura entre autres valeurs que :

- $\pi_{(5,5,2)} = 0$
- $\pi_{(5,5,3)} = 0.5$

et la projection donnera :

- $\pi_{(5,5)} = 0.5$

• D'où il ressort que la distribution de possibilité sur la superficie, inférée à partir des dimensions, dépend de la distribution de la hauteur ! Et où l'on s'aperçoit également que le résultat par cette voie est différent de celui qu'on obtiendrait en prenant directement la conjonction des distributions de longueur et largeur.

• En fait, le problème est évident : il faut que les distributions soient normalisées pour que ça marche.

Cependant, ceci n'est mentionné nulle part dans la théorie des mécanismes d'inférence.

De plus, ce problème prend toute son importance si on le voit avec le point 2) mentionné plus haut : certaines définitions ne garantissent pas la normalisation!

D. CONCLUSION.

Il ressort de tout ceci que les choses sont loin d'être acquises et évidentes.

D'une part, l'unanimité n'est pas faite sur certains points.

D'autre part, force est de constater que bien qu'un tel modèle réussisse à exprimer des nuances qui nous échappaient dans la logique binaire, il serait toutefois dangereux de s'y fier les yeux fermés : non seulement les résultats sont parfois surprenants et très différents de ce que notre intuition et notre vision humaine des choses aurait voulu mais ils sont même parfois incohérents sur un plan purement logique.

Rigueur formelle, résultats validant l'intuition, cohérence logique : on retrouve tout à fait les souhaits de SCHEFE [4] pour une théorie plus "robuste".

III. Le flou dans les systèmes experts (objectif 3).

A. LES ESPÉRANCES.

Plusieurs auteurs (dont DUBOIS [9], NEGOITA [37], YAGER [10], ZADEH [11]) ont vu dans la théorie des possibilités et la logique floue le moyen de développer des systèmes experts assez semblables, dans le principe, à ce que nous connaissons déjà avec la programmation logique mais avec l'énorme différence, toutefois, qu'ils pourraient gérer des données floues. Ainsi espère-t-on parvenir à raisonner de façon "automatique" en termes imprécis.

Il faut pour cela parvenir à représenter les faits et règles qui serviront de base au raisonnement, et d'autre part pouvoir inférer à partir d'eux. On voit immédiatement que tout ceci est dans le prolongement direct de ce que nous avons vu dans les deux chapitres précédents.

1) REPRÉSENTER DES FAITS ET DES REGLES.

1.1 : Les faits.

Un fait est de la forme "X is A" où :

- X est une variable, un attribut d'un objet, un objet
- A est un SEF.

comme par exemple : "John's height is tall".

Nous avons vu qu'une telle proposition induisait une distribution de possibilité $\Pi_X = A$ avec $\pi_X(x) = \mu_A(x)$.

1.2 : Les règles.

Une règle est de la forme : "If antécédent then conséquent" càd une expression conditionelle. En termes de SEF, nous aurons : "If X is A then Y is B".

Nous avons vu que la distribution de possibilité induite par cette proposition peut être, par exemple, $\Pi_{(X,Y)} = A^{*'} \oplus B^*$ avec $\pi_{(X,Y)}(x,y) = \min(1, 1-\mu_A(x)+\mu_B(y))$.

1.3 Des faits et des règles plus complexes.

On peut représenter des faits et des règles plus complexes, du type :

- (X is **A**) et (Y is **B**) et (Z is **C**) $\Rightarrow$ $\pi_{(X,Y,Z)}(x,y,z) = \min(\mu_A(x), \mu_B(y), \mu_C(z))$.

- (X is **A**) ou (Y is **B**) $\Rightarrow$ $\pi_{(X,Y)}(x,y,z) = \text{Max}(\mu_A(x), \mu_B(y))$

- If (X is **A**) et (Y is **B**) then (Z is **C**) $\Rightarrow$ $\pi_{(X,Y,Z)}(x,y,z) = \min(1, 1-\mu_H(x,y)+\mu_C(z))$, où $\mu_H(x,y) = \min(\mu_A(x), \mu_B(y))$.

- If (X is **A**) ou (Y is **B**) then (W is **C**) et (Z is **D**) $\Rightarrow$ $\pi_{(W,X,Y,Z)}(w,x,y,z) = \min(1, 1-\mu_H(x,y)+\mu_G(w,z))$, où :

$$\mu_H(x,y) = \text{Max}(\mu_A(x), \mu_B(y)).$$
$$\mu_G(w,z) = \min(\mu_C(w), \mu_D(z))$$

...etc.

Ce qui est important à noter, c'est que les faits et règles sont représentés sous forme de distributions de possibilité. Pour ajouter des faits à notre base, "il nous suffira" de déduire de nouvelles distributions de possibilité à partir de celles qui existent déjà à l'aide des mécanismes d'inférence.

2) RÉALISER DES INFÉRENCES.

On a vu les trois grands principes d'inférence : la conjonction / particularisation, la projection et la substitution.

Le mécanisme principal avec lequel nous allons travailler est la règle compositionelle d'inférence ou modus ponens généralisé.

Schématiquement, si on a :

- if X is **A** then Y is **B** et
- X is **C**

alors on peut déduire que Y is **D**, représentable également par une distribution de possibilité.

Il faut ici se rappeler qu'il est théoriquement possible de retraduire une distribution de possibilité en une expression en langage naturel ; ceci serait fait, dans un dernier stade, pour communiquer le résultat de la recherche à l'utilisateur.

3) EXEMPLE.

Voyons sur un exemple simpliste une illustration de cette démarche, en comparaison avec un SE classique.

3.1 : Contexte flou.

• Supposons, dans notre base de faits :

- "John's height is medium"
- avec medium défini par :

150	170	190
0.5	1	0.5

• Supposons une règle :

"If X is Tall then X is Fat", avec :

- Tall défini par :

150	170	190
0	0.5	1

- Fat défini par

60	80	100
0	0.5	1

- On sait que cette règle induit une distribution définie par (Tall*' ⊕ Fat*) :

	60	80	100
150	1	1	1
170	0.5	1	1
190	0	0.5	1

• On veut connaître le poids de John : John's weight is ?

• Dans un environnement flou, cette règle est tout-à-fait applicable avec le fait dont on dispose, ce qui nous donnerait : Y is (Medium o (Tall'* ⊕ Fat*)) càd :

60	80	100
0.5	1	1

• Comme on s'y attendait, on ne retrouve pas la distribution caractéristique de "Y is Fat". Notre résultat est une autre distribution, qu'il serait possible de retraduire en langage naturel, ce qu'on pourrait peut-être faire ici par " Le poids de John est moyen-à-gros".

Nous ne revenons pas sur les problèmes de ce type de logique, dont nous avons déjà parlé, le but étant ici d'illustrer le principe.

3.2 : Contexte binaire.

On aurait plutôt vu les choses comme :

- Fait : TailleMoyenne(John).
- Règle : If GrandeTaille(X) then GrosPoids(X)
- Question : GrosPoids(John) ?
- Réponse : Non, parce que la règle n'aurait tout simplement pas pu être activée.

4) RAFFINEMENTS.

4.1 Matching partiel.

• On peut quantifier les antécédents, de façon à offrir un matching partiel, par des règles du type :

- (a) "Si la plupart des conditions de X is **A**, Y is **B**, Z is **C**, W is **D** sont satisfaites, alors U is **F**."

ou encore

- (b) "Si au moins 2 des conditions....."

Dans le cas de (b), il s'agit d'un quantificateur absolu, qu'on représentera par un SEF des réels positifs; Tandis que dans le cas (a), le SEF sera dans l'intervalle [0,1].

De telles règles sont représentées par une distribution de possibilité semblable à ce que nous avons vu :

$\pi_{(U,W,X,Y,Z)}(u,w,x,y,z) = \min(1, 1-\mu_H(x,y,z,w)+\mu_F(u))$.

La différence réside dans le calcul de $\mu_H(x,y,z,w)$ qui, sans être complexe, est cependant un peu long et surtout très technique. Nous n'entrerons donc pas dans cette évaluation. Retenons que cela peut se faire.

• Avec une telle méthode de matching partiel, tous les éléments de l'antécédent sont considérés comme également importants. Certains auteurs [9] proposent un modèle où l'on peut pondérer les éléments par un degré d'importance. On peut également donner une mesure du degré de matching pour chaque élément de l'antécédent.

Ceci nous permet de définir des seuils de matching en-deçà desquels une règle n'est pas activable ou de choisir la meilleuree règle, celle qui "matche" le mieux.

4.2 : Degré de certitude.

Pour tenir compte du fait que l'on n'est pas toujours sûr des faits ou des règles qu'on pose, on propose d'y adjoindre un degré de confiance (ou de certitude) :

- X is **A** avec certitude α.
- If X is **A** then Y is **B** avec certitude α.

Partant de la constatation que plus on met de précision et moins on peut mettre de confiance dans ce qu'on affirme,on peut donc arriver à transformer le "X is **A** avec confiance α" en "X is **B** avec confiance 1" càd tout simplement "X is **B**", où **B** sera moins précis que **A**.

La formule de transformation proposée est : $\mu_B(x) = [\min(\alpha,\mu_A(x)) + (1-\alpha)]$.

Avec les cas particuliers :

- X is **A** avec confiance 1 $\Rightarrow$ X is **B** avec **B** = **A**.
- Xis **A** avec confiance 0 $\Rightarrow$ X is **B** avec **B** = U, l'univers du discours.

De même, une règle "if X is **A** then Y is **B** avec certitude α" induit une distribution de possibilité telle que $\pi_{(X,Y)}(x,y) = (\min(H(x,y),\alpha) + (1-\alpha))$,

où $H(x,y) = \min (1 , 1-\mu_A(x)+\mu_B(y))$.

5) CONCLUSIONS.

Nous avons vu, en principe, comment réaliser des systèmes experts flous.

5.1 : Quelles sont, en résumé, les caractéristiques que veulent atteindre de tels systèmes ?

5.1.1 : Unification sémantique ou matching flou.

Dans des systèmes classiques, lorsqu'on cherche à activer une règle, il faut que l'antécédent (ou le conséquent si on travaille en chaînage arrière) matche exactement le fait (ou le sous-objectif).

Dans l'exemple que nous avons vu, la règle n'était pas activée parce que "GrandeTaille" ne matchait pas "TailleMoyenne".

Pourtant, ces deux éléments ne sont pas sans rapport et on pourrait en déduire quelque chose concernant la taille, même si ce n'est pas le conséquent tel quel : c'est d'ailleurs ce que nous faisons dans nos raisonnements quotidiens.

C'est ce que l'on veut atteindre en permettant un matching flou : permettre l'activation de la règle par des éléments qui, sans être strictement semblables, ont cependant un rapport, et moduler la déduction en fonction du degré de similitude au lieu de se limiter à un phénomène de tout ou rien. C'est ce qui fait parler d'unification sémantique (au lieu d'une unification purement syntaxique).

5.1.2 : Matching partiel.

Dans le but de nuancer encore l'unification, on veut permettre l'activation d'une règle lorsqu'une partie seulement des éléments sont unifiés.

5.1.3 : Admettre des faits imprécis, flous.

5.1.4 : Admettre des faits incertains.

5.1.5 : Admettre des règles incertaines.

5.1.6 : Admettre les règles imprécises de la logique floue.

On peut dire qu'une règle est imprécise lorsque, pour des unifications différentes, elle produit des conséquences différentes, en suivant les lois de la logique floue.

5.1.7 : Produire des conclusions imprécises et/ou incertaines.

C'est le résultat des quatre points précédents.

5.2 : Quels sont les inconvénients que l'on peut entrevoir ?

• Rappelons la discussion (voir chapitres précédents) des résultats obtenus avec de telles méthodes de raisonnement approximatif, qui sont parfois contre-intuitifs.

• De plus, il paraît difficile d'asocier dans une même règle, à la fois des éléments flous et des éléments non-flous. Théoriquement, il est possible de transformer une donnée non-floue en une donnée floue, en donnant à sa distribution de possibilité des valeurs égales à 0 ou 1. Mais là encore, les résultats sont surprenants et décevants. On peut regretter que ce problème de "mixité" des données dans un même problème ne soit pas plus souvent envisagé dans la discussion.

B. QUELQUES RÉALISATIONS.

Nous nous proposons de voir, dans ce point, dans quelle mesure les réalisations pratiques ont suivi les souhaits théoriques du point précédent et comment elles rencontrent les différentes caractéristiques que l'on voudrait voir présentées par un SE dit "gérant le flou". Nous tenterons de dresser un tableau récapitulatif en fin de ce point.

1) MYCIN.

Références : [12],[13],[14],[15].

• C'est un des exemples bien connus de logiciels médicaux, capable de gérer des données de façon "non-catégorique".

Il est cité à titre de comparaison parce qu'en fait, il ne fait pas du tout référence à la théorie des SEF. On peut dire de façon simple qu'il gère l'incertitude mais pas le flou.

• Les règles sont du type :

SI :

- 1) l'infection qui demande traitement est une méningite
- et 2) le patient n'a pas de signe d'infection sérieuse de la peau ou des tissus mous
- et 3) les organismes n'ont pas été visualisés en culture
- et 4) le type d'infection est bactérienne

ALORS on peut penser que le responsable de l'infection est un staphylocoque coagulase-positif (0.75) ou un streptocoque A (0.5).

• On s'aperçoit que les données ne sont pas floues (on aurait peut-être pu y trouver intérêt aux points 2 et 4) et que, de même, la règle ne répond pas à la logique floue.

Il n'y a pas non plus de matching sémantique ou partiel.

2) FRIL (ARIES) ET LES SUPPORTS LOGIQUES.

Références : [16],[17],[18],[19].

• Il s'agit de logiciels généraux, qui offrent un support pour l'implémentation d'applications diverses et ne se limitent pas spécifiquement au domaine médical.

• FRIL est composé de faits de la forme :

fait(X) : [Sn, Sp]

et de règles de la forme :

règle(X) : [Sn, Sp].

Les faits et les règles sont exprimées comme en PROLOG (FRIL se sert d'ailleurs de son interpréteur, après traitement par un pré-processeur) si ce n'est l'adjonction des couples [Sn, Sp] avec Sn, Sp appartenant à [0,1].

Ces couples ont la signification suivante :

- Sn : pour *necessary support* , qui donne le degré minimum de certitude
- Sp : pour *possible support* , qui donne le degré maximum de certitude.

Le moteur d'inférence est donc celui de PROLOG et il existe de plus un module dans ce logiciel qui permet de calculer les supports [Sn, Sp] au fur et à mesure de l'activation des règles. On peut spécifier différentes stratégies de calcul.

Si plusieurs chemins ont été trouvés pour prouver l'objectif (et donc avec chacun son propre support), le module de calcul combine ces supports différents pour en retirer un seul.

• Jusqu'ici, on n'a pas parlé de SEF. Cela apparaît dans la possibilité de mimer une unification sémantique. En effet, lorsque deux valeurs se rapprochent sans être strictement les mêmes, par exemple height(X, tall) et height (X, average), il est possible d'ajouter une règle :

height(X,tall) : — height(X,average)

permettant de passer de l'un à l'autre et on donne au support [Sn, Sp] de cette règle supplémentaire des valeurs qui sont fonction de la ressemblance des deux valeurs.

