

Sc. N. 72/126^A

système de gestion
et d'exploitation documentaire
d'un corpus de dossiers médicaux

THESE

pour l'obtention du
doctorat de spécialité
mathématiques (3^e cycle)



soutenue devant le jury le 9 décembre 1972

par

patrick germain

jury : M. J. LEGRAS Président
Mme M. CREHANGE Examineurs
M. J. MARTIN
M. C. PAIR

système de gestion
et d'exploitation documentaire
d'un corpus de dossiers médicaux

THESE

pour l'obtention du
doctorat de spécialité
mathématiques (3e cycle)



soutenue devant le jury le 9 décembre 1972

par

patrick germain

jury : M. J. LEGRAS Président
Mme M. CREHANGE Examineurs
M. J. MARTIN
M. C. PAIR

Je remercie vivement Monsieur le Professeur J. LEGRAS, Directeur de l'Institut Universitaire de Calcul Automatique, de me faire l'honneur de présider le Jury.

Que Madame CREHANGE trouve, dans ce travail, l'expression de ma profonde reconnaissance pour l'attention, les encouragements et les nombreux conseils qu'elle m'a apportés pour sa réalisation.

Monsieur le Professeur J. MARTIN de la Faculté de Médecine de Nancy est à l'origine de ce problème, qu'il soit remercié pour les nombreux conseils et encouragements qu'il a toujours su me donner.

Ce travail a été réalisé sous la direction de Monsieur le Professeur PAIR ; je le remercie pour les orientations fructueuses qu'il a données à mon travail.

Enfin, je remercie Monsieur et Madame BUNEL, Madame Ducloy et Mademoiselle Le Maréchal pour la réalisation pratique de cette thèse, toutes les personnes de l'I.U.C.A., en particulier l'exploitation, pour leur aide technique.

S O M M A I R E

INTRODUCTION

CHAPITRE I

STRUCTURE DES DOSSIERS MEDICAUX

A - Définitions relatives aux graphes en général

B - Adaptation au dossier médical

- 1) Structure des dossiers
- 2) Informations complémentaires
- 3) Lexique
- 4) Exemple de dossier codé arborescent

CHAPITRE II

GESTION DES DOSSIERS MEDICAUX

I - Analyse de la gestion des dossiers médicaux

- I-1 Codage des dossiers
- I-2 Prise en compte de nouveaux dossiers dans le fichier
- I-3 Mise à jour des dossiers du fichier
- I-4 Traitements concernant le fichier lexique
- I-5 Exploitation médicale

CHAPITRE III

PROCESSUS D'INTERROGATION DES DOSSIERS MEDICAUX

A - Structure des questions

- 1) Analogie de la structure des questions avec celle des dossiers médicaux
- 2) Arborescence (ET/OU)
- 3) Adaptation aux questions

B - Découpages d'une question en questions élémentaires

- 1) Critères complémentaires
- 2) Elément de sélection
- 3) Elément de question
- 4) Sous-question

C - Mécanisme de la recherche d'un dossier pertinent

- 1) Principe de séparation
- 2) Principe d'évaluation
 - a) évaluation d'un critère complémentaire
 - b) pertinence d'un dossier pour un élément de sélection
 - c) pertinence d'un dossier pour une question

D - Itération du mécanisme

- 1) Intérêt
- 2) Procédure d'itération

E - Logique de stockage des informations extraites d'un dossier

CHAPITRE IV

LANGAGE D'INTERROGATION

A - Introduction

- 1) Objectifs généraux du langage
- 2) Description générale d'une question

B - Phase de déclaration

- 1) Déclaration des variables simples
- 2) Déclaration des piles
- 3) Déclaration des tableaux de statistiques

C - Phase de sélection

- 1) Introduction
- 2) Description du langage

D - Phase de traitement

- 1) Instructions de traitements arithmétiques et statistiques
- 2) Instructions de sortie de résultat

E - Exemple d'utilisation du langage

CHAPITRE V

ANALYSE DES MODULES DE TRADUCTION ET D'EXPLOITATION DU

SYSTEME D'INTERROGATION

A - Analyse générale

- 1) Analyse générale du programme d'interrogation
- 2) Implantation des variables, piles et tableaux de statistiques utilisés dans un train de question
- 3) Présentation générale de l'analyse des modules

B - Analyse du module de traduction : TRADUCT

C - Analyse d'un sous programme réalisant la fonction d'une instruction procédure particulière

CONCLUSION

BIBLIOGRAPHIE

I N T R O D U C T I O N

Les problèmes pratiques qui se posent quotidiennement aux cliniciens à l'échelon d'un service moyen de C.H.U. et, notamment, celui du dossier médical, peuvent bénéficier du considérable développement de l'informatique. La mise au point d'un système opérationnel de stockage et d'exploitation du dossier médical est une des conditions à remplir pour que le médecin puisse tirer le meilleur profit des informations contenues dans un dossier, en particulier, faire l'étude d'une maladie, de son évolution et de ses conséquences et d'analyser l'efficacité des diverses thérapeutiques.

De plus, un dossier de malade est un document individuel irremplaçable et il faut pouvoir le retrouver rapidement.

Pour constituer "un système de données" de ce type, le premier travail est de décider de la structure des dossiers. L'étendue du champ médical étudié, la complexité de la chronologie des événements, la multiplicité des actes médicaux (consultations, bilan, hospitalisation, étude de l'évolution, ...) imposaient d'autres solutions que celles du questionnaire préétabli, se traduisant par des articles de structure fixe.

Un mode de représentation du dossier médical a été défini par J. M. MARTIN (thèse de troisième cycle), d'après une idée du professeur J. MARTIN. Une structure arborescente constitue l'ossature du dossier ; un lexique composé de descripteurs représentant les termes et expressions utilisés en diabétologie habille cette structure. En outre, à chaque descripteur, peuvent être accolés, une date, un nombre ou adjectif qualifiant ou quantifiant la notion exprimée par ce dernier.

Dès nos premiers contacts avec les médecins, nous avons été très surpris du caractère très vague et du manque de précision de leurs questions. Nous avons eu de grandes difficultés à leur faire préciser les notions clés qui y étaient contenues et leurs relations logiques. Pour voir les difficultés de cette analyse, il suffit de penser à ce que veulent dire des expressions telles que "mode d'évolution...", "influence de... sur...".

Une collaboration avec les médecins ayant participé au codage des dossiers nous a permis de définir l'analyse logique d'un certain nombre de questions que nous avons jugé suffisamment représentatives de toutes celles susceptibles d'être posées.

Nous avons donc ensuite essayé de définir un langage qui ne soit, ni trop difficile (c'est-à-dire qui aurait une syntaxe trop riche et par-là même difficilement assimilable par le médecin), ni trop artificiel (c'est-à-dire ne permettant qu'une écriture trop générale des questions), de façon à permettre, sans trop d'erreurs, une description facile de l'analyse des questions.

CHAPITRE I

STRUCTURE DES DOSSIERS MEDICAUX

Rappelons brièvement le système de représentation d'un dossier médical tel qu'il a été défini par J. M. MARTIN (Thèse 3e cycle). Il repose sur l'hypothèse fondamentale de la structure arborescente des informations qui s'y trouvent. Nous donnons dans le paragraphe suivant quelques définitions, certaines relatives aux graphes en général, d'autres moins classiques (notions de père, de fils et de niveau), relatives à des structures arborescentes. Nous adaptons ensuite toutes ces définitions dans notre cas particulier.

A - DEFINITIONS RELATIVES AUX GRAPHES EN GENERAL

Graphe : Un graphe est un couple (E, Γ) formé d'un ensemble E (ensemble des points du graphe) et d'une relation binaire Γ dans E ; on note $\Gamma(x)$ l'ensemble des $y \in E$ tels que $x \Gamma y$

Famille : Etant donné un point x d'un graphe (E, Γ) , on appelle famille de x et on note $\Gamma(x)$ l'ensemble des points y tels que $x \Gamma y$

Chemin : On appelle chemin d'un graphe (E, Γ) toute suite $(u_1, u_2, \dots, u_n, u_p)$ d'arcs telle que l'extrémité terminale de chaque arc coïncide avec l'extrémité initiale de l'arc suivant pour $n = 1, 2, \dots, p$.
On peut aussi définir un chemin comme étant une suite de sommets. Soient x_{n-1} l'origine de u_n et x_n l'extrémité de u_n , un chemin est défini par la donnée de la suite (x_n) de points ($n = 0, 1, \dots, p$).

Longueur d'un chemin : La longueur d'un chemin $\mu = (v_1, v_2, \dots, v_k)$ est le nombre d'arcs de la séquence : $I(\mu) = k$.

Arborescence : On appelle arborescence un graphe (E, Γ) fini sans circuit tel que :

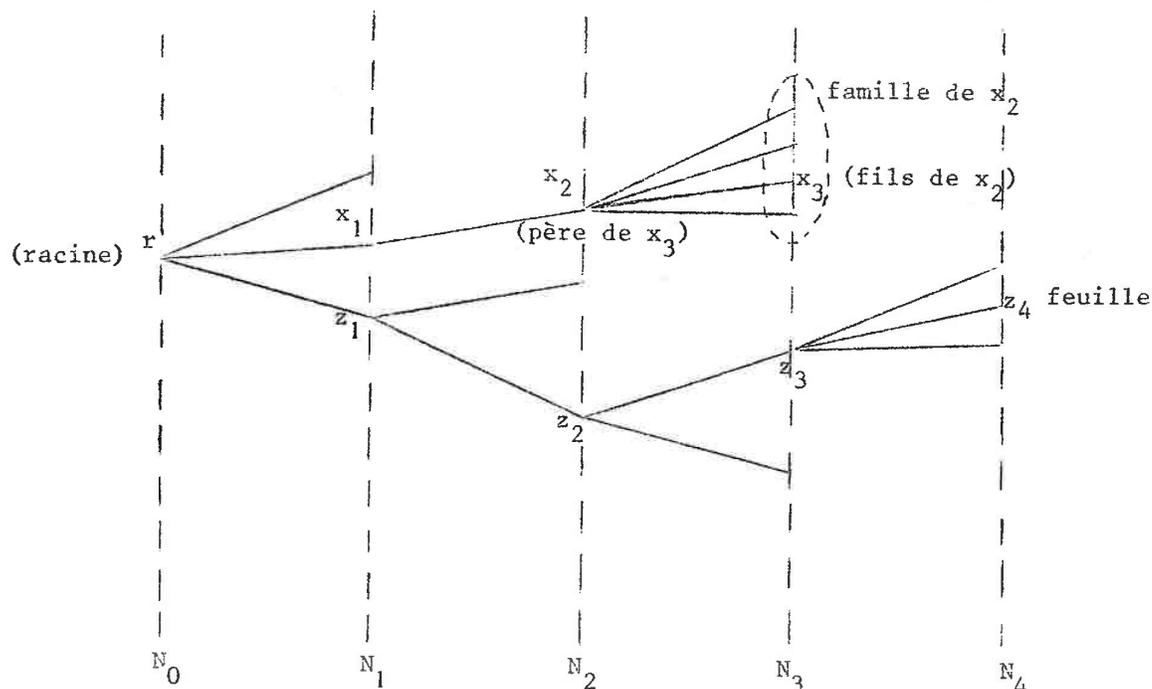
- a) il existe un point r qui n'est extrémité d'aucun arc ;
- b) tout point $x \neq r$ est l'extrémité d'un arc unique.

r s'appelle la racine de l'arborescence

Feuille : Un point z d'une arborescence (E, Γ) est une feuille si $\Gamma(z) \neq \emptyset$

Père et fils d'une famille : Etant donné une famille $\Gamma(x)$ d'une arborescence (E, Γ) , x est appelé père de la famille ; y tel que $y \in \Gamma(x)$ est appelé fils de x

Niveau d'une arborescence : Etant donné une arborescence (E, Γ) et un nombre n , on appelle niveau d'ordre n , noté N_n , l'ensemble des points x appartenant à E , tels que la longueur du chemin $(r \dots x)$ soit égale à n .



B - ADAPTATION AU DOSSIER MEDICAL

I - Structure des dossiers

Les dossiers médicaux que nous considérons sont structurés par une arborescence, étiquetée en chacun de ses sommets par des descripteurs appartenant à un lexique que nous définirons dans un paragraphe suivant.

Chaque rencontre du médecin et de son malade est représentée par un sommet s (sommet de premier niveau) de l'arborescence, et l'on place, dans les niveaux d'ordre supérieur, sous la dépendance de s , les événements médicaux qui y sont reliés.

Chaque noeud de l'arborescence reçoit un descripteur caractéristique du niveau. C'est par ces descripteurs que l'essentiel de l'information est introduit dans l'arborescence.

Le nombre de niveaux est limité à quatre, ils correspondent à un ordre de généralité décroissante de l'information. Les informations qui auraient pu exister aux niveaux supérieurs à 4 ne sont plus hiérarchisées, mais représentées dans des "informations complémentaires" en général rattachées au descripteur de niveau 4.

2 - Informations complémentaires

Les informations complémentaires ou addenda susceptibles d'être accolés aux sommets de l'arborescence sont destinés à préciser, dater ou qualifier les données médicales représentées par les descripteurs étiquetant ces sommets.

A la différence de l'arborescence, les addenda sont non structurés et n'impliquent aucune hiérarchie entre leurs constituants. Les addenda des niveaux 1, 2 et 3 reçoivent essentiellement des dates et des qualificatifs ; en revanche, les addenda rattachés aux descripteurs de quatrième niveau servent en fait à exprimer en bloc les informations qui pourraient exister aux niveaux supérieurs à quatre.

Nous avons été amené toutefois (comme nous le verrons au chapitre suivant), à limiter le nombre des éléments d'une information complémentaire (notamment pour en faciliter la recherche).

Seules les associations suivantes sont possibles :

- date et adjectif
- date et nombre
- nombre et adjectif
- adjectif et adjectif.

Dans cette dernière association, l'ordre d'apparition des adjectifs étant quelconque, nous verrons que la recherche de l'un de ces éléments présentera quelques difficultés.

Deux cas échappent à cette règle d'association : ils concernent l'écriture des intervalles de temps ou les intervalles de variations numériques. Nous utilisons le descripteur jusqa, ce qui porte à trois le nombre d'éléments d'une telle information complémentaire.

Règles concernant l'écriture des dates

Comme les règles d'association le montrent si une date existe dans une information complémentaire, elle est nécessairement le premier élément.

Toute date qualifiant un sommet de quatrième niveau peut avoir une précision inférieure à un jour ; elle sera alors suivie d'un repère horaire contenant l'heure et la minute.

Toute date accolée à un sommet de niveau quelconque qualifie chronologiquement toutes les informations qui en dépendent, sauf si celles-ci portent une date particulière, auquel cas la même règle s'applique à nouveau.

Règles d'écriture de l'en-tête d'un dossier

Les informations complémentaires suivantes, dont la recherche doit être rapide et dont l'existence est essentielle, sont placées en tête du dossier (derrière le descripteur DOSSI) et dans l'ordre indiqué :

- numéro du dossier
- date de codification du dossier
- nature du sexe
- date de naissance

3 - Lexique

C'est un ensemble de quelque 300 termes, appelés descripteurs, employés pour désigner les différentes notions contenues dans un dossier médical. Ils ont été choisis par des médecins pour être aussi évocateurs que possible de la notion qu'ils décrivent (OISIF signifie sans profession) et permettent au médecin de rédiger, soit directement à l'interrogatoire du malade, soit à partir des dossiers manuscrits préexistants, un dossier codé.

Ce lexique est composé de cinq groupes G1, G2, G3, G4 et G5, comprenant les descripteurs de niveau 1, 2, 3, 4 et les descripteurs non introducteurs de niveau. Chaque élément de G3 (descripteur de troisième niveau noté d3) est donné en même temps que la famille de descripteurs de quatrième niveau (d4) qui seuls peuvent être fils de ce d3.

Au moment de l'introduction en machine du dossier (cf. chapitre suivant), chaque descripteur est traduit en son code numérique qui rend compte de la structure du lexique.

Ce lexique est organisé : il est constitué de deux ensembles :

- un ensemble de descripteurs susceptibles d'être, soit des descripteurs introducteurs de niveau, soit des descripteurs non introducteurs de niveau. Il est partitionné en quatre sous-ensembles E1, E2, E3, E4 correspondant, respectivement, aux descripteurs introducteurs des 1er, 2e, 3e et 4e niveaux ;
- un ensemble de descripteurs non introducteurs de niveau, utilisés uniquement comme adjectif (élément d'une information complémentaire).

Notons qu'une perspective d'avenir pourrait être une véritable organisation sémantique du lexique (en particulier, les descripteurs de quatrième niveau seraient reliés par des relations plus ou moins simples).

4 - Exemple de dossier codé arborescent

Nous ne donnons la représentation arborescente que du début d'un dossier, (cf. page suivante).

Remarques : Nous n'avons donné qu'une description générale des dossiers soumis à notre étude. Nous donnons, en annexe I, une description plus détaillée des règles d'écriture. Il faut remarquer à cet effet qu'un certain nombre de règles implicites sont consacrées par l'usage. Elles sont consignées dans un petit fascicule : "Le Guide du Codeur" créé par les utilisateurs médecins.

DOSSI + 822 + D9720125 + HOMME + D9250725

JOURN + D9711103

ETCIV

ANAMN

EXAME

FAMIL

MEITE

+D950+JUSQ_A+D971

MARIE

MILIT

ACTIV

MODER

PROFE

SANSS

NAISS

ENFAN

+02+AVANT

BBRON TRAVA

+02+AVANT

TAMAT

+02+OBESE

ETIOL

+D9710823

SOFIUR APARU

+01+OBESE

FORME

SIGNE POIDS

+85

CHAPITRE II

GESTION DES DOSSIERS MEDICAUX

Nous nous proposons dans ce chapitre de donner une description des résultats de l'analyse de la gestion des dossiers médicaux. Ce système de gestion nécessite un certain nombre de tâches que nous présentons de façon arborescente, puis de façon modulaire.

La numération des paragraphes laisse entrevoir les différents niveaux de cette analyse et explique la présentation particulière de ce chapitre.

Ces tâches sont décrites dans un langage aussi proche que possible de la langue naturelle (tout au moins dans les premières phases). Nous utiliserons toutefois, pour mettre en évidence les procédures informatiques sous-jacentes, certaines instructions inspirées du langage ALGOL (exemple : si ... alors ... sinon...).

Pour faciliter la compréhension de cette analyse, nous arrêtons cette description au niveau des modules de chaque programme (une analyse détaillée de ces derniers sera donnée en annexe 5).

I - ANALYSE DE LA GESTION DES DOSSIERS

- I - Codage des dossiers
- 2 - Prise en compte de nouveaux dossiers dans le fichier
- 3 - Mise à jour des dossiers du fichier
- 4 - Traitements concernant le fichier lexique
- 5 - Exploitation médicale

I - Codage des dossiers

I.1 Remplissage d'un dossier d'observations par le médecin

I.2 Codage du dossier d'observations

I.3 Saisie sur cartes et premières vérifications

I.I - Remplissage d'un dossier d'observations par le médecin

- Nécessité d'un dossier d'observations :

L'absence de localisation et de structuration du dossier traditionnel nécessite un tri des documents et un déchiffrement qui sont hautement préjudiciables à la rapidité du codage. Il importe donc d'envisager une réorganisation des informations à l'intérieur d'un dossier manuscrit.

- Dossier d'observations

Il a été conçu par l'équipe du docteur DROUIN (service de diabétologie, Hôpital Jeanne d'Arc à Toul) pour être particulièrement adapté à la pratique quotidienne diabétologique du service.

Il est constitué d'une couverture cartonnée où figurent les renseignements, en principe, peu changeants, tels que : le nom, le numéro, de sécurité sociale, le numéro d'ordre du codage, la situation de famille ...

Dans ce cadre rigide se trouvent des intercalaires concernant l'examen clinique, l'interrogatoire et l'enquête alimentaire.

Il existe pour chaque série d'intercalaires (une série par séjour) différentes couleurs ; cet artifice permet d'identifier très simplement les différents séjours du malade.

Ce dossier présente l'avantage de donner un support de systématique d'exams, tout en laissant, bien entendu, au médecin la liberté de faire les exams qu'il juge utile.

Il possède les caractéristiques suivantes :

- Son plan est calqué sur celui du lexique pour établir un parallélisme entre le dépouillement de l'observation et le codage.
- Les rubriques de l'examen clinique sont précisées dans leur séquence.
- Enfin, les problèmes de datation y sont résolus selon la méthode utilisée dans le dossier codé.

Nous donnons en annexe II un exemple d'une partie de dossier d'observations.

- Remplissage du dossier d'observations

Pour chaque nouveau malade arrivant au service de diabétologie faire

- "Ouvrir" un dossier d'observations.
- Remplir la couverture cartonnée et donner un numéro d'anonymat au dossier.
En effet, en raison des problèmes de "Secr. Médical" liés au traitement sur ordinateur dans un service non médical (I.U.C.A.), il est nécessaire qu'aucune identification directe ne figure sur un dossier codé, et par conséquent dans le fichier. Chaque dossier codé sera alors uniquement répertorié par son numéro. Pour connaître les identités, il existe un fichier de correspondance de nom, prénom, état-civil (y compris le numéro INSEE et l'adresse du malade) avec le numéro du dossier.
- Reporter sur les intercalaires les différentes observations cliniques relevées lors de la consultation ou de l'hospitalisation.

Si au cours de son séjour à l'hôpital le malade meurt : on ferme son dossier d'observations et on le range dans un fichier "archives mortes".

Nous détaillerons dans un paragraphe ultérieur ce fichier qui présente un intérêt certain.

I.2 - Codage du dossier d'observations

- * Entrée - dossier d'observations (y compris ceux qui sont "fermés")
 - lexique (donne pour chaque concept médical le descripteur correspondant)
- * Sortie - bordereaux de codage.
- * Traitement.

Remarque :

Il convient de préciser dès maintenant que nous allons faire trois utilisations différentes du lexique :

- A) Une utilisation comme dictionnaire pour le traitement présent. Les descripteurs du lexique seront regroupés par chapitre en conformité avec la pratique du codage, et écrits sur des micro-fiches; il est possible grâce à un système de touches pré-réglées d'accéder rapidement à n'importe quelle partie du lexique.
- B) Une utilisation dans le sens descripteur → code numérique lors du traitement "machine" des dossiers.
- C) Une utilisation dans le sens descripteur → traduction en "texte clair" lors de l'édition des dossiers.

Pour ces deux dernières utilisations le lexique sera sur disque, les descripteurs étant rangés dans l'ordre alphabétique.

Le codage d'un dossier médical est réalisé à partir du dossier d'observations et à l'aide du lexique.

Bordereaux de codage

La répétition de certains descripteurs selon des séquences toujours identiques a permis la création d'un bordereau de codage ayant les caractéristiques suivantes :

- La disposition typographique choisie met en évidence la structure arborescente sous-tendant le dossier (en particulier, l'écriture des différents descripteurs est décalée du bord de la feuille pour mettre en évidence leurs niveaux).
- L'entête du bordereau relatif à la structure fixe des informations d'utilité administrative et médicale (numéro de dossier, nature du sexe de l'individu, date de naissance ...) a une disposition pré-établie.
- Certains chapitres concernant l'état-civil, la thérapeutique diététique, glycémiant, sont décrits de façon exhaustive par rapport au lexique ; le codeur n'aura qu'à, soit biffer éventuellement les descripteurs superflus, soit les qualifier ou les quantifier par des informations complémentaires.

Nous donnons en annexe III le bordereau de codage correspondant au dossier d'observations donné en annexe II.

Rendement du système de codage

Compte tenu du débit mensuel du service, ce travail nécessiterait l'emploi d'un médecin à plein temps. Etant donné le caractère relativement fastidieux de ce codage, il est impossible de demander ce travail à un médecin. Le problème a été résolu en répartissant cette tâche entre quatre médecins vacataires effectuant trois vacations hebdomadaires de trois heures chacune.

La pratique routinière du codage des observations - encore améliorabile, mais nous ne savons pas dans quelles limites - nécessite à l'heure actuelle : - 10 à 15 minutes pour une consultation
- 20 à 30 minutes pour une hospitalisation.

Actuellement, par mois en moyenne, une quarantaine de dossiers représentant 120 à 150 observations sont codés.

1.3 - Saisie sur cartes et premières vérifications

Les bordereaux de codage, servant des bordereaux de perforation, sont amenés par fournées (40 environ) chaque mois au service de perforation (Faculté de Médecine).

Nous avons essayé d'introduire le minimum de contrainte dans la perforation de ces bordereaux :

- Chaque dossier commence par le descripteur DOSSI perforé dans les cinq premières colonnes d'une carte, suivi de la description linéaire du dossier arborescent.

- Toute information (descripteur ou élément d'information complémentaire) est séparée de la précédente par au moins un caractère blanc.

Chaque bordereau donne lieu à un "dossier cartes" d'une dizaine à une centaine de cartes selon son importance. L'ensemble des "dossiers cartes" constitue le fichier "source-cartes".

- Premières vérifications

Les cartes sont, soit contrôlées sur une vérificatrice, soit listées sur l'ordinateur IO OIO de la Faculté de Médecine, et collationnées ensuite. Compte tenu de la souplesse de perforation et de ces vérifications, les erreurs imparties à la perforation sont relativement rares ; par contre les erreurs de vocabulaire et surtout de syntaxe sont plus fréquentes et nécessitent, comme nous allons le voir, un contrôle réalisé par un programme.

2 - Prise en compte de nouveaux dossiers dans le fichier

- 2.1 - Validation des dossiers codés. Regroupement des dossiers corrects.
- 2.2 - Traitement des dossiers codés. Adjonction au fichier.
- 2.3 - Copie du fichier.

2.I - Validation des dossiers codés; Regroupement des dossiers corrects.

2.1.1 - Validation des dossiers codés.

2.1.2 - Tri des dossiers corrects.

2.1.3 - Corrections éventuelles des bordereaux.

2.1.1 - Validation des dossiers

- Principe

Il convient de vérifier qu'à l'issue du codage de chaque dossier, la structure arborescente a été respectée, ainsi que les règles d'écriture représentant la syntaxe utilisée.

La représentation d'une arborescence sous une forme linéaire telle qu'elle apparaît dans un "dossier-cartes" est très compacte et n'offre pas une vue d'ensemble du dossier ; elle ne permet pas, surtout, d'en déceler aisément d'éventuelles anomalies.

Il nous a semblé intéressant de mettre en oeuvre le processus suivant : restituer le dossier en un "texte clair" à partir du "dossier-cartes". Cette édition présente un double intérêt :

- 1) Celui de contrôler la cohérence formelle du dossier : détection des erreurs éventuelles de syntaxe et de vocabulaire.
- 2) Celui de permettre un contrôle médical (tout au moins lors des premiers essais) en vérifiant que le dossier codé est bien l'image du dossier manuscrit.

- Réalisation

Nous donnons de plus amples détails concernant le programme d'édition des dossiers à la fin de ce chapitre et en annexe.

Nous nous contenterons dans ce qui suit de donner un aperçu des erreurs les plus fréquentes détectées par le programme.

- * Entrée - fichier "source-cartes".
 - fichier "lexique".
- * Sortie - listing des dossiers et libellés des erreurs éventuelles.
 - fichier source-bande 1.
- * Traitement.

Pour chaque "dossier-cartes" du fichier "source-cartes" vérifier :

- Que toutes les informations de l'en-tête du dossier (numéro de dossier, sexe de l'individu, date de naissance ...) sont présentes.
- Que la structure arborescente du dossier est respectée (c'est-à-dire que si un descripteur de niveau i suit un descripteur de niveau j avec $j < i$ on a $j=i-1$ quelque soit $j=1,2,3$ et quelque soit $i=2,3,4$).
- Que la structure du lexique est respectée (c'est-à-dire que chaque descripteur est utilisé à son bon niveau et qu'un descripteur de 4^e niveau est utilisé dans la famille à laquelle il appartient).
- Que l'orthographe des descripteurs ou l'écriture des informations complémentaires est correcte.

Dans tous les cas un libellé d'erreurs, spécifique à chaque type d'erreur, est imprimé et l'édition continue en séquence.

A l'issue du traitement chaque "dossier-cartes" correct est recopié dans le fichier "source-bande 1".

Nous donnons en annexe une édition partielle du dossier correspondant au bordereau de codage donné en exemple, accompagnée de libellés d'erreurs.

2.I.2 - Tri des dossiers corrects

- * Entrée - fichier "source-bande 1".
- * Sortie - fichier "source-bande 2".
- * Traitement - tri par numéro de dossier croissant.

2.1.3 - Corrections éventuelles des dossiers

Traitements des anomalies

- * Entrée - listings de dossiers incorrects.
 - lexicque.
 - bordereaux de perforation des dossiers incorrects.
- * Sortie - fichier des "dossiers-cartes" corrigés.
 - fichier des "dossiers-cartes" incorrects.
- * Traitement.