Pour celà, il faut disposer de la définition des SEF (tall et average) correspondants.

C'est certainement un progrès mais on ne peut pas vraiment appeler cela de l'unification sémantique au sens strict :

- d'une part parce qu'elle diffère assez nettement de celle décrite dans les systèmes experts travaillant sur les distributions de possibilité
- d'autre part, il faut ajouter une règle à chaque correspondance de valeurs que l'on désire, ce qui finit par surcharger le système.

• En ce qui concerne les autres points :

- il n'y a pas de matching partiel
- c'est surtout l'incertitude des faits que l'on gère ; le fait d'avoir un intervalle introduit une mesure d'imprécision mais il s'agit d'abord d'une imprécision sur l'incertitude, l'incertitude totale correspondant à un support [0,1].

(NB : c'est d'ailleurs ce point qui permet de ne pas se fixer l'hypothèse d'un monde fermé : un fait qui n'est pas dans la base n'est pas considéré comme faux mais est considéré comme ignoré, avec un support [0,1].

ARIES est du même genre, plus puissant parce que ses stratégies de calcul de support peuvent varier plus largement.

3) LIFTAN

Référence : [20]

• Ce système a pour but d'évaluer le risque de pathologie lombaire chez des travailleurs manuels soulevant des poids de façon répétitive.

• Pour y arriver, on procède en deux étapes :
- un ensemble de règles estime le risque en fonction de la fréquence, du poids, de la hauteur à laquelle il faut arriver et de la facilité de ramener la charge près du corps, tout celà en considérant que le travailleur est un homme jeune et en bonne santé
- ensuite, on pondère le risque obtenu en introduisant les caractéristiques du travailleur : âge, poids, force musculaire.

• Il est permis d'entrer des données "floues" : le poids, par exemple, ne se chiffre pas en kg mais est une des valeurs "léger", "moyen", "lourd", "très lourd".
De même pour les autres données.

• Les règles sont de la forme :

"SI le poids est moyen
et la fréquence élevée
et la distance par rapport au corps est faible
et la hauteur est faible
ALORS le risque est moyen".

A chaque élément i de l'antécédent, on donne un poids w_i (de 0 à 5) pour tenir compte des éléments plus importants dans l'estimation.

• D'autre part, pour chaque donnée en entrée, l'utilisateur doit préciser le degré de certitude c_i (de 0 à 1).
De cette façon, le degré de certitude du conséquent d'une règle est estimé par :

$$\sum_{\text{sur } i} (c_i * w_i) \;/\; \sum_{\text{sur } i} w_i.$$

• En conclusion :

- malgré l'apparence floue des données en entrée, le matching n'est pas sémantique. Il ne peut pas non plus être partiel.
- On tolère une imprécision dans les faits mais pas dans les règles. Avec la réserve, toutefois, que toutes les imprécisions ne sont pas permises : l'utilisateur doit choisir parmi un ensemble de termes qui lui sont présentés.
- L'incertitude est gérée au niveau des faits et des règles.
- Les conclusions sont donc également incertaines et imprécises (avec la même réserve).

4) ANALYSE D'IMAGES ECHOCARDIOGRAPHIQUES.

Références : [24],[25].

• Le but est de retrouver à quelle zone du coeur (ventricule, valvule,...etc) ou d'artefact ou d'espace mort se rapportent les images qu'on analyse.

• Les données en entrée sont floues et les données qui ne le seraient pas sont quand même considérées comme telles. On peut ainsi traiter des SEF, des éléments de SEF ou des attributs particuliers. Les SEF sont discrets ou représentent des nombres flous (qui sont des intervalles flous des réels).

A chacune de ces données est associé un degré de certitude (CF) :

- pour les SEF, il est obligatoirement égal à 1
- pour les éléments du SEF, il est égal à leur valeur de la fonction d'appartenance (μ_{SEF} (él.))
- pour les attributs autres, il est précisé par l'utilisateur.

• Les règles sont de la forme :

IF [ ] © [ ]...[ ] THEN D avec $\tau(R)$

où :

- [ ] est un pattern
- © est un connecteur logique : and, or...
- $\tau(R)$ est le degré de confiance dans la règle R ([0,1])
- K_i est la valeur du pattern i : voir ci-dessous.

Le degré de confiance postérieur (càd après activation) de R peut être trouvé par :

$\Gamma(R)$ = min (K_i, $\tau(R)$) si © = and

= min (Max(K_i), $\tau(R)$) si © = ou

• Un pattern a la forme : opérande {opérateur de relation, opérande} où :

- {} est optionnel.
- un opérande peut donc être :
  - un SEF : avec CF = 1
  - un élément de SEF : avec CF = μ_{SEF}(él)
  - un attribut particulier : avec CF attribué.

* forme 1 de pattern : opérande.

La valeur K_i du pattern est égale au CF de l'opérande.

* forme 2 de pattern : opérande - opérateur de relation - opérande càd :

(P1*, CF1) F* (P2*, CF2) où :

P1* et P2* sont les opérandes, CF1 et CF2 sont les degrés de certitude et F* est l'opérateur de relation.

La valeur K_i du pattern est alors égale à : min (CF1, CF2, P1*F*P2*) et dans le cas particulier où P1* et P2* sont des SEF (discrets ou nombres flous), alors K_i = P1*F*P2* car CF1 = CF2 = 1.

Deux cas sont particulièrement fréquents :

- les opérandes sont des nombres flous et l'opérateur donne l'équivalent des opérateurs relationnels classiques (>=, <, =...etc)
- les opérandes sont des SEF discrets et F* est une relation donnant un degré de ressemblance entre deux SEF.

• On pose un seuil minimum d'activation (α) tel qu'une règle dont le Γ(R) est < α n'est pas activée.

Une fois qu'une règle a été activée, l'action D sur les données est effectuée et le degré de confiance attaché à D est posé égal à Γ(R).

Si plusieurs règles ont la même conclusion D, on prend celle dont le Γ(R) est le plus grand.

• En conclusion :

- les données peuvent être floues ou incertaines mais elles ne peuvent pas être les deux : voir données.

- les règles peuvent être incertaines mais elles ne sont pas imprécises selon la définition que nous en avons posée puisque leur conséquence est toujours la même si elle est activée et que leur logique n'est pas floue.

On note une confusion des notions d'incertitude et de flou puisque le degré de certitude a posteriori d'une règle peut être fonction, dans certains cas (voir forme 2 du pattern) du degré de matching qui, lui, est une notion de flou.

En effet, si on prend le cas fréquent où F* est une relation floue de ressemblance entre deux SEF, la valeur K_i du pattern dépendant de cette relation influencera le degré de certitude $\Gamma(R)$. Il n'est cependant pas sûr que ceci rende la modélisation plus complexe, au contraire.

- Dans ce même cas, on note qu'il s'agit d'une forme d'unification sémantique puisqu'il est possible d'unifier la règle avec des patterns différents mais il est particulier dans le sens où la conséquence de la règle ne varie pas avec ces unifications. Ceci est révélateur du fait que la logique de la règle n'est pas floue.

- On dispose également d'une forme de matching partiel puisqu'on peut avoir des connecteurs "ou" dans l'antécédent de la règle.

-Les conclusions peuvent être incertaines ($\Gamma(R)$ attaché à D) et/ou imprécises puisque D peut être une action sur les données qui peuvent être des SEF.

C. CONCLUSION.

• Les espérances théoriques nous proposaient la réalisation de systèmes très riches et très nuancés. Il suffit, pour s'en convaincre, de voir la liste des caractéristiques que visent à atteindre de tels systèmes.

• D'un point de vue plus pratique, les choses sont beaucoup plus limitées. On ne trouve encore que peu d'applications et, à l'examen, celles-ci sont loin d'offrir tout ce qu'on souhaiterait : on se reportera au tableau-résumé page suivante.

• Cependant, la question se pose de savoir s'il faut vraiment vouloir "coller" au tableau théorique, où les pierres d'achoppement sont, encore une fois, les difficultés rencontrées dans la logique floue (dont la sémantique de l'implicateur) et la difficulté de traiter par les mêmes règles des données non-floues.

Peut-être est-il alors préférable de "maîtriser" la sémantique d'une règle par un moyen semblable à celui que l'on trouve dans le SE d'echocardiographie. Ce système semble d'ailleurs bien être, dans ce que nous avons présenté, le plus riche dans l'application du flou et dans ce qu'il permet, tout en évitant les pièges soulignés plus haut. Conséquence de cette maîtrise de la sémantique de la règle, non-laissée à la logique floue : il n'y a pas de variabilité automatique de la conséquence. Seul varie le degré de certitude.

• D'autre part, nous retiendrons du tableau comparatif que l'incertitude est plus facilement gérée que l'imprécision.

• Le matching sémantique est sans doute le point qui pourrait nous apporter le plus de bénéfices. Il nous permettrait de ne plus faire de l'activation d'une règle un phénomène de tout ou rien :

- d'abord cela libère de l'exigence d'avoir exactement la même valeur pour que le matching soit possible
- dans un deuxième temps, on pourrait répercuter cette différence au niveau de la conclusion mais cela ramène à la question centrale : comment ?

TABLEAU COMPARATIF DES POSSIBILITÉS OFFERTES PAR LES SE.

	MYCIN	FRIL	LIFTAN	ECHOCARDIO.
Matching sémantique	-	±(1)	-	±
Matching partiel	-	-	-	±
Faits imprécis	-	±(2)	±(3)	+(4)
Faits incertains	+	+	+	+(4)
Règles imprécises(log.fl.)	-	±(2)	-	-
Règles incertaines	+	+	+	+
Conclusions imprécises	-	±(2)	±(3)	+
et / ou incertaines	+	+	+	+

(1) : Matching sémantique élémentaire

(2) : Il s'agit plutôt d'une imprécision sur l'incertitude

(3) : L'imprécision est limitée par le système

(4) : L'un ou l'autre

CONCLUSION

Nous allons maintenant essayer de tirer les idées maîtresses de ce que nous avons vu.

Un schéma de diagnostic.

Finalement, on peut proposer les points essentiels suivants pour un schéma de diagnostic.

• **Tout d'abord**, on recueille les plaintes, les antécédents, les signes cliniques et les résultats des examens complémentaires du patient. Appelons tout cela les symptômes du patient.

Il apparaît nécessaire d'en établir un relevé non seulement soigné mais également nuancé, entendons par là qu'il faut joindre aux symptômes au minimum ce que nous appelerons leur degré de sévérité. Cette sévérité ne doit pas seulement tenir compte de l'intensité mais aussi des caractéristiques du symptôme comme par exemple les circonstances d'apparition d'une douleur. Il n'est d'ailleurs pas toujours facile de réaliser un mapping entre ces caractéristiques, essentiellement non-quantitatives, et un degré de sévérité. Mais cela peut déjà constituer une bonne approximation.

Si, à l'inverse, on ne note que la présence ou l'absence des symptômes, càd qu'on les voit de façon binaire, on s'expose à ce que le tableau que l'on dresse soit caricatural et insuffisant dans les informations qu'il nous donne puisque des maladies peuvent parfois présenter les mêmes symptômes, différant seulement par leur intensité. Ceci nous introduit dans le point suivant.

• **En face de la situation présente d'un patient,** il y a ce qu'on peut appeler la connaissance médicale càd l'ensemble des maladies connues, parmi lesquelles se trouve(nt) celle(s) dont est atteint le patient, et qu'on décrit par l'ensemble des symptômes qu'elle réunit.

On dira, par exemple, qu'une maladie m peut se présenter sous deux tableaux t_1 et t_2, respectivement dans 30 % et 70 % des cas et que l'on retrouve dans t_1 :

- toujours les symptômes s_1 et s_2 d'intensités respectivement forte et faible
- moins souvent s_3, d'intensité forte
- presque jamais s_4, d'intensité moyenne...etc

On voit ainsi se dégager d'une part que des symptômes sont plus typiques que d'autres et, par là, plus importants pour le diagnostic, et d'autre part que l'intensité d'un symptôme n'est pas nécessairement proportionnel à son importance.

Il faudra donc, dans la description d'une maladie, dissocier ces deux points, sévérité et importance, d'une façon ou d'une autre.

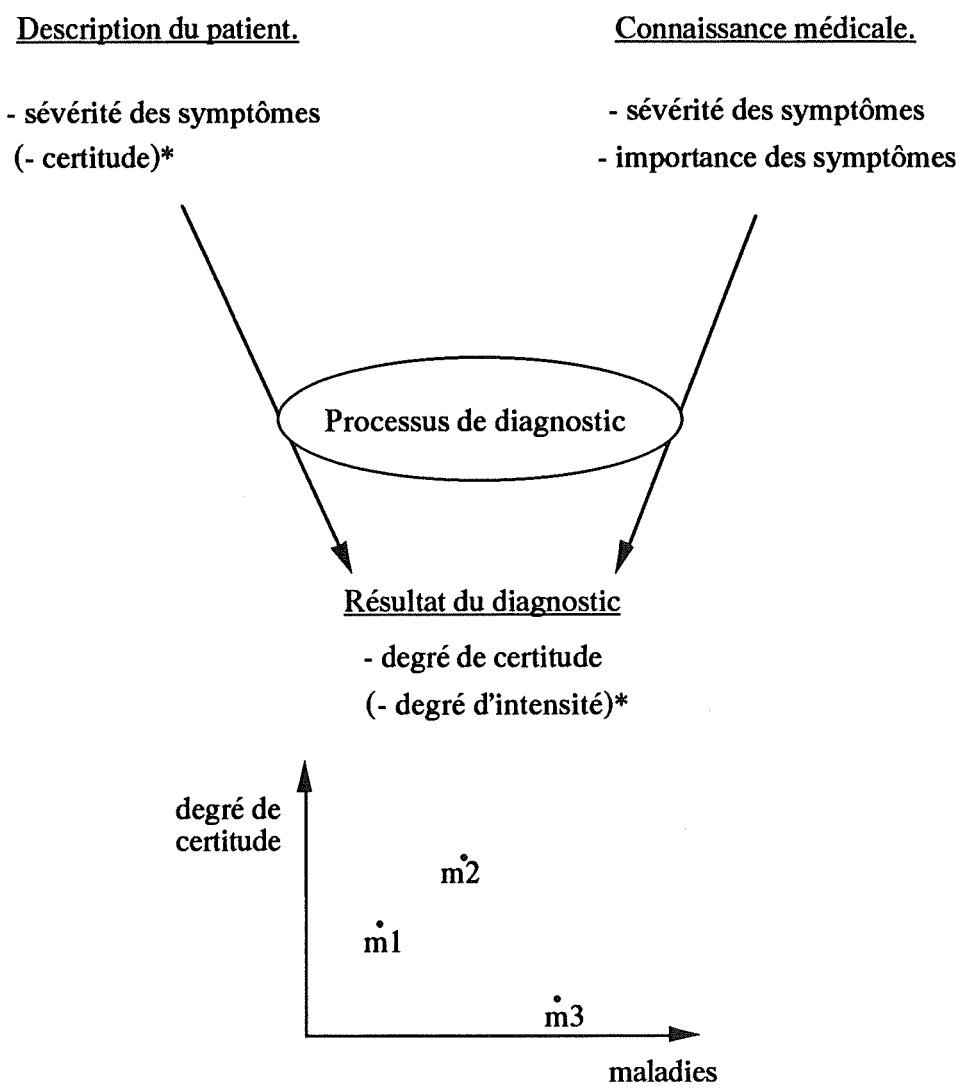
L'importance d'un symptôme sera basé sur sa fréquence et/ou son caractère discriminatif et/ou sa spécificité. Ces trois interprétations ne sont bien sûr pas équivalentes mais le plus souvent, on n'ira pas jusque là et on sera déjà très heureux de trouver une quelconque indication de l'importance.

• **Le processus de diagnostic** est difficile à cerner : la médecine n'est-elle pas "l'art de guérir" ? Connaissance, analogie, déduction, prudence, expérience...vont tour à tour y intervenir pour dégager les maladies les plus vraisemblables, avec une estimation du degré de certitude. On dira : " c'est sans doute m_1, mais on ne peut pas tout à fait exclure m_2 ; ce n'est par contre sûrement pas m_3".

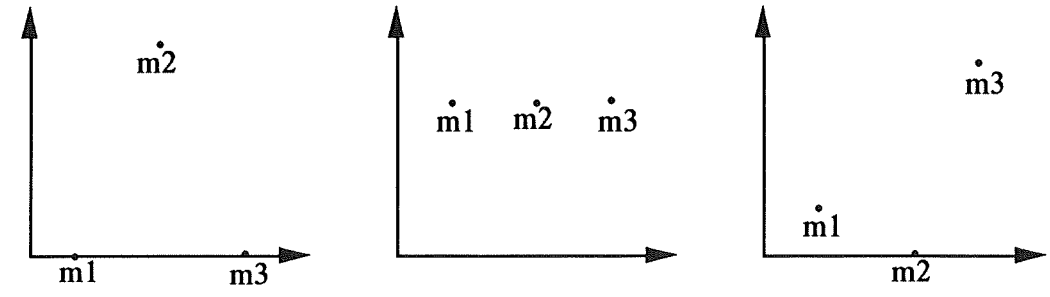
Quoi qu'il en soit, nous attendons du diagnostic qu'il ait certaines qualités, dont nous pouvons mettre en évidence les trois suivantes :

- ce n'est pas, comme nous venons de le voir, un résultat binaire au niveau de la certitude : une maladie n'est pas toujours sûrement présente ou sûrement absente.
- Le résultat doit cependant être suffisament discriminatif et ne doit pas déclarer, sans distinction, un grand nombre de maladies possibles.
- Le résultat doit être "suffisament correct" : un degré de certitude n'est pas absolu et est lui-même sujet à caution mais on ne peut pas admettre que le classement soit trop discordant par rapport à la réalité.

On peut essayer de résumer tout ceci par le schéma suivant :



ET NON PAS :



* : Ces éléments jouent également un rôle en réalité mais on peut, dans cette grossière approximation, les laisser de côté.

Comment se comportent les méthodes floues ?

A. Méthodes non-floues

Les inconvénients de méthodes non-floues ont déjà été évoquées dans le point précédent.

Sans doute le problème principal est-il d'obtenir un résultat correct, malgré les approximations binaires.

B. Comparaison matricielle.

• Sur le principe,on y reconnaît pas mal de points de notre "schéma idéal" de diagnostic.

- La modélisation des symptômes, que ce soit ceux des patients ou dans la description des maladies, peut y être très nuancée. On se reportera, pour s'en convaincre, à l'exposé de cette méthode.
- Dans le résultat, on peut en attendre les trois caractéristiques sus-citées : non-binaire, discriminatif et correct.
- On trouve un vecteur de pondération, reflétant l'importance des symptômes, bien que tout ne soit pas résolu avec cela, comme nous l'avons déjà fait remarquer. Ainsi, on ne peut pas refléter, par exemple, une situation où un symptôme est rare mais affirme à coup sûr la présence d'une maladie.

• Pratiquement, nous avons déjà fait remarqué les difficultés qu'il y aurait à l'appliquer en réalité :

- il faudrait gérer l'information manquante et, ce qui est lié, il faudrait sortir de la standardisation des inputs tels que nous l'impose la méthode jusqu'à présent.
- On se trouverait rapidement devant un nombre et une taille de matrices trop grands.

- Il s'agit, et ceci est valable pour beaucoup d'autres méthodes sauf peut-être les systèmes experts, de modèles qui offrent peu de possibilités d'explications quant à leur résultat. On leur soumet des données et il est difficile de suivre le chemin de la déduction et de justifier pourquoi une telle maladie est avancée comme vraisemblable et d'autres pas. Cela peut constituer un réel obstacle psychologique à leur utilisation.

C. Composition de relations.

Côté positif, il est possible de bien modéliser les symptômes du patient.

Côté négatif, par contre, on note deux points importants, qui peuvent expliquer les problèmes que nous avons noté avec cette méthode :

- la matrice S x M, qui est sensée représenter les maladies ne fait aucune différence entre intensité et importance des symptômes. Un seul auteur prend d'ailleurs position sur la nature des chiffres de cette relation et penche pour une mesure de l'importance. L'autre parle de "association symptôme-maladie".
- Le processus de diagnostic, par composition Max-min, nous produit des résultats non-discriminatifs, voire erronés.

D. Variables linguistiques.

Ici, le problème est un peu différent : le résultat se veut binaire puisqu'il s'agit de classer un questionnaire dans la classe pathologique ou dans la classe normale. Le problème se résume donc à éviter trop de résultats incorrects.

- Pour la modélisation des symptômes, on retrouve un poids $\gamma_{jk}^{(i)}$ qui donne une mesure de la force et du sens dans lequel va cette réponse. Il s'agit donc en quelque sorte d'une mesure de la ressemblance entre les données du questionnaire et les données-types.
- On retrouve également un vecteur de pondération des questions, qu'on peut assimiler à une mesure de l'importance des symptômes.
- Nous avons noté que le processus de "diagnostic" par fonction discriminative semblait donner de meilleurs résultats que lorsqu'on envisage un calcul de possibilité reposant sur un principe Max-min.

E. CONNECTIVITÉ ET DISCRIMINATION FLOUES.

Il s'agit de deux approches différentes, possédant des caractéristiques différentes :

a) discrimination.

- Les symptômes du patient sont considérés comme binaires.
- La description de la maladie ne fait intervenir que la fréquence avec laquelle les symptômes sont présents et ceux-ci sont de nouveau considérés comme binaires.
- Le processus de diagnostic se base sur le pouvoir discriminatif des symptômes.

Le fait de se trouver dans un contexte d'abdomen aigu, où les symptômes sont plus souvent marqués que dans d'autres domaines, doit atténuer un peu les conséquences mais on peut craindre quand même des erreurs de jugement dues au fait qu'on ne prend pas en compte l'intensité des symptômes.

b) connectivité.

- Ici, on considère les symptômes de façon variable :
  - on tient compte de l'intensité des symptômes lors de l'établissement des chaînes de connectivité et des groupes représentatifs de chaque niveau.
  - Par contre, pour la description du patient, on les considère de façon binaire.
- La mesure de l'importance des symptômes reflète quelque chose d'assez particulier : nous avions déjà évoqué la fréquence et le pouvoir discriminatif; nous avons maintenant l'interconnection entre symptômes. Seront gratifiés d'une grande importance des symptômes fortement interconnectés càd apparaissant ensemble et ce, indépendemment de leur fréquence d'apparition.

Cette méthode nous apparaît sûrement non-binaire et discriminative. La correction doit dépendre des modalités pratiques mais il semblerait qu'on puisse obtenir de bons résultats.

Et la théorie des possibilités ?

• On aborde ici une autre façon de nuancer les données. Jusqu'à présent, dans les SEF et les relations floues, un symptôme était qualifié par un degré d'intensité. Ici, c'est toute une distribution.

Y gagne-t-on en modélisation ?

Ce n'est pas sûr. Ce degré de flou supplémentaire apparaît plutôt comme une perte de précision dans l'information, qui ne semble pas nécessaire dans notre domaine. Dire, par exemple, qu'un patient a de la température, avec sa distribution de possibilité à l'appui, nous apprend moins que de donner directement un degré de sévérité. Et il en va de même pour de nombreux exemples.

On peut, bien sûr, pour aller à l'encontre de ceci, donner une distribution plus précise, ce qui n'est alors qu'une autre façon de donner un degré de sévérité.

• Pour ce qui est de la modélisation des maladies, on imagine des compositions de propositions, avec des règles et des connecteurs.

- On se rappellera tous les problèmes que nous avons rencontrés lorsqu'on laisse la sémantique de la règle à charge de la logique floue, surtout avec l'implication mais aussi parce que et se traduit par le minimum et ou par le maximum.

En effet, si on a une règle du type :

SI une personne présente x et y et z, ALORS elle a la maladie m,

une personne présentant très peu x et très peu y et très peu z "matchera" de la même façon que quelqu'un qui présente très peu x et très fort y et très fort z. Comment, dès lors, arriver à structurer les différents symptômes en importance?

D'autant que le matching partiel n'apporte que peu de solutions, parce qu'il considère tous les éléments d'une condition comme d'égale importance.

On a les problèmes équivalents dans le cas de conditions reliées par des ou.

De façon générale, on peut dire qu'il apparaît difficile de modéliser des règles exprimant la conséquence d'une conjonction ou d'une disjonction de conditions, ce qui semble devoir être fréquemment le cas dans un problème de diagnostic.

- De même, pour conclure à une maladie, il faudra bien en arriver à une donnée binaire : oui ou non, y a-t-il cette maladie, avec un degré-résultat pour chaque éventualité.

Nous avons également déjà signalé les difficultés de gestion des données binaires dans un contexte flou, faisant qu'on ne sait pas très bien que conclure des distributions de possibilité : faut-il prendre en compte le degré de possibilité "oui" qui apparaît souvent peu sensible, le degré de possibilité "non", ou les deux?

Bref, tout ceci dresse un tableau assez négatif de ce type d'approche.

• Les choses semblent aller mieux lorsqu'on gère soi-même la sémantique des règles, comme nous l'avons vu dans certains systèmes experts, et il apparaît de même plus facile d'y intégrer des données précises.

Les problèmes d'antécédents composés restent par contre entiers.

En conclusion.

• On peut finalement voir le problème en termes de modélisation et de manipulation floues.

- La modélisation peut se faire essentiellement de deux façons :
  - l'élément ou la relation entre éléments sont associés à un degré appartenant à [0, 1], nuançant ainsi la situation binaire {0, 1}
  - l'élément ou la relation sont décrits par une distribution de possibilité.

- La manipulation, elle, peut être floue ou non-floue.
  - Par manipulation floue, nous entendons des méthodes propes à cette théorie : la composition Max-min (pour la première façon de modéliser) et tout le chapitre du raisonnement flou (pour la théorie des possibilités).
  - Par manipulation non-floue, au contraire, nous entendons des méthodes classiques et non-propres au domaine du flou mais adaptées à celui-ci (ex: la composition matricielle). Nous y incluons également les systèmes experts qui proposent une adaptation au flou mais sans reposer sur la logique floue.

• Or, nous avons fait remarquer, tout au long de ce survol, que les méthodes de manipulation floues donnaient des résultats peu encourageants.

C'est assez évident pour le calcul Max-min et cela se répète dans les systèmes experts : on a beaucoup moins de problèmes dès qu'on ne laisse pas la sémantique de la règle à la logique floue et qu'on gère soi-même les conclusions et leur degré de certitude.

Les méthodes classiques adaptées au flou étaient dans l'ensemble bien meilleures, tout d'abord par rapport à celles propres au flou mais aussi par rapport à ces mêmes méthodes classiques qui travailleraient sur des données binaires.

• En d'autres termes :

-la théorie des SEF nous offre une modélisation sous forme de degrés appartenant à [0, 1], qui améliore les performances du diagnostic posé par des méthodes classiques adaptées, parce qu'il est posé en termes moins catégoriques et plus nuancés.(1)

- Par contre, lorsqu'on fait suivre cette même modélisation par une manipulation typiquement floue, on obtient des résultats beaucoup moins bons que dans (1).

- D'autre part, la modélisation sous forme de distribution de possibilité nous fait entrevoir la possibilité d'un matching sémantique, de tout grand intérêt pour affiner la déduction mais qui,de nouveau, inséré dans tout le contexte de la manipulation par le raisonnement flou, se révèle décevant à cause des problèmes qu'on y rencontre.

- La même idée de matching sémantique, évitant le phénomène de tout-ou-rien qu'est l'activation d'une règle, se révèle par contre être un enrichissement lorsqu'on l'intègre dans une gestion moins floue.

• Ceci nous amène à tirer, un peu schématiquement, les enseignements suivants :

1) La théorie des SEF est d'un grand apport dans la modélisation des données mais elle déçoit dans la manipulation de celles-ci.

2) Il y a une tendance, en effet, dans ces techniques de manipulation floues, à perdre le contrôle de ce que l'on fait et à vouloir remplacer l'analyse du problème par des lois automatiques de calcul.

Or, il apparaît inversément que dans les méthodes que nous avons rencontrées, les meilleures avaient chaque fois poussé fort loin l'analyse du problème avant d'y introduire le flou.

En effet, lorsqu'une conclusion est difficile à tirer à cause de la complexité du problème et des éléments qui y interviennent, cela ne peut pas être résolu uniquement en rendant plus imprécis les mécanismes de déduction.

La référence au flou doit, en conclusion, faire naître la nuance qui permet d'affiner des déductions autrement caricaturales mais elle ne doit pas remplacer toute analyse : l'imprécision n'est pas un but en soi.

BIBLIOGRAPHIE.

[1] L.A. ZADEH, Fuzzy Sets as a Basis for a Theory of Possibility in Fuzzy Sets and Systems 1 (1978) 3-28.

[2] L.A. ZADEH, Calculus of Fuzzy Restrictions in Fuzzy Sets and their Applications to Cognitive and Decision Processes, Academic Press, New York.

[3] L.A. ZADEH, A Theory of Approximate Reasoning in Machine Intelligence, Vol. 9, John Wiley, New York 1979.

[4] MAMDANI, GAINES, Fuzzy Reasoning and its Applications, Academic Press, London 1981.

[5] R. YAGER, Reasoning with Conjunctive Knowledge in Fuzzy Sets and Systems 28 (1988) 69-83.

[6] R. GILES, A Formal System for Fuzzy Reasoning in Fuzzy Sets and Systems 2 (1979) 233-257.

[7] B. NATVIG, Possibility Versus Probability in Fuzzy Sets and Systems 10 (1983) 31-36.

[8] MIZUMOTO, ZIMMERMAN, Comparison of Fuzzy Reasoning Methods in Fuzzy Sets and Systems 8 (1982) 253-283.

[9] DUBOIS, PRADE, TESTEMALE, Weighted Fuzzy Pattern Matching in Fuzzy Sets and Systems 28 (1988) 313-331

[10] R. YAGER, Approximate Reasoning as a Basis for Rule-Based Expert Systems in IEEE Transactions on Systems,Man and Cybernetics, Vol 14, N° 4, 1984.