Ce traitement est réalisé manuellement au service de perforation de la Faculté de Médecine.

Pour chaque listing de dossiers incorrects faire :

- Recherche du bordereau de perforation du dossier concerné.

Pour chaque erreur détectée faire :

- Si ce n'est pas une erreur de perforation alors :

Si l'erreur peut être corrigée au niveau du service de perforation (par exemple : erreur au niveau de la sémantique : information complémentaire mal écrite ou orthographe incorrecte d'un descripteur et dont la correction ne présente pas d'ambiguïté) alors la correction est faite sur place.

Sinon si c'est une erreur de syntaxe alors le bordereau et le listing du dossier incorrect sont envoyés au service de diabétologie.

2.2 - Traitement des dossiers - Adjonction au fichier

- Principe

Le traitement que nous allons décrire consiste à construire à partir du dossier codé les cinq tableaux d'exploitation $NI_i (i=1 \text{ à } 4)$ et AA définis en annexe 4 bis.

Chaque dossier est rangé dans deux enregistrements, le premier contenant la dimension des tableaux d'exploitation, le deuxième les tableaux proprement dits.

Nous appelons fichier "dossiers traités I" le fichier constitué par ces enregistrements rangés par ordre croissant des numéros de dossiers. Les nouveaux dossiers (c'est-à-dire correspondants à des nouveaux malades pris en compte au service de diabétologie) à adjoindre au fichier "dossiers traités I" peuvent avoir des numéros inférieurs aux numéros de dossiers déjà enregistrés. Pour les insérer parmi ces derniers nous utilisons un fichier de travail "dossiers traités 2" servant aussi de "fichier copié".

Si i est le numéro du dossier à adjoindre au fichier "dossiers traités I" nous réalisons les opérations suivantes.:

- recopie des dossiers de numéros inférieurs à i de "dossiers traités I" dans "dossiers traités 2".
- Traitement du dossier i ; enregistrement du dossier à la suite dans "dossiers traités 2".
- Recopie des dossiers de numéros supérieurs à i de "dossiers traités I" dans "dossiers traités 2".

Il suffira ensuite de faire une copie de "fichier dossiers traités 2", dans "fichier dossiers traités I".

Nous appelons fichier "archive bande I" le fichier contenant les tableaux d'exploitation relatifs aux dossiers d'observations "clos". Ce fichier n'est pas trié, tout nouveau dossier d'observations est alors ajouté à la suite.

- Réalisation

- ⊗ Entrée - fichier "source bande 2".
 - fichier "lexique" (utilisation descripteur \rightarrow code numérique).
 - fichier "dossiers traités I".
 - fichier "archive bande I".
- ⊗ Sortie - fichier "dossiers traités 2".
 - fichier "archive bande I".
 - fichier "anomalies".
- ⊗ Traitement.

Pour chaque dossier du fichier "source-bande 2" faire

*Phase d'initialisation (module INIT)

*Pour chaque information du dossier faire :

- Reconnaître son type : descripteur ou élément d'information complémentaire de type, date, heure, numérique ou adjectif (module CARSUIV).
- Exécuter les traitements éventuels spécifiques à chaque type (module de traitement des descripteurs et des informations complémentaires respectivement DESC et INFO).
- Ranger le résultat des traitements dans NI1 et AA.
- Réorganiser le tableau NI3 (module CLAS).

Si le dossier est issu d'un dossier d'observations "clos" faire :
. Mettre à jour le fichier "archive bande 1".

Sinon (soit i le numéro du dossier) faire :

- . Recopier les dossiers de numéros inférieurs à i de "dossiers traités 1" dans dossiers traités 2.
- . Enregistrer à la suite le dossier traité numéro i.
- . Recopier les dossiers de numéros supérieurs à i de "dossiers traités 1" dans dossiers traités 2.

Remarque

Si un dossier est déjà présent dans un des deux fichiers avec un même numéro, la mise à jour n'est pas faite et un libellé indique l'erreur et le numéro du dossier.

Nous donnons en annexe (5) le détail des modules INIT, CARSUIV, DESC, INFO et CLAS.

2.3 - Copie des fichiers

* Entrée - fichier "dossiers traités 2".

* Sortie - fichier "dossiers traités 1".

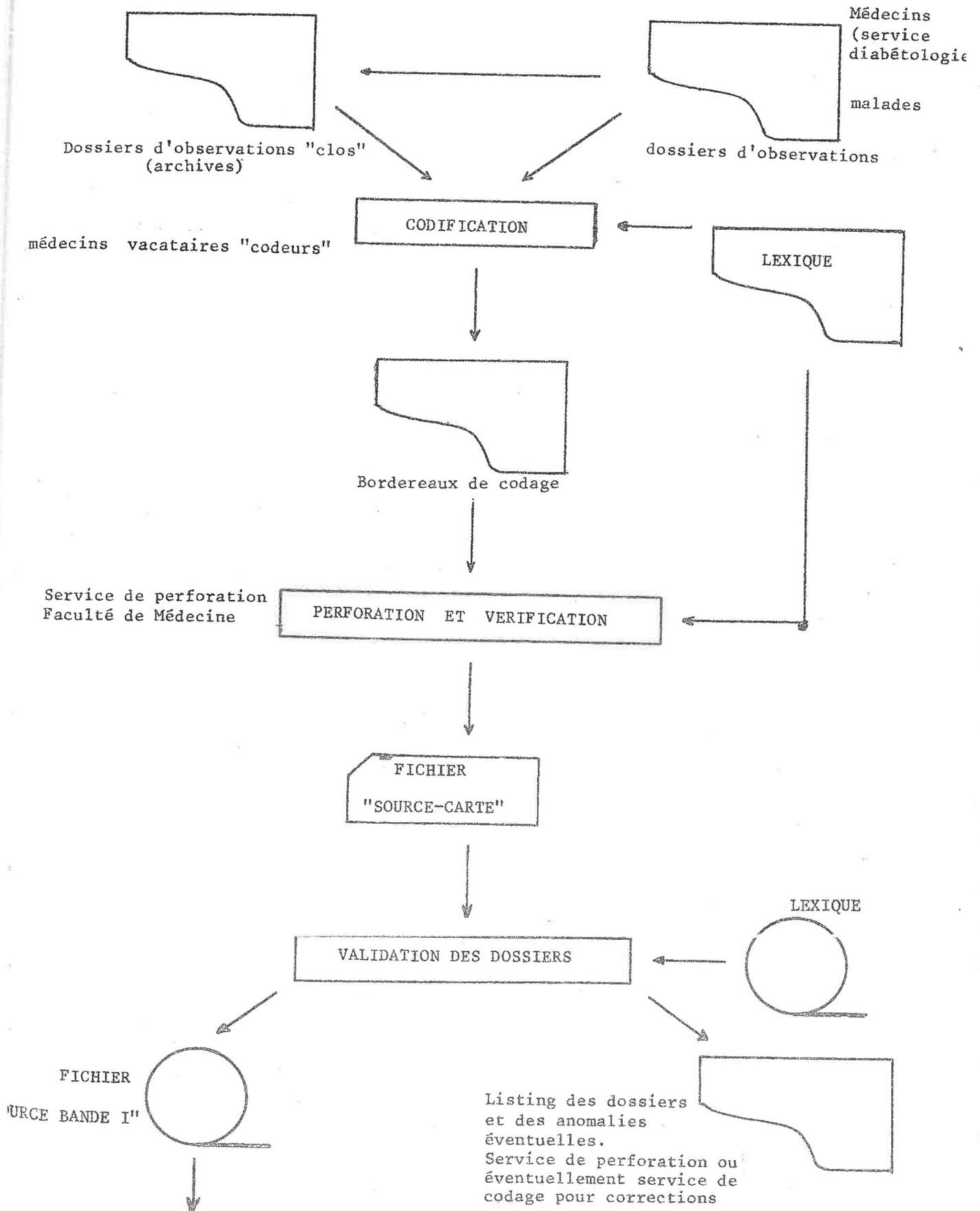
* Traitement - copie.

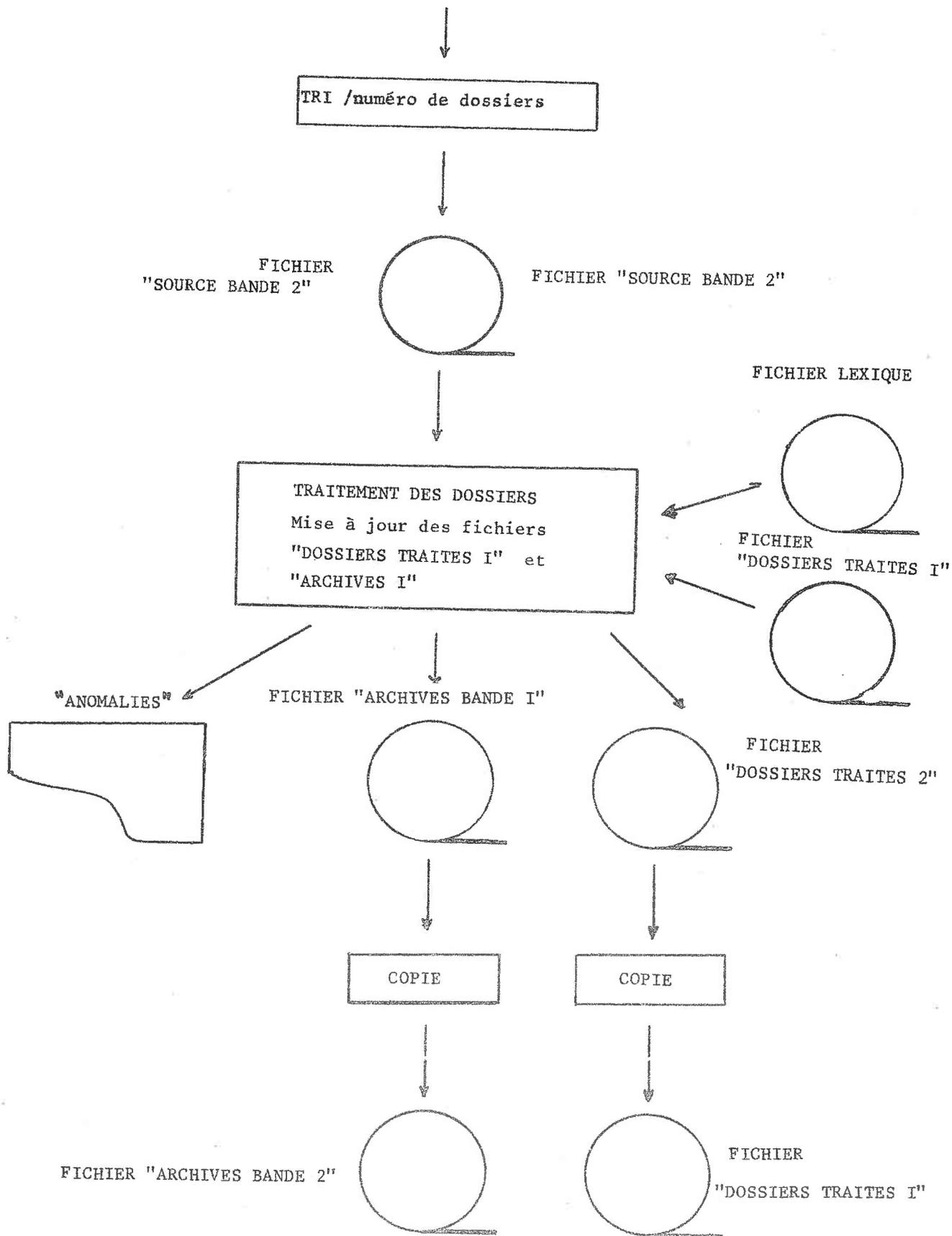
* Entrée - fichier "archive bande 1".

* Sortie - fichier "archive bande 2".

* Traitement - copie.

CHAINE DE TRAITEMENT DES DOSSIERS MEDICAUX





3 - Mise à jour des dossiers du fichier

3.1 - Mise à jour des bordereaux d'observations.

3.2 - Bordereaux de mise à jour.

3.3 - Saisies des mises à jour.

3.4 - Mise à jour proprement dite.

Remarques

Nous nous préoccupons des deux types de mise à jour suivants, qui sont les seuls à avoir une signification médicale :

- 1) Adjonction à un dossier de nouvelles rencontres du malade et de son médecin
- 2) Retrait et archivage d'un dossier du fichier.

Bien que le problème ne se soit jamais posé jusqu'à présent, étant donné que le système ne fonctionne en routine dans le service que depuis un an, il faut envisager le moment où un dossier ne sera plus utile.

Logiquement, le passage d'un dossier en "archives mortes" peut avoir trois causes, compte tenu du fait que la maladie diabétique ne peut-être guérie et les malades doivent rester sous surveillance jusqu'à la fin de leur vie.

La première cause de la non-réapparition d'un malade au service est son décès. Dans ce domaine, nous sommes extrêmement mal informés. A part une enquête de temps à autre (assez peu fructueuse), le service n'est pas averti du décès des malades.

La deuxième cause est le changement d'adresse du malade, l'emmenant dans une autre région. Dans ce cas, le dossier ancien est fréquemment réclamé par le nouveau service où le malade est consulté.

Enfin, il arrive que le malade change de service de consultation : cette situation est relativement voisine de la précédente. En effet, pour suivre correctement le malade, il est préférable de connaître exactement ses antécédents et en règle générale, lorsqu'un patient passe dans un autre service (volonté qui est toujours respectée) une demande de documents est faite.

En conséquence, il semble raisonnable au point de vue des archives, et à partir du moment où le fichier devient trop volumineux de considérer qu'un malade qui n'a pas été revu au bout d'un délai de 3 à 5 ans, correspond à l'une des catégories précédentes, et que son dossier, tout en étant conservé soit mis à part.

3.I - Mise à jour du bordereau d'observations

Pour chaque nouvelle visite à l'hôpital d'un ancien malade:

- Rechercher le dossier d'observations, vérifier et mettre à jour les renseignements relatifs à l'état civil du malade figurant sur la couverture cartonnée
- Ajouter une série d'intercalaires où l'on transcrira les observations cliniques lors de la nouvelle consultation ou hospitalisation.

Si le malade décède au cours de la nouvelle hospitalisation alors on "ferme" son dossier d'observations et on le range dans le fichier "archives mortes".

3.2 - Bordereau de mise à jour

- * Entrée - nouveaux intercalaires du dossier d'observations.
_lexique (utilisation concept médical → descripteur).
- *Sortie - bordereaux de codage.
- * Traitement.

Pour chaque nouvel intercalaire ajouté à un dossier d'observations - créer un bordereau de codage. L'entête est constitué de l'indicatif SUITE, suivi du numéro du dossier à mettre à jour, et d'un certain nombre de renseignements tels que la date de naissance, le sexe, permettant de faire des vérifications complètes. Les caractéristiques du codage des nouveaux intercalaires sont identiques à celles des dossiers d'observations des nouveaux malades.

Pour chaque dossier d'observations correspondant à un malade non revenu au service de diabétologie au bout d'un délai de 3 à 5 ans :
- Créer un bordereau de suppression de dossier composé d'un descripteur RETIR suivi du numéro du dossier, de la date de naissance et du sexe du malade. Ces renseignements permettent de faire les mêmes vérifications que dans le cas d'un ajout.

3.3 - Saisies des mises à jour

- * Entrée - bordereaux de mise à jour,
- * Sortie - fichier "M.A.J. source carte";
- * Traitement.

Chaque bordereau de mise à jour fait office de bordereau de perforation. Les contraintes à la perforation sont les mêmes que pour les bordereaux de codage de nouveaux dossiers.

Pour chaque bordereau de mise à jour faire :

Si c'est un bordereau d'adjonction alors l'indicatif SUITE est perforé à la suite du descripteur DOSSI; ainsi que les renseignements identifiant le dossier. L'arborescence représentant la nouvelle rencontre du médecin et de son malade est perforée linéairement comme en I.3.

Sinon c'est un bordereau de suppression. L'indicatif RETIR est perforé à la suite du descripteur DOSSI, ainsi que les renseignements complémentaires d'identification cités en 3.2.

3.4 - Mises à jour proprement dites

- 3.4.1 - Validation des mises à jour. Regroupement des mises à jour correctes.
- 3.4.2 - Traitement des mises à jour.
- 3.4.3 - Copie des fichiers.

3.4.1 - Validation des mises à jour. Regroupement des mises à jour correctes

- Validation des mises à jour

- * Entrée - fichier "mise à jour, source carte"
 - fichier lexique
- * Sortie - fichier "mise à jour, source bande 1"
 - listings des mises à jour et des anomalies éventuelles
- * Traitement.

Pour chaque "mise à jour carte" du fichier "mise à jour source cartes"

- Vérifier que les renseignements identifiant le dossier sont tous présents et que l'ordre est respecté.

Si c'est une adjonction alors faire le même contrôle de syntaxe et de vocabulaire que pour un nouveau dossier (2.1). A l'issue du contrôle, chaque "mise à jour carte" correcte est recopiée dans le fichier "mise à jour source bande 1".

- Regroupement des mises à jour correctes

- * Entrée - fichier "mise à jour source bande 1"
- * Sortie - fichier "mise à jour source bande 2"
- * Traitement - tri par numéro croissant des mises à jour de dossiers

Nous réalisons le même traitement des anomalies qu'avec les bordereaux relatifs à des nouveaux dossiers.

3.4.2 - Traitement des mises à jour

Le fichier "dossiers traités I" est à accès séquentiel ; toute mise à jour d'un dossier, c'est à dire d'un enregistrement du fichier nécessite l'utilisation d'un fichier intermédiaire "dossiers traités 2".

Le fichier "archive bande I" n'étant pas trié selon l'ordre croissant des numéros de dossier, la mise à jour de ce fichier consiste simplement à ajouter les dossiers "clos" au bout de ce dernier.

* Entrée - fichier "mise à jour source bande 2".
- fichier "dossiers traités I".
- fichier lexique.

* Sortie - fichier "dossiers traités 2".
- fichier "archive bande I".
- fichier "anomalies".

* Traitement.

Pour chaque mise à jour du fichier "mise à jour bande 2" faire

Si c'est une suppression alors retirer le dossier concerné du fichier "dossiers traités I" et mettre à jour le "fichier "archive bande I".

Sinon c'est une adjonction alors mettre à jour les tableaux d'exploitation $NI(i=1 \text{ à } 4)$ et AA du dossier concerné en réalisant les mêmes traitements qu'en 2.2.

Si le malade est décédé au cours de son nouveau séjour alors retirer le dossier du fichier "dossiers traités I" et le recopier dans le fichier "archive bande I".

Sinon recopier le dossier mise à jour dans le fichier "dossiers traités 2".

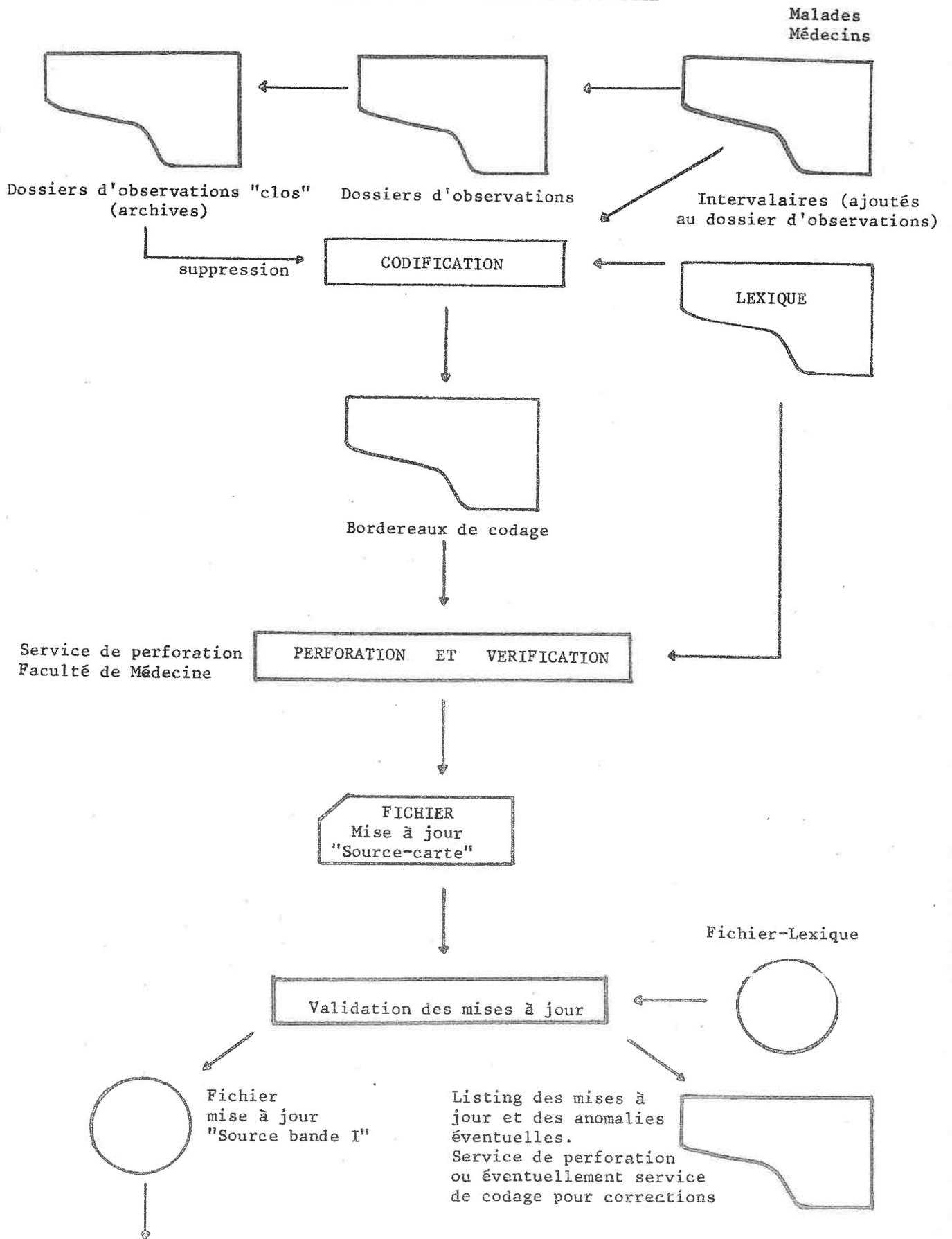
Remarque

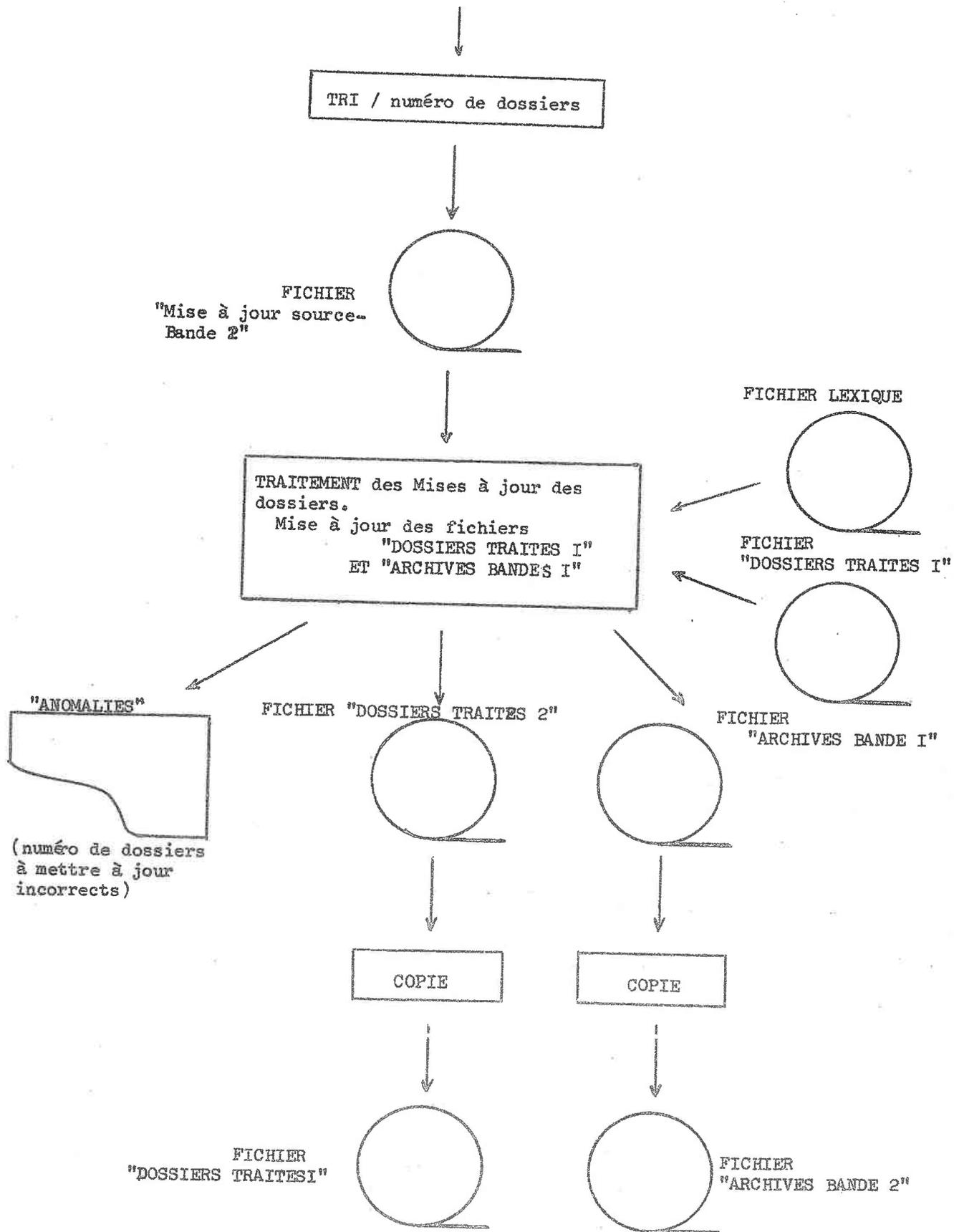
L'adjonction de nouvelles hospitalisations ou consultations à des anciens dossiers est une opération simple. Compte tenu de leurs structures arborescentes il suffira d'ajouter de nouvelles "branches arborescentes" (de racine un descripteur de niveau I) à l'arborescence sous-tendant un dossier. Au niveau de la mise à jour des tableaux d'exploitation, exception faite du module INIT d'initialisation, tous les autres modules (et leurs enchaînements seront les mêmes que ceux vus lors de la création de nouveaux tableaux.

3.4.3 - Copie des fichiers

Le traitement est le même qu'en 2.3

CHAINE DE MISE A JOUR DU FICHIER





4 - Traitements concernant le fichier lexique

4.1 - Mise à jour du fichier "lexique".

4.2 - Etude de la fréquence d'utilisation des descripteurs.

4.1 - Mise à jour du fichier "lexique".

Rappelons qu'à chaque descripteur de niveau 3 est associée une famille de descripteurs de niveau 4 qui seuls peuvent être fils de ce descripteur. La configuration du lexique (dont nous donnons une partie en annexe 6) est telle qu'un regard de chaque descripteur se trouve le code numérique et s'il est de niveau 4 un pointeur indiquant le rang dans le lexique du descripteur "père" des niveaux 3 et enfin sa traduction en texte clair.

- * Entrée - fichier "lexique".
- fichier "modifications".
- * Sortie - fichier "lexique".
- * Traitement.

Pour chaque descripteur à retirer faire :

- Recherche dichotomique du descripteur dans le lexique.
- Décrémenter de un les pointeurs des descripteurs de 4^e niveau dont la valeur est supérieure au rang du descripteur à retirer.
- Tasser le lexique et décrémenter de un sa dimension.

Pour chaque descripteur à ajouter faire :

- Recherche dichotomique du rang I auquel il doit être inséré.
- Insertion du descripteur, de son code numérique et de sa signification en clair.
- Incrémenter de un les pointeurs des descripteurs de niveau 4 dont la valeur est supérieure ou égale au rang du descripteur à ajouter.
- Si le descripteur est de niveau 4 alors faire : valeur du pointeur = rang dans le lexique du descripteur "père" de niveau 3.
- Incrémenter de un la dimension du lexique.

Remarques

Le problème de la mise à jour du lexique s'est posé au fur et à mesure de la mise en place du système d'exploitation.

Après avoir codé une quarantaine de dossiers, et surtout après que ces dossiers aient été restitués sous forme d'une édition en texte clair, les médecins-codeurs ont trouvé le lexique initial trop restrictif, ne permettant pas de prendre en compte toute l'information valide et explicite d'un dossier. Nous avons alors procédé à d'importantes mises à jour qui ont entraîné la création d'environ 170 nouveaux descripteurs introducteurs de niveau.