[11] L.A. ZADEH, The Role of Fuzzy Logic in the Managment of Uncertainty in Expert Systems in Fuzzy Sets and Systems 11 (1983) 199-227.

[12] R. DUDA, E. SHORTLIFFE, Expert Systems Research in Science, Vol 220, N° 4594, 1983.

[13] E. SHORTLIFFE, Consultation Systems for Physicians in Expert Systems and AI Applications in Readings in Artificial Intelligence, WEBER and NILSSON, Tioga Publishing Company, 1981.

[14] SHORTLIFFE, BUCHANAN, FEIGENBAUM, Knowledge Engineering for Medical Decision Making : A Review of Computer-Based Clinical Decision Aids in Proceedings of the IEEE, Vol 67, N° 9, 1979.

[15] SHORTLIFFE, Computer-Based Medical Consultation : MYCIN, New York, Elsevier / North Holland 1976.

[16] BALDWIN, Evidential Support Logic Programming in Fuzzy Sets and Systems 24 (1987) 1-26.

[17] D'AMBROSIO, Truth Maintenance with Numeric Certainty Estimates in The Third Conference on Artificial Intelligence Applications ; Computer Society of the IEEE, 1987.

[18] APPELBAUM, RUSPINI, Aries : an Approximate Reasoning Inference Engine in Approximate Reasoning in Expert Systems, North Holland Amsterdam 1985

[19] Fril - A support Logic Programming System in Fuzzy Sets and Systems 28 (1988) 384-387.

[20] KARWOWSKI, MULHOLLAND, WARD, A Fuzzy Knowledge Base of an Expert System for Analysis of Manual Lifting Tasks in Fuzzy Sets and Systems 21 (1987) 363-374.

[21] KAUFMANN, Introduction à la théorie des sous-ensembles flous. Masson, 1975.Tome 1.

[22] SKALA, On Many-Valued Logics, Fuzzy Sets, Fuzzy Logics and their Applications in Fuzzy Sets and Systems 1 (1978) 129-149.

[23] L.A. ZADEH, Calculus of Fuzzy Restrictions in Fuzzy Sets and their Applications

[24] BUCKLEY, SILER, TUCKER, A Fuzzy Expert System in Fuzzy Sets and Systems 20 (1986) 1-16.

[25] BUCKLEY, SILER, Echocardiogram Analysis Using Fuzzy Sets and Relations in Fuzzy Sets and Systems 26 (1988) 39-48.

[26] FORDON, BEZDEK, The Application of Fuzzy Set Theory to Medical Diagnosis in Advances in Fuzzy Set Theory and Applications, North Holland 1979.

[27] ESOGBUE, ELDER, Fuzzy Sets and the Modelling of Physician Decision Processes part I in Fuzzy Sets and Systems 2 (1979) 279-291.

[28] ESOGBUE, ELDER, Fuzzy Sets and the Modelling of Physician Decision Processes part II in Fuzzy Sets and Systems 3 (1980) 1-9.

[29] ESOGBUE, ELDER, Measurement and Valuation of a Fuzzy Mathematical Model for Medical Diagnosis in Fuzzy Sets and Systems 10 (1983) 223-242.

[30] JUNIAN, YAOZENG, CHENGHAN, The Establishment of a Fuzzy Mathematic Method in Differential Diagnosis of Acute Abdomen in Approximate Reasoning in Expert Systems, North Holland 1985.

[31] SANCHEZ, Medical Diagnosis and Composite Fuzzy Relations in Advances in Fuzzy Set Theory and Applications Nort Holland 1979.

[32] VILA, DELGADO, On Medical Diagnosis Using Possibility Measures in Fuzzy Sets and Systems 10 (1983) 211-222.

[33] ADLASSNIG, Fuzzy Set Theory in Medical Diagnosis in IEEE Transactions on Systems, Man, Cybernetics, Vol 16, N° 2, 1986.

[34] SAITTA, TORASSO, Fuzzy Characterization of Coronary Disease in Fuzzy Sets and Systems 5 (1981) 245-258.

[35] KRUSINSKA, LIEBHART, A Note on the Usefulness of Linguistic Variables for Differentiating between some Respiratory Diseases in Fuzzy Sets and Systems 18 (1986) 131-142.

[36] NORRIS, PILSWORTH, BALDWIN, Medical Diagnosis from Patient Records : a Method using Fuzzy Discrimination and Connectivity Analyses. in Fuzzy Sets and Systems 23 (1987) 73-87.

[37] NEGOITA, Expert Systems and Fuzzy Systems in Benjamin/Cummings Menlo Park 1985.

Annexe au mémoire : un programme "boîte à outils" réalisant les principales opérations définies dans la théorie des SEF.

Ce programme comporte quatre grandes procédures :

- l'entrée de nouvelles relations floues.
- la modification de relations floues existantes.
- l'édition des valeurs de relations floues ou d'ensembles vulgaires existants. On peut demander d'éditer les résultats de préférence sous forme graphique mais ce ne sera accordé que dans certains cas : relations floues de une ou deux dimensions et pour lesquelles le nombre de valeurs ne dépasse pas dix dans chacune des dimensions. Si une édition graphique doit être refusée, la procédure réalisera automatiquement une présentation sous forme de liste.
- la réalisation d'opérations sur des relations existantes. On pourra ainsi demander l'union, l'intersection, la somme bornée, la différence bornée, le complément, la projection, l'extension cylindrique, la distance, la déduction d'ensembles vulgaires de niveau alpha et la transposition de relations à deux dimensions.

Il est bien sûr possible d'étendre ce programme par diverses fonctionnalités comme par exemple :

- la gestion de textes de commentaires libres sur la signification d'une relation, son origine lorsqu'elle résulte d'opérations, la correspondance entre les numéros des valeurs dans chaque dimension et leur signification réelle...
- la gestion et l'affichage des noms des fichiers correspondant aux différentes relations.
- la pondération des distances.

etc.

La plupart des procédures sont simples et ne requièrent pas de longs commentaires. Nous nous bornerons à voir le principe de l'enregistrement des relations et à expliquer les points plus délicats du programme, selon la numérotation et en complément d'autres commentaires qui se retrouvent dans le listing du programme.

A. Principe de l'enregistrement de relations n-aires floues.

1) Limites.

Soit une relation n-aire R, définie sur l'univers U_1 x...x U_n :

- elle portera un nom de 0 à 6 caractères inclus.
- n doit être compris entre 1 et 5 inclus; n est appelé le nombre de dimensions de la relation R.
- pour chaque dimension i (de 1 à n), on admet un nombre de valeurs différentes pour l'univers U_i compris entre 2 et 100 inclus. Chaque valeur portera un numéro entier.

Ces limites sont bien au-delà de ce qu'on a pu rencontrer dans l'application de la théorie des SEF au domaine médical.

2) Stockage des valeurs d'une relation.

Pour stocker les valeurs d'une relation (appelons la "rel") et pouvoir les réutiliser facilement, on va tenir les structures suivantes :

1) Un descripteur :
   - fichier de un seul record.
   - Ce record a pour structure :
     - nom : de la relation.
     - nbrdim : nombre de dimensions de la relation.
     - TabDim : array [1..5] avec TabDim[i] est le nombre de valeurs de l'univers U_i pour i allant de 1 à nbrdim et TabDim[i] = 1 pour i > nbrdim.
       Les nbrdim dimensions occupent toujours les nbrdim premières positions de TabDim. La seule exception à cette loi n'est que temporaire: durant l'opération de projection, entre les procédures PreparOp et Projection.
       Toutes ces informations nous seront utiles lors des différentes opérations.
   - Ce fichier porte le nom de la relation suivi du suffixe "ds" (pour DeScripteur) ; dans notre exemple : "relds".

2) Le fichier de valeurs de la relation :
   - Ce fichier porte le nom de la relation suivi de "dt" (pour DaTa); ici : "reldt".
   - Il est composé d'autant de records qu'il n'y a d'éléments dans la relation càd TabDim[1] *...* TabDim[nbrdim] records.
   - Chaque record est un réel compris dans l'intervalle [0,1].

- Les records sont rangés par ordre de coordonnées. Par exemple, pour une relation à deux dimensions, on aurait : (1,1),...,(1,TabDim[2]),(2,1),...,(TabDim[1],TabDim[2]).

3) Il est important de noter que ces caractéristiques sont des prérequis pour la bonne exécution du programme. Elles sont également respectées par les différentes procédures du programme.

B.Principe de l'enregistrement des ensembles vulgaires de niveau alpha.

* Le descripteur a les caractéristiques suivantes :
  - fichier de un seul record.
  - Ce record a pour structure :
    - nom : nom de l'ensemble.
    - orig : nom de la relation floue d'origine.
    - nbrdim : nombre de dimensions de la relation d'origine.
    - niveau : quel niveau alpha avait été demandé lors de la création.
  - Il porte le nom choisi pour l'ensemble suivi de "ds".

* Le fichier des éléments :
  - est composé de records qui sont des strings donnant la coordonnée d'un élément
  - porte le nom de la relation suivi de "vl" (pour VuLgaire).

C. Quelques commentaires sur les procédures du programme.

(1 *) CalculPos*

retourne la position de l'élément dont les coordonnées sont données par TabCoord, dans le fichier de valeurs décrit par nbrdim et TabDim.

(a) TabBloc[i] indique de combien de positions il faut avancer dans le fichier quand on incrémente TabCoord[i] de 1.

(b) Interm donne le résultat de CalculPos et est calculé en deux étapes :

1) de combien faut-il avancer pour arriver à la position fictive (TabCoord[1],...,TabCoord[nbrdim - 1],0).

2) puis passage à (TabCoord[1],...,TabCoord[nbrdim - 1],TabCoord[nbrdim]).

(* 2 *) EntrDonnées répond au schéma suivant :

- 2.1 : Procédure d'entrée du descripteur : construit "relds".

- 2.2 : Procédure d'entrée des données : construit "reldt" en fonction des informations déjà enregistrées dans "relds".

(* 3 *) ModifDonnées.

Voir Listing.

(* 4 *) SortieDonnées.

4.1 : MessageImposs : voir listing.

4.2 : VerifSortie.

Cette procédure effectue les opérations suivantes :

- vérification de l'existence du descripteur de la relation dont on donne le nom.
- vérification de l'existence du fichier de valeurs de cette même relation.
- vérifie s'il s'agit d'une relation floue ou d'un ensemble vulgaire.

Elle produit les résultats suivants :

- OKS = 10 ssi :
  - il existe bien le descripteur
  - et il existe bien le fichier de valeurs
  - et il s'agit d'une relation floue.
- OKS = 11 ssi :
  - il existe bien le descripteur
  - et il existe bien le fichier de valeurs
  - et il s'agit d'un ensemble vulgaire.
- OKS = 12 ssi il n'existe pas de descripteur.
- OKS = 13 ssi il existe un descripteur mais le fichier de valeurs est absent.

4.3 : VerifGraphOK.

Précondition : le nom correspond à une relation qu'il est possible d'éditer (voir 4.2).

Résultat :

- OKG = 0 ssi il est possible d'éditer le résultat de façon graphique.
- OKG = 2 ssi ce n'est pas possible parce que le nombre de dimensions dépasse 2.
- OKG = 3 ssi ce n'est pas possible parce que le nombre de valeurs dans une dimension dépasse 10, bien que le nombre de dimensions soit <= 2.

4.4 : EditGraphe.

Précondition : la relation est éditable graphiquement (4.2 et 4.3).

Affiche graphiquement à l'écran les résultats d'une relation éditable graphiquement.

4.5 : EditListe.

Précondition : la relation est éditable (4.2).

Edite selon la préférence signalée par l'utilisateur dans DemandeEcran :

- sur écran : par blocs de 20 valeurs
- sur imprimante : toutes les valeurs en une fois.

Les valeurs d'une relation floue sont affichées précédées de leur coordonnée.

(5 *) OperDonnées.*

5.1 : PreparOp.

Projection : pour qu'un string donnant les dimensions sur lesquelles on projette soit valide, il faut :

- que le nombre de dimensions de projection soit inférieur au nombre de dimensions de la relation initiale (i < ElDesc1.nbrdim)
- et que les différentes parties du string (entre les virgules) représentent des entiers
- et que chaque entier soit compris entre 1 et 5
- et que les dimensions de projection existent bien dans la relation initiale (ElDesc1.TabDim[TabDimProj[j]] <> 1).

5.10 : Projection.

- Pour rappel, le descripteur construit dans PreparOp fait exception à la règle que nous avons donnée càd que les x dimensions d'une relation occupent les x premières positions du tableau TabDim. On rétablira la règle en fin de procédure en "compactant" le tableau TabDim : voir listing.
- L'idée des boucles for imbriquées est la suivante :
  - 1) avec i, j, k, l et m, on parcourt les différents éléments de la relation-résultat à construire.
  - 2) Pour chacun d'entre eux, on parcourt les éléments dans les boucles (for a....for e) de 1 jusque consi. Cela revient à parcourir uniquement ceux qui disparaissent dans la projection puisque lorsque la dimension i est gardée dans la projection, consi vaut 1.
  - 3) On construit TabCoord en prenant les valeurs de coordonnées
    - de l'élément à construire (i,j,k,l,m) si la dimension est gardée
    - ou de l'élément du point 2) (a,b,c,d,e) si la dimension disparaît.

- 4) On prend le maximum de tous ces éléments des points 2) et 3), et on l'inscrit dans la relation-résultat. On reprend à l'élément suivant du point 1).

Cela revient à prendre chaque élément-résultat et à chercher l'élément de valeur maximum parmi tous les éléments de la relation-argument possédant les mêmes valeurs de coordonnées pour les dimensions maintenues dans la projection.