Actuellement, l'exploitation des dossiers tournant en routine, il est nécessaire que le lexique ait une version définitive. Il serait en effet désastreux d'envisager des mises à jour "brutales" de ce dernier car cela aboutirait : soit à rendre partiellement inutilisables les dossiers traités et concernés par la mise à jour, soit à coder une nouvelle fois ces dossiers.

Toute modification du lexique n'est faite que lorsque les utilisateurs rencontrent des difficultés imperieuses de codage et avec beaucoup de prudence.

4.2 - Etude de la fréquence d'utilisation des descripteurs

Nous disposons de cinq tableaux de compteurs TC_i (i = 1 à 5) correspondants respectivement aux cinq tableaux d'exploitation (nous attribuons la valeur i = 5 pour les descripteurs non introducteurs de niveau). Le code numérique affecté à chaque descripteur permet de déterminer aisément le niveau de ce dernier (exemple : les descripteurs non introducteurs de niveaux sont codés de 1 à 99, les descripteurs de 3^e niveau de 300 à 399,....).

Nous associons à chaque élément (compteur) d'un tableau de compteur le code numérique d'un descripteur.

Une correspondance automatique est facile à réaliser en traduisant les codes numériques des descripteurs de même niveau i d'une valeur TR_i telle que la valeur du plus petit code soit 1 (exemple : les codes des descripteurs de 3^e niveau après translation seront compris entre 1 et 100) il leur sera associé les 100 premiers éléments de TC₃.

- * Entrée - fichier "dossiers traités I".
 - fichier "lexique" (utilisé seulement pour transcoder).
- * Sortie - tableaux de fréquence d'utilisation des descripteurs.
- * Traitement.

Pour chaque dossier du fichier faire :

Pour chaque tableau d'exploitation i d'un dossier faire :

Pour chaque occurrence d'un code numérique dans ces tableaux faire :

$$\begin{aligned} \text{code numérique} &= \text{code numérique} - \text{TR}_i \\ \text{TC}(\text{code numérique}) &= \text{TC}(\text{code numérique}) + \text{I}. \end{aligned}$$

Nous éditerons ensuite en face de chaque élément des différents tableaux le code numérique correspondant obtenu en faisant la translation inverse.

Remarques

Ces dénombrements ne sont généralement pas faits sur tout le fichier, mais sur les derniers dossiers enregistrés pendant les six ou douze derniers mois. La recherche des adjectifs (descripteurs non introducteurs de niveau) dans le tableau AA étant plus longue et plus coûteuse, et présentant moins d'intérêt sera faite moins souvent.

Les résultats obtenus offrent un triple intérêt :

- 1) Celui de mettre en évidence les descripteurs introducteurs de niveau peu ou jamais utilisés : soit parce que le descripteur correspond à un symptôme ou une maladie rarement ressentie, soit parce que ce descripteur a une signification redondante ou sans intérêt. Dans ce dernier cas cette étude se révèle intéressante pour faire des mises à jour du lexique en toute connaissance de cause.
- 2) Celui de déceler les différences de codage entre différents codeurs et de permettre une meilleure homogénéisation du codage.
- 3) Celui de permettre une optimisation dans la recherche de plusieurs descripteurs d'une question en fonction de leur probabilité d'apparition dans un dossier (on recherche en premier le descripteur dont l'occurrence est la moins probable).

4) Enfin, à plus long terme celui de permettre une évolution des thérapeutiques en fonction de l'évolution des techniques. Le résultat du dénombrement des descripteurs relatifs à des nouveaux médicaments illustrerait l'utilisation de ces derniers.

5 - Exploitation médicale

5.1 - Edition des dossiers médicaux.

5.2 - Systèmes d'interrogation du fichier.

5.1 - Edition des dossiers médicaux

Initialement nous avons prévu l'édition des dossiers dans un but de contrôle de la validité des dossiers codés (2.1). Rapidement cet outil s'est révélé un outil fondamental pour une utilisation médicale en routine.

Un même malade n'est pas toujours vu par le même médecin, en particulier à la consultation. Le médecin consultant avec un dossier d'observations ne peut connaître le malade en quelques minutes ; cette édition lui donne une idée très précise et rapide du malade.

Nous avons essayé de soigner la présentation typographique afin d'obtenir un document agréable à l'oeil.

Pour chaque dossier à éditer faire :

Pour chaque descripteur introducteur de niveau du dossier :
- Editer sa traduction en texte clair en décalant l'impression par rapport au bord de la feuille d'imprimante pour mettre en évidence son niveau.

Pour chaque information complémentaire l'éditer en regard du descripteur auquel elle se rattache en utilisant des sous-programmes d'édition spécifique à chaque type.

Nous donnons en annexe (7) une partie de l'édition du dossier dont nous avons donné le dossier d'observations et le bordereau de codage en exemple.

5.2 - Système d'interrogation du fichier

Les systèmes d'interrogation sont le principal intérêt, voire même la raison d'être d'un tel fichier. Si la gestion des dossiers médicaux telle que nous l'avons vue, est devenue routinière et ne pose plus de problèmes majeurs, un certain nombre de difficultés restent à résoudre quant à l'interrogation des dossiers.

La compréhension réciproque entre médecins et informaticiens n'est pas toujours parfaite et cela se vérifie chaque fois qu'un médecin même initié cherche à interroger le fichier.

Le système d'interrogation actuel tel qu'il a été décrit par Monsieur Jean-Marie MARTIN (thèse 3^e cycle) nécessite pour chaque question l'écriture d'un programme, rédigé en fortran et constitué par l'assemblage de module d'interrogation, ces derniers réalisant certaines "fonctions d'interrogations élémentaires".

Dès l'origine ce système avait été défini comme provisoire, il avait l'avantage d'être efficace et rapidement exploitable, et par la suite d'inciter le médecin à interroger son dossier. Il présentait par contre deux inconvénients:

pour l'informaticien d'avoir à écrire un programme différent pour chaque question différente .

pour le médecin de ne pas connaître, sous réserve d'apprendre un langage de programmation, les outils dont il dispose pour formuler une question. Le plus souvent c'était alors l'informaticien qui prenait en charge l'analyse des questions tant au point de vue médical qu'informatique.

Un prolongement logique de ce travail est de définir un langage d'interrogation offrant d'une part, un outil plus rigide, en un certain sens car il obligerait le médecin à formuler lui-même sa question de façon précise en y dégageant les différentes notions clefs et les relations logiques qui peuvent exister entre elles, et plus simple, car il permettrait d'écrire et de modifier d'une façon plus souple et plus rapide une question.

Il est certain que l'utilisation d'un tel langage ne pourra se faire qu'en étroite collaboration (tout du moins au début) entre médecins et informaticiens.

La description de ce langage fait l'objet du chapitre suivant.

II - DIFFICULTES RENCONTREES LORS DE LA MISE EN PLACE
DE CETTE GESTION

Il s'agissait au départ de réaliser un programme permettant de construire pour chaque dossier les cinq tableaux d'exploitation. Cependant, rapidement il est apparu de nombreuses difficultés au niveau de la saisie des informations, de leur validation et, plus tard, au niveau de la mise à jour et de l'archivage des dossiers traités.

Nous allons examiner du côté médecins et du côté informaticiens les différents problèmes rencontrés tout au long de la mise en place de ce système de gestion.

- Côté médecins

Si la structure arborescente n'a pas posé de problèmes, laissant lors du codage toute latitude, pour ouvrir un niveau lorsque c'est nécessaire ou pour permettre de réaliser aisément des adjonctions successives au fur et à mesure des hospitalisations et des consultations subies par un même malade, la pratique du codage laisse apparaître deux difficultés: une au niveau du lexique, l'autre au niveau des règles d'écriture des informations complémentaires.

A) Problèmes liés au lexique:

Un effort de mémorisation du lexique s'impose au départ si l'on veut éviter des consultations fréquentes du lexique. Toutefois ces consultations restent inévitables pour empêcher des glissements de sens lorsque la signification exacte du descripteur ne serait pas respectée scrupuleusement.

Il ressort de ceci que l'opération codage est donc fastidieuse, la présence de descripteurs mnémotechniques allège toutefois cette contrainte. Le lexique initialement beaucoup trop restreint a subi, comme nous l'avons vu d'importantes mises à jour et a été très étoffé. Il semble qu'actuellement un compromis ait été trouvé entre un lexique ayant une taille trop réduite et étant trop peu spécifique pour qu'un dossier reflète de façon fidèle l'état du malade, et un lexique trop vaste rendant difficile le travail du codeur et coûteux le traitement machine.

B) Problèmes liés aux règles d'écriture des addenda

Initialement les règles d'écriture des informations complémentaires étaient extrêmement libres et le nombre des éléments de ces informations n'était pas fixé.

Ainsi pour qualifier un descripteur, de longues chaînes d'éléments d'informations complémentaires pouvaient être assemblées. (exemple OSCIL + NORMA + MEMBR + INFER + BILAT) et de nombreux descripteurs de niveau 4 étaient utilisés abusivement comme adjectifs. Il en résultait alors des difficultés très lourdes lors de l'interrogation car l'exploration de ces longues chaînes était coûteuse et peu commode, d'autant plus que rien dans les règles d'écriture n'interdisait d'intervertir les descripteurs utilisés. Nous avons donné au paragraphe I les associations permises. Nous pouvons remarquer qu'il subsiste l'association de deux qualificatifs, association qui créera quelques difficultés lors de l'interrogation des dossiers.

- Côté informaticiens

Précisons tout de suite que les utilisateurs médecins avec lesquels nous avons travaillé étaient extrêmement ouverts et motivés pour une telle gestion. Les difficultés rencontrées ont été la plupart du temps rapidement aplanies, et le système de gestion a pu être rapidement opérationnel.

Toutefois n'ayant aucune expérience de codification des dossiers médicaux, les médecins se sont montrés au départ très prudents et ont procédé au "coup par coup", attendant de voir, à l'issue du listing de dossiers, ce que donnait telle ou telle modification.

Le cahier des charges, fixé au départ pour la codification des dossiers, a été fréquemment sujet à des modifications (il a fallu par exemple,

alors que le système était opérationnel depuis six mois, prévoir la prise en compte d'une datation au niveau de l'heure et de la minute. Cette précision de la datation à tous les niveaux a ensuite été limitée au 4^e niveau pour éviter une précision abusive).

En outre, le changement de codeurs tous les six mois, entraînait de nouvelles interprétations et erreurs de règles d'écriture : il semble que le "guide du codeur" ait résolu ce problème.

Enfin, nous nous sommes rapidement aperçus que la validité et la précision de la saisie des informations étaient étroitement liées à la motivation du médecin codant ses observations.

Nous avons souvent remarqué que la saisie et la codification des informations par de nouveaux externes auxquels on n'avait pas montré l'intérêt d'un tel travail, étaient souvent incomplètes et entachées d'erreurs.

Si l'on veut que ce travail survive, il sera nécessaire que chaque membre de l'équipe fasse l'effort de se mettre au courant des techniques des autres spécialités.

Nous pouvons espérer qu'avec l'aide de moyens informatiques suffisants en matériel et en personnel, ce travail pourra déboucher vers l'analyse automatique de textes médicaux ; nous aurons alors supprimé la contrainte importante que représente le codage manuel des dossiers médicaux.

CHAPITRE III

PROCESSUS D'INTERROGATION DES DOSSIERS MEDICAUX

Une analyse rapide des questions rédigées par des cliniciens laisse entrevoir deux problèmes distincts :

- un problème de traitements des informations qui auront été extraites des dossiers pertinents.
- un problème de sélection des dossiers pertinents pour une question posée.

Illustrons ces deux problèmes sur la question suivante :

"Réaliser un tableau de contingence entre le laps de temps séparant l'apparition du diabète de celle de l'artérite des membres inférieurs et le type de diabète chez des femmes mariées ayant eu soit une psychopathie, soit une polynévrite, mais pas de cérébroscélrose.

On aura mesuré lors de la détection du diabète un taux de cholestérol supérieur à 3g/litre, et un taux d'uricémie supérieur à 0,05g/litre. Le laps de temps sera considéré de 1 an à 10 ans par tranche de 1 an."

Pour déterminer les dossiers pertinents pour une telle question, il importe de mettre en évidence les différentes notions sélectives de type :

- a) qualitatif : femme mariée, diabétique, ayant eu une artérite des membres inférieurs, etc...
- b) quantitatif : taux de cholestérol supérieur à 3g/litre, taux d'uricémie supérieur à 0,05g/litre.

Dans la phase de sélection il est nécessaire de prévoir des fonctions de recherche d'informations dans un dossier et de définir une logique de stockage de ces dernières, c'est-à-dire que nous mettrons en évidence des instructions de rangement dans des variables ou des listes de variables.

Dans l'exemple précédent nous devrons mémoriser les dates d'apparition du diabète, de l'artérite et le type de diabète.

La phase "traitement des informations" extraites des dossiers pertinents consiste dans ce cas à calculer le laps de temps séparant les deux dates, à mettre à jour le tableau de contingences. Lorsque le fichier sera complètement examiné nous éditerons ce tableau.

Nous nous proposons dans ce chapitre de décrire les outils utilisés pour résoudre le problème de sélection des dossiers pertinents et celui du stockage des informations. Nous laissons de côté le problème du traitement des informations mémorisées, beaucoup plus classique, nous le reprendrons au chapitre suivant lors de la définition du langage d'interrogation.

A - Structure des questions

1 - Analogie de la structure des questions avec celle des dossiers médicaux.

Compte tenu de la représentation des dossiers médicaux, une idée naturelle pour structurer de telles questions semble être de donner à l'ensemble des différentes notions qu'elles contiennent une structure arborescente.

Chaque sommet de niveau n est porteur d'un descripteur appartenant au lexique L (lexique utilisé lors du codage des dossiers) et caractérisant ce niveau. Il peut lui être accolé une expression booléenne relative aux informations complémentaires du dossier.

Comme le montre l'exemple cité au début de ce chapitre, une question ne peut pas être exprimée par une simple arborescence. Il est nécessaire d'introduire une structure arborescente particulière, permettant de faire intervenir des opérateurs booléens : une question sera une arborescence où les sommets appartenant à une même famille seront reliés par une expression booléenne.

Exemple : La question dont nous venons de parler se représente sous la forme suivante :

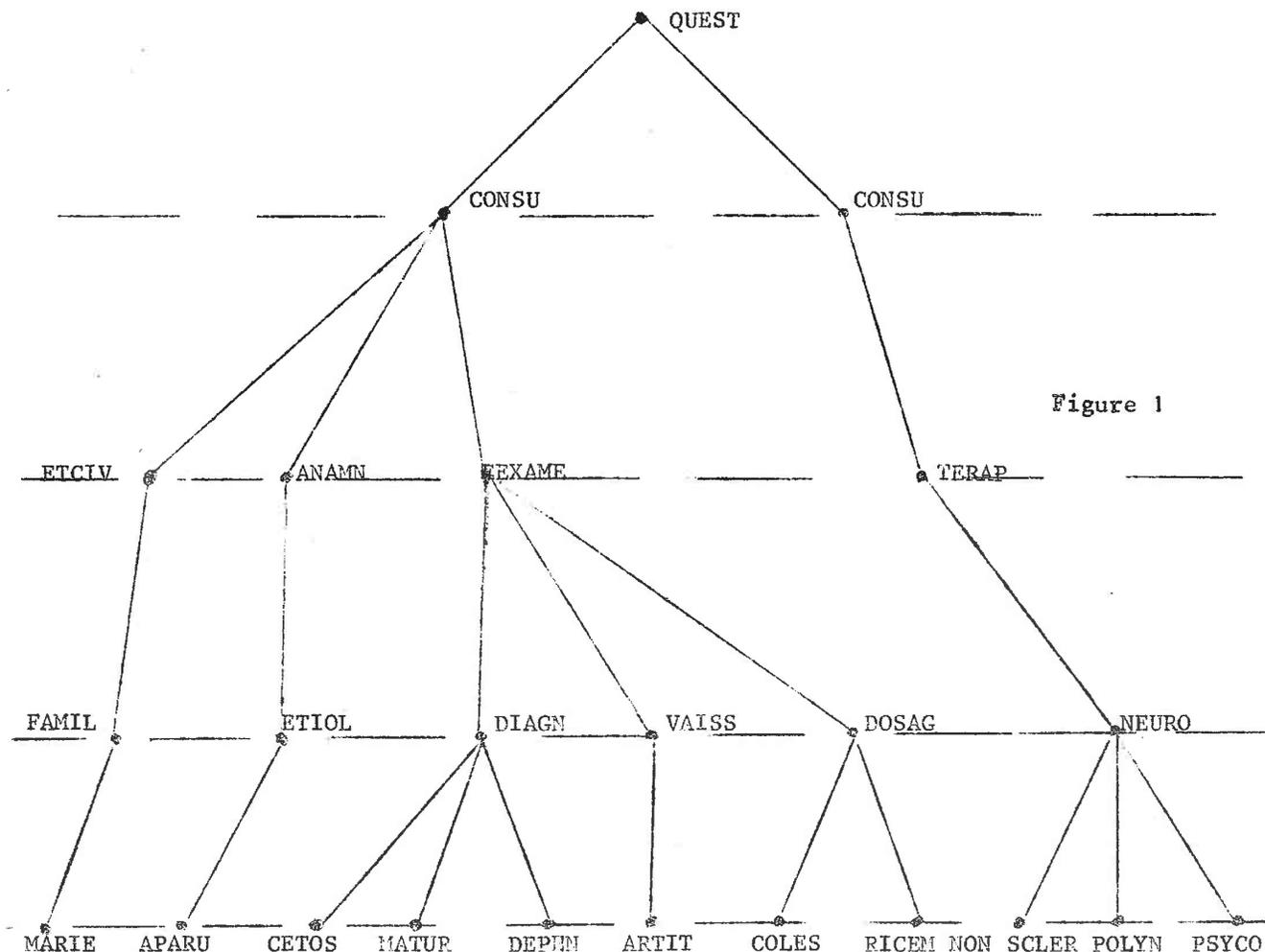


Figure 1

. CETOS, MATUR, DEPEN représentent respectivement les trois types de diabète auxquels nous nous intéressons et ont pour signification cétose diabétique, diabète de la maturité et diabète peu glucido-dépendant.

. Le descripteur FEMME ne figure pas dans le chapitre état-civil car il fait partie de l'en-tête du dossier. Nous avons vu que sa recherche était accélérée.

. Il faut compléter cet arbre par :

- dans chaque famille, des opérateurs booléens entre ses membres :

((CETOS OU DEPEN) OU MATUR)

((PSYCO OU POLYN) ET NON SCLER) ; les autres expressions booléennes reliant les sommets appartenant à une même famille, ne comportant que l'opérateur ET,

- Certains sommets peuvent être qualifiés par une expression booléenne faisant intervenir les informations complémentaires susceptibles d'y être attachées ;

Exemple : Au sommet étiqueté par le descripteur COLES nous accolerons la relation : taux de cholestérol > 3g/litre, au sommet étiqueté par RICEM, la relation : taux d'uricémie supérieur à 0,05 g/litre et enfin au sommet étiqueté par artit la relation : adjectifs qualifiant ARTIT = membre et inférieur.

(1) Nous verrons plus loin que cela ne sera utile qu'aux niveaux 3 et 4.

Remarques :

. (i) le nombre de niveaux est en réalité cinq, si on considère la racine de l'arborescence (QUEST) comme niveau.

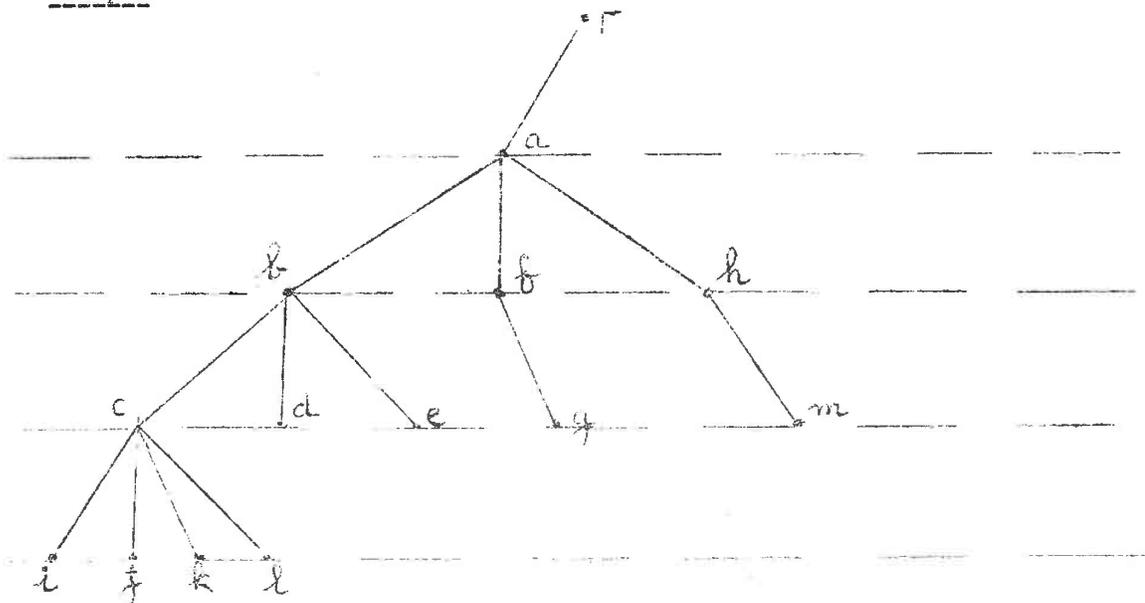
Nous nous sommes fixé arbitrairement certains descripteurs de 1er et de 2e niveau. Si le choix des descripteurs ANAMN et ETCIV étaient implicites, nous verrons que l'utilisateur pourra traduire son indifférence dans le choix des descripteurs de 1er et de 2e niveau.

. (ii) l'étude du champ des questions pouvant être effectivement posée nous a amené à restreindre la généralité des expressions booléennes citées ci-dessus. Nous aborderons ce problème au paragraphe suivant car il importe d'abord de préciser les outils théoriques et les notations que nous allons utiliser tout au long de ce chapitre.

2 - Arborescence (ET/OU)

Nous appelons arborescence (ET/OU) une arborescence (E, Γ) (avec $E = \bigcup_{i=1}^n E_i$ où E_i représente l'ensemble des sommets de niveau i et Γ une relation binaire dans E) où les éléments de $\Gamma(x_i) \forall x_i \in E_i$ sont reliés par une expression booléenne d'opérateurs booléens ET et OU

Exemple:

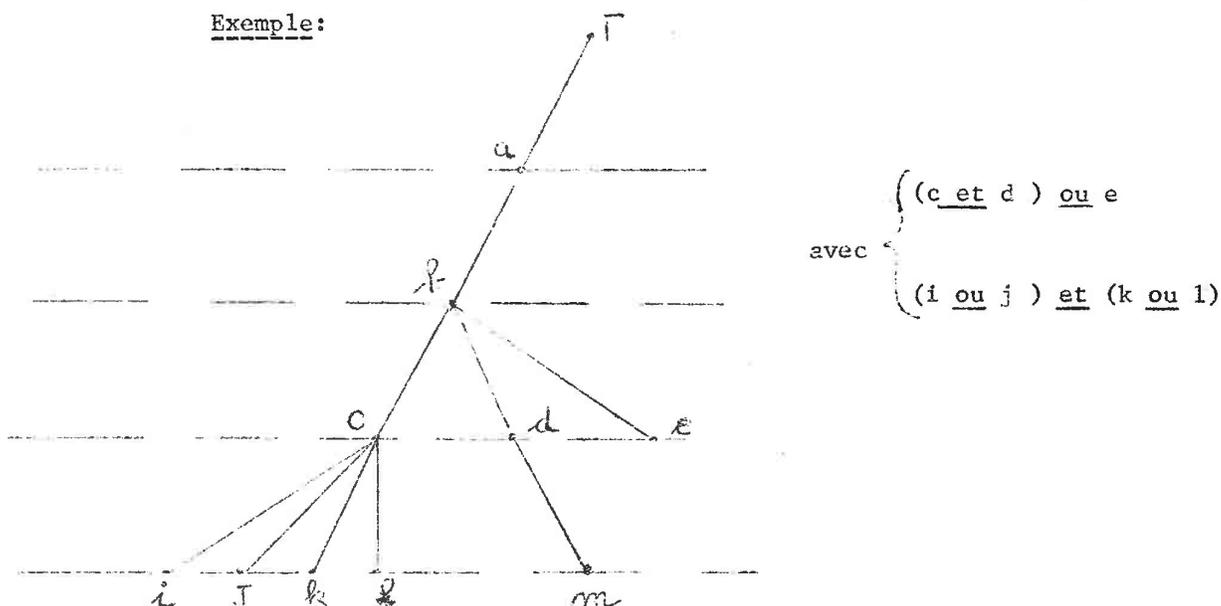


L'arborescence ci-dessus munie des expressions booléennes suivantes: b ou (f et h) ; (c et d) ou e ; (i ou j) et (k ou l) est une arborescence (ET/OU) à quatre niveaux.

Les deux définitions suivantes sont relatives à des arborescences (ET/OU) particulières, non réduites à une racine et telle que $\forall x \in E_1$ ou $x \in E_2$ $\Gamma(x) \neq \emptyset$

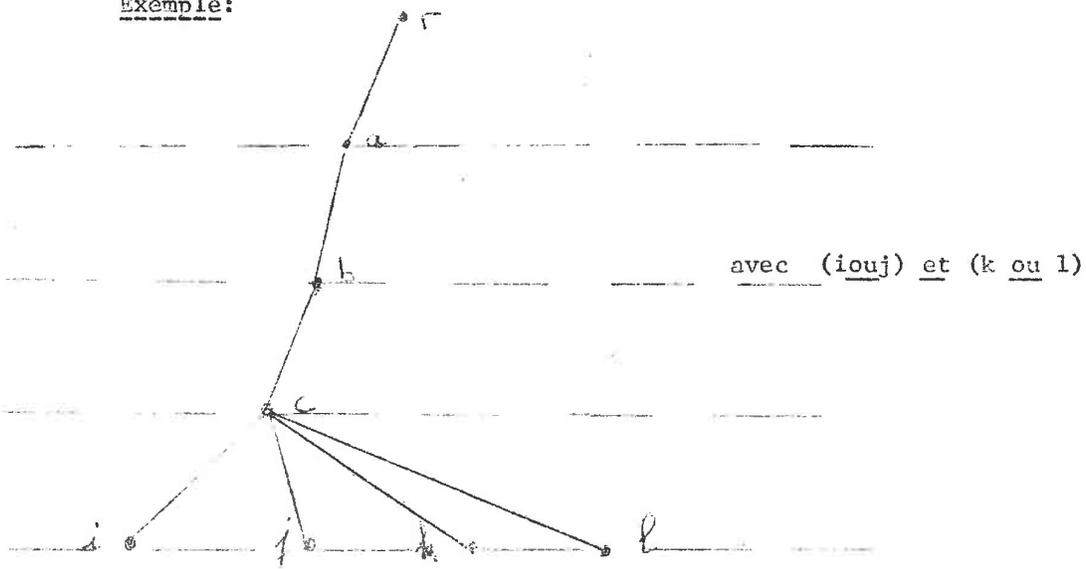
Nous appelons rameau (ET/OU) (E, Γ) une arborescence (ET/OU) ayant un seul noeud de 1er niveau et un seul noeud de 2e niveau.

Exemple:



Nous appelons Branche (ET/OU) $(E; \Gamma)$ un rameau (ET/OU) (E, Γ) n'ayant qu'un seul sommet de niveau 3.

Exemple:



3 - Adaptation aux questions

Chaque question sera structurée par une arborescence du type (ET / OU) (E, Γ) à quatre niveaux à laquelle nous avons apporté certaines modifications dictées par le contexte médical.

a) Chaque arborescence devra être telle que $\forall x \in E_1$ ou $x \in E_2$ (ensemble des sommets de 1er et 2^e niveau) $\Gamma(x) \neq \emptyset$. En effet cela n'aurait aucun sens médical de rechercher uniquement des notions aussi peu spécifiques que celles exprimées aux niveaux 1 et 2 (nous verrons en outre que cette restriction facilite la recherche des dossiers pertinents).

b) D'une façon générale l'utilisateur médecin ne précise pas les descripteurs qu'il veut voir figurer aux sommets de niveau 1 ou (inclusif) de niveau 2 ; une solution lourde et fastidieuse serait par exemple de faire figurer dans l'arborescence (ET / OU) structurant la question tous les descripteurs de 1er niveau ou (inclusif) de 2e niveau. Le lien horizontal entre les sommets de 1er niveau et entre les sommets de 2e niveau serait l'opérateur OU.

Nous avons préféré créer deux descripteurs INDIF1 et INDIF2, que l'utilisateur emploiera pour traduire son indifférence dans l'étiquetage de certains sommets respectivement de niveaux 1 et 2.

c) Nous avons imposé en accord avec les utilisateurs que les liens horizontaux entre les sommets de 1er niveau et entre les sommets de 2e niveau ne soient représentés que par l'opérateur ET; l'expérience montre en effet que l'opérateur OU n'a pas ici d'intérêt pratique, comme l'illustre la question suivante : "Rechercher les individus ayant eu au cours d'une hospitalisation certaine maladie X ou au cours d'une consultation d'autres maladies Y ; une telle alternative n'a pas de sens au point de vue médical.

d) Enfin, il est nécessaire de pouvoir faire figurer dans la représentation arborescente, l'absence de certaines notions. Si ce problème ne se pose jamais pour des notions à caractère très général (placées au niveau 1, comme par exemple : consultation, hospitalisation, ou au niveau 2 comme état-civil, anamnèse, examen...) nous le rencontrons souvent pour des notions plus spécifiques.

Nous avons convenu de faire précéder tout descripteur de niveau 3 ou 4, illustrant une notion que l'utilisateur ne veut voir apparaître dans un dossier, de l'opérateur booléen NON.

Les sommets auxquels sera attaché l'opérateur NON sont nécessairement des feuilles de l'arborescence (au niveau 3 ou 4) et il ne leur sera accolé aucune condition booléenne.

B - Découpages d'une question en questions élémentaires

L'algorithme de recherche des dossiers pertinents pour une question, que nous développerons au paragraphe suivant, est basé sur un principe de séparation d'une question en parties élémentaires. Pour rendre plus clair ce paragraphe, nous présenterons ici, sans les justifier, ces différents découpages.

I - Critères complémentaires.

Nous avons vu qu'à chaque sommet de l'arborescence "question" peut être accolée une expression booléenne où les primaires sont des relations booléennes devant être vérifiées par les éléments d'informations complémentaires (ou addenda.) du dossier et où les opérateurs booléens sont ET et OU.

Nous appelons critères complémentaires de telles relations booléennes. On peut donc accoler à chaque sommet une expression booléenne de critères complémentaires.

2 - Éléments de sélection

Nous appelons éléments de sélection toute question structurée pour une branche (ET/OU).

Exemple:

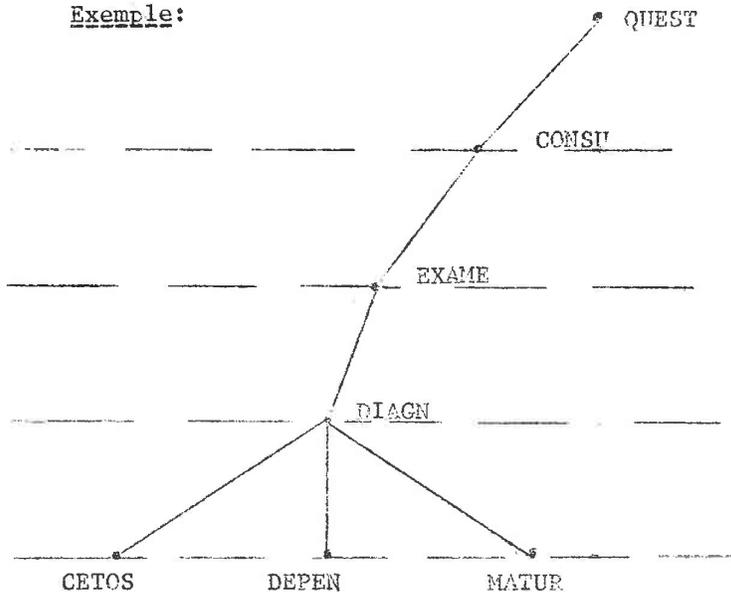


FIG.2

avec . ((CETOS OU DEPEN) OU MATUR)
 . et éventuellement, accolées à certains sommets, des expressions booléennes de critères complémentaires.

3 - Elément de question.

Nous appelons élément de question toute question structurée par un rameau (ET/OU).

Exemple:

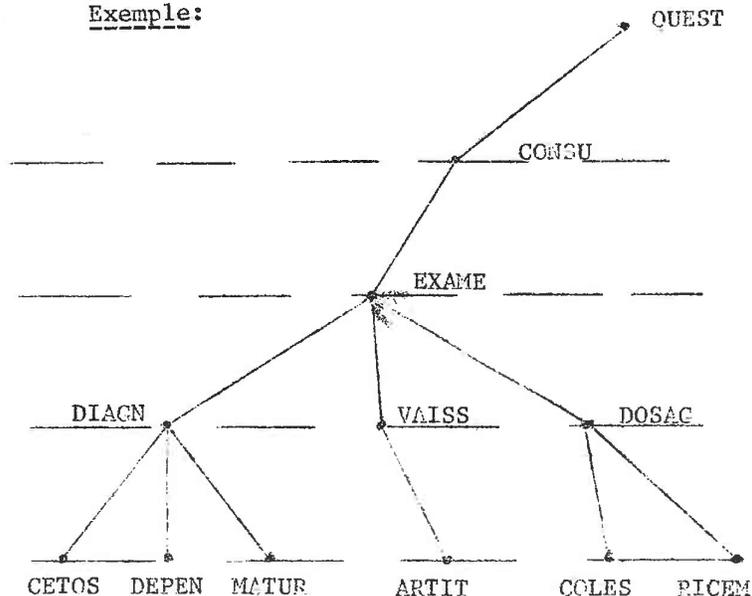


FIG. 3

Il faut ajouter:

- .1'E.B. qui relie les sommets de niveau 3:
(DIAGN ET VAISS) ET DOSAC)
- .les E.B. qui relient les sommets de niveau 4:
(CETOS OU DEPEN) OU MATURE)
COLES ET RICHEM
- . accoler au sommet étiqueté par le descripteur ARTIT, l'E.B. des critères complémentaires signifiant que ARTIT doit être qualifié par MEMBR et IMPER
- . accoler aux sommets étiquetés par COLES et RICHEM, les deux critères complémentaires suivants: taux de cholestérol 3g/l
taux d'urémie 0,05g/l.

4 - Sous-question

Nous appelons sous question toute question structurée par une arborescence (ET/OU) n'ayant qu'un seul sommet de niveau I.

C - MECANISME DE LA RECHERCHE D'UN DOSSIER PERTINENT

La représentation particulière de l'arborescence structurant un dossier, sous forme de tableaux (cf. annexe 4bis) a été conçue en vue de permettre un accès privilégié au 3^e niveau. Toute recherche de structure arborescente dans l'arborescence "dossier" nécessitera donc une exploitation de cette dernière à partir des sommets de 3^e niveau coïncidant avec ceux de la structure recherchée. Pour évaluer la pertinence d'un dossier pour une question, il convient de mettre en évidence dans cette dernière des éléments de sélection ; la recherche correspondante dans l'arborescence "dossier" sera rapide.

L'algorithme de recherche d'un dossier pertinent repose donc sur deux principes :

- La séparation d'une question en un certain nombre d'éléments de sélection,
- L'évaluation de la pertinence d'un dossier pour chaque élément de sélection.

- Principe de séparation

Cette séparation s'effectue en plusieurs étapes :

a) Décomposition d'une question en éléments de question

Soit une arborescence (ET / OU) (Q, \bar{q}) avec $Q = \bigcup_{i=0}^3 Q_i$ (Q_i étant l'ensemble des sommets de niveau i) et telle que $\forall x \in Q_1$ ou $x \in Q_2$ $\bar{q}(x) \neq \emptyset$. Soit $k_1 = \text{card}(Q_1)$ et $k_2 = \text{card}(Q_2)$: il est possible de mettre en évidence dans une telle structure $k_1 \times k_2$ rameaux (Q_r, \bar{r}) où $Q_r = \bigcup_{i=0}^3 Q_{ri}$ (Q_{ri} ensemble des sommets de niveau i du rameau et où \bar{r} est la restriction de \bar{q} à Q_r).

Une question apparaît alors comme une expression booléenne d'éléments de question, de seul opérateur ET. Un dossier sera pertinent pour une question s'il est pertinent pour chaque élément de question.

b) Décomposition d'un élément de question en éléments de sélection

Soit un rameau (ET / OU) (Q_r, \bar{r}) défini précédemment, et soit $k_3 = \text{card}(Q_{r3})$; il est possible de mettre en évidence dans un tel rameau k_3 branches (ET / OU) (Q_b, \bar{b}) où $Q_b = \bigcup_{i=0}^3 Q_{bi}$ (Q_{bi} ensemble des sommets de niveau i de la branche) et où \bar{b} est la restriction de \bar{r} à Q_b .

Un élément de question peut-être considéré comme une expression booléenne d'éléments de sélection reliés par les opérateurs ET et OU, prenant la valeur vraie si le dossier est pertinent pour cet élément.

Un dossier sera pertinent pour un élément de question si l'expression booléenne d'éléments de sélection a la valeur vraie

c) Fonction père - Critère de paternité

Nous venons de voir que l'évaluation de la pertinence d'un dossier pour une question se fera par les évaluations successives de la pertinence du dossier pour les constituants élémentaires de la question. Mais il importe que ces différentes évaluations ne soient pas faites indépendamment les unes des autres : on doit tenir compte des relations de

filiation qui lie plusieurs sommets de 3^e niveau de plusieurs éléments de sélection à un même sommet de 2^e niveau, et plusieurs sommets de 2^e niveau de plusieurs éléments de question à un même sommet de 1^{er} niveau.
Exemple : Revenons à la représentation arborescente (Lig. 1) de la question précédente et considérons par exemple les deux éléments de sélection où les sommets de 3^e niveau sont respectivement étiquetés par DIAGN et VAISS. Lors de l'évaluation de la pertinence d'un dossier pour ces deux éléments de sélection, il sera nécessaire, compte tenu de cette représentation de prendre en compte le fait que ces deux sommets ont un même père (sommet de 2^e niveau étiqueté par EXAME). Cette même contrainte apparaît au niveau des éléments de sélection où les sommets de 2^e niveau sont respectivement étiquetés par ETCIV, AMAMN, EXAME, de même père, sommet de 1^{er} niveau étiqueté par CONSU, mais n'existe pas entre ces derniers et l'élément de sélection dont le sommet de 2^e niveau est étiqueté par TERAP.

Nous appelons fonction père toute fonction qui associe à un sommet de niveau i d'une arborescence, le sommet prédécesseur de niveau $i - 1$; elle s'écrit : père (x_i).

Nous appelons critère de paternité une relation exprimant que deux sommets x_i, y_i de niveau $i = 3$ (ou 2) appartenant à 2 éléments de sélection (ou respectivement à 2 éléments de question) ont un même prédécesseur (père) dans l'"arborescence dossier".

Il s'écrit : père (x_i) = père (y_i).

Dans l'exemple précédent, si x_3 et y_3 sont les sommets respectivement étiquetés par DIAGN et VAISS le critère de paternité entre les deux éléments de sélection s'écrit : père (x_3) = père (y_3).

- Principe d'évaluation

a) Evaluation d'un critère complémentaire

1) - Fonction de recherche et d'exploration d'informations complémentaires

Nous avons défini un certain nombre de fonctions de recherche extrayant en un moment "x" donné, des éléments d'informations complémentaires de types différents. Nous avons vu (chapitre 1er) qu'un élément d'information complémentaire lorsqu'il était de type date ou de type numérique (entier ou réel) pouvait apparaître soit, sous la forme d'une seule valeur, soit, sous la forme d'un intervalle de valeurs. L'utilisateur doit donc avoir la possibilité de rechercher la borne inférieure ou (inclusif) la borne supérieure d'un intervalle de valeurs.

Nous appelons (fdp, fdg) (fhp, fhg) les fonctions recherchant pour un sommet "x" d'une arborescence "dossier" respectivement la borne inférieure de l'intervalle de dates (heures) ou la date fdp (x), (l'heure fhp (x)), la borne supérieure du même intervalle ou la date fdg (x), (l'heure fhg (x)) accolée au sommet x, sinon à un prédécesseur de x.

Nous appelons fvp, fvg les fonctions recherchant pour un sommet "x" d'une arborescence "dossier" respectivement la borne inférieure de l'intervalle de valeurs numériques ou la valeur numérique (fvp (x)), la borne supérieure du même intervalle ou la valeur fvg (x) accolée au sommet "x".

Si l'élément d'information recherché par l'une de ces fonctions est absent nous avons convenu de donner au résultat une valeur négative. En effet, les valeurs numériques mesurées au cours des observations ne

sont jamais négatives, il sera donc, par conséquent, facile de repérer lors de traitements ultérieurs ces marques d'absence d'information.

Enfin signalons que chaque sommet de l'arborescence "dossier" peut aussi être qualifié par un ou deux éléments de type adjectif (l'ordre d'apparition étant quelconque). Nous avons créé une fonction fa recherchant pour un sommet "x" l'adjectif ou les deux adjectifs, s'ils existent, qualifiant le sommet.

Nous écrirons $fa(x) = adj_1 \cup adj_2$, où adj_1 et adj_2 représentent respectivement le code interne du 1er et 2e adjectif accolé au sommet x. Si un seul adjectif est présent alors adj_2 contient une valeur négative, si aucun n'est présent adj_1 contient aussi une valeur négative.

2) - Structure d'un critère complémentaire

. < critère complémentaire $\mathcal{C}(x)$ > : : = < valeur d'une fonct. de recherche au sommet "x" > < opérateur de relation > < valeur >
 . < valeur d'une fonct. de recherche au sommet "x" > : : = fdp(x) | fdg(x) | fhp(x) | fhg(x) | fvp(x) | fvg(x) | fa(x)
 . < opérateur de relation > : : = > | ≥ | < | ≤ | = | ≠
 . < valeur > : : = < constante > | < valeur d'une fonction de recherche en un sommet exploré avant le sommet "x" > (*)
 . < constante > : : = < date > | < valeur numérique > | < descripteur >

(*) Nous verrons que c'est plus compliqué dans le cas d'un adjectif.
Exemple : Soit x le sommet de l'arborescence (ET / OU) (fig. 1), étiqueté par le descripteur COLES, le critère complémentaire relatif au taux de cholestérol s'écrit : $fvp(x) > 3$, ou bien $fvg(x) > 3$, selon que l'utilisateur souhaite, lorsque le taux est mesuré par un intervalle de valeurs, considérer la borne inférieure ou la borne supérieure de l'intervalle.

Si le taux de cholestérol est mesuré par une seule valeur, le résultat sera le même en utilisant $fvp(x)$ ou $fvg(x)$.

3) - Evaluation proprement dite du critère

Avant toute évaluation d'un critère complémentaire $\mathcal{C}(x)$ il faut vérifier que :

- (1) La valeur de la fonction de recherche au sommet "x" n'est pas négative (c'est-à-dire que l'élément d'addenda sur lequel porte le critère est présent dans le dossier).
- (2) La valeur du second membre n'est pas négative (ceci pourrait arriver si l'élément d'information complémentaire recherché en un sommet exploré avant le sommet "x" était absent).

a) Pertinence d'un dossier pour un élément de sélection

Soit un dossier structuré par une arborescence (F, Γ) et soit un élément de sélection structuré par une branche (ET / OU) (Q_b, Γ_b) ; nous appelons L : l'application "étiquetage" qui associe à tout sommet x de ces structures, le code interne L(x) du descripteur l'étiquetant ; si ce dernier est précédé de l'opérateur NON la valeur prise par la fonction est $-L(x)$.

Nous allons examiner la pertinence de ce dossier pour cet élément de sélection dans les deux cas suivants :

1er cas : Le sommet de 3e niveau x_3 n'est pas étiqueté par un descripteur précédé de NON

Le dossier sera pertinent si les conditions numérotées $\alpha, \beta, \gamma, \delta$ sont vérifiées.

α) Il existe un sous graphe (nécessairement une branche) (E_b, X_b) où X_b est la restriction de Γ à E_b et un isomorphisme Ψ qui fait correspondre à tout sommet x_i (de niveau i ; $i = 0, 1, 2, 3$) de la branche (ET / OU) le sommet image $\Psi(x_i)$ de l'"arborescence dossier".

En tenant compte du fait que l'exploration dans l'"arborescence dossier" est faite à partir du 3e niveau nous avons :

$$\Psi(x_3) \Gamma_b^{-1} x_2 \Gamma_b^{-1} x_1 \Gamma_b^{-1} x_0 = \Psi(x_3) X_b^{-1} \Psi(x_2) X_b^{-1} \dots \Psi(x_1) X_b^{-1} \Psi(x_0)$$

x_0 et $\Psi(x_0)$ désignant respectivement la racine de la branche (ET / OU) (Q_b, Γ_b) et de la branche (E_b, X_b) .

β) Soit $\mathcal{L}x$, l'élément de sélection structuré par la branche (ET / OU)

(Q_b, Γ_b) . Les critères de paternité s'ils existent, aux sommets x_i ($i = 2$ ou (inclusif) 3) sont vérifiés : c'est-à-dire que pour tout élément de sélection $\mathcal{L}y$ de sommets y_i , précédant dans l'arborescence $\mathcal{L}x$ et appartenant à la même sous-question, ou (inclusif) au même élément de question nous avons :
- père $(x_i) =$ père (y_i) pour i respectivement égal à 1, 2 et 3.

γ) $\forall x_i$ pour $i = 1, 2, 3$ l'étiquetage est le même aux sommets x_i de la branche (ET / OU) (Q_b, Γ_b) et aux sommets $\Psi(x_i)$ de la branche (E_b, X_b) :
 $L(x_i) = L(\Psi(x_i))$.

Il faut remarquer que si $L(x_1) =$ code numérique de INDIF1 ou si $L(x_2) =$ code numérique de INDIF2, nous considérons que l'égalité est vérifiée quel que soit l'étiquetage respectivement de $L(x_1)$ et $L(x_2)$.

L'expression booléenne de critères complémentaires $\Psi(x_i)$ accolée aux sommets x_i est vérifiée par les éléments des informations complémentaires accolées aux sommets $\Psi(x_i)$.

δ) Nous avons vu que les sommets de niveau 4, $x_4 \in \Gamma_b(x_3)$ sont reliés par une expression booléenne E.B. d'opérateurs ET et OU.

Nous considérons comme primaire booléen tout sommet de niveau 4 auquel nous adjoignons son étiquetage $L(x_4)$ ou $-L(x_4)$ et, si elle existe, l'expression booléenne de critères complémentaires $\Psi(x_4)$.

Nous attribuons la valeur VPAIE à ce primaire booléen :

soit : s'il existe un sommet de niveau 4, $y_4 \in X_b(\Psi(x_3))$ tel que l'étiquetage soit le même que celui de x_4 : $L(x_4) = L(y_4)$, et si l'expression booléenne de critères complémentaires $\Psi(x_4)$ est vérifiée par les éléments d'informations complémentaires accolés au sommet y_4 .

soit : s'il n'existe pas de sommet $y_4 \in X_b(\Psi(x_3))$ avec : $L(y_4) = -L(x_4)$.

La condition δ est vérifiée si la valeur de l'expression booléenne E.B. a la valeur VPAIE.

2^e cas : Le sommet de 3^e niveau $x_3 \in Q_{b3}$ est étiqueté par un descripteur précédé de NON.

Il est plus facile dans ce cas d'examiner la "non-pertinence" du dossier à cet élément de sélection. Il sera non-pertinent si les conditions α, β, γ , sont vérifiées.

• α, β sont les mêmes conditions que dans le 1^{er} cas.

• $\gamma, \forall x_i$ pour $i = 1, 2$: l'étiquetage doit être le même aux sommets x_i de la branche (ET / OU) (Q_b, \bar{Q}_b) et aux sommets $\Psi(x_i)$ de la branche (E_b, \bar{Y}_b), c'est-à-dire $L(x_i) = L(\Psi(x_i))$.

Si elle existe, l'expression booléenne de critères complémentaires $\Psi(x_i)$ accolée aux sommets x_i doit être vérifiée par les éléments des informations complémentaires accolées aux sommets $\Psi(x_i)$.

L'étiquetage du sommet de 3^e niveau doit-être le même à un opérateur NON près : $-L(x_3) = L(\Psi(x_3))$.

c) Pertinence d'un dossier pour une question

Nous avons vu que les évaluations de la pertinence d'un dossier pour des éléments de sélection d'un élément de question ou d'une sous question ne doivent pas être faites indépendamment les unes des autres. Ainsi nous allons voir que la "non-pertinence" d'un dossier pour un élément de sélection pourra remettre en question la pertinence de ce dernier pour des éléments de sélection étudiés auparavant et appartenant au même contexte, c'est-à-dire au même élément de question ou à la même sous-question.

Pour chaque sous-question d'une question tant que le dossier est pertinent faire

éta 2 : Pour chaque élément de question d'une sous-question faire

éta 1 : Pour chaque élément de sélection d'un élément de question faire

Evaluer la pertinence du dossier pour cet élément de sélection (ce dernier étant considéré comme un primaire booléen prenant la valeur VRAIE si le dossier est pertinent, FAUX sinon)

Si le dossier n'est pas pertinent faire

Si l'élément de sélection est relié par un critère de paternité à au moins un autre élément de sélection Ψ lt, déjà examiné faire

Si Ψ lt appartient au même élément de question (critère de paternité accolé au sommet de niveau 3) alors aller à éta 1 (passer au rameau suivant dans le dossier)

Si Ψ lt appartient à la même sous-question (critère de paternité accolé au sommet de niveau 2) alors aller à éta 2 (passer à la sous-arborescence (de racine un sommet de niveau 1) suivante).

Evaluer l'expression booléenne d'éléments de sélection ;
Si la valeur est FAUX alors dossier non-pertinent.

D - ITERATION DU MECANISME- Intérêt

Nous avons vu qu'un dossier médical est généralement constitué de plusieurs consultations et hospitalisations et qu'à l'intérieur de chacune d'elles le malade a pu subir par exemple un ou plusieurs examens ou thérapeutiques. Il semble donc intéressant de pouvoir recommencer la recherche de certaines notions d'une question parmi toutes les rencontres du médecin avec le malade.

L'expérience nous a montré qu'au cours d'un même examen ou d'une même thérapeutique il n'était jamais réalisé deux traitements identiques (par exemple : mesure du poids et de la taille, différents dosages de cholestérol, lipides, glucides, etc...).

Comme cela est vrai également pour l'anamnèse et l'état-civil (représenté aussi par un descripteur de niveau 2), l'itération des parties de questions (voir définition ci-dessus) pertinentes pour ces notions se placera toujours au niveau d'un ou plusieurs éléments de question.

- Procédure d'itération

Nous appelons partie d'un dossier structuré par une arborescence (F, \mathcal{C}) la restriction de ce dossier à une arborescence partielle de (F, \mathcal{C}) de même racine.

Nous appelons partie de question un ou plusieurs éléments de question reliés par l'opérateur ET.

Nous nous proposons de rechercher l'ensemble des parties d'un dossier pertinent pour toute partie de question (Nous nous limiterons à des parties de question soit disjointes, soit emboîtées de la façon suivante " $\{ \{ \}$ ").

Exemple : Dans la question vue au début du chapitre, l'utilisateur pourra demander la recherche de toutes les valeurs supérieures à 3g/litre du taux de cholestérol et supérieures à 0.05g/litre du taux d'uricémie.

Procédure : Dans une première phase nous examinons la pertinence du dossier pour toute la question et nous repérons, pour chaque partie de question dont on doit itérer l'examen, l'endroit N_3 iter dans le dossier (c'est-à-dire la branche, ou plus exactement, l'adresse du sommet de 3^e niveau dans le tableau NI_3 du dossier) à partir duquel il faudra reprendre la recherche.

β Si le dossier est pertinent alors

Pour chaque partie de question dont on itère l'examen faire

Déterminer à partir de l'adresse du sommet N_3 iter une partie, de dossier, pertinente

S'il en existe une alors mémoriser les informations extraites du dossier (comme nous allons le voir), si nécessaire l'adresse du sommet N_3 iter.

E - LOGIQUE DE STOCKAGE DES INFORMATIONS EXTRAITES D'UN DOSSIER

Les informations extraites d'un dossier sont essentiellement soit, des informations complémentaires, soit, des descripteurs dont on désire mémoriser l'existence.

Une méthode de stockage de ces informations serait d'avoir au préalable, lors de l'examen d'un dossier, marqué tous les sommets pertinents de l'arborescence structurant le dossier où l'on désire extraire des informations ; puis le dossier étant pertinent il suffirait d'aller explorer ces sommets et d'y rechercher et mémoriser les renseignements souhaités.

Cette solution permettrait de ne stocker du premier coup que des informations sûres ; par contre elle présenterait le gros inconvénient de ne pas permettre de situer dans le temps ou de façon quantitative des notions l'une par rapport à l'autre.

Exemple : "Dénombrer les dossiers relatifs à des individus ayant eu une cataracte après l'apparition du diabète".

Pour traduire cette relation de postériorité entre l'évènement "apparition de la cataracte" et celui "apparition du diabète", il est nécessaire de mémoriser la valeur "date" au moment de l'examen du dossier.

La méthode que nous utilisons nécessite donc une mémorisation dynamique des informations, c'est-à-dire, un rangement dans des variables ou liste de variables, au moment de l'examen du dossier.

Les informations stockées durant cet examen seront conservées si le dossier est pertinent, retirées sinon.

Il nous a semblé naturel d'utiliser des piles plutôt que des tableaux monodimensionnels pour stocker des listes de valeurs car :

- nous ne nous soucierons jamais du rang d'une valeur dans une liste : le traitement sera toujours relatif à l'ensemble des valeurs, mais jamais à une particulière.
- ce seront toujours, comme nous allons le voir, les dernières valeurs mémorisées qui seront retirées le cas échéant.

Gestion des piles

Nous distinguerons deux types de piles : les piles locales dont nous connaissons tous les éléments après chaque examen de dossiers pertinents et les piles globales dont les éléments seront connus une fois l'étude complète du fichier terminée.

Une pile locale sera remise à zéro avant chaque examen de dossier, une pile globale sera mise à jour (complétée) après chaque examen de dossier.

Chacune de ces piles comporte trois pointeurs : un pointeur P_m indiquant le niveau dans la pile des informations mémorisées, un pointeur P_s indiquant le niveau des informations sûres, et enfin un pointeur P_i indiquant le niveau initial de la pile avant chaque examen de dossier (uniquement pour les piles globales), et avant chaque examen de partie de dossier.

Nous donnons un exemple de gestion de ces piles dans ce qui suit, mais nous pouvons déjà remarquer qu'avant chaque examen de dossier ou partie de dossier nous aurons : $P_m = P_s = P_i$ et qu'à la fin $P_i = P_s = P_m$.

Exemple : Pour faciliter la compréhension de la gestion de ces piles nous allons considérer une question où les informations sont mémorisées au cours de l'examen d'un même élément de sélection : Hts. Les informations recherchées sont les six valeurs du cycle glycémique (glycémie à jeun, avant le déjeuner, après le déjeuner, etc...) et la valeur de la glycosurie. L'élément de sélection est structurée par une branche (ET / OU) où les sommets de niveau 4 sont uniquement reliés par l'opérateur ET et étiquetés par les

six descripteurs AJEUN, AVREP, etc... correspondant aux six valeurs du cycle glycémique et le descripteur GLYCO.

Les valeurs du cycle glycémique sont mémorisées dans une pile locale PL (si nous ne souhaitons pas conserver ces valeurs d'un dossier à un autre), les valeurs de la glycosurie et de la glycémie à jeun dans une pile globale PG (pour la raison inverse).

Nous appelons Eltq l'élément de question auquel appartient Elts.

Nous étudions cette gestion dans les deux cas suivants :

- a) nous recherchons les dossiers pertinents pour cette question et, pour ceux-là, nous mémorisons les valeurs du cycle glycémique et de la glycosurie correspondante (par exemple mesurées à l'apparition du diabète).
- b) dans les dossiers pertinents nous recherchons l'ensemble des parties pertinentes pour l'élément de question Eltq.

. Pgi, Pgs, Pgm sont les pointeurs de la pile PG; Pli, Pls, Plm sont les pointeurs de la pile PL.

1er cas : Pgi à la valeur Pgi₀ avant l'examen du dossier

Pgm = Pgs = Pgi₀ ; Plm = Pls = 0

Pour chaque dossier du fichier faire

- Si le dossier est pertinent pour les éléments de question précédant Eltq dans la question alors
- Pour chaque élément de sélection appartenant à Eltq et précédant Elts dans Eltq faire
 - . Evaluer la pertinence du dossier
 - Lors de l'évaluation de la pertinence du dossier pour Elts Pour chaque valeur recherchée (valeur du cycle glycémique ou de la glycosurie) faire
 - . mémoriser dans les piles PL ou PG la valeur ou l'absence de valeur
 - . incrémenter les pointeurs Plm et Plg :
Plm = Plm + 1 ; Pgm = Pgm + 1
 - Si le dossier est pertinent pour cet Elts alors
(Pls = Plm ; Pgs = Pgm sinon Plm = Pls ; Pgm = Pgs)
 - Pour chaque élément de sélection appartenant à Eltq et placé après Elts dans Eltq faire
 - . évaluer la pertinence du dossier
 - Si le dossier est pertinent pour l'élément de question Eltq et pour les éléments de question placés après Eltq dans la question alors
(Pgi = Pgs = Pgm sinon Pgm = Pgs = Pgi ; Plm = Pls = 0)*

Remarque : Le pointeur Pli ne présente pas d'intérêt dans ce cas car le niveau initial de la pile PL est toujours nul avant chaque examen du dossier.