L'avantage de cette façon de procéder est qu'elle est tout-à-fait générale et permet d'autoriser toutes les combinaisons de projection voulues.

```
PROGRAM MANIPFLOU;

 (* Un certain nombre de commentaires se trouveront dans l'annexe écrite, *)
(* selon la numérotation indiquée dans ce code.                            *)


 Type st1 = string[1];
      st3 = string[3];
      st4 = string[4];
      st6 = string[6];
      st8 = string[8];
      st20 = string[20];
      Tab5 = array [1..5] of integer;
      TpElDesc = record
                   nom : st6;
                   nbrdim : integer;
                   TabDim : Tab5;
                   end;
      TpDesc = file of TpElDesc;
      TpElRel = record
                  valeur : real
                  end;
      TpRel = file of TpElRel;
      TpElRelBis = record
                     coord : st20
                     end;
      TpRelBis = file of TpElRelBis;
      TpElDescBis = record
                      nom : st6;
                      orig : st6;
                      nbrdim : integer;
                      niveau : real
                      end;
      TpDescBis = file of TpElDescBis;


 Var choix : char;

 (* 1 *)
 Function CalculPos(nbrdim : integer; TabDim, TabCoord : Tab5) : integer;

  Var Tabbloc : Tab5;
       interm, i, j : integer;

   Begin

   interm := 0;
   For i:= 1 to 5 do Tabbloc[i] := 1;
   For i:= 1 to 4 do
     begin
     For j := (i + 1) to nbrdim do Tabbloc[i] := Tabbloc[i] * TabDim[j]
     end;
         (* a *)

   For i:= 1 to (nbrdim - 1) do
```

```
                interm := interm + (TabCoord[i] - 1) * Tabbloc[i];
    interm := interm + TabCoord[nbrdim];
          (* b *)
    CalculPos := interm;
    End;


 (* 2 *)
 Procedure EntrDonnees;

 Var nom : st6;
     erreur : integer;

 (* 2.1 *)
 Procedure DemandeDescripteur(var nom : st6 ;var erreur : integer);

 Var strdim : st1;
     strvaldim : st3;
     rep : char;
     i, nbrdim, code : integer;
     TabDim : Tab5;
     Desc : TpDesc;
     ElDesc : TpElDesc;

 Begin
 erreur := 0;
 Clrscr;
 GotoXY (23,2);
 Write ('ENTREE DU DESCRIPTEUR DES DONNEES');
 GotoXY (23,3);
 Write ('-----------------------------------');
 GotoXY (3,6);
 Write ('Entrez le nom de la relation : ');
 ClrEol;
 Read (nom);
 Assign (Desc,nom + 'ds');
 {$I-} Reset(Desc) {$I+};
 If IOresult = 0 then begin
                        GotoXY (3,23);
                        Write ('Il existe déjà une relation de ce nom.');
                        Write (' Voulez-vous réellement l''écraser ?');
                        Repeat
                           begin
                           GotoXY (3,24);
                           Write ('Votre réponse (o/n) : ');
                           ClrEol;
                           Read(rep)
                           end
                         until (rep = 'o') or (rep = 'n') or (rep = 'O') or
                               (rep = 'N');
                       If (rep = 'n') or (rep = 'N')
                                then begin
                                     Close (Desc);
                                     erreur := 1;
                                     Exit
```

```
                                        end ;
                        GotoXY (3,23);ClrEol;
                        GotoXY (3,24);ClrEol;
                        end;

Repeat
   begin
   GotoXY (3,9);
   Write ('Entrez le nombre de dimensions (relation n-aire : n <= 5) : ');
  ClrEol;
   Read (strdim);
   Val (strdim,nbrdim,code)
   end
  until (code = 0) and (1 <= nbrdim) and (nbrdim <= 5);

 For i := 1 to 5 do TabDim[i] := 1;

 For i := 1 to nbrdim do
   begin
   Repeat
     begin
     GotoXY (3, 9 + 3*i);
     Write ('Entrez le nombre de valeurs (<= 100) de la dimension ',i,': ');
    ClrEol;
     Read (strvaldim);
     Val (strvaldim, TabDim[i], code)
     end
    until (code = 0) and (2 <= TabDim[i]) and (TabDim[i] <= 100);
   end;

 Rewrite (Desc);
 ElDesc.nom := nom;
 ElDesc.nbrdim := nbrdim;
 For i := 1 to 5 do ElDesc.tabDim[i] := TabDim[i];
 Write (Desc, ElDesc);
 Close (Desc)
 End;



 (* 2.2 *)
 Procedure DemandeDonnees(nom:st6);

 Var chiffre : real;
     i,j,k,l,m,code : integer;
     strchiffre : st4;
     Desc : TpDesc;
     ElDesc : TpElDesc;
     Rel : TpRel;
     ElRel : TpElRel;

 Begin
 Assign (Desc, nom + 'ds');
 Reset (Desc);
 Read (Desc,ElDesc);
```

```
Clrscr;
GotoXY (32,2);
Write ('ENTREE DES DONNEES');
GotoXY (32,3);
Writeln ('--------------------');
Assign (Rel, nom + 'dt');
Rewrite (Rel);
For i:= 1 to ElDesc.TabDim[1] do
  begin
  For j:= 1 to ElDesc.TabDim[2] do
    begin
    For k:= 1 to ElDesc.TabDim[3] do
      begin
      For l:= 1 to ElDesc.TabDim[4] do
        begin
        For m:= 1 to ElDesc.TabDim[5] do
          begin
          repeat
            begin
            GotoXY (1,whereY);
            case Eldesc.nbrdim of
             1 : write (i,' = ');
             2 : write (i,',',j,' = ');
             3 : write (i,',',j,',',k,' = ');
             4 : write (i,',',j,',',k,',',l,' = ');
             5 : write (i,',',j,',',k,',',l,',',m,' = ');
             end;
            ClrEol;
            read(strchiffre);
            Val (strchiffre,chiffre,code);
            end
          until (code = 0) and (0 <= chiffre) and (chiffre <= 1);
          Writeln;
          ElRel.valeur := chiffre;
          Write (Rel,ElRel)
          end
        end
      end
    end
  end;
Close (Desc);
Close (Rel)
End;


Begin  (* de 2 : entrdonnees *)
DemandeDescripteur(nom,erreur);
If erreur = 1 then Exit;
DemandeDonnees(nom)
End;   (* de 2 : entrdonnees *)

(* 3 *)
Procedure ModifDonnees;

Var code,i,j,erreur,posvirg : integer;
```

```
      valeur : real;
      strvaleur : st4;
      rep : char;
      arret : boolean;
      nom : st6;
      strcoord : st20;
      Tabstrcoord : array [1..5] of string[20];
      TabCoord : Tab5;
      Desc : TpDesc;
      ElDesc : TpElDesc;
      Rel : TpRel;
      ElRel : TpElRel;


Begin

ClrScr;
GotoXY (29,2);
Write ('MODIFICATION DES DONNEES');
GotoXY (29,3);
Writeln ('————————————————————————————————————');
Repeat
    begin
    GotoXY (3,5);
    Write ('Introduisez le nom de la relation à modifier : ');
    ClrEol;
    GotoXY (50,5);
    Readln (nom);
    rep := 'I';
    Assign (Desc, nom + 'ds');
    {$I-} Reset (Desc) {$I+};
    If Ioresult <> 0 then
       begin
       GotoXY (1,22);
       Writeln ('Il n''y a pas de relation de ce nom.');
       Write ('Taper "A" pour annuler et "R" pour recommencer.');
       Repeat
          begin
          GotoXY (1,24);
          Write ('Votre choix : ');
          ClrEol;
          Read(rep);
          end
        until (rep = 'A') or (rep = 'a') or (rep = 'r') or (rep = 'R')
        end;
    GotoXY (1,22); ClrEol;
    GotoXY (1,23); ClrEol;
    GotoXY (1,24); ClrEol
    end

 until (rep = 'A') or (rep = 'a') or (rep = 'I');

If (rep = 'A') or (rep = 'a') then Exit;

Assign (Rel, nom + 'dt');
```

```
 {$I-} Reset (Rel) {$I+};
 If Ioresult <> 0 then
    begin
    GotoXY (3,10);
    Writeln ('Le fichier de valeurs ',nom,' n''existe pas; modification impossi
e.');
    GotoXY (3,14);
    Writeln ('Taper une touche quelconque pour retourner au menu principal.');
   Repeat until keypressed;
    Exit
    end;

 Read (Desc,ElDesc);
 Close (Desc);
 GotoXY (1,10);
 Writeln ('Donner les coordonnées de la valeur à modifier en les séparant uniq
ent');
 Writeln ('par une virgule.');
 Writeln;
 Writeln ('Exemple : 2,1 ou 4 ou 2,4,5');
 Writeln;
 Writeln ('Taper "stop" pour arrêter.');
 GotoXY (1,23);
 Writeln ('Frapper une touche quelconque pour continuer.');
 Repeat until keypressed;
 Clrscr;
 Arret := false;

 Repeat      (* repeat n° 1 *)
    begin
    writeln;
    writeln;
    Repeat    (* repeat n° 2 : assure que la coordonnée entrée est valide   *)
       begin
        GotoXY (1,whereY);
        write ('coordonnée : ');
        ClrEol;
        read (strcoord);
        if strcoord = 'stop'
          then begin
               arret := true;
               erreur := 0
               end
          else begin
               strcoord := concat (strcoord,',');
               i := 0;
               erreur := 0;
               posvirg := pos(',',strcoord);
               while posvirg <> 0 do
                 begin
                 i := i + 1;
                 if i <= ElDesc.nbrdim then
                    begin
                    Tabstrcoord[i] := copy(strcoord,1,posvirg - 1);
                    delete (strcoord,1,posvirg);
```

```
                    posvirg := pos (',',strcoord)
                    end
                end;
                  (* tronçonne le string de coordonnée en ses différentes *)
                 (* parties séparées par des virgules                      *)

            if i <> ElDesc.nbrdim then erreur := 1;
                        (* détecte la 1° cause d'erreur *)

            For j := 1 to ElDesc.nbrdim do
                begin
                Val(TabStrCoord[j],TabCoord[j],code);
                if (code <> 0) or (TabCoord[j] > ElDesc.TabDim[j]) or
                  (TabCoord[j] < 1)
                    then erreur := 1
                end;
                        (* détecte les deux autres causes d'erreur *)

            end;

      end
     until erreur = 0;

    If arret = false then
       begin
       writeln;
       repeat begin
              GotoXY (1,WhereY);
              erreur := 0;
              write ('valeur nouvelle : ');
              ClrEol;
              read (strvaleur);
              Val(strvaleur,valeur,erreur)
              end
          until (erreur = 0) and (valeur <= 1) and (0 <= valeur);
       Seek (Rel, Calculpos(ElDesc.nbrdim,ElDesc.TabDim,TabCoord) - 1);
                          (* recherche de l'élément dans le fichier *)
       ElRel.valeur := valeur;
       Write (Rel,ElRel)  (* modification de l'élément *)
       end
   end  (* du repeat n° 1 *)

 until arret = true;    (* sort quand a lu 'stop' *)

Close (Rel)
End;


(* 4 *)
Procedure SortieDonnees;


Var choix : char;
    nom : st6;
    OKG,OKS : integer;
```

```
 (* 4.1 *)
 Procedure MessageImposs(num : integer);
        (* Affiche le message correspondant à la raison de l'impossibilité *)

    begin
    ClrScr;
    case num of
     1,12 : begin
            GotoXY (20,12);
            write ('Il n''y a pas de relation de ce nom.')
            end;
     13 : begin
          GotoXY (12,12);
          write ('Le fichier de valeurs de cette relation n''existe pas.')
          end;
     2 : begin
         GotoXY (2,12);
         write ('La relation possède plus de 2 dimensions : présentation ');
        write ('graphique impossible.')
         end;
     3 : begin
         GotoXY (5,12);
         write ('La relation possède trop de valeurs pour une présentation ');
        write ('graphique.')
         end;
     11 : begin
          GotoXY (2,12);
          write ('Le nom donné correspond à un ensemble vulgaire : pas de ');
          write ('présentation graphique.')
           end
     end;
    GotoXY (10,22);
    Write ('Frapper une touche quelconque pour continuer.');
    Repeat until keypressed
    end;


 (* 4.2 *)
 Procedure VerifSortieOK (nom : st6; var OKS : integer);
      (* Analyse s'il est possible (OKS = 10 ou 11) ou non (OKS = 12 ou 13) *)
       (* d'éditer la relation : voir annexe pour la signification          *)


Var Rel : TpRel;
     RelBis : TpRelBis;
     Desc : TpDesc;
     DescBis : TpDescBis;

 Begin

 OKS := 10;
 Assign (Desc, nom + 'ds');
 {$I-} Reset (Desc) {$I+};
```

```
 If IOresult <> 0 then begin
                        Assign (DescBis,nom + 'ds');
                        {$I-} Reset (DescBis) {$I+};
                        If IOresult <> 0
                          then begin
                               OKS := 12;
                               Exit
                               end
                          else begin
                               OKS := 11;
                               Close (DescBis)
                               end;
                        end
                   else begin
                        Close (Desc)
                        end;

 Assign (Rel, nom + 'dt');
 {$I-} Reset (Rel) {$I+};
 If IOresult <> 0 then begin
                        Assign (RelBis, nom + 'vl');
                        {$I-} Reset (RelBis) {$I+};
                        If Ioresult <> 0 then OKS := 13
                                         else begin
                                              Close (RelBis);
                                              OKS := 11
                                              end
                        end
                   else Close (Rel);


 End;

 (* 4.3 *)
 Procedure VerifGraphOK(nom:st6;var OKG : integer);
    (* Analyse s'il est possible d'éditer graphiquement une relation qu'il *)
     (* est possible d'éditer : voir annexe                                *)

Var Desc : TpDesc;
     ElDEsc : TpElDesc;

 Begin
 OKG := 0;
 Assign (Desc, nom + 'ds');
 Reset (Desc);
 Read (Desc ,ElDesc);
 If ElDesc.nbrdim > 2
     then OKG := 2
     else begin
          If Eldesc.TabDim[1] > 10 then OKG := 3;
          If ElDesc.TabDim[2] > 10 then OKG := 3
          end;
 Close (Desc);
 End;
```

```
(* 4.4 *)
Procedure EditGraphe(nom: st6);

Var Rel : TpRel;
    ElRel : TpElRel;
    Desc : TpDesc;
    ElDesc : TpElDesc;
    margeh,margev,i,j : integer;

Begin
Assign (Desc,nom + 'ds');
Reset (Desc);
Read (Desc,ElDesc);
Assign (Rel, nom + 'dt');
Reset (Rel);

If ElDesc.nbrdim = 1
  then
    begin
    Clrscr;
    GotoXY (26,1);
    Write ('Valeurs de la relation ',nom,'.');
    margeh := (80 - ((ElDesc.TabDim[1]+1) * 6)) div 2;
    margev := 10;
    GotoXY(margeh,margev);
    For i := 1 to ElDesc.TabDim[1] do write (i:4,'  ');
    GotoXY(margeh - 1,margev + 1);
    For i := 1 to ElDesc.TabDim[1] do write ('--------');
    GotoXY (margeh - 1, margev + 2);
    Write ('| ');
    For i := 1 to ElDesc.TabDim[1] do
       begin
       Read (Rel,ElRel);
       write (ElRel.valeur:3:1,' | ')
       end;
    GotoXY(margeh - 1,margev + 3);
    For i := 1 to ElDesc.TabDim[1] do write ('--------');
    Close (Rel);
    Close (Desc);

    GotoXY (1,23);
    Write ('Tapez la touche prtsc pour imprimer.');
    GotoXY (1,24);
    Write ('Tapez une touche quelconque pour continuer');