2^e cas : Nous supposerons que les pointeurs sont initialisés aux valeurs suivantes : Plm = Pls = Pli₀ ; Pgm = Pgs = Pgi₀. Pgi₀ et Pli₀ correspondent ici respectivement aux valeurs Pgi et Pls obtenues après l'examen d'un dossier (1er cas).

Dans cet exemple, une partie de dossier pertinent sera nécessairement structurée par une arborescence partielle réduite à un rameau.

Pour chaque partie du dossier faire

- . Pgi = Pgs ; Pli = Pls
- . le même traitement que dans le 1er cas, en remarquant que l'affectation repérée par une étoile se transforme en :
Plm = Pls = Pli

Remarque : L'exemple précédent se généralise facilement pour plusieurs piles remplies lors d'examen de dossier selon des éléments de sélection différents, et pour des parties de question plus importantes.

CHAPITRE IV

LANGAGE D'INTERROGATION

A - INTRODUCTION

I - OBJECTIFS GENERAUX DU LANGAGE

Notre but a été de fournir aux utilisateurs (médecins informatiens en l'occurrence) un langage relativement simple permettant de décrire facilement les différentes phases de la mise en oeuvre d'une question. C'est à dire que nous nous sommes préoccupés de fournir l'ensemble d'outils, permettant de passer aisément de la question manuscrite formulée par le clinicien à la question codée "assimilable" par l'ordinateur.

Les principales difficultés rencontrées sont dues tout d'abord à la grande diversité de structures des questions posées et à leurs caractères flous. En outre le degré de finesse de ces questions est extrêmement variable. Le langage doit, donc, décrire les questions les plus fines, mais aussi, permettre de réaliser des études statistiques des phénomènes généraux.

Le nombre de questions exprimées jusqu'à présent est relativement faible. Il nous a donc fallu extrapoler les outils qu'il était nécessaire de définir pour répondre au problème.

Nous avons essayé de réaliser un compromis entre un langage trop écriqué, se révélant incomplet lors de l'écriture de nouvelles questions et, un langage à syntaxe plus riche, plus général, mais plus difficilement assimilable par un "non-initié" et posant de sérieux problèmes de compilation.

2 - DESCRIPTION GENERALE D'UNE QUESTION

Nous avons vu, au chapitre précédent que dans chaque question nous pouvions mettre en évidence une phase de sélection des dossiers pertinents et, le cas échéant, une phase de traitement des informations recueillies lors de la sélection.

Ceci nous amène à définir deux langages, chacun adapté à une de ces phases: un langage de sélection et un langage de traitement ou de commande.

a) Langage de sélection

Le langage de sélection doit permettre de décrire aisément, compte-tenu de leurs structures arborescentes, tous les éléments de sélection d'une question et toutes les conditions booléennes associées en certains sommets de ces structures. Il doit aussi permettre l'appel des fonctions de recherche de certaines informations contenues dans un dossier et leur mémorisation dans des variables simples ou dans des piles.

Nous avons choisi une syntaxe s'apparentant à celle qui a été définie lors de l'écriture des dossiers médicaux, c'est à dire s'appuyant sur une structure arborescente et à laquelle nous avons ajouté un certain nombre d'instructions (instructions d'affectation, appels de fonctions de recherche, relations booléennes...) que nous détaillerons dans un paragraphe ultérieur, et qui réalisent les objectifs cités plus haut.

b) Langage de traitement

Il met en évidence deux phases de traitement; l'une relative aux traitements qui seront réalisés après chaque examen de dossier pertinent : traitement local, l'autre relative à ceux qui seront réalisés après l'étude

complète du fichier du traitement global.

(Exemple: Supposons que l'on veuille calculer la moyenne des poids maximaux chez le diabétique. Pour chaque dossier de diabétique, le traitement local consiste à rechercher parmi la pile locale contenant les valeurs du poids d'un individu, la valeur maxima. Ce maximum est une variable globale, qui sera cumulée au cours de chaque traitement local. Le fichier étant complètement examiné le traitement global consiste à diviser la variable globale cumulée par le nombre de dossiers explorés.)

Nous pouvons remarquer que si les traitements locaux sont relatifs à des variables locales ou globales, les traitements globaux n'interviennent que sur les variables globales.

Ce langage est formé d'instructions ayant une syntaxe très simple, et réalisant des traitements élémentaires sur des tableaux de statistiques (dénommement, édition). Lorsqu'il paraîtra insuffisant pour un utilisateur, ce dernier aura la possibilité d'écrire ces deux phases de traitements en FORTRAN (langage dans lequel sont traduites les phases de sélection et de traitement). Il devra également prendre en charge l'enchaînement des questions. Le langage de sélection et les instructions du langage de traitement se présentent alors comme une aide à la programmation.

En effet nous verrons que la phase de traitement et les différentes fonctions incluses dans les instructions de traitement sont écrites modulairement, et par conséquent facilement utilisables par un informaticien.

c) Structure d'une question

- Structure générale

La majorité des questions qui nous ont été posées, sont soit des questions simples, soit des questions laissant apparaître une question principale, suivie d'une ou plusieurs questions secondaires. Ces dernières sont plus sélectives que la question principale et leurs phases de sélection ont en commun la phase de sélection de la question principale.

Tout dossier non-pertinent pour une question principale ne sera pas pris en compte par les questions secondaires. Ces dernières sont indépendantes les uns des autres, c'est-à-dire qu'un dossier sera interrogé successivement par toutes les questions secondaires quel que soit le résultat de chacune d'elles.

Toute question a la structure générale suivante:

. <question> ::= <question principale> <liste des questions secondaires> . La liste des questions secondaires est optionnelle.
 . <liste des questions secondaires> ::= <une question secondaire> |
 <une question secondaire> <liste des questions secondaires>

- Structure détaillée

Nous sommes amenés lors de l'écriture du langage de sélection et de traitement à utiliser un certain nombre de variables, de piles et de tableaux de statistiques. Il apparaît donc nécessaire de préciser avant leurs utilisations leur portée (locale ou globale) et leur structure (notamment pour les tableaux de statistiques.)

Cette phase de déclaration sera alors commune à la question principale et aux questions secondaires.

. <question principale> ::= QUEST <numéro de question> <phase de déclaration> <phase de sélection> <phase de traitement>
 . <question secondaire> ::= <phase de sélection> <plan de traitement>

d) Exemple

Considérons la question suivante:

"Etude de la répartition de l'âge des femmes mariées diabétiques, à l'apparition du diabète, en fonction de leur profession."

"Parmi celles où il apparaît après cinquante ans, étudier la répartition des circonstances d'apparition de ce dernier."

Il apparaît dans cette question une question principale et une question secondaire. Nous allons mettre en évidence la phase de déclaration des variables communes à ces deux questions et leur phase de sélection et de traitement respective.

-Phase de déclaration:

Il est nécessaire de mémoriser la date de naissance, la date de l'apparition du diabète, la profession du sujet et la circonstance d'apparition du diabète. Nous aurons à définir quatre variables qui seront locales. Il faut, en outre, déclarer deux tableaux de statistiques de type global, l'un à deux dimensions (question principale), l'autre à une dimension (question secondaire) et définir les modes de partition des ensembles sur lesquels porte l'étude.

- Phase de sélection: (question principale)

Le langage de sélection devra permettre la mise en évidence des critères de sélection suivants: sujet de sexe féminin, marié, diabétique et la recherche de la date d'apparition du diabète et de la profession du malade.

- Phase de traitement: (question principale)

Le traitement local consiste à calculer l'âge du sujet à la date d'apparition du diabète et à mettre à jour le tableau de statistiques. Le traitement global consiste, par exemple, soit à étudier directement ce tableau, soit à le ranger sur bande magnétique pour faire des travaux ultérieurs de statistique (corrélation, calcul de X^2 par exemple).

- Phase de sélection: (question secondaire)

Le seul critère de sélection est, ici, l'âge à l'apparition du diabète, supérieur à 50 ans.

- Phase de traitement: (question secondaire)

Le traitement local consiste à mettre à jour le tableau de statistiques; le traitement global est semblable à celui de la question principale.

B - PHASE DE DECLARATION

L'utilisateur peut utiliser dans ce langage des variables simples, des piles et des tableaux de statistiques.

La phase "déclaration" commence par le symbole de base DECLARER suivi de trois sortes de déclaration.

. < phase de déclaration > : : = DECLARER < déclaration des variables simples > | < déclaration des piles > | < déclaration des tableaux de statistiques >

L'une au moins de ces trois déclarations devra être présente.

I - DECLARATION DE VARIABLES SIMPLES

Ces variables peuvent être de cinq types différents: réel, entier, code, date et heure. La déclaration du type d'une variable est implicite et réalisée par l'intermédiaire de l'identificateur qui la désigne.

Le nombre des variables utilisées dans une question n'étant jamais très élevé, aussi pour des raisons de simplification dans la reconnaissance du type des variables lors de la traduction, nous avons apporté des restrictions dans le choix des identificateurs.

En outre, le programme de traduction des questions étant écrit en grande partie en FORTRAN, nous avons été contraint de limiter a priori le nombre des variables pouvant être utilisées.

a) Variable entière:

susceptible de prendre toutes valeurs entière, positive, négative, ou nulle.

. < identificateur de variable entière > : : = I E < numéro >

b) Variable code:

L'ensemble des valeurs prises par ces variables est formé par l'ensemble des codes internes des descripteurs du lexique (ces codes sont des nombres entiers de 2 à 4 chiffres).

L'intérêt de ces variables est de permettre la mémorisation lors de la phase sélection de l'existence de descripteurs.

. < identificateur des variables code > : : = I C < numéro >

Exemple: la variable ICI mémorisant l'existence du descripteur ABDOM dans un dossier contient son code numérique 98.

c) Variable date:

Elle contient une date, son contenu s'écrit de la façon suivante:

. < date > : : = < année > < mois > < jour >
 . < année > : : = < suite de 4 chiffres >
 . < mois > : : = < suite de 2 chiffres >
 . < jour > : : = < suite de 2 chiffres >

Si le mois ou (inclusif) le jour ne sont pas notés, ils sont remplacés par les deux chiffres 00.

Exemple: la date mai 1972 s'écrit: I9720500
 . < identificateur des variables dates > : : = I D < numéro >

d) Variable heure:

Elle contient une heure, son contenu s'écrit:

. < date horaire > : : = < heure > < minute >
 . < heure > : : = < suite de 2 chiffre >
 . < minute > : : = < suite de 2 chiffres >

Si la minute n'est pas notée, elle est remplacée par 00
8 h. 50 s'écrit: 0850

. < Identificateur des variables heure > : : = IH < numéro >

e) Variable réelle:

susceptible de prendre toutes valeurs réelle, positive, négative ou nulle.

. < Identificateur des variables réelles > : : = R < numéro >
Nous avons limité le nombre des variables de chacun des types à 30.

. < numéro > : : = I | 2 | - - - - | 29 | 30

Des variables seront remises à zéro après chaque étude de dossier.

Ce sont des variables locales, d'autres seront sauvegardées d'un dossier à l'autre et permettront de réaliser des cumuls de valeur: ce sont les variables globales.

La déclaration du type local est implicite, l'utilisateur devra par contre déclarer explicitement les variables de type global.

. < déclaration des variables simples > : : = GLØB < liste d'identificateur des variables >

. < liste d'identificateur des variables > : : = < identificateur des variables > | < identificateur des variables > . < liste d'identificateur des variables >

. < identificateur des variables > : : = < identificateur des variables entières > | < identificateur des variables réelles > | < identificateur des variables code > | < identificateur des variables date > | < identificateur des variables heure >

Exemple: DECLARER GLØB ICI, ID2, RI;

On apprend par cette déclaration que les variables code ICI date ID2 et réelle RI sont globales.

2 - DECLARATION DES PILES

Les différents types de piles sont les mêmes que ceux que nous avons vus pour les variables simples. La déclaration du type des composants d'une pile est implicite et est faite de la même façon: c'est l'identificateur définissant le nom de la pile qui précise le type.

Pour les mêmes raisons que précédemment nous avons restreint le choix des identificateurs et limité le nombre des piles pouvant être utilisées.

. < Identificateur des piles entières > : : = IPE < numéro >

. < identificateur des piles heures > : : = IPH < numéro >

Pour ces deux types de piles: < numéro > ; : = I 2-----5

. < identificateur des piles dates > : : = IPD < numéro >

. < identificateur des piles code > : : = IPC < numéro >

. < identificateur des piles réelles > : : = RP < numéro >

- < numéro > : : = I | 2 | - - - - - | 10

a) Piles locales et piles globales:

Nous avons vu au paragraphe précédent qu'une pile pouvait être
-soit locale à un dossier c'est-à-dire que tous ses éléments sont connus après chaque examen de dossier pertinent. Elle est remise à zéro avant chaque examen.

-soit globale à tout le fichier c'est-à-dire que l'on ne connaît tous ses éléments qu'après l'étude complète du fichier. Elle est mise à jour lors de chaque examen de dossier pertinent. le symbole de base GLØB précède la liste des identificateurs de la pile globale.

b) Dimension des piles:

Compte-tenu des deux types d'utilisation des piles, nous avons fixé les dimensions des piles à deux valeurs: 100 et 1000. (Précisons tout de suite que ces valeurs ainsi que le nombre de piles disponibles fixé précédemment ne sont pas figés et peuvent être facilement modifiés).

Dans chaque type de piles la distinction de ces dernières selon leur dimension est réalisée d'après le numéro qui figure à la fin de l'identificateur. (les identificateurs des piles de type entier et heure se terminant par 1, 2, 3, et ceux des pile de type date, code et réel se terminant par un numéro de 1 à 5 corespondent à des piles de dimension 100; les autres piles sont dimensionnées à 1000).

Dans chaque pile le rang de la dernière. valeur mémorisée figure dans le premier élément.

Etant donné que le nombre d'informations susceptibles d'être mémorisées lors de l'étude d'un dossier ou du fichier est très variable, il était nécessaire d'offrir à l'utilisateur plus de souplesse dans l'utilisation de ces piles.

Ces dernières étant rangées en mémoire par type et selon l'ordre croissant des numéros d'identificateur, l'utilisateur pourra "déborder" d'une pile sur celles dont le numéro d'identificateur est immédiatement supérieur; Il devra, alors, dans ce cas, déclarer la taille des piles "débordantes" et ne pas utiliser celles qui sont en mémoire consécutives à ces piles. La liste des déclarations des tailles de piles est précédé du nom réservé PILE.

- . < déclaration de pile > : : = < déclaration des piles globales >
- < déclaration des dimensions des piles >
- . < déclaration des piles globales > : : = GLØB < liste d'identificateurs de piles >
- . < liste d'identificateurs de piles > : : = < identificateurs de piles >
- ; | < identificateurs de piles > : < liste d'identificateurs de piles >
- . < identificateurs de piles > : : = < identificateurs de piles entieres > |
- < identificateurs de piles réelles > | < identificateurs de piles >
- heures > | < identificateurs de piles date > | < identificateurs de pile >
- code >
- . < déclaration de dimensions de piles > : : = FILE < liste de piles dimensionnées >
- . < liste de piles dimensionnées > : : = < piles dimensionnées > ;
- < piles dimensionnées > ; < liste de piles dimensionnées >
- . < pile dimensionnées > : : = < identification de piles > (< taille >
-);
- . < taille > : : = < nombre entier >

Exemple:

```
GLØB IPD1, IPC2, RP2.;
FILE IPD1 (200), IPD3 (300);
```

L'utilisateur ne peut utiliser les piles IPD2 et IPD4, IPD5 La déclaration des dimensions des piles doit être faite pour un type donné dans l'ordre croissant des numéros d'identificateurs. Dans tous les cas de débordement de piles non prévus par l'utilisateur, un message d'erreur est imprimé et selon le cas soit on passe au dossier suivant soit on arrête l'étude de la question.

c) Piles parallèles:

Nous aurons à réaliser, dans la suite du chapitre des traitements sur des piles gérées simultanément et de façon identique. (c'est à dire que les pionteurs indiquant les niveaux des piles s'incrémentent ou se décrémentent de la même quantité et en même temps). Pour éviter la redéfinition de telles piles nous les appellerons : piles parallèles.

Supposons que l'on désire stocker une liste de couples d'informations, composées des poids d'un individu et des dates respectives au moment des mesures. Nous utiliserons deux piles parallèles R1 (poids) et R

Cette notion apparaîtra intéressante chaque fois que nous désirerons mémoriser une liste de couples, de triplets ou plus d'informations.

Exemple:

Supposons que l'on désire stocker une liste de couples d'informations, composée des poids d'un individu et des dates respectives au moments des mesures. Nous utiliserons deux piles parallèles RP1 (poids) et BD2 (date) telles que leurs pointeurs pointent respectivement sur les valeurs poids et âge qui se correspondent. Si l'une des deux informations est absente, nous avons vu que nous concrétisons cette absence en mettant une valeur négative.

3 - DECLARATIONS DE TABLEAUX DE STATISTIQUES

a) Tableaux de statistiques:

Les études statistiques nous sont, en général, demandées sous forme de tableaux soit à 1 dimension, soit à 2 dimensions (ils sont alors appelés "tableaux croisés ou à double entrée"). Nous allons mettre en évidence la structure de ces deux tableaux à travers les deux exemples suivants:

Exemple 1 :

"Etude de la répartition du type d'apparition du diabète en fonction du sexe"

Le tableau statistique correspondant à cette étude comporte des rangées "lignes" et des rangées "colonnes". Il pourra être associé par exemple, aux rangées "lignes" la partition des malades selon leur sexe, aux rangées "colonnes" la partition selon le type d'apparition du diabète.

A chaque rangée "ligne ou colonne" est associé une possibilité offerte par la partition correspondante. Pour cet exemple, les deux rangées "lignes" sont relatives respectivement aux malades de sexe masculin et de sexe féminin; chaque rangée "colonne" peut être relative à un type ou comme nous le verrons à une expression booléenne des types d'apparition du diabète.

L'appartenance d'un élément à ce tableau est caractérisée par son appartenance à une partie du produit des 2 partitions.

Exemple 2:

"Etude de la répartition du poids des individus à l'âge d'apparition du diabète"

Le tableau statistique correspondant à cette étude ne comporte qu'un seul type de rangées ("lignes" ou "colonnes").

Le mode de partition des individus selon la valeur de leur poids à l'apparition du diabète peut être tel que les intervalles des valeurs obtenues soient réguliers ou irréguliers. A chaque rangée "ligne" ou "colonne" est associé un intervalle ou une expression booléenne d'intervalles de valeurs.

Un élément appartient au tableau s'il appartient à une partie de la partition.

A l'issue de ces deux exemples nous dégageons deux notions inspirées du système S.A.D de Th. GIRARD (2) celles de caractéristiques et de caractères.

Nous appelons caractéristique le mode de partition d'un ensemble E (dans notre cas E sera l'ensemble des malades du fichier) et caractères

- La première classe de travaux consiste à réaliser un dénombrement de "cas" déterminés par la combinaison d'un caractère "ligne" et d'un caractère "colonne" (exemple : "enfant pesant à la naissance plus de 4 kg et né après l'apparition du diabète" constitue un cas).
- Dans la deuxième classe nous mettrons en évidence les mêmes cas que précédemment, mais pour chacun d'eux nous cumulons la valeur d'un élément d'information complémentaire du dossier (nous pourrions cumuler ici le nombre d'enfants).

c) Mise à jour des tableaux

Il peut paraître surprenant que nous abordions ce problème dans ce chapitre, mais nous verrons que le mode de mise à jour de ces tableaux conditionne largement leur mode de déclaration. Compte tenu du fait que nous ne sommes assurés en général de la validité des informations mémorisées lors de l'examen d'un dossier qu'à la fin de cet examen, tout tableau de statistiques sera mis à jour lors de la phase "traitement", c'est-à-dire lorsque nous aurons déterminé la pertinence du dossier à la question.

Ceci nous impose de mémoriser lors de la phase "sélection" la valeur ou la liste des valeurs (valeurs numériques, dates ou descripteurs) du champ caractérisant le mode de partition.

A chaque caractéristique nous associons soit un identificateur de variable simple, soit un identificateur de pile (selon que nous recherchons une ou plusieurs occurrences du champ, le type de ces identificateurs étant conforme au type de la partition), soit enfin une des deux fonctions suivantes : la fonction SEXE prenant la valeur HOMME ou FEMME et la fonction AGE (paramètre) où le paramètre est soit une variable, soit une pile selon que l'on recherche l'âge ou la liste des âges aux dates spécifiées dans la variable ou dans la pile.

Exemple : "Etude de la répartition du poids, après l'apparition du diabète, en fonction de la taille d'un individu".

L'étude est faite sur tout le fichier.

1er cas : Nous supposons qu'il s'agit d'un adulte et que la taille ne varie pas tout au long de l'examen.

A la caractéristique "valeur du poids après l'apparition du diabète" nous associons l'identificateur de pile réelle RP1, à la caractéristique "valeur de la taille" l'identificateur de variable réelle R1. A l'issue de la phase sélection les valeurs des poids et de la taille sont mémorisées dans la pile et la variable.

Au cours de la phase traitement et lors de la mise à jour des tableaux de statistique le traitement suivant sera réalisé :

Pour chaque élément du sommet de la pile RP1 (h) tant que
h ≠ 0 faire :

- Constituer avec la variable R1 un couple de valeurs.
- Sélectionner les caractères ligne et colonne, c'est-à-dire l'élément du tableau à incrémenter.
- Décrémenter le pointeur de pile : $h = h - 1$.

2èm cas : Nous supposons qu'il s'agit d'un enfant et que la taille varie.

A la caractéristique "valeur de la taille" nous associerons l'identificateur de pile réelle RP2.

Nous réalisons le traitement suivant :

- Pour chaque élément du sommet de la pile RP1 (h) tant que
- h ≠ 0 faire :
- Constituer avec l'élément RP2 (h) un couple de valeurs.
 - Sélectionner l'élément du tableau à incrémenter.
 - Décrémenter les pointeurs de pile : h = h - 1.

Remarque

Il est nécessaire dans ce dernier cas que les deux piles soient parallèles.

D'autre part, nous ne prenons pas en compte les couples de valeurs dont l'une est négative (valeur traduisant une absence d'information).

d) Notion de dimension

Conjointement à la déclaration d'un tableau de statistiques, il apparaît nécessaire de préciser la ou les caractéristiques associées et de définir les différents caractères. Pour permettre une édition des tableaux beaucoup plus souple pour l'utilisateur, il est avantageux d'ajouter les libellés accompagnant chaque descripteur de caractères.

Nous rejoignons alors la notion de dimension, donnée par Jean-François DUFOURD (thèse de 3^e cycle) pour un tableau de statistiques.

Nous donnons ci-dessous un exemple de tableau relatif à la question vue au paragraphe précédent, ainsi que les déclarations des dimensions correspondantes. Les caractères de la caractéristique "valeur du poids" sont associées aux rangées "lignes", ceux de la caractéristique "valeur de la taille" aux rangées "colonnes".

	TAILLE ≥ 150	TAILLE ≥ 160	TAILLE ≥ 170	TAILLE ≥ 180
POIDS ≥ 50				
POIDS ≥ 60				
POIDS ≥ 90				

Les dimensions seront alors déclarées :

DIM PDS / RP1 / ZONES 50, 60, 90, 130 / 'POIDS ≥ /,

DIM TAIL / RP2 / (150, 190, 10) / 'TAILLE ≥ /,

où RP1 et RP2 sont les identificateurs des piles contenant la liste des valeurs du poids et de la taille.

Plusieurs tableaux pouvant avoir une dimension commune, pour éviter la répétition des déclarations de dimensions il nous a semblé souhaitable de séparer la déclaration proprement dite des tableaux de statistiques de celle de leurs dimensions.

Nous distinguerons donc les déclarations de dimensions (exemple de déclarations ci-dessus) et les déclarations proprement dites des tableaux

de statistiques (exemple : T1 (PDS, TAIL), T2 (TAIL), T3 (TAIL, PDS)...) associant à chaque tableau une ou deux dimensions.

- . < déclar. de dim. > ::= < liste de dim. >
- . < liste de dim. > ::= < dim. > ; < dim. > , < liste de dim. >

- Chaque dimension est précédée du symbole de base DIM. > /
- . < dim. > ::= DIM < nom de dim. > ! < définition de dim. >
 - < nom de dim. > ::= < lettre > | < lettre > < suite d'au plus 3 caractères alphanumériques. >
 - . < définition de dimension > ::= < ident. associée à la caractéristique > / < définition de partition numérique >
 - < ident. associée à la caractéristique > / < définition de partition lexicographique >
 - < définition de partition booléenne >
 - . < ident. associée à la caractéristique > ::= < ident. de variable > | < ident. de pile > | < nom de fonction >
 - < ident. de variable > ::= < ident. de variable réelle > | < ident. de variable entière > | < ident. de variable date > | < ident. de variable code >
 - < ident. de pile > ::= < ident. de pile réelle > | < ident. de pile entière > | < ident. de pile date > | < ident. de pile code >
 - < nom de fonction > ::= SEXE | AGE (< paramètre >)
 - < paramètre > ::= < ident. de variable date > | < ident. de pile date >
 - . < définition de partition > ::= < définition de partition numérique > | < définition de partition lexicographique > | < définition de partition booléenne >

Nous allons maintenant reprendre en détail les différents types de partition.

- Définition de partition numérique

Nous avons vu que l'ensemble des valeurs susceptibles d'être prises par la variable ou la pile de type numérique ou date associée à la caractéristique pouvait être partagée soit en intervalles de même longueur (partition régulière), soit en intervalles de longueur variable (partition irrégulière).

Soit \mathcal{L} cet ensemble de valeurs, il s'écrit :

$$\mathcal{L} = \bigcup_{i=1}^n a_i, a_i + 1$$

avec $a_i \in \mathbb{R}$ si la variable ou la pile est réelle,
 $a_i \in \mathbb{N}$ si la variable ou la pile est de type entier,
 $a_i \in \mathcal{D}(\mathbb{N})$ (où $\mathcal{D}(\mathbb{N})$ désigne l'ensemble des entiers représentant des dates) si la variable ou la pile est de type date.

- . A chaque intervalle $[a_i, a_i + 1[$ nous associons un caractère i .
- . Si $a_{i+1} - a_i = a_i - a_{i-1} = \text{lt. q. } a_{i+1} > a_i \forall i$ la partition est régulière, elle est caractérisée par une borne inférieure, une borne supérieure de l'ensemble et la longueur commune des intervalles.
- . Si $a_{i+1} - a_i$ est de longueur quelconque, la partition est irrégulière, elle est caractérisée par la donnée d'une liste de bornes, la première borne étant la borne inférieure de l'ensemble, la dernière la borne supérieure de l'ensemble. La description de cette liste est précédée du nom réservé ZONES
- . < définition de partition numérique > ::= < définition de caractères

numériques > / libellé

Le libellé est facultatif.

- < définition de caractères numériques > ::= (< borne inférieure > < borne supérieure > < pas >) | 'ZONES' < liste de bornes >
- $\left\{ \begin{array}{l} \text{< borne >} \\ \text{< borne inférieure >} \\ \text{< borne supérieure >} \\ \text{< pas >} \end{array} \right\} ::= \text{< nombre entier > | < nombre réel > | < date >}$
- < liste de bornes > ::= < borne > < borne > +
- < borne > ::= < nombre entier > | < nombre réel > | < date >

Le signe < >+ indique la possibilité de répéter la quantité entre chevrons au moins une fois (dans notre cas ce nombre est égal à n).

• Écriture du libellé : si le libellé est présent il est édité, concaténé à la borne inférieure de l'intervalle [ai, ai+1[définissant le caractère i, en tête de chaque rangée "ligne" ou "colonne". S'il est absent nous éditons toutefois la borne inférieure de l'intervalle.

• < libellé > ::= ' < chaîne d'au plus dix caractères alphanumériques > '

• Exemple de déclaration de dimensions à partition numérique

Reprenons l'exemple vu précédemment :

DIM PDS / RP1 / ZONES 50,60, 90, 130 / 'POIDS', '/',

DIM TAIL / RP2 / (150, 190, 10) / 'TAILLE', '/';

la première dimension définit 3 caractères numériques c'est-à-dire 3 classes de poids, la deuxième dimension 4 classes. Les libellés associés à ces caractères sont mis en évidence sur le dessin précédent.

- Définition de partition lexicographique

Nous avons vu que l'ensemble des valeurs susceptibles d'être prises par la variable ou la pile de type code associée à la caractéristique est défini par un ensemble de descripteurs introducteurs de niveau ou adjectifs, tous en général relatifs à une même notion. A la place d'un descripteur nous pouvons trouver le symbole RESTE définissant la classe des éléments n'appartenant pas à l'ensemble.