    Repeat until keypressed;
    end
  else
    begin
    ClrScr;
    GotoXY (26,1);
    Write ('Valeurs de la relation ',nom,'.');
    margeh := (80 - ((ElDesc.TabDim[2]+1) * 5)) div 2;
    margev := (24 - ((ElDesc.TabDim[1]+1) * 2)) div 2;
    GotoXY(margeh + 3,margev);
    For i := 1 to ElDesc.TabDim[2] do write (i:4,'  ');
```

```
    GotoXY(margeh + 3,margev + 1);
    For i := 1 to ElDesc.TabDim[2] do write ('--------');
    Write ('-');
    For i := 1 to ElDesc.TabDIm[1] do
      begin
      GotoXY (margeh, whereY + 1);
      Write (i:2,' ! ');
      For j := 1 to ElDesc.TabDim[2] do
       begin
       Read (Rel,ElRel);
       write (ElRel.valeur:3:1,' ! ')
       end;
      GotoXY (margeh + 3,whereY + 1);
      For j := 1 to ElDesc.TabDIm[2] do write ('--------');
      Write ('-');
      end;
    Close (Rel);
    Close (Desc);

    GotoXY (1,23);
    Write ('Tapez la touche prtsc pour imprimer.');
    GotoXY (1,24);
    Write ('Tapez une touche quelconque pour continuer');
    Repeat until keypressed;
    end;

End;

(* 4.5 *)
Procedure EditListe (nom : st6;OKS : integer);

Var numero,i,nbrdim : integer;
    Rel : TpRel;
    RelBis : TpRelBis;
    ElRel : TpElRel;
    ElRelBis : TpElRelBIs;
    Desc : TpDEsc;
    ElDesc : TpElDesc;
    ecran : boolean;
    TabCoord : Tab5;
    TabStrCoord : array [1..5] of st20;
    coord : st20;


(* 4.5.1 *)
Procedure DemandeEcran (var ecran : boolean);

 Var rep : char;

 Begin
 Repeat
   begin
   ClrScr;
   GotoXY (10,12);
   Write ('Voulez-vous les résultats sur écran : E');
```

```
    GotoXY (10,14);
    Write ('                    ou sur imprimante : I');
    GotoXY (50,22);
    Write ('Votre réponse : ');
    ClrEol;
    Read (rep)
    end
 until (rep = 'E') or (rep = 'e') or (rep = 'I') or (rep = 'i');
 If (rep = 'E') or (rep = 'e') then ecran := true
                               else ecran := false

 End;        (* de DemandeEcran *)

Begin        (* de EditListe    *)

DemandeEcran(ecran);
Case OKS of
  11 : begin
       Assign (RelBis , nom + 'v1');
       Reset (RelBis);
       If ecran
         then begin
              ClrScr;
              GotoXY (26,1);
              Write ('Valeurs de la relation ',nom,'.');
              numero := 2;
              While not eof (RelBis) do
                begin
                Read (RelBis, ElRelBis);
                If numero < 22 then numero := numero + 1
                               else begin
                                    GotoXY (10,24);
                                    Write ('Frapper une touche quelconque '
;
                                    Write ('pour continuer.');
                                    Repeat until keypressed;
                                    numero := 1;
                                    Clrscr;
                                    GotoXY (26,1);
                                    Write ('Valeurs de la relation ',nom,'.
');
                                     end;
                GotoXY (5,numero);
                Write (ElRelbis.coord);
                end;
              GotoXY (10,23);
              Write ('Affichage terminé.');
              GotoXY (10,24);
              Write ('Frapper une touche quelconque pour continuer.');
              Repeat until keypressed
              end
          else begin
              writeln (Lst,'Valeurs de la relation ',nom,'.');
              writeln (Lst,'-------------------------------------------');
              writeln (Lst);
```

```
            while not eof (RelBis) do
               begin
               Read (RelBis,ElRelBis);
               Writeln (Lst,ElRelBis.coord)
               end
            end;
      Close (RelBis)
      end;

10 : begin
      Assign (Rel, nom + 'dt');
      Reset (Rel);
      Assign (Desc , nom + 'ds');
      Reset (Desc);
      Read (Desc, ElDesc);          (* TabCoord joue le rôle de compteur :      *)
    nbrdim := ElDesc.nbrdim;     (* à tout moment,                            *)
      If ecran                     (* (TabCoord[1],...,TabCoord[nbrdim])        *)
       then begin                  (* correspond à la coord. de l'él. suivant *)
            ClrScr;                (* qui sera lu dans le fichier de valeurs. *)
            GotoXY (26,1);
            Write ('Valeurs de la relation ',nom,'.');
            numero := 2;
            For i:= 1 to 5 do TabCoord[i] := 1;
            While not eof (Rel) do
              begin
              Read (Rel, ElRel);
              If numero < 22 then numero := numero + 1
                             else begin
                                  GotoXY (10,24);
                                  Write ('Frapper une touche quelconque ');
                                  Write ('pour continuer.');
                                  Repeat until keypressed;
                                  numero := 2;
                                  ClrScr;
                                  GotoXY (26,1);
                                  Write ('Valeurs de la relation ',nom,'.');
                                  end;
              GotoXY (5,numero);

              coord := '(';
              For i := 1 to ElDesc.nbrdim do
                    begin
                    Str (TabCoord[i], TabStrCoord[i]);
                    Coord := Concat (coord, TabStrCoord[i],',')
                    end;
              Delete (coord, length(coord), 1);
              coord := Concat (coord, ')');
              Write (coord,' = ',ElRel.valeur:4:2);
                 (* compose la valeur de coord. à afficher à l'écran *)

              If TabCoord[nbrdim] + 1 <= ElDesc.TabDim[nbrdim]
               then TabCoord[nbrdim] := TabCoord[nbrdim] + 1
               else
                begin
                TabCoord[nbrdim] := 1;
```

```
            If TabCoord[nbrdim - 1] + 1 <= ElDesc.TabDim[nbrdim - 1]
             then TabCoord[nbrdim - 1] := TabCoord[nbrdim - 1] + 1
             else
              begin
              TabCoord[nbrdim - 1] := 1;
              If TabCoord[nbrdim - 2] + 1 <= ElDesc.TabDim[nbrdim - 2]
               then TabCoord[nbrdim - 2] := TabCoord[nbrdim - 2] + 1
               else
                begin
                TabCoord[nbrdim - 2] := 1;
                If TabCoord[nbrdim - 3] + 1 <= ElDesc.TabDim[nbrdim - 3]
                 then TabCoord[nbrdim - 3] := TabCoord[nbrdim - 3] + 1
                 else
                  begin
                  TabCoord[nbrdim - 3] := 1;
                  TabCoord[nbrdim - 4] := TabCoord[nbrdim - 4] + 1
                  end
                end
              end
            end
          end;
              (* incrémente le compteur TabCoord sans dépasser les *)
              (* valeurs de TabDim                                 *)

          GotoXY (10,23);ClrEol;
          Write ('Affichage terminé.');
          GotoXY (10,24); ClrEol;
          Write ('Frapper une touche quelconque pour continuer.');
          Repeat until keypressed

        end

else  (* de ecran *)
      begin
      For i:= 1 to 5 do TabCoord[i] := 1;
      writeln (Lst,'Valeurs de la relation ',nom,'.');
      writeln (Lst,'-------------------------------------------');
      writeln (Lst);
      While not eof (Rel) do
       begin
       Read (Rel, ElRel);
       coord := '(';
       For i := 1 to ElDesc.nbrdim do
              begin
              Str (TabCoord[i], TabStrCoord[i]);
              Coord := Concat (coord, TabStrCoord[i],',')
              end;
       Delete (coord, length(coord), 1);
       coord := Concat (coord, ')');
       Writeln (Lst,coord,' = ',ElRel.valeur:4:2);

       If TabCoord[nbrdim] + 1 <= ElDesc.TabDim[nbrdim]
        then TabCoord[nbrdim] := TabCoord[nbrdim] + 1
        else
         begin
```

```
                    TabCoordEnbrdim] := 1;
                    If TabCoordEnbrdim - 1] + 1 <= ElDesc.TabDimEnbrdim - 1]
                     then TabCoordEnbrdim - 1] := TabCoordEnbrdim - 1] + 1
                     else
                      begin
                      TabCoordEnbrdim - 1] := 1;
                      If TabCoordEnbrdim - 2] + 1 <= ElDesc.TabDimEnbrdim - 2]
                       then TabCoordEnbrdim - 2] := TabCoordEnbrdim - 2] + 1
                       else
                        begin
                        TabCoordEnbrdim - 2] := 1;
                        If TabCoordEnbrdim - 3] + 1 <= ElDesc.TabDimEnbrdim - 3]
                         then TabCoordEnbrdim - 3] := TabCoordEnbrdim - 3] + 1
                         else
                          begin
                          TabCoordEnbrdim - 3] := 1;
                          TabCoordEnbrdim - 4] := TabCoordEnbrdim - 4] + 1
                          end
                        end
                      end
                    end
                  end
                end;
                Close (Desc);
                Close (Rel);
      end
   end
end;   (* de EditListe *)



Begin     (* de SortieDonnées *)

Repeat    (* repeat n° 1 : recommence jusqu'à ce qu'on demande l'arrêt *)
 begin
  ClrScr;
  Repeat (* repeat n° 2 : impose un choix dans ce qui est proposé *)
     begin
     GotoXY (31,2);
     Write ('EDITION DE RESULTATS');
     GotoXY (31,3);
     Writeln ('----------------------');
     GotoXY (12,10);
     Write ('SORTIE AVEC PREFERENCE GRAPHIQUE..... : G');
     GotoXY (12,12);
     Write ('SORTIE EN LISTE..... : L');
     GotoXY (12,14);
     Write ('ARRET..... : A');
     GotoXY (50,20);
     Write ('Votre choix : ');
     ClrEol;
     Read (choix)
     end
   until (choix = 'G') or (choix = 'g') or (choix = 'L') or (choix = 'l') or
         (choix = 'A') or (choix = 'a');
```

```
  If (choix <> 'A') and (choix <> 'a') then
    begin
    ClrScr;
    GotoXY (20,12);
    Write ('Nom de la relation : ');
    ClrEol;
    Read (nom);

    VerifSortieOK(nom,OKS);
    If (OKS = 10) or (OKS = 11)
     then
       begin
       Case choix of
        'G','g' : begin
                  If OKS = 11 then begin
                                   MessageImposs (OKS);
                                   EditListe (nom, OKS)
                                   end
                              else begin
                                   VerifGraphOK (nom,OKG);
                                   If OKG = 0 then EditGraphe(nom)
                                              else begin
                                                   MessageImposs(OKG);
                                                   EditListe (nom,OKS)
                                                   end
                                   end
                  end;
        'L','l' : EditListe(nom,OKS);
        end;
       end
     else MessageImposs (OKS)
    end
  end
until (choix = 'A') or (choix = 'a');

End; (* de SortieDonnées *)


(* 5 *)
Procedure OperDonnees;

Var nomop,select : char;
    nom1,nom2,nom3 : st6;
    coderreur : integer;


(* 5.1 *)
Procedure PreparOp (nomop : char; var nom1,nom2,nom3 : st6; var coderreur : inte
ger);


Var val1,val2,val3,val4,strdim :st1;
    nbrdim,code,dimproj,posvirg,i,j,nbrdimproj,erreur : integer;
    strdimproj : st8;
```

```
    stralpha : st4;
    Desc1, Desc2, Desc3 : TpDesc;
    ElDesc1, ElDesc2, ElDesc3 : TpElDesc;
    Rel1,Rel2,Rel3 : TpRel;
    ElRel1,ElRel2,ElRel3 : TpElRel;
    rep : char;
    Desc3Bis : TpDescBis;
    Rel3Bis : TpRelBis;
    ElDesc3Bis : TpElDescBis;
    ElRel3Bis : TpElRelBis;
    alpha :real;
    StrTabVal : array [1..5] of st3;
    TabVal,TabDimProj : Tab5;
    TabStrDimProj : array [1..5] of st1;



Begin

(* I : Demande des noms et vérifications d'existence *)

coderreur := 0;
Clrscr;
Case nomop of
  'u','i','s','d','m','t' :
     begin
     GotoXY (10,10);
     Write ('Nom de la première relation - argument : ');
     Read (nom1);
     GotoXY (10,12);
     Write ('Nom de la seconde relation - argument : ');
     Read (nom2)
     end;
  else
     begin
     GotoXY (10,11);
     Write ('Nom de la relation - argument : ');
     Read (nom1);
     nom2 := ''
     end;
  end;


If (nomop <> 't') and (nomop <> 'T') then
   begin
   GotoXY (10,14);
   Write ('Nom de la relation - résultat : ');
   Read (nom3)
   end;


If (nom1 = nom3) or (nom2 = nom3) then begin
                                       coderreur := 7;
                                       Exit
                                       end;
```

```
Assign (Desc1, nom1 + 'ds');
{$I-} Reset (Desc1) {$I+};
If Ioresult <> 0 then begin
                        coderreur := 1;
                        Exit
                        end
                   else begin
                        Read (Desc1, E1Desc1);
                        Close (Desc1)
                        end;
Assign (Rel1, nom1 + 'dt');
{$I-} Reset (Rel1) {$I+};
If Ioresult <> 0 then begin
                        coderreur := 2;
                        Exit
                        end
                    else Close (Rel1);


If (nomop = 'u') or (nomop = 'i') or (nomop = 's') or (nomop = 'd') or
   (nomop = 'm') or (nomop = 't')
 then begin
      Assign (Desc2, nom2 + 'ds');
      {$I-} Reset (Desc2) {$I-};
      If IoResult <> 0 then begin
                            coderreur := 3;
                            Exit
                            end
                       else begin
                            Read (Desc2, E1Desc2);
                            Close (Desc2)
                            end;
      Assign (Rel2, nom2 + 'dt');
      {$I-} Reset (Rel2) {$I+};
      If IoResult <> 0 then begin
                            coderreur := 4;
                            Exit
                            end
                       else Close (Rel2);
      end;

If (nomop <> 't') and (nomop <> 'T')
 then
   begin
   If (nomop = 'v') or (nomop ='V') then begin
                                          Assign (Desc3bis, nom3 + 'ds');
                                          {$I-} Reset (Desc3bis) {$I+}
                                          end
                                    else begin
                                          Assign (Desc3, nom3 + 'ds');
                                          {$I-} Reset (Desc3) {$I+}
                                          end;
   If IoResult = 0 then begin
                        GotoXY (10,20);
```

```
                        Write ('Il existe déjà une relation ',nom3,'. ');
                    GotoXY (10,21);
                    Write ('Voulez-vous réellement l''écraser (o/n) ?');
                    Repeat begin
                           GotoXY (10,22);
                           Write ('Votre réponse : ');
                           ClrEol;
                           Read (rep);
                           end
                       until (rep = 'o') or (rep = 'O') or (rep = 'n')
                             or (rep = 'N');
                    If (nomop = 'v') or (nomop ='V')
                      then Close (Desc3bis)
                      else Close (Desc3);
                    If (rep = 'n') or (rep = 'N') then
                       begin
                       coderreur := 5;
                       Exit
                       end
                    end
   end;

 (* II : Demande les informations complémentaires nécessaires pour les      *)
(*       de niveau alpha, extension et projection                           *)

 Case nomop of
   'v' : begin
         Clrscr;
         Repeat begin
                GotoXY (10,12);
                Write ('Niveau alpha = ');
                ClrEol;
                Read (stralpha);
                Val (stralpha, alpha, code)
                end
            until (code = 0) and (alpha <= 1) and (alpha >= 0)
         end;
   'e' : begin
         ClrScr;
         Repeat begin
                GotoXY (10,10);
                Write ('Donnez le nombre désiré (<= 5) de dimensions : ');
               ClrEol;
               Read (strdim);
               Val (strdim, nbrdim, code);
               end
           until (code = 0) and (nbrdim <= 5) and (nbrdim > ElDesc1.nbrdim);
             (* max. 5 dimensions et plus que la relation initiale  *)

          For i := (ElDesc1.nbrdim + 1) to nbrdim do
                 begin
                 GotoXY (10, WhereY + 2);
                 repeat begin
                        GotoXY (WhereX, WhereY);
                        Write ('Nombre de valeurs de la dimension ',i,' : ');
```

```
                        ClrEol;
                        Read (StrTabVal[i]);
                        Val (StrTabVal[i],TabVal[i],code)
                        end
                    until (code = 0) and (TabVal[i] <= 100) and (TabVal[i] > 1);
                end;
            end;
'p'   :  begin
             ClrScr;
             GotoXY (3,8);
             Write ('Donnez les numéros des dimensions de projection');
            Writeln (' séparés les uns des autres par');
            Writeln (' une virgule et sans espace.');
            Writeln;
            Write ('Exemple : 1,3 ou 2 ou 1,2,4');
            repeat begin           (* recommence jusqu'à ce que le string de *)
                   erreur := 0;    (* dimensions soit valide : voir annexe   *)
                   GotoXY (10,14);
                   Write ('Dimensions de projection : ');
                   ClrEol;
                   Read (strdimproj);
                   strdimproj := concat (strdimproj,',');
                   i := 0;
                   Posvirg := Pos (',',strdimproj);
                   While posvirg <> 0 do
                      begin
                      i := i + 1;
                      if i <= 5 then
                          TabstrDimProj[i] := copy (strdimproj,1,posvirg - 1);
                      Delete (strdimproj,1,posvirg);
                      Posvirg := Pos (',',strdimproj)
                      end;
                         (* tronçonne le string en ses différentes parties *)
                         (* séparées par des virgules                      *)

                   (* i = le nombre de parties trouvées *)

                   If i < ElDesc1.nbrdim
                    then
                      begin
                      nbrdimproj := i;
                      For j := 1 to nbrdimproj do
                         begin
                         Val (TabstrDimProj[j], TabDimproj[j], code);
                         if code <> 0
                          then erreur := erreur + 1
                          else
                           begin
                           if (TabDimProj[j] > 5) or (TabDimProj[j] < 1)
                            then erreur := erreur + 1
                            else
                             begin
                             If ElDesc1.TabDim[TabDimProj[j]] = 1
                                then erreur := erreur + 1
                             end
```

```
                                    end
                                  end;
                                end
                              else erreur := 1;
                            end
                        until (erreur = 0)
                    end;
    end;


  (* III : Vérification de la compatibilité des opérations pour l'opération *)

Case nomop of
    'u','i','s','d','t' :
       begin
       If ElDesc1.nbrdim <> ElDesc2.nbrdim
           then begin
                coderreur := 6;
                Exit
                end
           else begin
                for i:= 1 to 5 do
                 begin
                 if ElDesc1.TabDim[i] <> ElDesc2.TabDim[i] then begin
                                                                coderreur := 6;
                                                               Exit
                                                               end

                 end
                end
       end;   (* obligatoirement même format pour les 2 relations *)
     'r' : If ElDesc1.nbrdim <> 2 then begin
                                        coderreur := 8;
                                        Exit
                                        end;

     'm' :
         begin
         If (ElDesc1.nbrdim <> 1) or (ElDesc2.nbrdim <> 2) or
                (ElDesc1.TabDim[1] <> ElDesc2.TabDim[1])
            then begin
                 coderreur := 6;
                 Exit
                 end
         end;
      end;



  (* IV : Ecriture du descripteur de la relation résultat *)

  If (nomop <> 't')
   then
     begin
     If (nomop = 'v') then begin
                           Assign (Desc3bis, nom3 + 'ds');
                           Rewrite (Desc3bis)
```

```
                        end
                    else begin
                        Assign (Desc3, nom3 + 'ds');
                        Rewrite (Desc3)
                        end
    end;

Case nomop of
   'u','i','s','d','c' :
      begin
      ElDesc3.nom := nom3 ;
      ElDesc3.nbrdim := ElDesc1.nbrdim;
      For i := 1 to 5 do ElDesc3.TabDim[i] := ElDesc1.TabDIm[i]
      end;
   'm' :
      begin
      ElDesc3.nom := nom3 ;
      ElDesc3.nbrdim := 1;
      ElDesc3.TabDim[1] := ElDesc2.TabDim[1];
      For i := 2 to 5 do ElDesc3.TabDim[i] := 1
      end;
   'e' :
      begin
      ElDesc3.nom := nom3 ;
      ElDesc3.nbrdim := nbrdim;
      For i := 1 to ElDesc1.nbrdim do ElDesc3.TabDim[i] := ElDesc1.TabDim[i];
     For i := (ElDesc1.nbrdim + 1) to nbrdim do ElDesc3.TabDim[i] := TabVal[i]
      For i := (nbrdim + 1) to 5 do ElDesc3.TabDim[i] := 1
      end;
   'r' : begin
           ElDesc3.nom := nom3;
           ElDesc3.nbrdim := ElDesc1.nbrdim;
           ElDesc3.TabDim[1] := ElDEsc1.TabDim[2];
           ElDesc3.TabDim[2] := ElDesc1.TabDim[1];
           For i := 3 to 5 do ElDesc3.TabDim[i] := ElDesc1.TabDim[i]
           end;
   'p' :
      begin
      ElDesc3.nom := nom3 ;
      ElDesc3.nbrdim := nbrdimproj;
      For i := 1 to 5 do ElDesc3.TabDim[i] := 1;
      For i := 1 to nbrdimproj do
         begin
         ElDesc3.TabDim[TabDimProj[i]] := ElDesc1.TabDim[TabDimProj[i]]
         end
      end;    (* pour les dimensions i disparues dans la projection, TabDim[i]
              (* est remis à 1                                                  *
  'v' :
      begin
      ElDesc3bis.nom := nom3 ;
      ElDesc3bis.orig := nom1 ;
      ElDesc3bis.nbrdim := ElDesc1.nbrdim;
      ElDesc3bis.niveau := alpha
      end;
   end;
```

```
If (nomop <> 't')
 then
   begin
   If (nomop = 'v') then begin
                         write (Desc3bis,E1Desc3bis);
                         close (Desc3bis)
                         end
                    else begin
                         write (Desc3,E1Desc3);
                         close (Desc3)
                         end
   end;


End;  (* de PréparOp *)


(* 5.2 *)
Procedure Message(coderreur : integer; nomop : char);


Begin
Clrscr;
Case coderreur of
   1 : begin
       GotoXY (10,12);
       if (nomop = 'u') or (nomop = 'i') or (nomop = 'd') or (nomop = 's') or
         (nomop = 'm') or (nomop = 't')
      then
          writeln ('La première relation - argument n''existe pas.')
       else
          writeln ('La relation argument n''existe pas')
       end;

   2 : begin
       GotoXY (3,12);
       if (nomop = 'u') or (nomop = 'i') or (nomop = 'd') or  (nomop = 's') o
         (nomop = 'm') or (nomop = 't')
      then
          begin
          write ('Le fichier de valeurs de la première relation - argument');
         writeln (' n''existe pas.')
         end
      else
          begin
          write ('Le fichier de valeurs de la relation - argument');
          writeln (' n''existe pas.')
          end;
       end;
   3 : begin
       GotoXY (10,12);
       writeln ('La seconde relation - argument n''existe pas.')
       end;
```

```
   4 : begin
       GotoXY (3,12);
       writeln ('Le fichier de valeurs de la seconde relation n''existe pas.')
      end;
   5 : begin
       GotoXY (10,12);
       writeln ('Refus d''écraser la relation résultat.')
       end;
   6 : begin
       GotoXY (3,12);
       write ('Les dimensions des relations - arguments sont incompatibles ');
      write ('avec ');
       GotoXY (30,13);
       write ('l''opération demandée.')
       end;
   7 : begin
       GotoXY (2,12);
       write ('Il n''est pas permis de mettre le résultat dans une des ');
      write ('relations initiales.')
       end;
   8 : begin
       GotoXY (9,12);
       write ('La relation doit posséder deux dimensions pour cette opération.'
      end;
   end;
GotoXY (3,22);
Write ('Frapper une touche quelconque pour revenir au menu initial.');
Repeat until keypressed
End;       (* Message *)



(* 5.3 *)
Procedure Union;

Var nom1,nom2,nom3 : st6;
    Rel1, Rel2, Rel3 : TpRel;
    ElRel1, ElRel2, ElRel3 : TpElRel;
    nomop : char;
    coderreur : integer;

Begin
nomop := 'u';
PreparOp(nomop,nom1,nom2,nom3,coderreur);
If coderreur <> 0
   then begin
        Message (coderreur,nomop);
        Exit
        end
   else begin
        Assign (Rel1, nom1 + 'dt');
        Assign (Rel2, nom2 + 'dt');
        Assign (Rel3, nom3 + 'dt');
        Reset (Rel1);
        Reset (Rel2);
```

```
        Rewrite (Rel3);
        While not eof(Rel1) do
            begin
            Read (Rel1,ElRel1);
            Read (Rel2, ElRel2);
            If ElRel1.valeur > ElRel2.valeur then write (Rel3, ElRel1)
                                             else write (Rel3, ElRel2)
            end;
        Close (Rel1);
        Close (Rel2);
        Close (Rel3)
        end
End;


(* 5.4 *)
Procedure Intersection;

Var nom1,nom2,nom3 : st6;
    Rel1, Rel2, Rel3 : TpRel;
    ElRel1, ElRel2, ElRel3 : TpElRel;
    nomop : char;
    coderreur : integer;

Begin
nomop := 'i';
PreparOp(nomop,nom1,nom2,nom3,coderreur);
If coderreur <> 0
   then begin
        Message (coderreur,nomop);
        Exit
        end
   else begin
        Assign (Rel1, nom1 + 'dt');
        Assign (Rel2, nom2 + 'dt');
        Assign (Rel3, nom3 + 'dt');
        Reset (Rel1);
        Reset (Rel2);
        Rewrite (Rel3);
        While not eof(Rel1) do
            begin
            Read (Rel1,ElRel1);
            Read (Rel2, ElRel2);
            If ElRel1.valeur < ElRel2.valeur then write (Rel3, ElRel1)
                                             else write (Rel3, ElRel2)
            end;
        Close (Rel1);
        Close (Rel2);
        Close (Rel3)
        end
nd;


* 5.5 *)
rocedure SommeBornee;
```

```
 nom1,nom2,nom3 : st6;
 Rel1, Rel2, Rel3 : TpRel;
 ElRel1, ElRel2, ElRel3 : TpElRel;
 nomop : char;
 coderreur : integer;
 somme : real;

in
op := 's';
parOp(nomop,nom1,nom2,nom3,coderreur);
coderreur <> 0
then begin
     Message (coderreur,nomop);
     Exit
     end
else begin
     Assign (Rel1, nom1 + 'dt');
     Assign (Rel2, nom2 + 'dt');
     Assign (Rel3, nom3 + 'dt');
     Reset (Rel1);
     Reset (Rel2);
     Rewrite (Rel3);
     While not eof(Rel1) do
        begin
        Read (Rel1,ElRel1);
        Read (Rel2, ElRel2);
        somme := ElRel1.valeur + ElRel2.valeur;
        If somme < 1 then ElRel3.valeur := somme
                     else ElRel3.valeur := 1;
        write (Rel3, ElRel3)
        end;
    Close (Rel1);
    Close (Rel2);
    Close (Rel3)
    end
;

5.6 *)
cedure Diffbornee;

 nom1,nom2,nom3 : st6;
 Rel1, Rel2, Rel3 : TpRel;
 ElRel1, ElRel2, ElRel3 : TpElRel;
 nomop : char;
 coderreur : integer;
 diff : real;

.n
op := 'd';
arOp(nomop,nom1,nom2,nom3,coderreur);
oderreur <> 0
hen begin
     Message (coderreur,nomop);
     Exit
```

```
      end
  .se begin
      Assign (Rel1, nom1 + 'dt');
      Assign (Rel2, nom2 + 'dt');
      Assign (Rel3, nom3 + 'dt');
      Reset (Rel1);
      Reset (Rel2);
      Rewrite (Rel3);
      While not eof(Rel1) do
         begin
         Read (Rel1,ElRel1);
         Read (Rel2, ElRel2);
         diff := ElRel1.valeur - ElRel2.valeur;
         If diff > 0  then ElRel3.valeur := diff
                      else ElRel3.valeur := 0;
         write (Rel3, ElRel3)
         end;
    Close (Rel1);
    Close (Rel2);
    Close (Rel3)
    end
;


5.7 *)
cedure Complement;

 nom1,nom2,nom3 : st6;
 Rel1, Rel3 : TpRel;
 ElRel1, ElRel3 : TpElRel;
 nomop : char;
 coderreur : integer;

gin
nop := 'c';
eparOp(nomop,nom1,nom2,nom3,coderreur);
 coderreur <> 0
 then begin
      Message (coderreur,nomop);
      Exit
      end
 else begin
      Assign (Rel1, nom1 + 'dt');
      Assign (Rel3, nom3 + 'dt');
      Reset (Rel1);
      Rewrite (Rel3);
      While not eof(Rel1) do
         begin
         Read (Rel1,ElRel1);
         ElRel3.valeur := 1 - ElRel1.valeur;
         write (Rel3, ElRel3)
         end;
    Close (Rel1);
    Close (Rel3)
    end
```

```
End;


(* 5.8 *)
Procedure Maxmin;

Var nom1,nom2,nom3 : st6;
    Rel1, Rel2, Rel3 : TpRel;