La partition de cet ensemble est caractérisé par une liste de descripteurs éventuellement suivie du symbole RESTE.

La description de cette liste est précédée du nom réservé DESC

• < définition de partition lexicographique > ::= < définition de caractère lexicographique > / < liste de libellé >

• < définition de caractères lexicographiques > ::= DESC < descripteur > < descripteur >+

Les descripteurs sont ceux qui appartiennent à l'ensemble vu plus haut.

• Écriture du libellé : si la liste des libellés est absente, nous éditons pour chaque caractère i un libellé formé du descripteur associé au caractère. Si l'utilisateur désire faire éditer d'autres libellés, il donnera une description de la liste des libellés telle que le premier soit associé au caractère numéro 1, le deuxième au caractère numéro 2, et ainsi de suite.

• < liste de libellé > ::= < libellé > < libellé > +

Le nombre de répétition de la quantité entre < > est la même que ci-dessus.

. Exemple de déclaration de dimension à partition lexicographique

Soit la question "répartition du type des signes électriques d'insuffisance coronarienne en fonction du sexe chez un diabétique".

Les descripteurs LESIO (courant de lésion), NECRO (infarctus) et ISCHI (infarctus limité) recouvrent la notion "signes électriques d'insuffisance coronarienne. La variable code ICI contient à l'issue de la phase sélection le code d'un de ces descripteurs.

Les deux dimensions suivantes seront déclarées :

DIM SIGN / ICI / DESC LESIO, NECRO, ISCHI/ ,

DIM SEX / SEXE / DESC HOMME, FEMME/;

pour un diabétique de sexe masculin ayant un infarctus nous sélectionnerons les caractères numéro 2 de la première dimension et le caractère numéro 1 de la deuxième .

Remarque :

Lorsque la partition lexicographique nécessite plus de cinq descripteurs, nous réarrangeons par ordre croissant la liste des codes numériques de ces derniers pour faire une exploration dichotomique.

- Définition de partition booléenne

L'ensemble des valeurs susceptibles d'être prises par la variable ou la pile associée à la caractéristique recouvre les deux ensembles précédents, Selon le type de la variable ou de la pile, chaque caractère est défini par un ou plusieurs critères lexicographiques ou numériques. Ces derniers sont reliés entre eux par l'opérateur booléen OU et ET, aucune priorité n'existant entre les opérateurs. En l'absence des parenthèses l'expression booléenne de conditions est évaluée de gauche à droite.

- < définition de partition booléenne > ::= BOOL < liste de caractères booléens > / < liste des libellés >
- < liste de caractères booléens > ::= < caractères booléens > | < caractère booléen > / < liste de caractères booléens >
- < caractères booléens > ::= < E.B. de critère lexicographique ou numérique >
- < critère lexicographique > ::= < prim. lex. > < opérateur de comparaison > < second lex. >
- < prim. lex. > ::= < identificateur de variable code > | < identification de pile code > | SEXE
- < second lex. > ::= < identificateur de variable code > | < descripteur >
- < opérateur de comparaison > ::= # | =
- < critère numérique > ::= < prim. num. > < opérateur de relation > < second numérique >
- < prim. num. > ::= < identificateur de variable numérique > | < identificateur de pile numérique > | AGE | < paramètre date >
- < opérateur de relation > ::= > | >= | < | <= | ≠
- < second numérique > ::= < identificateur de variable numérique > | < identificateur de pile numérique > | < constante >
- < constante > ::= < nombre entier > | < nombre réel > | < date >

Le type de l'identificateur et celui de la constante devront se correspondre.

. Écriture du libellé : si la liste des libellés est présente, la description est la même que pour la dimension à partition lexicographique : le premier libellé de la liste est associé au premier caractère, le deuxième libellé au deuxième, et ainsi de suite. Si la liste des libellés est absente nous éditons pour chaque caractère le libellé suivant : CRITERE concaténé à i (rang du caractère dans la liste).

. Exemple de déclaration de dimension à partition booléenne

```
DIM FORM / BOOL (R1 ≤ 50 ET R2 ≤ 150), ((R1 ≥ 90 ET R2 ≤ 150) OU
(R1 ≤ 50 ET R2 ≥ 170)), (R1 ≥ 70 ET R1 ≤ 90 ET R2 ≥ 160 ET R2
≤ 180), (R1 ≥ 90 ET R2 ≥ 80) ;
```

la valeur du poids et de la taille d'un malade est mémorisée respectivement dans la variable R1 et R2. Si nous avons mémorisé dans la variable R1 : 76 kg et dans R2 : 176 cm nous sélectionnerons le caractère numéro 3.

- Déclaration proprement dite de tableaux de statistiques

La construction de tableaux de dénombrement pour chaque dossier aurait peu de sens car l'échantillonnage serait trop faible. Nous avons donc considéré qu'implicitement tous les tableaux de statistiques auraient le type global.

Chaque tableau sera caractérisé par la donnée d'un ou de deux noms de dimensions qui sont associés aux rangées "ligne" et "colonne".

```
< déclaration proprement dite de tableaux > :: = TAB < liste de
tableaux de statistiques >
< liste de tableaux de statistiques > :: = < tableaux de statistiques >
< liste de tableaux de statistiques > | < tableaux de statistiques > ;
< tableaux de statistiques > :: = < identificateur de tableaux >
( < liste de noms de dimension > )
< liste de noms de dimension > :: = < nom de dimension > |
< nom de dimension > < nom de dimension >
< identificateur de tableaux > :: = < T > < entier >
< entier > :: = 1 | 2 | 3 | 4 | 5.
```

nous avons prévu en effet, la possibilité d'utiliser dans une même question cinq tableaux de statistiques. Le nombre maximum de rangées "ligne" ou "colonne" a été fixé à 20 ; en réalité chaque tableau aura une ligne et une colonne de plus pour loger éventuellement les pourcentages lignes et colonnes. Ces estimations nous ont paru suffisantes jusqu'à présent, elles pourront être facilement modifiées ultérieurement si cela est nécessaire.

. Exemple de déclaration de tableaux de statistiques

Considérons la question suivante : "Etudes de la répartition du type d'apparition du diabète et de l'âge au moment de l'apparition en fonction du sexe.

Nous utilisons deux tableaux de statistiques T1 et T2 ayant une dimension commune.

Nous déclarons trois dimensions :

```
DIM DIAB / ICI / DESC ACCID, OPERA, MALAD, AUTRE /,
```

```
DIM AGE / AGE (IO1) / (20, 80, 10) / 'AGE = ' /,
```

```
DIM SEX / SEXE / DESC HOMME, FEMME /;
```

Nous déclarons deux tableaux :

```
TAB T1(DIAB, SEX), T2(AGE, SEX) ;
```

C) PHASE DE SELECTIONI - INTRODUCTION

Le langage de sélection que nous allons définir permet de décrire toute phase de sélection appartenant à une question principale ou à une question secondaire. Nous avons vu que ce langage devait avoir une syntaxe facilitant la traduction des différents outils (arborescence (ET / OU), fonctions de recherche, critères complémentaires...) mis en évidence au chapitre III.

Nous allons donner ex-abrupto l'écriture dans ce langage de sélection, de la phase de sélection d'une question principale et d'une question secondaire. Nous avons, en réalité, repris pour la question principale, la question vue au chapitre précédent, afin que le lecteur puisse se reporter à la description arborescente de la phase de sélection et comprendre la signification des descripteurs étiquetant cette dernière.

Nous avons toutefois complété cette question pour lui donner un caractère aussi général que possible et pour qu'elle rende compte de toutes les possibilités du langage. Pour cette raison elle pourra apparaître artificielle pour un médecin.

Question 1 principale : "Réaliser un tableau de contingence entre le laps de temps séparant l'apparition du diabète de celle de l'artérite des membres inférieurs et le type de diabète chez des femmes mariées ayant eu une atteinte coronarienne, mais n'ayant pas de manifestations endocriniennes. Ces malades auront soit une psychopathie, soit une poly-névrite, mais pas de cérébroscélrose. On aura mesuré lors de la dilatation du diabète un taux de cholestérol supérieur à 3g/litre, et un taux d'uricémie supérieur à 0,05g/litre".

Le laps de temps sera considéré de 1 an à 10 ans par tranche de 1 an

Question 1 secondaire : "Etude de la probabilité d'amputation des membres inférieurs en fonction de l'âge du malade à l'apparition du diabète et pour lequel le laps de temps défini précédemment est inférieur à trois ans". Nous considérerons des malades âgés entre 30 et 60 ans, par tranche de 10 ans.

Ecriture de la phase de sélection des questions principale et secondaire. Question principale

SELECT (SEXE = FEMME)

CONSU - ANAMN - ETIOL - APARU/DATP → ID1/
 ETCIV - FAMIL - MARIE
 EXAME - [DIAGN - {CETOS/DES → IC1/OU
 MATUR/DES → IC1/OU
 DEPEND/DES → IC1/ET
 VAISS - ARTIT AVEC (ADJ = MEMBR) ET
 (ADJ = INFER) ET
 (DATP ID1)/DATP → ID2/ET
NON ENDOC ET

```

DOSAG - { COLES-AVEC (VALP > 3) ET
          RICHEM AVEC (VALP > 005) }
CONSU-TERAP - NEURO- { NON SCLER ET
                     (POLYN OU
                     PSYCO) } ##

```

• Question secondaire

```

SELECT (ID3 < 30 000)†
CONSÜ-EXAME-VAISS-AMPUT AVEC (ADJ=MEMBR) ET
                          (ADJ=INFER) ##

```

ID3 est la variable de type date contenant la valeur du laps de temps. Rappelons que 3.00.00 est l'écriture sous la forme année, mois, jour de 3 ans. Cette variable est remplie lors de la sélection du dossier selon la question principale.

2 - DESCRIPTION DU LANGAGE

a) Structure générale d'une phase de sélection

Nous avons vu au chapitre 1 que certains éléments d'information complémentaire (tels que par exemple, la date de naissance du malade, son sexe, etc,...) dont l'existence dans le dossier est essentielle, sont logés en tête du dossier et sont, par conséquent, d'accès rapide. Il semble donc naturel, lorsque la pertinence d'un dossier dépend de la valeur de ces informations, de leur donner un caractère très sélectif. Nous placerons alors en tête de la phase sélection les conditions booléennes relatives à ces informations afin qu'elles soient les premières évaluées.

Cette sélection rapide d'un dossier pourrait être généralisée de la façon suivante : lors de la construction d'un dossier nous placerions en tête du dossier non seulement les informations complémentaires que nous venons de citer, mais aussi un mot "silhouette" où à chaque position binaire serait associé un mot clé (descripteur), reflétant une information originale (maladie ou traitement particulier) du dossier. Chaque position binaire contiendrait le bit 1, si l'information existe, 0 sinon.

Toute question commencerait éventuellement par une expression booléenne de "mot - clés" et une première sélection consisterait à vérifier que le dossier examiné a la silhouette recherchée.

Exemple : nous avons placé en tête de la "phase sélection" de la question principale la condition (SEXE = FEMME) ; si cette condition n'est pas vérifiée, le dossier sera non pertinent pour la question.

Le même raisonnement s'applique pour des "phases sélection" de questions secondaires comportant des conditions booléennes, sur des résultats obtenus lors de la phase traitement de la question principale.

Ces résultats sont en mémoire, donc d'accès rapide au moment de l'évaluation de la pertinence du dossier. Les conditions booléennes sur ces résultats seront donc placés en tête de la phase sélection.

Exemple : la condition booléenne relative au laps de temps contenu dans ID3 sera écrite en tête de la phase sélection (ID3 > 30000).

Toute phase de sélection commence par le symbole SELECT, se termine par le symbole ## et possède la structure générale suivante :

```

• < phase de sélection > : : = SELECT < E.B. de critères > < question
arborescente > ##
• < critère > : : = < identificateur > < opérateur de relation >
< constante >

```

- $\langle \text{identificateur} \rangle ::= \langle \text{identificateur de variable} \rangle / \text{SEXE} \mid \text{AGE} \langle \text{identificateur de variable date} \rangle \mid$
- opérateur de relation $::= \langle \mid \leq \mid > \mid \geq \mid = \mid \neq \mid$
- $\langle \text{constante} \rangle ::= \langle \text{nombre entier} \rangle \mid \langle \text{nombre réel} \rangle \mid \langle \text{date} \rangle \mid \langle \text{descripteur} \rangle$

L'expression booléenne de critères est optionnelle, les opérateurs booléens sont ET et OU ; en l'absence de parenthèses, elle sera évaluée de gauche à droite.

b) Ecriture de la question arborescente

La structure arborescente (telle que nous l'avons définie au paragraphe III) structurant une question et étiquetée par les descripteurs du lexique, auquel nous aurons adjoint les descripteurs INDIF1 et INDIF2, sera décrite sous une forme linéaire.

L'opérateur représenté par le symbole " - " traduit le lien vertical entre deux sommets de l'arborescence.

Les expressions booléennes, d'opérateurs ET et OU reliant respectivement les descripteurs de niveau 3 et de niveau 4 seront écrites entre crochets. Il n'y a aucune priorité entre les opérateurs ; en l'absence de parenthèses l'expression sera évaluée de gauche à droite. L'opérateur OU sera toujours pris dans son sens inclusif et les expressions booléennes seront toujours complètement évaluées.

Compte tenu du fait que le seul lien horizontal reliant les sommets de 1er niveau d'une part, et de 2è niveau d'autre part, est l'opérateur ET, nous avons convenu, pour simplifier et alléger l'écriture de l'arborescence, de ne pas le représenter mais d'écrire :

- chaque descripteur de niveau 1 en début de ligne éventuellement derrière le symbole ITER (voir paragraphe ultérieur),
- Chaque descripteur de niveau 2 en retrait par rapport au début de la ligne, l'opérateur " - " le précédant pouvant être omis.

En outre pour rendre plus lisible cette écriture, chaque descripteur de niveau trois et quatre, et chaque critère complémentaire seront écrits également en retrait par rapport au début de la ligne comme le montre l'exemple précédent.

Rappelons que l'utilisateur n'est pas tenu dans une question, d'indiquer le type de rencontre du médecin avec le malade (représenté par les descripteurs de niveau 1 et 2), il lui suffira d'utiliser les descripteurs INDIF1 et INDIF2.

Il peut aussi partitionner sa question en sous-question ou en éléments de question, comme il le désire. Il pourra par exemple, soit souhaiter que tous les événements médicaux contenus dans une question aient lieu lors d'une même rencontre du médecin avec le malade, soit au contraire, exprimer une condition moins restrictive c'est-à-dire que ces événements aient lieu au cours d'une ou (inclusif) plusieurs rencontres de même type (par exemple : des consultations) ; il utilisera alors plusieurs descripteurs de niveau 1 identiques (CONSU) et il placera sous la dépendance de ces derniers les événements médicaux qui y sont reliés. Par exemple, dans la question principale précédente, nous avons mis en évidence une partition en deux sous-questions correspondant à deux consultations. Un dossier sera pertinent pour cette question si les événements médicaux contenus dans les deux consultations figurent soit, dans une seule consultation, soit, respectivement dans deux consultations du dossier.

Par contre si un utilisateur souhaite préciser explicitement que

les événements aient lieu lors de deux ou plusieurs rencontres différentes mais de même type, il indiquera qu'elles devront avoir lieu à des dates différentes (nous verrons comment dans un paragraphe ultérieur),
Exemple : Reprenons la description arborescente de la question principale en traduisant l'indifférence de l'utilisateur dans le choix des descripteurs de 1er niveau (nous ne donnons qu'une description qualitative : les instructions de rangement et les critères complémentaires seront vus dans des paragraphes ultérieurs).

INDIFL - ANAMN - ETIOL - APARU
 ETCIV - FAMIL - MARIE
 EXAME - {DIAGN - {CETOS OU
 MATUR OU
 DEPEN} ET
 VAISS - ARTIT ET
 CORON ET
 NON ENDOC ET
 DOSAG - {COLES ET
 RICEM}}
 INDIFL - TERAP - NEURO - {NON SCLER ET
 {POLYN OU
 PSYCO}}

c) Critères complémentaires et instructions de rangement

Ces deux notions sont relatives aux éléments d'informations complémentaires contenus dans un dossier. Il convient de rappeler et de préciser tout d'abord, les fonctions de recherche de ces derniers.

- Fonctions de recherche des éléments d'informations complémentaires

Chaque fonction de recherche (vu au chapitre III) est repérée par un identificateur. Lors de la traduction du langage, la reconnaissance d'un identificateur génère l'appel de la fonction correspondante.

L'élément d'informations complémentaires recherché est celui qui est accolé au sommet, étiqueté par le descripteur précédent dans la description linéaire de la question, l'identificateur de la fonction.

Les fonctions fdp et fdp (recherche de la date ou de la borne inférieure respectivement supérieure de l'intervalle de dates qualifiant directement ou non le descripteur) sont repérées par les identificateurs DATP et DATG, les fonctions fhp et fhg ayant le même rôle que les précédentes mais recherchant l'heure sont repérées par les identificateurs HRP et HRC, les fonctions fvp et fvg (recherche de la valeur numérique ou de la borne inférieure respectivement supérieure de l'intervalle de valeurs numériques qualifiant le descripteur) sont repérées par VALP et VALG, enfin la fonction fa (recherche de l'adjectif ou des adjectifs accolés au descripteurs) est représentée par l'identificateur ADJ.

Rappelons qu'à chacune de ces fonctions, c'est-à-dire à chacun de ces identificateurs est associé un mot réservé, destiné à recevoir l'élément d'informations complémentaires recherché (un double mot est associé à la fonction fa car nous avons vu que nous recherchions d'un seul coup les deux adjectifs, s'ils existent, qualifiant un sommet).

Si l'information recherchée est absente, nous mémorisons une marque d'absence se traduisant concrètement par une valeur négative.

- Critères complémentaires

Nous avons vu qu'à chaque sommet de l'arborescence question"

pouvait être accolée une expression booléenne de critères complémentaires.

Dans la description linéaire chaque descripteur est alors suivi, dans l'ordre, d'un symbole de base AVEC et d'une expression booléenne de critères complémentaires. Les opérateurs utilisés sont là aussi, uniquement les opérateurs ET et OU (aucune priorité n'existant entre eux). En l'absence de parenthèses l'expression sera évaluée de gauche à droite.

Au niveau du langage, un critère complémentaire s'écrit :

- . < critère complémentaire > ::= < critère complémentaire lexicographique > | < critère complémentaire numérique >
- . < critère complémentaire lexicographique > ::= ADJ < opérateur de comparaison > < valeur 1 >
 - . < opérateur de comparaison > ::= # | =
 - . < valeur 1 > ::= < identificateur de variable code > < descripteur > < >
- . < critère complémentaire numérique > ::= < identificateur de fonction numérique > < opérateur de relation > < valeur 2 >
 - . < identification de fonction numérique > ::= DATP | DATG | HRP | HRG | VALP | VALG
 - . < opérateur de relation > ::= # | > | >= | < | ≤ | =
 - . < valeur 2 > ::= < identificateur de variable numérique > | < constante >
 - . < identificateur de variable numérique > ::= < identificateur de variable date > | < identificateur de variable heure > | < identificateur de variable entière > | < identificateur de variable réelle >

Exemple : Pour traduire le critère complémentaire numérique : taux de cholestérol supérieur à 3g / litre nous avons fait le choix de considérer, dans le cas où ce taux serait donné sous la forme d'un intervalle de valeurs, la borne inférieure : COLES AVEC (VALP > 3) ; nous aurions pu également considérer la borne supérieure et écrire : COLES AVEC (VALG > 3) ou bien si l'on veut faire apparaître un critère plus précis (par exemple 3g/litre < taux de cholestérol < 5g/litre nous pouvons considérer les deux bornes : COLES AVEC (VALP > 3) ET (VALG < 5).

Lors de son évaluation, un critère complémentaire prendra la valeur FAUX dans les trois cas suivants :

- l'information recherchée par la fonction de recherche, est absente ;
- la variable du second membre contient une valeur négative (c'est-à-dire qu'aucune information n'a été rangée dans cette variable) ;
- enfin, le critère n'est pas vérifié.

Nous avons vu au chapitre précédent que l'évaluation des critères complémentaires lexicographiques présentait un cas particulier. En effet, étant donné qu'il n'est pas possible pour un utilisateur, de préciser sur quel adjectif du double mot portera le critère, nous évaluerons ce dernier en examinant successivement, s'ils existent, les deux adjectifs.

En outre, pour faciliter une éventuelle mémorisation de ces derniers, nous avons convenu, après chaque expression booléenne de critères complémentaires de valeur vraie, de réaliser le traitement suivant : les adjectifs sont réarrangés dans le double mot selon leur ordre d'apparition dans les critères lexicographiques de valeurs VRAIE (où l'opérateur de comparaison est " = ").

Exemple : Reprenons l'expression booléenne de critères complémentaires relatifs aux éléments d'informations accolés au sommet étiqueté par ARTIT : ARTIT AVEC (ADJ = MEMBE) ET (ADJ = INFER) ET (DATP > ID1).

Si cette expression booléenne a la valeur VRAIE, les adjectifs contenus dans le double mot, associés à l'identificateur ADJ seront rangés dans l'ordre : code du descripteur MEMBR, code du descripteur INFEP.

e) Instructions de rangement

Ces instructions ont pour intérêt de ranger dans des variables ou dans des piles, les éléments d'informations complémentaires extraits d'un dossier, par les fonctions de recherche vues précédemment.

La liste des instructions sera placée entre deux symboles "/" et suivra soit l'expression booléenne de critères complémentaires, si elle existe, soit immédiatement le descripteur étiquetant le sommet à partir duquel on a recherché les éléments d'informations.

- liste d'instructions de rangement : : = < instructions de rangement > |
< instructions de rangement > . < liste d'instructions de rangement >
- < instructions de rangement > : : = < ident. f > → < ident. v >

Le symbole "→" n'appartenant pas à l'ensemble des caractères typographiques sur CII, sera remplacé lors de la perforation par le caractère "=".

< ident. f > : : = < ident. de fonct. de recherche > | DES

La reconnaissance de l'identificateur DES génère la recherche du code numérique du descripteur précédant, dans la description de la "phase sélection", la liste d'instructions de rangement. Ce code est rangé dans un mot associé à l'identificateur DES. Si le descripteur est précédé de l'opérateur NON nous rangeons l'opposé du code.

- < ident. v > : : = < ident. de variable > | < ident. de pile >
- < ident. de variable > : : = < ident. de variable numérique > |
< ident. de variable code >
- < ident. de pile > : : = < ident. de pile numérique > | < ident. de pile code >

La mémorisation des éléments d'informations complémentaires du type adjectif est réalisée de la façon suivante :

- La première instruction de rangement de la liste, relative à l'identificateur de fonction ADJ, entraîne la mémorisation du 1er adjectif contenu dans le double mot, la deuxième instruction mémorise le 2è adjectif.

- D'une façon générale il est conseillé à l'utilisateur d'ordonner les adjectifs du double mot en faisant précéder la liste d'instructions de rangement d'une expression booléenne des critères lexicographiques relatifs à ces derniers.

Exemple 1 : Reprenons l'exemple précédent en recherchant des sujets ayant eu de l'artérite aux membres inférieurs ou supérieurs et en souhaitant mémoriser la localisation (INFEP OU SUPER). Nous écrirons ARTIT AVEC (ADJ = INFEP) OU (ADJ = SUPER) ET (ADJ = MEMBR) / ADJ → IC1 /

Si la valeur de l'expression booléenne est VRAIE alors le double mot associé à ADJ contient dans l'ordre : code numérique de INFEP ou de SUPER, code numérique de MEMBR. La variable code IC1 contiendra le contenu du 1er mot du double mot.

Exemple 2 : Nous avons vu qu'un utilisateur peut exprimer le souhait que des événements médicaux aient lieu, par exemple, lors de consultations différentes, il utilisera alors les repères datés :

CONSU / DAT → ID1 / - EXAME....

CONSU AVEC (DAT ≠ ID1) - EXAME....

f) Instructions ITER ... FIT

Nous avons vu qu'il était possible d'"itérer" la recherche de l'ensemble des parties d'un dossier pertinent pour un ou plusieurs éléments de question. Ces dernières pouvant appartenir, comme nous allons le voir, à une ou plusieurs sous-questions, mais devant se suivre dans la description de la phase sélection ; ils seront encadrés par les symboles de bases ITER et FIT.

Les deux types d'utilisation permis sont :

- Le symbole ITER précède un descripteur de 1er niveau : nous "itérons" alors la recherche des parties pertinentes pour la sous-question correspondante, à laquelle il pourra être ajouté une ou plusieurs sous-questions suivantes. Le symbole FIT sera placé à la fin du dernier élément de question de la dernière sous-question itérée;

- Le symbole ITER précède un descripteur de 2^{ème} niveau : nous réitérons la recherche des parties pertinentes pour l'élément de question correspondant, auquel il pourra être ajouté un ou plusieurs éléments de question appartenant à la même sous-question et même une ou plusieurs sous-questions suivantes. Le symbole FIT sera placé à la fin du dernier élément de question itérée.

Plusieurs instructions ITER ... FIT pourront être utilisées dans une même phase de sélection ; nous allons rappeler cependant quelques restrictions relatives à leur emploi :

- Ces instructions ne pourront se chevaucher ; elles seront toujours considérées comme s'imbriquant de la façon suivante : ITER ... ITER ... FIT ... FIT
- Pour éviter une gestion trop lourde des informations mémorisées dans des piles lors de ces instructions nous avons limité à deux, le nombre d'itérations incluses l'une dans l'autre.

Exemple : Dans la question principale vue au début du paragraphe, supposons que l'utilisateur souhaite mémoriser les taux de cholestérol et d'uricémie dans deux piles réelles. Nous allons envisager le cas où l'on recherche ces valeurs à l'intérieur d'une même consultation et celui où l'on recherche ces valeurs dans tout le dossier. Nous ne répèterons pas le début de la phase de sélection qui reste inchangée.

1er cas : Pour avoir accès à l'élément de sélection de sommet de niveau 3 étiqueté par DOSAG, nous le transformons en élément de question :

```
EXAME - [ DIAGN - [ CETOS / DES → ICI / OU
           MATUR / DES → ICI / OU
           DEPEN / DES → ICI / ] ET
           VAISS - APETIT AVEC (ADJ = MEMBR) ET
                       (ADJ = INFER) ET
                       (DATP > ID1) / DATP → ID2 / ET
```

CORON ET
NON ENDOC

```
ITER EXAME - DOSAG - [ COLES AVEC (VALP > 3) / VALP → RP1 / ET
                    RICHEM AVEC (VALP > 005) / VALP → RP2 / ] FIT
```

2^{ème} cas : La recherche de taux de cholestérol et d'uricémie s'effectuant dans tout le dossier, l'élément de sélection, doit-être transformé en sous-question.

La première partie de la description précédente est conservée ; nous écrirons à la suite :

```
ITER CONSU - EXAME - DOSAG - [ COLES AVEC (VALP > 3) / VALP → RP1 / ET
                             RICHEM AVEC (VALP > 005) / VALP → RP2 / ]
```

D - PHASE DE TRAITEMENT

Nous avons mis en évidence deux types de traitement :

- Les traitements locaux réalisés après chaque examen de dossiers pertinents ; ces traitements sont relatifs soit à des variables locales, soit à des variables globales.
- Les traitements globaux réalisés une fois l'étude du fichier terminée ; ils n'intéressent que les variables globales.

Le langage utilisé pour décrire ces deux types de traitement est extrêmement simple et se compose d'instructions qui seront interprétées et exécutées une à une.

Ces instructions sont essentiellement des procédures permettant à l'utilisateur d'effectuer un certain nombre d'opérations élémentaires telles que :

- Des traitements arithmétiques et statistiques.
 - Des sorties sur imprimante ou sur disque en format standard.
- . $\langle \text{phase de traitement} \rangle ::= \langle \text{phase de traitement local} \rangle \mid \langle \text{phase de traitement global} \rangle \#$
- . $\langle \text{phase de traitement local} \rangle ::= \text{FAIRE} \langle \text{liste d'inst. procédures} \rangle$
 - . $\langle \text{phase de traitement global} \rangle ::= \text{TRAIT} \langle \text{liste d'inst. procédures} \rangle$
 - . $\langle \text{liste d'inst. procédures} \rangle ::= \langle \text{inst. procédures} \rangle ; \langle \text{liste d'inst. procédures} \rangle \mid \langle \text{inst. procédure} \rangle$
 - . $\langle \text{inst. procédures} \rangle ::= \langle \text{code instruction} \rangle (\langle \text{liste de paramètre} \rangle)$
 - . $\langle \text{code instruction} \rangle ::= \langle \text{code inst. traitement arithmétique et statistique} \rangle \mid \langle \text{code inst. édition} \rangle$

Nous allons détailler les différentes instructions existantes dans chacune des trois opérations que nous venons de citer.

a) Instructions de traitements arithmétiques et statistiques

- Opérations arithmétiques élémentaires

Elles sont repérées par les codes instructions ADD, DIF, DIV et MUL correspondant respectivement aux opérations : addition, soustraction, division et multiplication.

La liste des paramètres comprend dans l'ordre : deux paramètres opérands et un paramètre résultat.

- . $\langle \text{liste des paramètres} \rangle ::= \langle \text{paramètre opér.} \rangle , \langle \text{paramètre opér.} \rangle , \langle \text{paramètre résultat} \rangle$
- . $\langle \text{paramètre opér.} \rangle ::= \langle \text{ident. de variable} \rangle \mid \langle \text{ident. de pile} \rangle \mid \langle \text{constante} \rangle$
- . $\langle \text{paramètre résult.} \rangle ::= \langle \text{ident. de variable} \rangle \mid \langle \text{ident. de pile} \rangle$
- . $\langle \text{constante} \rangle ::= \langle \text{nombre réel} \rangle \mid \langle \text{nombre entier} \rangle$

Remarque 1 : Par souci d'efficacité, nous avons éliminé les opérations concernant uniquement des constantes, c'est-à-dire que seulement un des deux paramètres opérands peut-être une constante.

Remarque 2 : Lorsqu'un des paramètres opérands est un identificateur de pile, l'autre un identificateur de variable, le traitement est réalisé pour chaque élément de la pile.

Remarque 3 : Lorsque les deux paramètres opérands sont des identificateurs de pile, ces dernières doivent nécessairement être des piles "parallèles".

Exemple 1 :

- L'opérateur $ID3 = ID2 - ID1$ s'écrit : DIF (ID2, ID1, ID3),
- L'incrément de la variable ID4 : ADD (ID4, 1, ID4).

Exemple 2 :

- La multiplication des éléments de la pile RP1 par la variable R1 s'écrit : MUL (RP1, R1, RP2) où RP2 est l'identificateur de pile résultat.

- Opérations arithmétiques concernant les piles

Nous avons retenu les quatre opérations : le cumul, la moyenne, la recherche du maximum et du minimum des éléments d'une pile, opérations dont les codes respectifs sont : CUM, MOY, MAX et MIN.

. < liste des paramètres > : : = < ident. de pile > ₁ < ident. de variable >

Exemple : RP1 est un identificateur de pile réelle contenant la liste des poids d'un individu, la recherche du maximum des éléments s'écrit MAX (RP1,R2) Le résultat sera dans R2.

Remarque : Il sera facile d'ajouter d'autres opérations telles que variance, écart-type... Dans toutes ces opérations arithmétiques le type des paramètres devra être le même.

- Construction de mise à jour de tableau de statistiques

Le code instruction est : CONS.

. < liste des paramètres > : : = < ident. de tab. stat. > | < ident. de tab. stat. > ₁ < ident. de variable > | < ident. de tab. stat. > ₂ < ident. de pile >

Illustrons l'utilisation de cette instruction à l'aide d'un exemple : Soit la question suivante : "Répartition de l'âge et du poids d'un individu à l'apparition du diabète en fonction du type de ce dernier".

Nous déclarons :

```
DIM EX / ICI / DESC ACCID, DEPIS, MALAD, OPERA, ACETO, MAMAN /,
DIM EY / AGE (ID1) / (20, 90, 10) / 'AGE >= ' /,
DIM EZ / RPI / ZONES 80, 85, 90, 100, 110, 120 / 'POIDS >= ' / ;
TAB T1 (EX, EY) , T2 (EX, EZ).
```

Supposons que nous ayons, lors de la phase sélection, dans la variable code ICI le descripteur MALAD, dans la variable date ID1 une date correspondant à un âge de 55 ans et enfin dans la variable RPI la valeur 93,5.

L'instruction CONS (T1), génère les opérations suivantes :

- Sélection du caractère pertinent de la dimension EX (caractère n° 3),
- Sélection du caractère pertinent de la dimension EY (caractère n° 4),
- Incrément de un de l'élément de tableau T(3, 4).

Si l'utilisateur souhaite incrémenter l'élément sélectionné d'un tableau non pas de 1, mais du contenu d'une variable (ayant reçu une valeur lors de la phase sélection), il fera suivre l'identificateur du tableau de statistique d'un identificateur de variable ou de pile. Dans ce dernier cas, plusieurs recherches d'éléments sont générés au cours d'une même instruction ; le 1er élément de la pile est incrémenté au 1er élément du tableau sélectionné et ainsi de suite.

Exemple : L'instruction CONS (T2, RP1) génère la sélection de l'élément de tableau T2(3, 3) et l'incrémentation : $T2(3, 3) = T2(3, 3) + RP1$.

- Calcul des pourcentages lignes, colonnes et du pourcentage total d'un tableau de statistiques

Les codes instructions sont respectivement PLIC, PCOL et PTOT. Ces instructions ont pour rôle d'adjoindre au tableau de statistiques respectivement, une rangée ligne ou une rangée colonne ou l'une et l'autre ; chaque élément contient le rapport de la somme des contenus des éléments d'une rangée à la somme des contenus des éléments de tout le tableau.

. < liste de p ramètre > : : = < ident. de tab. stat. >

b) Instructions de sortie de résultats et possibilité de création d'un sous-fichier

Les sorties des résultats sont réalisées soit sur imprimante, soit sur disque. Les deux instructions de sortie sur imprimante ont pour code IMPR et EDIT, l'impression des résultats s'effectuant en format standard ; l'instruction de sortie sur disque a pour code DISQ, et présente un double intérêt : si le code est suivi d'une liste de paramètres (identificateur de variable ou de pile) de recopier sur disque le contenu de ces variables, si le code n'est suivi d'aucun paramètre, de recopier le dossier examiné dans un fichier prévu à cet effet.

- Instruction codée IMPR

. < liste des paramètres > : : = < ident. > | < ident. > . < liste de paramètres >
 . < ident. > : : = < ident. de variable > | < ident. de pile >

Chaque impression de variable ou de pile est réalisée sur une ligne différente, le format standard est le format FORTRAN F8.3.

L'impression du contenu d'une variable a le modèle suivant :
 ident. de variable = valeur (si l'identificateur est du type date, le contenu de la variable sera imprimé sous la forme : jour / mois / année).

L'impression des éléments d'une pile s'effectue de la façon suivante :

ident. de pile = liste des valeurs des composants de la pile (chaque valeur étant séparée de la précédente par au moins un espace ; si la liste ne tient pas sur une ligne, elle se prolonge sur les lignes suivantes).

Exemple : Soient les variables date ID1 contenant la valeur 19720806 et la pile réelle RP1 contenant les valeurs réelles suivantes : 14, 2, 13, 1 ; 10, 1 l'ordre IMPR (ID1, RP1) fait apparaître l'impression suivante :

ID1 = 06 / 08 / 1972
 RP1 = . . . 14,2 13,1 10,1

- Instruction codée EDIT

. < liste des paramètres > : : = < ident. de table de stat. >

Cette instruction permet, à chaque appel, l'édition d'un tableau de statistiques. Celle-ci est déjà bien préparée car il faut remarquer que tous les libellés susceptibles d'apparaître en tête des rangées lignes et colonnes ont été spécifiés lors de la déclaration des dimensions.

Des totaux marginaux horizontaux et verticaux seront imprimés sur les dernières rangées ligne et colonne.

L'édition de chaque tableau est réalisée selon un format standard. Si le nombre de colonnes est supérieur à 10, le tableau sera imprimé "par tranche" avec rappel de libellé en tête des rangées ligne.

Exemple : Reprenons l'exemple vu précédemment et donnons un exemple d'édition du tableau de statistiques T1.

L'ordre EDIT (T1) génère l'édition suivante :

	AGE ≥ 20	AGE ≥ 30		AGE ≥ 80	TOT. HORIZ
ACCID					
DEPIS					
MALAD					
MAMAN					
TOT. VERT.					

- Instruction codée DISQ

Nous avons vu que cette instruction offrait à l'utilisateur deux intérêts :

1) - Celui de conserver des résultats sur disque (les identificateurs de variable ou de piles ou de tableau de statistiques, contenant ces résultats sont les paramètres de l'instruction). L'utilisateur pourra ensuite les reprendre et réaliser des traitements arithmétiques, statistiques plus complexes ou des éditions plus élaborées que celles permises par le langage. Une seule instruction de ce type sera utilisée lors de la phase "traitements globaux" d'une question.

• < liste des paramètres > : : = < ident. > | < ident. > _ < liste de paramètres >
 • < ident. > : : = < ident. de variable > | < ident. de pile > | < ident. de tab. stat. >

L'ensemble des résultats globaux demande pour une question ou une sous-question sera rangé dans un enregistrement logique.

2) - Celui de créer un sous-fichier composé des dossiers pertinents pour une question. Nous avons limité l'usage de cette instruction aux phases "traitements globaux" des questions principales.

La pose d'un train de questions pourrait, par exemple, se dérouler de la façon suivante :

- L'utilisateur pose au fichier de dossiers la question principale ; lors de la phase "traitements globaux" l'instruction DISQ crée un sous-fichier de dossiers pertinents pour cette question.
- L'utilisateur pose au sous-fichier les questions secondaires relatives à la question principale.

Remarques : Il apparaît nettement que ce langage est d'utilisation simple et facile pour un non-informaticien, mais qu'il est par contre insuffisant pour un initié. Il ne serait pas raisonnable de définir un langage interprétatif possédant des outils plus puissants et une syntaxe plus riche : la phase d'interprétation serait beaucoup trop lourde et trop coûteuse pour le but recherché.

Nous nous sommes donc orientés vers un système s'adressant à la fois aux deux types d'utilisateurs cités ci-dessus. Pour obtenir une généralisation d'emploi, nous avons modularisé les programmes de traduction et d'exploitation du langage.

Nous donnons au chapitre suivant une analyse générale de ces modules, mais nous pouvons d'ores et déjà mettre en évidence leurs fonctions principales et leur enchaînement.

Un module de traduction traduit la phase déclaration et la phase sélection et traitement de chaque question principale et des questions secondaires associées. Ce module vérifie la syntaxe du langage, traduit la structure arborescente sous forme de tableaux et génère les appels des fonctions de recherche.

Pour chaque dossier du fichier un module de sélection détermine la pertinence de ce dernier à chaque question du train. Enfin divers modules de traitement correspondent aux fonctions que nous avons décrites au cours de la phase traitement.

Lorsqu'un utilisateur informaticien souhaitera réaliser sur des résultats des traitements plus élaborés que ceux qui existent il pourra soit, si cela est suffisant, mémoriser lors des traitements globaux, ces résultats sur disque et les traiter ultérieurement dans un programme écrit en FORTRAN par exemple ; soit écrire la phase de traitement (local et global) en FORTRAN. Il devra prendre en charge l'enchaînement des phases d'une question. Il aura alors à sa disposition pour faciliter la programmation tous les modules cités précédemment.

E - EXEMPLE D'UTILISATION DU LANGAGE

Nous donnons deux exemples d'utilisation du langage :

Exemple 1 : il donne la description complète de la question principale et de la question vue au paragraphe C :

QUEST- 1

DECLARER

BIM LAPS / ID3 / (10000, 100000, 10000) / 'LAPS // ' / ,

DIM DIAB / IC1 / DESC CETOS, MATUR, DEPEN / ,

DIM AGE / AGE (ID1) / (30, 60, 10) / 'AGE // ' / ;

TAB T1 (LAPS, DIAB) , T2 (AGE) ;

SELECT (SEXE = FEMME)

CONSU - ANAMN - ETIOL - APARU / DATP → ID1 /

ETCIV - FAMIL - MARIE

EXAME - [DIAGN - [CETOS / DES → IC1 / OU

MATUR / DES → IC1 / OU

DEPEN / DES → IC1 / ET

VAISS - ARTIT AVEC (ADJ = MEMBR) ET

(ADJ = INFER) ET

(DATP > ID1) / DATP → ID2 / ET

CORON ET

NON ENDOC ET

DOSAC - [COLES AVEC (VALP > 3) ET

PICEM AVEC (VALP > 0.05)]]

CONSU - TERAP - NEURO - [NON SCLEP ET

(POLYN OU

PSYCO)] #

FAIRE DIF (ID1, ID2, ID3) , CONS (T1) ;

TRAIT EDIT (T1) ; #

SELECT (ID3 > 30000)

CONSU - EXAME - VAISS - AMPUT AVEC (ADJ = MEMBR) ET

(ADJ = INFER) #

FAIRE CONS (T2) ;

TRAIT EDIT (T2), ECR (T2) ; #

Exemple 2 : "présence ou absence du diabète en fonction de l'hérédité et en fonction du nombre moyen de rations glucidiques absorbées par jour".

QUEST - 2

DECLARER

DIM GLUC / R1 / (100, 300, 20) / "RAT. GLUC. = " / ,

DIM DIAB / IC1 / BOOL APARU, NON APARU / ,

DIM HERED / IPC1 / DESC PATER, MATER, FREERE, SOEUR, ONNAT, TAMAT / ;

TAB T1 (DIAB, GLUC) , T2 (DIAB, HERED);

SELECT

INDIFI - ANAMN - ETIOL - [APARU / DES → IC1, DAT → ID1 / OU

NON APARU / DES → IC1 /] ET

HERED - [PATER / DES → IPC1 / OU
 MATER / DES → IPC1 / OU
 FRERE / DES → IPC1 / OU
 SOEUR / DES → IPC1 / OU
 ONMAT / DES → IPC1 / OU
 TAPAT / DES → IPC1]

INDIF1 - ITER - INDIF2 - ALIME AVEC (DAT ≥ ID1) / VAL → RP1 / FIT ~~#~~

FAIRE MOY (RP1, R1) , CONST (T1), CONST (T2) ;

TRAIT EDIT (T1), EDIT (T2) ; ~~#~~

CHAPITRE V

ANALYSE DES MODULES DE TRADUCTION ET
D'EXPLOITATION DU SYSTEME D'INTERROGATION

A - ANALYSE GENERALE

Nous avons vu que les questions étaient groupées par train, un train de questions comprenant une question principale et plusieurs questions secondaires. L'interrogation des dossiers du fichier est réalisée en trois étapes .

- 1) Traduction de chaque question du train : module TRADUCT (NERR) où NERR est une variable positionnée à 1, s'il y a au moins une erreur de vocabulaire ou de syntaxe, à 0 sinon.
- 2) Sélection des dossiers pertinents pour la question et mémorisation, dans des variables et piles, des informations recherchées : module SELECT (NPERT) où NPert = 1 si le dossier est pertinent, 0 sinon.
- 3) Traitement des informations recueillies ; les traitements locaux sont pris en compte par le module ACTFAIR, les traitements globaux par le module ACTRAIT.

Ces modules seront écrits en grande partie en FORTRAN. (Seuls les modules de reconnaissance de symboles et de caractères seront écrits en langage symbolique).

Il importe donc que toutes les variables, piles (en réalité, ce sont des tableaux à une dimension) et tous les tableaux à deux dimensions soient déclarés au début du programme principal au cours duquel les modules précédents sont activés. Une phase d'initialisation d'un certain nombre de variables et de tableaux suit ces déclarations. Ces initialisations et déclarations sont indépendantes des trains de questions et sont obligatoirement écrites en tête de tout programme d'interrogation.

I - Analyse générale du programme d'interrogation

- Déclarations et initialisations de variables et tableaux utilisés dans tout programme d'interrogation
- Pour chaque question du train faire

TRADUCT (NERR) (Précisons qu'il existe une variable NERR par question) <u>si</u> NERR = 1 <u>et</u> question = question principale <u>aller a fin</u> programme
--

- Pour chaque dossier du fichier faire

Pour chaque question du train telle que NERR ≠ 1 faire

 . SELECT (NPER1)

 . si NPERT = 0 faire

si question = question principale aller a éti 1

sinon aller a éti 2

sinon ACTFAIR

 éti 2 : fin pour chaque

 éti 1 : fin pour chaque

- Pour chaque question du train telle que NERR ≠ 1 faire

ACTRAIT

fin programme

Remarque : Lorsqu'un utilisateur ne souhaitera pas utiliser le langage de traitement relatif aux traitements locaux et globaux, il écrira alors, lui-même, en Fortran, son programme d'interrogation. S'il désire utiliser les modules TRADUCT, SELECT et tous ceux relatifs aux instructions procédures de la phase traitement, il devra écrire en tête de son programme les déclarations et initialisations dont nous avons parlé précédemment. Il assurera nécessairement l'enchaînement de ces modules.

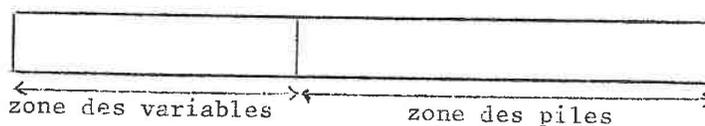
2 - Implantation des variables, piles et tableaux de statistiques utilisés dans un train de questions

Toutes les variables simples, piles et tous les tableaux de statistiques, susceptibles d'être utilisés dans un train de questions, sont déclarées dans le programme d'interrogation.

Variables simples et piles

Les variables et les piles de type entier, code, date et heure sont rangées dans un tableau entier NV, celles du type réel dans un tableau réel RV.

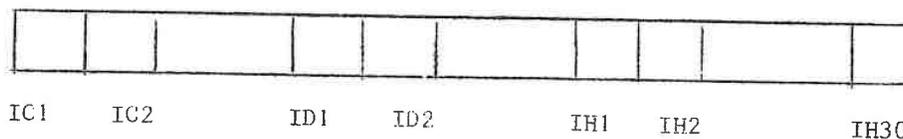
Ces tableaux sont remplis de la façon suivante :



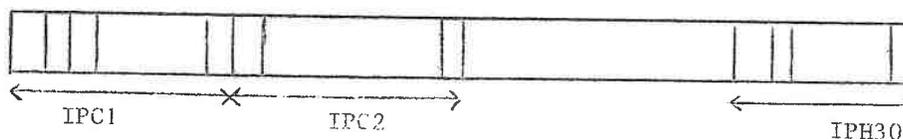
A l'intérieur d'un même tableau, les valeurs des variables et des piles sont rangées dans l'ordre alphabétique croissant de leurs identificateurs, et, pour un même type, dans l'ordre numérique croissant des numéros. (Exemple ID1, ID2).

Exemple :

Zone des variables de NV



Zone des piles de NV



Rappelons que le premier élément de chaque pile contient le rang dans le tableau de la dernière information mémorisée. Ceci permet, lors de la phase traitement, un emploi de piles beaucoup plus souple.

Il est adjoint à la déclaration des piles, celle d'un tableau NPIL indiquant, pour chaque pile, sur une ligne, un certain nombre de renseignements que nous allons préciser.

Ces renseignements sont rangés dans l'ordre alphabétique et numérique croissant des identificateurs de pile.

- La première colonne indique si les piles sont globales (indicateur positionné à 1) ou locales (indicateur positionné à 0)

- Les trois colonnes suivantes contiennent les valeurs des pointeurs p_i , p_s et p_m (cf. chapitre III) indiquant dans NV ou RV, selon le type, respectivement : le niveau des informations avant tout nouvel examen ou toute nouvelle itération d'examen de dossier, le niveau des informations "sûres" après tout examen ou toute itération d'examen, le rang du premier élément de la pile.
- La quatrième colonne indique, pour chaque pile, la taille maximale qu'elle peut avoir, sans "déborder" sur des piles de types différents. Cette valeur est de plus en plus petite pour des piles dont les numéros d'identificateurs sont de plus en plus grands. Cette colonne est initialisée au début du programme d'interrogation.
- La cinquième colonne indique, pour les piles dont la taille n'est pas standard, la valeur de cette dernière. Rappelons que nous avons fixé la taille des piles à 100 ou à 1000 éléments selon le numéro de l'identificateur.
- Enfin, la dernière colonne contient un indicateur prenant la valeur -1 si la pile est annexée à des piles "débordantes", 0 sinon.

indicateur de pile globale	p_i	p_s	p_m	taille max.	taille non standard	indicateur de pile annexée
----------------------------------	-------	-------	-------	-------------	------------------------	----------------------------------

Tableaux de statistiques

Nous avons vu qu'un utilisateur pouvait utiliser, dans un même train de questions, au maximum cinq tableaux de statistiques à une ou deux dimensions. Nous déclarons, au début du programme d'interrogation, cinq tableaux T1, T2, T3, T4, T5 à 22 lignes et 22 colonnes ; en effet, le nombre maximum de caractères utilisés pour chaque dimension est de 20, et il faut prévoir deux rangées supplémentaires pour le calcul éventuel des pourcentages ligne et colonne.

3 - Présentation générale de l'analyse des modules

Nous nous proposons, dans les paragraphes qui vont suivre, de donner une analyse relativement modulaire des macro-modules TRADUCT, SELECT, ACTFAIR et ACTRAIT.

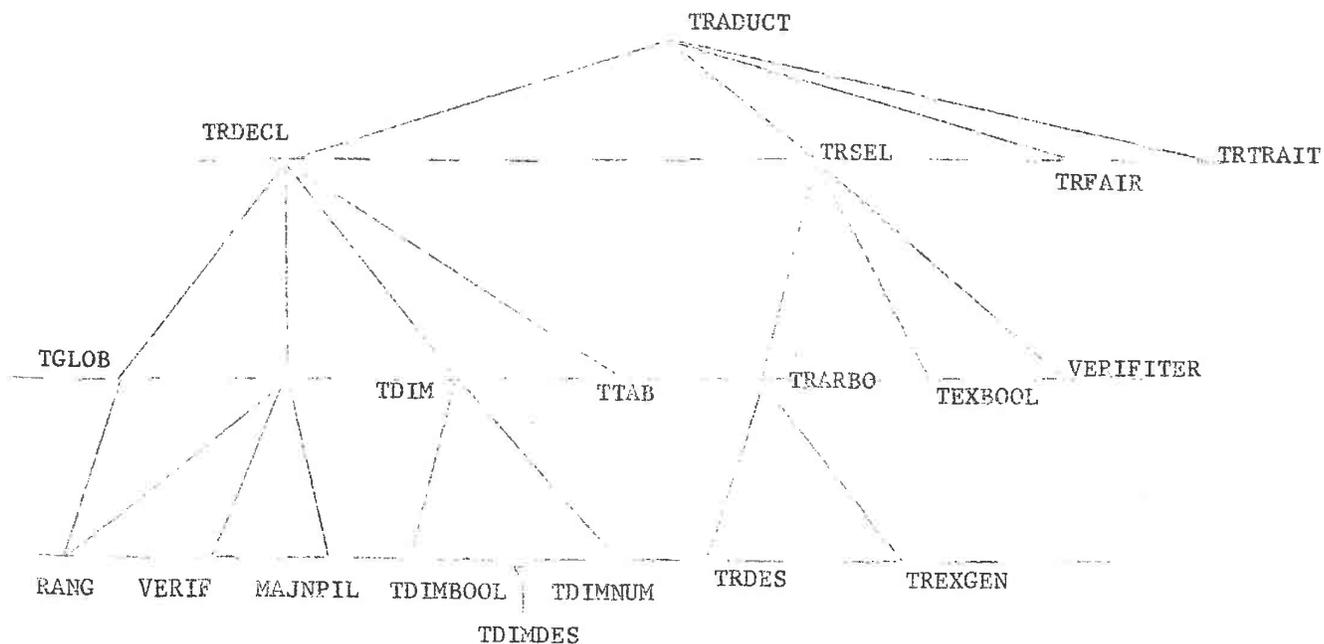
L'analyse d'un module sera toujours présentée en deux parties :

- La première commençant par UTILISE suivi du nom et d'une description sommaire de sous programmes utilisés, et du nom et de la structure des tableaux apparaissant, soit en paramètres explicites, soit en COMMON.
- La deuxième commençant par TRAITEMENT suivi de l'analyse proprement dite du module.

Nous donnons, avant chaque analyse de ces macro-modules, un graphe des modules (pour la relation "utilise").

B - ANALYSE DU MODULE DE TRADUCTION : TRADUCT

Graphe des modules de la relation "utilise"



ETAPE 1 MODULE TRANDUCT (NERR)

UTILISE

- Les modules TRDECL (NERR), TRSEL (NERR) dont les buts sont respectivement de traduire la phase "déclaration" et la phase "sélection" d'un train de questions
- Eventuellement, les modules TRFAIR (NERR), TRTRAIT (NERR) traduisent la phase des traitements, respectivement locaux et globaux, lorsque l'utilisateur utilise le langage de traitement.

TRAITEMENT

Il consiste, lors de la reconnaissance des symboles de base, DECLARE, SELECT, FAIRE et TRAIT, à appeler le module correspondant.

Remarque :

Dans tous les modules de traduction, nous allons être amené à reconnaître et à traduire les identificateurs de variables simples et de piles. Pour ce faire, nous ferons appel à un sous-programme :

RANG (IDENT, NTYP, MRG, NERR) où

- IDENT transmet, au sous programme, l'identificateur de variable ou de pile
- NTYP, MRG et NERR sont trois paramètres résultats
 - . NTYP indique le type de l'identificateur (il prend les valeurs 1, 2, 3, 4, 5 si le type est réel, entier, code ou heure)
 - . MRG indique à la fois le genre : identificateur de variable (MRG > 0) ou identificateur de pile (MRG < 0) et, dans le premier cas, le rang dans NV ou RV du contenu de la variable, dans le deuxième cas, l'opposé du rang dans NPIL de la ligne associée à l'identificateur de pile
 - . NERR a comme résultat 1 si l'identificateur est incorrect, 0 sinon

Chaque identificateur de variables ou de piles est traduit sur deux mots contenant, dans l'ordre, les valeurs de NTYP et MRG.

Nous donnons, en annexe 2, l'analyse de ce sous programme.

ETAPE 2 MODULE TRBECL (NERR)

UTILISE quatre modules TCLOB (NERR), TPILE (NERR), TDIM (NERR) et
 TTAB (NERR) dont les noms indiquent qu'ils ont pour rôle de tra-
 duire les déclarations, respectivement, des variables et piles
 globales, des dimensions de pile, des déclarations des dimensions
 de tableau de statistique et enfin des déclarations proprement
 dites de tableaux de statistiques.

TRAITEMENT Ces modules sont respectivement activés lors de la reconnais-
 sance des symboles de base GLOB, PILE, DIM et TAB.

ETAPE 3 MODULE TCLOB (NERR)

UTILISE - Le tableau NPIL et un tableau NGLOB à une dimension dans lequel
 seront rangés les rangs dans NV ou RV (selon leur type) des va-
 riables globales
 - Le sous programme RANG

TRAITEMENT Pour chaque variable de type global faire

```

. RANG (NTYP, NRG, NERR)
. si NERR ≠ 1 alors
    si NTYP = 1 alors NGLOB (i) = - NRG
    sinon NGLOB (i) = NRG
    i = i + 1
. sinon faire erreur n° 1
  
```

Pour chaque pile globale faire

```

. RANG (NTYP, NRG, NERR)
. si NERR ≠ 1 alors        NPIL (- NRG, 1) = 1
. sinon erreur n° 1
  
```

Nous donnons, en annexe 8, la signification des erreurs.

ETAPE 3 MODULE TPILE (NERR)

Il consiste :

- 1) à vérifier, pour chaque pile dont on a déclaré la taille, que cette dernière est inférieure à la taille maximale permise et qu'elle n'est pas annexée par une autre pile de déclaration antérieure, (réalisée par VERIF).

Exemple : la déclaration suivante : FILE RP1(300), RP3(200) est incorrecte.

La dimension standard des piles RP1, RP2... RP5 étant de 100, il apparaît que la pile RP1 "annexe" les piles RP2 et RP3.

La déclaration de RP3 est donc incorrecte.

- 2) à mettre à jour le tableau NPIL, c'est-à-dire à y faire figurer la taille de la pile et à positionner les indicateurs de piles annexées (réalisé par MAJNPIL)

TRAITEMENT Pour chaque pile dimensionnée faire

. reconnaître l'identificateur : EANG(MTYP, NRG, NERR)

. si NERR ≠ 1 alors

VERIF

NPIL(-NRG, 7) ≠ -1 pile non annexée	N	O	O
taille de la pile : taille maxima (NPIL(-NRG, 5))	-	N	O
NERR = 1 ; erreur n° 2	X		
NERR = 1 ; erreur n° 3		X	
MAJNPIL			X

MAJNPIL (NERR)

. ranger la taille dans NPIL (-NRG, 6)

. pour $i = \text{NRG}, -\text{NRG}+1, \dots$ faire

si NPIL (i, 7) = -1 alors NERR = 1 ; erreur n° 4

sinon NPIL (i, 7) = -1 (marque de pile annexée)

ETAPE 3 MODULE TDIM (NERR)

UTILISE

1) un tableau entier NDIM destiné à mémoriser une partie des renseignements contenus dans la déclaration de dimension, deux tableaux NPAR et PAR à une dimension, contenant la traduction de la définition des caractères, et enfin un tableau entier NLIB à une dimension, contenant les libellés.