    ElRel1, ElRel2, ElRel3 : TpElRel;
    nomop : char;
    coderreur,depart,position : integer;
    max,min : real;
    Desc : TpDesc;
    ElDesc : TpElDesc;

Begin
nomop := 'm';
PreparOp(nomop,nom1,nom2,nom3,coderreur);
Assign (Desc,nom2 + 'ds');
Reset (Desc);
Read (Desc, ElDesc);
Close (Desc);
If coderreur <> 0
   then begin
        Message (coderreur,nomop);
        Exit
        end
   else begin
        depart := 0;
        Assign (Rel1, nom1 + 'dt');
        Assign (Rel2, nom2 + 'dt');
        Assign (Rel3, nom3 + 'dt');
        Reset (Rel2);
        Rewrite (Rel3);
        While not eof(Rel2) do
           begin
           max := 0;
           Reset (Rel1);
           depart := depart + 1;
           position := depart;
           While not eof (Rel1) do
             begin
             Read (Rel1,ElRel1);
             Seek (Rel2,position - 1);
             Read (Rel2, ElRel2);
             If ElRel1.valeur < ElRel2.valeur then min := ElRel1.valeur
                                              else min := ElRel2.valeur;
            If min > max then max := min;
            position := position + ElDesc.TabDim[2];
            end;
          ElRel3.valeur := max;
          write (Rel3, ElRel3);
          Close (Rel1)
          end;
       Close (Rel2);
```

```
        Close (Rel3)
        end
End;


(* 5.9 *)
Procedure ExtensionCyl;

Var nom1,nom2,nom3 : st6;
    Desc1,Desc3 : TpDesc;
    ElDesc1,ElDesc3 : TpElDesc;
    Rel1, Rel2, Rel3 : TpRel;
    ElRel1, ElRel2, ElRel3 : TpElRel;
    nomop : char;
    coderreur,insert,idem,i,j : integer;

  Begin
  nomop := 'e';
  PreparOp(nomop,nom1,nom2,nom3,coderreur);
  If coderreur <> 0
     then begin
          Message (coderreur,nomop);
          Exit
          end
     else begin
          Assign (Rel1, nom1 + 'dt');
          Assign (Rel3, nom3 + 'dt');
          Assign (Desc1, nom1 + 'ds');
          Assign (Desc3, nom3 + 'ds');
          Reset (Desc1);
          Reset (Desc3);
          Read (Desc1,ElDesc1);
          Read (Desc3,ElDesc3);
          Reset (Rel1);
          Rewrite (Rel3);

          Insert := 1;
          For j := (ElDesc1.nbrdim + 1) to 5 do
                                Insert := insert * ElDesc3.TabDim[j];
              (* calcule le nombre de fois qu'il faudra dupliquer un élément *)

          While not eof(Rel1) do
              begin
              Read (Rel1,ElRel1);
              ElRel3.valeur := ElRel1.valeur;
              For i := 1 to insert do write (Rel3, ElRel3);
              end;
         Close (Rel1);
         Close (Rel3);
         Close (Desc1);
         Close (Desc3);
         end
  End;

  (* 5.10 *)
```

```
Procedure Projection;

Var nom1,nom2,nom3 : st6;
    Desc1,Desc3 : TpDesc;
    ElDesc1,ElDesc3 : TpElDesc;
    Rel1, Rel2, Rel3 : TpRel;
    ElRel1, ElRel2, ElRel3 : TpElRel;
    nomop : char;
    max : real;
    TabCoord : Tab5;
    coderreur,a,b,c,d,e,i,j,k,l,m,v,w,x,y,z,cons1,cons2,cons3,cons4,cons5 : inte
ger;
    t,u,posvide,posnonvide : integer;
    arret,echange : boolean;

Begin
nomop := 'p';
PreparOp(nomop,nom1,nom2,nom3,coderreur);
If coderreur <> 0
   then begin
        Message (coderreur,nomop);
        Exit
        end
   else begin
        Assign (Rel1, nom1 + 'dt');
        Assign (Rel3, nom3 + 'dt');
        Assign (Desc1, nom1 + 'ds');
        Assign (Desc3, nom3 + 'ds');
        Reset (Desc1);
        Reset (Desc3);
        Read (Desc1,ElDesc1);
        Read (Desc3,ElDesc3);
        Reset (Rel1);
        Rewrite (Rel3);

        cons1 := ElDesc1.TabDim[1] - ElDesc3.TabDim[1] + 1;
        cons2 := ElDesc1.TabDim[2] - ElDesc3.TabDim[2] + 1;
        cons3 := ElDesc1.TabDim[3] - ElDesc3.TabDim[3] + 1;
        cons4 := ElDesc1.TabDim[4] - ElDesc3.TabDim[4] + 1;
        cons5 := ElDesc1.TabDim[5] - ElDesc3.TabDim[5] + 1;
            (* si la dimension i est gardée dans la projection, consi = 1 *)

        For i := 1 to ElDesc3.TabDim[1] do          (* voir annexe *)
         begin
         For j := 1 to ElDesc3.TabDim[2] do
          begin
          For k := 1 to ElDesc3.TabDim[3] do
           begin
           For l := 1 to ElDesc3.TabDim[4] do
            begin
            For m := 1 to ElDesc3.TabDim[5] do
             begin
             max := 0;
             For a := 1 to cons1 do
              begin
```

```
          For b := 1 to cons2 do
           begin
           For c := 1 to cons3 do
            begin
            For d := 1 to cons4 do
             begin
             For e := 1 to cons5 do
              begin
              if cons1 = 1 then v := i
                           else v := a;
              if cons2 = 1 then w := j
                           else w := b;
              if cons3 = 1 then x := k
                           else x := c;
              if cons4 = 1 then y := l
                           else y := d;
              if cons5 = 1 then z := m
                           else z := e;
              TabCoord[1] := v;
              TabCoord[2] := w;
              TabCoord[3] := x;
              TabCoord[4] := y;
              TabCoord[5] := z;

              Seek (Rel1,CalculPos(ElDesc1.nbrdim,ElDesc1.TabDim,TabCoord) -

              Read (Rel1,ElRel1);
              If ElRel1.valeur > max then max := ElRel1.valeur
              end
             end
            end
           end
          end;
        ElRel3.valeur := max;
        Write (Rel3, ElRel3);
        end
      end
     end
    end
  end;

 t := 0;                (* restructure le descripteur de façon à ce que *)
posvide := 0;          (* les nbrdim dimensions occupent les nbrdim     *)
arret := false;        (* premières positions de TabDim                 *)

while arret = false do
  begin
  while (t < 5) and (posvide = 0) do       (* positionne posvide sur *)
    begin                        (* le 1° él. de TabDim = 1 à partir de t *)
    t := t + 1;
    If ElDesc3.TabDim[t] = 1 then posvide := t
    end;
      (* sortie si on a parcouru les 5 dim. ou si on a trouvé *)
      (* posvide                                              *)
```

```
    if posvide <> 0
     then          (* positionne posnonvide                              *)
        begin      (* sur le 1° él. de TabDim <> 1 et suivant posvide *)
        u := t;
        posnonvide := t;
        while (u < 5) and (posnonvide = t) do
          begin
          u := u + 1;
          If ElDesc3.TabDim[u] <> 1 then begin
                                         echange := true;
                                         posnonvide := u
                                         end
          end;


        If echange = true
           then begin
                ElDesc3.TabDIm[posvide] := ElDesc3.TabDim[posnonvide];
                ElDesc3.TabDim[posnonvide] := 1
                end;
        posvide := 0;
        echange := false;
        end
     else
        arret := true

    end;

  Reset (Desc3);
  Write (Desc3,ElDesc3);
  Close (Rel3);
  Close (Rel1);
  Close (Desc1);
  Close (Desc3);
  end;



.11 *)
edure Distance;

nom1,nom2,nom3 : st6;
ElDesc1,ElDesc3 : TpElDesc;
Rel1,Rel2 : TpRel;
ElRel1, ElRel2 : TpElRel;
nomop : char;
coderreur : integer;
dist : real;

n
p := 't';
arOp(nomop,nom1,nom2,nom3,coderreur);
oderreur <> 0
hen begin
    Message (coderreur,nomop);
```

```
     Exit
     end
lse begin
     Assign (Rel1, nom1 + 'dt');
     Assign (Rel2, nom2 + 'dt');
     Reset (Rel1);
     Reset (Rel2);
     dist := 0;
     While not eof (Rel1) do
        begin
        Read (Rel1,ElRel1);
        Read (Rel2,ElRel2);
        If ElRel1.valeur > ElRel2.valeur
          then dist := dist + ElRel1.valeur - ElRel2.valeur
          else dist := dist + ElRel2.valeur - ElRel1.valeur
        end;
     Close (Rel1);
     Close (Rel2)
     end;

scr;
oXY (25,12);
.e ('La distance entre ',nom1,' et ',nom2,' est : ',dist:7:2);
oXY (3,20);
te ('Frappez une touche quelconque pour revenir au menu initial.');
eat until keypressed;


5,12 *)
tedure NiveauAlpha;

 nom1,nom2,nom3 : st6;
 coord : st20;
 Desc1 : TpDesc;
 Desc3Bis : TpDescBis;
 ElDesc1 : TpElDesc;
 ElDesc3bis : TpElDescBis;
 Rel1: TpRel;
 Rel3bis : TpRelBis;
 ElRel1 : TpElRel;
 ElRel3bis : TpElRelBis;
 nomop : char;
 coderreur,i,nbrdim : integer;
 alpha : real;
 TabCoord : Tab5;
 TabStrCoord : array [1..5] of st20;

n
op := 'v';
arOp(nomop,nom1,nom2,nom3,coderreur);
oderreur <> 0
hen begin
     Message (coderreur,nomop);
     Exit
     end
```

```
else begin
     Assign (Rel1, nom1 + 'dt');
     Assign (Rel3bis, nom3 + 'vl');
     Assign (Desc1, nom1 + 'ds');
     Assign (Desc3bis, nom3 + 'ds');
     Reset (Desc1);
     Reset (Desc3bis);
     Read (Desc1,ElDesc1);
     Read (Desc3bis,ElDesc3bis);
     Reset (Rel1);
     Rewrite (Rel3bis);
     alpha := ElDesc3bis.niveau;
     nbrdim := ElDesc1.nbrdim;

     For i:= 1 to 5 do TabCoord[i] := 1;
     While not eof (Rel1) do
        begin
        Read (Rel1,ElRel1);
        If ElRel1.valeur >= alpha
          then
           begin
           coord := '(';
           For i := 1 to ElDesc1.nbrdim do
               begin
               Str (TabCoord[i], TabStrCoord[i]);
               Coord := Concat (coord, TabStrCoord[i],',')
               end;
           Delete (coord, length(coord), 1);
           coord := Concat (coord, ')');
           ElRel3bis.coord := coord;
           Write (Rel3bis, ElRel3bis)
           end;            (* construction d'un string coord à partir de *)
                          (* TabCoord, pour l'affichage                    *)

          (* TabCoord joue le rôle de compteur ; cfr sup : proc. Sortie *)
        If TabCoord[nbrdim] + 1 <= ElDesc1.TabDim[nbrdim]
          then TabCoord[nbrdim] := TabCoord[nbrdim] + 1
          else
           begin
           TabCoord[nbrdim] := 1;
           If TabCoord[nbrdim - 1] + 1 <= ElDesc1.TabDim[nbrdim - 1]
            then TabCoord[nbrdim - 1] := TabCoord[nbrdim - 1] + 1
            else
             begin
             TabCoord[nbrdim - 1] := 1;
             If TabCoord[nbrdim - 2] + 1 <= ElDesc1.TabDim[nbrdim - 2]
              then TabCoord[nbrdim - 2] := TabCoord[nbrdim - 2] + 1
              else
               begin
               TabCoord[nbrdim - 2] := 1;
               If TabCoord[nbrdim - 3] + 1 <= ElDesc1.TabDim[nbrdim - 3]
                then TabCoord[nbrdim - 3] := TabCoord[nbrdim - 3] + 1
                else
                  begin
                  TabCoord[nbrdim - 3] := 1;
```

```
                    TabCoord[nbrdim - 4] := TabCoord[nbrdim - 4] + 1
                    end
                end
            end
        end
    end;

s: (Def   d

;

5.12 *)
cedure Transposition;
 nom1,nom2,nom3 : st6;
 Rel1, Rel3 : TpRel;
 ElRel1, ElRel3 : TpElRel;
 nomop : char;
 TabCoord : Tab5;
 Desc : TpDesc;
 ElDesc : TpElDesc;
 coderreur,position,i,j : integer;

gin
nop := 'r';
eparOp(nomop,nom1,nom2,nom3,coderreur);
 coderreur <> 0
 then begin
      Message (coderreur,nomop);
      Exit
      end
 else begin
      Assign (Desc,nom1 + 'ds');
      Reset (Desc);
      Read (Desc,ElDesc);
      Close (Desc);
      Assign (Rel1, nom1 + 'dt');
      Assign (Rel3, nom3 + 'dt');
      Reset (Rel1);
      Rewrite (Rel3);
      For i := 1 to 5 do TabCOord[i] := 1;
      For j := 1 to ElDesc.TabDim[2] do
        begin
        For i := 1 to ElDesc.TabDim[1] do
          begin
          TabCoord[1] := i;
          Seek (Rel1,CalculPos(ElDesc.nbrdim,ElDesc.TabDim,TabCoord) -1);
         Read (Rel1, ElRel1);
         Write (Rel3, ElRel1)
         end
       end;
    Close (Rel1);
     Close (Rel3)
```

```
      end
;


5.13 *)
cedure AffichMenuOp( var select : char);

in
Scr;
oXY (24,2);
te ('MENU DES OPERATIONS REALISABLES');
oXY (24,3);
teln ('------------------------------------------');
oXY (15,6);
te ('UNION.......................................... : U');
oXY (15,7);
te ('INTERSECTION................................... : I');
oXY (15,8);
te ('SOMME BORNEE................................... : S');
oXY (15,9);
te ('DIFFERENCE BORNEE.............................. : D');
oXY (15,10);
te ('COMPLEMENT..................................... : C');
oXY (15,11);
te ('COMPOSITION MAX - MIN.......................... : M');
oXY (15,12);
te ('PROJECTION..................................... : P');
oXY (15,13);
te ('EXTENSION CYLINDRIQUE.......................... : E');
oXY (15,14);
te ('DISTANCE....................................... : T');
oXY (15,15);
te ('ENSEMBLE VULGAIRE DE NIVEAU ALPHA.............. : V');
oXY (15,16);
te ('TRANSPOSITION (2 DIM.)......................... : R');
oXY (15,17);
te ('ARRET.......................................... : A');

oXY (50,21);
te ('Votre choix : ');
dln (select);

;


in  (* de OperDonnées *)
eat
egin
repeat
   AffichMenuOp(select)
 until (select = 'u') or (select = 'U') or (select = 'i') or (select = 'I')
 or (select = 's') or (select = 'S') or (select = 'd') or (select = 'D')
  or (select = 'c') or (select = 'C') or (select = 'm') or (select = 'M')
 or (select = 'p') or (select = 'P') or (select = 'e') or (select = 'E')
  or (select = 't') or (select = 'T') or (select = 'v') or (select = 'V')
```

```
 or (select = 'r') or (select = 'R') or (select = 'a') or (select = 'A');

case select of
   'u','U' : Union;
   'i','I' : Intersection;
   's','S' : SommeBornee;
   'd','D' : DiffBornee;
   'c','C' : Complement;
   'm','M' : Maxmin;
   'p','P' : Projection;
   'e','E' : ExtensionCyl;
   't','T' : Distance;
   'v','V' : NiveauAlpha;
   'r','R' : Transposition
    end;

end
til (select = 'a') or (select = 'A')

d;   (* de OperDonnées *)




in    (* de ManipFlo *)
eat ClrScr;
    GotoXY (34,5);
    Write ('MENU PRINCIPAL');
    GotoXY (34,6);
    Write ('--------------');
    GotoXY (20,8);
    Write ('Entrée de nouvelles données          :  E');
    GotoXY (20,10);
    Write ('Modification de données existantes   :  M');
    GotoXY (20,12);
    Write ('Sortie de résultats                  :  S');
    GotoXY (20,14);
    Write ('Réalisation d''opérations             :  O');
    GotoXY (20,16);
    Write ('Arrêt du programme                   :  A');
    GotoXY (50,22);
    Write ('Votre choix : ');
    ClrEol;
    Read (choix);

    Case choix of
      'e','E' : EntrDonnees;
      'm','M' : ModifDonnees;
      's','S' : SortieDonnees;
      'o','O' : OperDonnees;
     end;

ntil (choix = 'A') or (choix = 'a')
```