Description du tableau NDIM

En face de chaque nom de dimension figurent, sur une même ligne :

- la taille de la dimension (calculée au moment de la traduction de la définition de caractères) ;
- le type de partition : soit numérique régulière ou irrégulière, soit lexicographique, soit booléenne (codée respectivement 1, 2, 3 et 4) ;
- soit la traduction sur deux mots de l'identificateur de variable ou de pile associé à la caractéristique, soit la traduction de la fonction AGE (paramètre date), soit la traduction de la fonction SEXE. (Nous verrons dans ce qui suit comment sont traduites ces deux fonctions) ;
- un pointeur pc indiquant l'adresse du début de la définition des caractères, soit dans NPAR si la partition est lexicographique ou numérique (les bornes étant alors des dates ou des entiers), soit dans PAR si la partition est booléenne ou numérique (bornes réelles) ;
- un pointeur plib indiquant dans NLIB l'adresse du début de la liste des libellés susceptibles d'apparaître en tête des rangées ligne ou colonne associés à cette dimension (si l'utilisateur n'a pas précisé de libellé, ce pointeur a la valeur 0).

Les fonctions AGE (paramètre date) et SEXE sont traduites sur deux mots (en réalité, un seul suffirait pour SEXE).

Le premier mot contient le code de la fonction, le deuxième mot, pour la fonction AGE contient, soit le rang NRG dans NV de la valeur de la variable date, soit le rang -NRG dans NPIL de la ligne associée à l'identificateur de pile date.

nom de dim	taille de dim	type de partition	trad. ident. de var. ou de pile ou de AGE ou de SEXE		pc	plib
←-----→	←-----→	←-----→	type ; réel ou autre AGE, SEXE	adresse dans NV, RV ou NPIL	←-----→	←-----→

Description des tableaux NPAR et PAR

Si la partition est numérique, chaque liste de bornes est séparée de la suivante par un mot séparateur (contenant la valeur -1)

Si la partition est lexicographique, chaque descripteur est représenté par son code numérique, (le descripteur RESTE a le code 0). Le même séparateur est présent entre chaque liste.

Enfin, si la partition est booléenne, chaque expression booléenne est traduite en notation post fixée et est séparée de la suivante par le même séparateur. Chaque opérande de cette expression, c'est-à-dire chaque relation, est traduite sur cinq mots.

Rappelons que le premier membre de cette relation est, soit un identificateur de variable ou de pile, soit la fonction AGE ou SEXE. Il est traduit, comme nous venons de le voir, sur deux mots.

L'opérateur de relation : <|<|>|>|=| ≠ est codé dans le 3ème mot. Enfin, le deuxième membre pouvant être, soit un identificateur de variable ou de pile, soit une constante entière, réelle, date ou descripteur est traduit sur deux mots. (Pour une constante, le premier mot contient la valeur 0 et, si c'est un descripteur, il est traduit par son code).

Description du tableau NLIB

Chaque libellé est logé sur trois mots, des caractères \perp le complètent éventuellement.

2) les modules : TDIMBOOL (NERR, NODIM) traduisant la liste des caractères booléens, TDIMDES (NERR, NODIM) traduisant la liste des caractères lexicographiques et TDIMNUM (NERR, NODIM, NTYP) traduisant la liste des caractères numériques.

NODIM indique, au module, le rang du nom de dimension dans NDIM
 NTYP indique, au module, le type de l'identificateur représentant la caractéristique. Nous donnons l'analyse de ces trois modules en annexe 11.

TRAITEMENT

Pour chaque déclaration de dimension alors

- ranger le nom de dimension dans la 1ère colonne de NDIM
 (soit NOD son rang)

- ranger le type de partition dans NDIM

si le type de partition est booléen alors TDIMBOOL(NERR,
 NOD)

sinon si la caractéristique est représentée par un identificateur de variable ou de pile alors

[. RANG (NTYP, NRG, NERR) ;
 . ranger NTYP et NRG dans NDIM ;
aller a éti

sinon si la caractéristique est représentée par la fonction

[AGE ou SEXE alors
 . traduire la fonction ; positionner NTYP et NRG

éti : si le type de partition est lexicographique alors

TDIMDES (NERR, NOD)

sinon si le type de partition est numérique alors

TDIMNUM (NERR, NOD, NTYP)

sinon NERR = 1 ; erreur n° 5

ETAPE 3

MODULE TTAB (NERR)

UTILISE

les tableaux NDIM et NTAB.

Description de NTAB

Ce tableau indique, sur chacune de ses lignes et dans l'ordre des numéros d'identificateurs de tableau de statistique, le ou les rangs dans NDIM des noms de dimensions qui lui sont associées.

rang dans IDIM du 1er nom de dimension	rang dans IDIM du 2ème nom de dimension
--	---

Si le tableau n'a qu'une dimension, le deuxième mot contient la valeur 0.

TRAITEMENT

Pour chaque tableau de statistique déclaré faire

- repérer le numéro POST de l'identificateur de tableau
- rechercher, dans la 1ère colonne de NDIM, le ou les deux noms de dimension
- s'ils existent alors
 - . repérer le ou les rangs de ces noms dans la colonne
 - . les ranger dans NTAB au rang POST
- sinon NERR = 1 ; erreur n° 9

ETAPE 2

MODULE TRSEL (NERR)

UTILISE

- Les modules TRARBO (NERR) et VERIFITER (NERR) dont les buts sont respectivement de traduire l'arborescence question et de vérifier la syntaxe de l'instruction ITER...FIT
- Le sous programme TEXBOOL (EXCRI, IPCRI, NERR) traduisant l'expression booléenne de relation écrite derrière le symbole SELECT. L'expression booléenne est rangée dans le tableau à une dimension EXCRI. Nous verrons aussi que dans ce tableau seront rangées les expressions booléennes de critères complémentaires.

TRAITEMENT

- . IPCRI = 1
- . si l'expression booléenne de relation écrite derrière SELECT existe faire
 - . mettre dans EXCRI une marque d'existence
 - . incrémenter IPCRI
 - . TEXBOOL (EXCPI, IPCRI, NERR)
- . TRARBO (NERR)
- . VERIFITER (NERR)

ETAPE 3MODULE TRARBO (NERR)UTILISE

- 1) quatre tableaux NL1, NL2, NL3 et NL4 dans lesquels est représentée l'arborescence (ET/OU) structurant la question.
- deux tableaux EXCRI et INST dans lesquels sont rangées les expressions booléennes de critères complémentaires et les instructions de rangement ;
- enfin, un tableau NIITER où sont indiqués le début et la fin des parties de questions à réitérer.

Description des tableaux NL1 et NL2

Ces deux tableaux décrivent les niveaux 1 et 2 de l'arborescence. La 1ère colonne de ces deux tableaux contient le code numérique du descripteur de niveau correspondant. Les 2ème et 3ème colonnes contiennent deux pointeurs pexi et prgi ($i = 1, 2$) indiquant, respectivement, le rang dans EXCRI du début de l'expression booléenne de critères complémentaires et le rang dans INST du début de la liste d'instructions de rangement accolées au descripteur correspondant.

Le tableau NL1 n'a que ces trois colonnes.

Le tableau NL2 a une 4ème colonne supplémentaire dans laquelle figure le pointeur "père 2" indiquant le rang dans NL1 du code du descripteur "père".

Description des tableaux NL3 et NL4

Ces deux tableaux n'ont qu'une dimension et contiennent, respectivement, la traduction en notation post fixée de l'expression booléenne

de descripteurs de niveau 3 (d3) et de descripteurs de niveau 4 (d4). Chaque opérande de l'expression booléenne de d3 est traduit sur six mots, et de l'expression booléenne de d4 sur trois mots.

Le premier mot d'un opérande contient le code numérique du descripteur, le 2ème et 3ème mots, les pointeurs pxi et prgi (i = 3, 4) vus précédemment.

Les trois mots complétant la traduction de l'opérande de l'expression booléenne de d3 comprennent : un pointeur père 3 indiquant, dans NL2, le rang du code numérique du descripteur "père" et deux pointeurs : pfils et dfils ayant pour valeur les rangs dans NL4 du début et de la fin de l'expression booléenne de d4 ayant pour père le d3.

Description des tableaux EXCRI et INST

Chaque critère complémentaire est traduit sur quatre mots.

L'identificateur de fonction de recherche VALP, VALG, DATP, DATG, HRP, HRG et ADJ est codé sur un mot.

Nous avons déjà vu comment nous codions l'opérateur de relation et le second membre de critère.

Chaque expression booléenne est suivie d'un mot séparateur.

Chaque instruction de rangement est codée sur trois mots : l'identificateur de fonction de recherche sur un mot et le deuxième membre (identificateur de variable ou de pile) sur deux mots.

Chaque liste d'instructions de rangement est suivie aussi d'un séparateur.

Description du tableau NTITER

Ce tableau a deux colonnes et autant de lignes qu'il y a d'instructions ITER...FIT dans une question.

Chaque ligne contient dans l'ordre les rangs dans NL3 du premier descripteur de niveau 3, suivant le symbole ITER et du dernier précédant le symbole FIT.

2) Deux modules : TRDES (NIV, NERR) et TREXGEN (NIV, NERR)

Le premier module traduit chaque descripteur de niveau NIV par son code numérique et traduit les liens horizontaux et verticaux existant avec les autres descripteurs.

Le deuxième est un module général de traduction d'expression booléenne en notation post fixée. Chaque primaire booléen peut être, soit un

critère complémentaire, soit un descripteur de niveau 3 suivi éventuellement d'une expression booléenne de critère et d'une liste d'instructions de rangement et d'une expression booléenne de descripteurs de niveau 4.

Ce module traduit, simultanément, l'expression booléenne de descripteurs de niveau 3, les expressions booléennes de descripteurs de niveau 4 relatives à chaque d3 et les expressions booléennes de critères complémentaires relatives à chaque d3 et d4.

C'est une extension du module TEXBOOL, ce dernier étant utilisé pour des expressions booléennes simples.

Nous donnons, en annexe 12, l'analyse de ces deux modules.

TRAITEMENT

Pour chaque sous question d'une question faire

- si le descripteur de niveau 1 est précédé de ITER alors
 - . mémoriser l'existence de ITER : NTITER (ITR, 1) = -1
 - . incrémenter ITR
- NIV = 1 ; TRDES (NIV, NERR)
 - si il existe une expression booléenne complémentaire et éventuellement une liste d'instructions de rangement alors TREXGEN (NIV, NERR)
- pour chaque élément de question de la sous question faire
 - . si le descripteur de niveau 2 est précédé de ITER alors
 - . mémoriser l'existence de ITER :
 - NTITER (ITR, 1) = -1
 - . incrémenter ITR
 - . NIV = 2 ; TRDES (NIV, NERR)
 - si il existe une expression booléenne de critères complémentaires et éventuellement une liste d'instructions de rangement alors TREXGEN (NIV, NERR)
 - . NIV = 3 ; TREXGEN (NIV, NERR)

si l'élément de question se termine par au moins
un symbole FIT alors

ITI = ITR

pour chaque symbole FIT faire

pour IIT = ITI pas - 1 jusqu'à 1 faire

si il n'existe pas d'élément

NTITER (IIT, 2) = 0 alors

erreur n° 12

sinon aller à éti

éti : NTITER (IIT, 2) = rang dans NL3 du
dernier d3 mémorisé; ITI = IIT

ETAPE 3 MODULE VERIFITER (NERR)

UTILISE tableau NTITER

TRAITEMENT vérifier qu'il ne manque pas un symbole FIT
Soit ITER le rang dans NTITER du dernier symbole ITER mémo-
risé.
si NTITER (ITER, 2) = 0 faire NERR = 1 ; erreur n° 12

ETAPE 2 MODULES TRFAIRE (NERR) et TRTRAIT (NERR)

UTILISE les tableaux NFAIRE, NTPAIT et NOPE

Description des tableaux NFAIRE, NTRAIT

La seule différence qui existe entre ces tableaux et qui existe
entre ces deux modules est que le premier est relatif aux ins-
tructions procédures appartenant aux traitements locaux, le
deuxième aux instructions procédures appartenant aux traitements
globaux.

Nous avons vu que la majorité de ces instructions est commune aux deux types de traitement : la description de ces deux tableaux, comme celle des deux modules est donc identique.

La 1ère colonne de ces tableaux contient le code numérique de l'instruction procédure ; la reconnaissance de ce code génère l'appel du sous programme correspondant.

La 2ème colonne contient un pointeur POPE indiquant le rang dans NOPE du premier opérande de l'instruction.

Description du tableau NOPE

C'est un tableau à une seule ligne, commun aux deux modules, et qui contient la traduction de la liste des opérandes d'une instruction.

Comme nous l'avons déjà vu, chaque identificateur de variable pile ou chaque constante est traduit sur deux mots. Tout identificateur de tableau de statistique est traduit par son numéro.

Chaque liste d'opérande est suivie d'un séparateur (mot contenant la valeur 0).

TRAITEMENT

Pour chaque instruction procédure faire

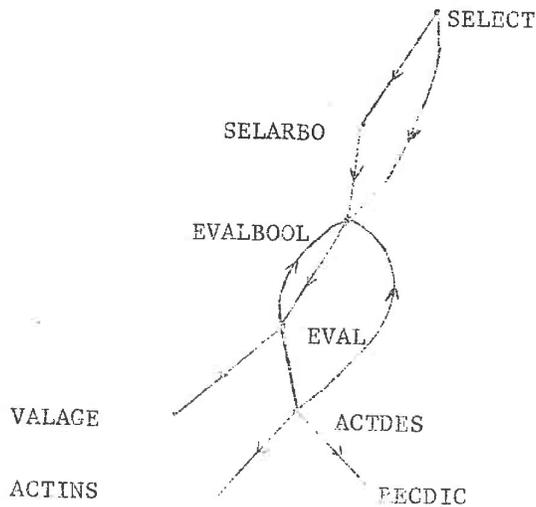
- ranger le code de l'instruction dans la 1ère colonne de NTRAIT ou de NFAIRE (selon le type de traitement)
- si l'instruction est suivie d'opérandes alors
 - . ranger le pointeur POPE dans la 2ème colonne
 - sinon ranger la valeur 0
- pour chaque opérande faire
 - [. traduire cet opérande
 - [. le ranger dans NOPE
 - [. mettre à jour le pointeur POPE
- positionner un séparateur dans NOPE

Exemple : Les paramètres de l'instruction CONS (T1, IE1) sont traduits et rangés dans NOPE de la façon suivante :

.....	1	NTYP=2	NRANG
NOPE	↓	↓	↓	
	n° de l'iden- tificateur	type entier	rang dans NV du contenu de IE1	

C - ANALYSE DU MODULE DE SELECTION SELECT

Grappe des modules ou sous programmes de la relation "utilise"



Pour rendre l'analyse qui va suivre moins lourde, nous avons utilisé des sous programmes récursifs, et nous nous sommes quelque peu écarté des contraintes de programmation (en particulier en FORTRAN)

1 - Module SELECT (NPERT)

UTILISE

- pratiquement tous les tableaux vus lors de l'analyse du module TRADUC. Nous les citons dans les sous programmes appelés par SELECT.
- INI3 et ISUP3 indiquent le rang, dans NI3 (c'est-à-dire le sommet de 3ème niveau du dossier), à partir duquel on explore le tableau (c'est-à-dire le dossier)

- FINDOS = I3, c'est-à-dire la taille du tableau NI3 (indicateur de fin de dossier)
- NDEB et NFIN contiennent le rang dans NL3, respectivement, du 1er descripteur et du dernier descripteur de 3ème niveau de la partie de question réitérée
- le sous programme EVALBOOL (NTRAV, IPOINT, INAT, NRES) évaluant une expression booléenne traduite en post fixée dans un tableau NTRAV à partir du rang IPOINT. Le type de l'expression booléenne (c'est-à-dire le type des opérandes : relation, critère complémentaire ou descripteurs de niveau 3 ou 4), est indiqué dans INAT. NRES = 1 si l'évaluation est vraie, 0 sinon
- le module SELARBO (NDEB, NFIN, NPERT), déterminant la pertinence d'un dossier pour une question ou une partie de question dont le 1er descripteur de niveau 3 (et le dernier descripteur de niveau 3), est au rang NDEB (respect, NFIN) dans NL3.

TRAITEMENT

```

s'il existe une expression booléenne de relation écrite
derrière le symbole SELECT alors INAT = 1
[ EVALBOOL (EXCRI, NEK, NAT, NPERT)
si NPERT = 0 aller a fin programme
sinon ISUP3 = INI3 = 1 ; NDEB = 1 ; NFIN = IL3 (dimension
du tableau NL3)
SELARBO (NDEB, NFIN, NPERT) ; INI3 = ISUP3
si NPERT = 0 alors
[ . pour chaque pile faire
pgm = pgs = pgi ; plm = pls = 0
. aller a fin programme
sinon pour chaque pile globale faire
pgi = pgs = pgm
si ITR (dimension du tableau NITER) = 0
aller a fin programme

```

sinon

commentaire : dans ce qui suit, nous recherchons l'ensemble des parties d'un dossier pertinentes pour une ou plusieurs parties de question. Nous avons autorisé la possibilité d'avoir deux instructions ITER..FIT imbriquées l'une dans l'autre (exemple [()]. Dans ce cas, nous évaluons d'abord l'instruction "contenue" ; si aucune partie du dossier n'est pertinente pour la partie de questions intéressée par cette instruction, nous n'évaluons pas l'instruction "contenant"

. i = ITR

éti : . NDEB = NTITER (i, 1) ; NFIN = NTITER (i, 2)

. pour chaque partie du dossier tant que INI3 FINDOS
faire

SELARBO (NDEB, NFIN, NPERT)

si NPERT = 1 alors

. NIT = 1

. pour chaque pile globale faire

pgi = pgs = pgm

sinon

. pour chaque pile faire

pgm = pgs = pgi ; plm = pls = pli

si NTITER (i,2) = NTITER (i-1,2) alors

si NIT = 0 alors

i = i-2 aller a éti

sinon i = i-1 aller a éti

fin programme

2 - Sous programme EVALBOOL (NTRAV, IPOINT, INAT, NRES)

Nous avons vu que l'intérêt de ce module est de traduire une expression booléenne, traduite en post fixée, dans un tableau NTRAV à partir du rang IPOINT. Nous avons précisé précédemment quels étaient les différents types d'expressions booléennes évaluées.

UTILISE

- Les tableaux "utilisés" dans le sous programme
EVAL (NTRAV, IPOINT, INAT, NRES) (cf. annexe n° 13)
- Une pile de travail NPT dans laquelle sera évaluée l'expression booléenne
- Deux variables de travail NOP1 et NOP2

TRAITEMENT

Pour chaque élément de l'expression booléenne faire

si c'est un opérande alors

EVAL (NTRAV, IPOINT, INAT, NRES)

ranger NRES sur la pile NPT (h) ; (h = h + 1)

si c'est un opérateur alors

dépiler NPT (h) (h = h - 1)

ranger le résultat dans NOP1

dépiler NPT (h) (h = h - 1)

ranger le résultat dans NOP2

si c'est l'opérateur OU alors

NPT (h) = NOP1 + NOP2 (h = h + 1)

sinon NPT(h) = NOP1 x NOP2 (h = h + 1)

sinon si marque de fin d'expression booléenne alors

si NPT (h) = 0 alors NRES = 0 sinon NRES = 1

fin programme

3 - Sous programme SELARBO (NDEB, NFIN, NPRT)

Nous avons défini en I les paramètres NDEB et NFIN.

UTILISE

- Le sous programme EVALBCOL (cf. paragraphe précédent)
- INI3 et ISUP3 indiquant le rang dans NI3 à partir duquel on explore ce tableau
- FINDOS = I3, c'est-à-dire la taille du tableau NI3

TRAITEMENT Pour chaque expression booléenne de niveau 3 (c'est-à-dire pour
chaque élément de sélection tant que INI3 < FINDOS faire

```

IPOINT = NDEB ; INAT = 3 ;
EVALBOOL (NL3, IPOINT, INAT, NPERT)
si IPOINT > NFIN ou NPERT = 0 alors
    aller a fin sous programme

```

fin sous programme

D - ANALYSE D'UN SOUS PROGRAMME REALISANT LA FONCTION D'UNE INSTRUCTION
PROCEDURE PARTICULIERE

1 - Analyse générale des modules ACFAIR. et ACTRAIT

Les instructions procédures contenues dans la phase des traitements locaux et des traitements globaux sont activées de la même façon. Rappelons qu'à l'issue de la phase de traduction (modules TRFAIRE (NERR) et TRTRAIT (NERR)), les codes de ces instructions sont rangés dans les tableaux NFAIRE et NTRAIT, leurs opérandes sont traduits et rangés dans le tableau NOPE. Le pointeur POPE indique, pour chaque instruction, l'adresse dans NOPE du début de la liste des opérandes.

Nous donnons l'analyse générale du module ACFAIRE ; celle du module ACTRAIT est identique.

UTILISE

- Les tableaux NFAIRE et NOPE
- Le pointeur POPE
- Tous les sous programmes réalisant les fonctions des instructions procédures

TRAITEMENT Pour chaque code d'instruction procédure contenu dans NTRAIT faire

```

- se brancher au sous programme correspondant par l'inter-
  médiaire d'un aiguillage (GO TO calculé en Fortran) et
- transmettre le contenu du pointeur POPE

```

Nous ne donnons pas l'analyse de tous les sous programmes appartenant au module ACFAIRE nous nous bornons à donner l'analyse d'un sous programme ACONS, réalisant la fonction originale de l'instruction CONS : c'est-à-dire la construction ou mise à jour d'un tableau statistique.

2 - Analyse du sous programme ACONS (POPE)

UTILISE

- Les tableaux NOPE, NDIR, NPAR et PAR
- Le sous programme RECELEM (NRDIM, NELEM, NBR) recherchant le ou les numéros des caractères sélectionnés d'une dimension.

Commentaire : précisons que NOPE (POPE) contient le numéro (i) de l'identificateur du tableau de statistique et que NOPE (POPE + 1), NOPE (POPE + 2) contiennent éventuellement la traduction d'un identificateur de variable ou de pile. Nous examinons, dans le cas où le tableau de statistique a une dimension et dans celui où il a deux dimensions, les éventualités suivantes :

- si un seul élément est sélectionné :
 - . incrémenter de 1 l'élément ou
 - . incrémenter du contenu d'une variable, ce même élément
- si plusieurs éléments sont sélectionnés au cours d'un même appel de la variable :
 - . incrémenter de 1 ou du contenu d'un élément d'une pile parallèle, chaque élément

TRAITEMENT

Rechercher, au rang i, dans NTAB, le rang NRDIM1 (dans NDIR) du nom de dimension associée aux rangées ligne RECELEM (NRDIM1, NELEM1, NBRL) où NRDIM 1 est le rang dans NDIR de la dimension NELEM1 contient la liste des numéros de caractères sélectionnés NBRL désigne le nombre de caractères sélectionnés

Si le tableau n'a qu'une dimension alors

si NBRL = 1 alors

NOL = NELEML (1)

si NOPE (POPE + 1) = 0 (fin de liste) alors

Ti (NOL) = Ti (NOL) + 1

sinon

- rechercher la valeur de la variable dans NV
ou RV au rang NOPE (POPE + 2)

Ti (NOL) = Ti (NOL) + (contenu de la variable)

sinon pour k = 1 pas 1 jusqua NBRL faire

NOL = NELEML (k)

si NOPE (POPE + 1) = 0 alors

Ti (NOL) = Ti (NOL) + 1

sinon

- rechercher, dans NPIL, le rang dans NV ou RV du
premier élément de la pile

- Ti (NOL) = Ti (NOL) + (contenu de l'élément k+1
de la pile

sinon le tableau a deux dimensions

rechercher le rang NRDIM 2 du nom de dimensions asso-
ciées aux rangées colonnes

RECELEM (NRDIM 2, NELEMC, NBRC)

si NBRL = 1 et NBRC = 1 alors

NOL = NELEML (1) ; NOC = NELEMC (1)

si NOPE (POPE + 1) = 0 alors

Ti (NOL, NOC) = Ti (NOL, NOC) + 1

sinon

- rechercher la valeur de la variable dans NV
ou RV au rang NOPE (POPE + 2)

Ti (NOL, NOC) = Ti (NOL, NOC) + (contenu de
la variable

si NBRL \neq i et NBRC = 1 (respectivement le contraire)alors

pour k = 1 pas 1 jusqa NBRL (resp. NBRC) faire

NOL = NELEM (k) (resp. NOC = NELEMC (k))

si NOPE (POPE + 1) = 0 alors

Ti (NOL, NOC) = Ti (NOL, NOC) + 1

sinon si NOPE (POPE + 2) > 0 (rang de la variable dans NV ou RV) alors

Ti (NOL, NOC) = Ti (NOL, NOC) + (contenu de la variable)

sinon si NOPE (POPE + 2) < 0

rechercher, dans NPIL, le rang dans NV ou RV du premier élément de la pile.

C O N C L U S I O N

Le système de stockage et de gestion des dossiers médicaux est opérationnel depuis mai 1971 et, peu à peu, lors des consultations de malade, l'édition en clair des dossiers par l'ordinateur remplace le traditionnel dossier manuscrit.

Nous nous sommes d'ailleurs aperçus que les médecins étaient, au début, beaucoup plus sensibilisés par ces problèmes d'édition que par l'interrogation et l'exploitation statistique du "système de données".

Le langage de questions a donc été mis au point à partir d'un nombre de questions relativement restreint par rapport à la diversité de celles susceptibles d'être posées. Néanmoins, il permet de répondre à toutes les questions envisagées jusqu'ici, mais il est certain qu'il ne pourra être complètement défini qu'après un long usage des médecins, mais aussi des informaticiens.

Compte tenu du manque de précision et souvent de l'ambiguïté des questions qui nous ont été posées, il semble prématuré de penser que les médecins pourront, dès maintenant, interroger directement leurs fichiers pour en obtenir des réponses valables.

En revanche, nous sommes persuadés qu'un dialogue approfondi et de fréquents échanges entre informaticiens et statisticiens, d'une part, et médecins, d'autre part, permettront à ces derniers de définir l'analyse logique de leurs questions.

C'est sur la base d'une telle collaboration que les médecins pourront utiliser le langage d'interrogation et l'affiner, en proposant des améliorations.

ANNEXE 1

REGLES D'ECRITURE DU DOSSIER CODE
=====

Nous reproduisons ici les règles d'écriture telles qu'elles ont été formalisées dans la thèse de J.M. Martin, en ayant ajouté les restrictions d'écriture concernant les informations complémentaires.

Ces règles sont écrites dans la syntaxe de Backus avec laquelle nous avons pris certaines libertés lorsque ce mode d'écriture entraîne des lourdeurs.

$\langle \text{Dossier} \rangle ::= \text{DOSSI} \langle \text{en-tête du dossier} \rangle [\langle \text{écriture arborescente du dossier} \rangle]$

Le signe [] enfermant une expression, implique la présence possible de cette expression.

1) EN-TETE DU DOSSIER CODE

Cet en-tête est obligatoirement constitué des éléments suivants pris dans cet ordre et séparés les uns des autres par le signe + considéré comme séparateur.

- Le numéro du dossier dans le fichier des dossiers codés
- La date de codage du dossier (cf. règles d'écriture des dates)
- Le sexe du malade (adjectif HOMME ou FEMME)
- La date de naissance du malade (cf. règles d'écriture des dates).

2) ECRITURE ARBORESCENTE DU DOSSIER

$\langle \text{écriture arborescente du dossier} \rangle ::= \langle \text{suite de 1er niveau} \rangle$

Nous écrirons, par abus de langage, pour condenser trois règles en une seule (en considérant N égal dans l'ordre 1, 2, 3 dans les deux membres de l'expression) :

- < suite de N^{ème} niveau > ::= < descripteur de niveau N > [< information complémentaire >] [< suite de (N + 1)^{ème} niveau >] |
- < suite de N^{ème} niveau > < descripteur de niveau N > [< information complémentaire >] [< suite de (N + 1)^{ème} niveau >]
- < suite de 4^{ème} niveau > ::= < descripteur de niveau 4 > [< information complémentaire >] | < suite de 4^{ème} niveau > < descripteur de niveau 4 > [< information complémentaire >]

3) ECRITURE DES INFORMATIONS COMPLEMENTAIRES

- < information complémentaire > ::= [< date >] [< heure >] < adjectif > |
 [< date >] [< heure >] < adjectif >
 < adjectif > |
 [< date >] [< heure >] < nombre >
 < adjectif > |
 [< date >] [< heure >] < nombre > |
 [< date >] [< heure >]
 | < date > + JUSQA < date > |
 | [< date >] [< heure >] < nombre >
 + JUSQA < nombre >

- < date > ::= + D < Millésime > [< Mois >] [< Jour >]
 < Millésime > ::= < suite de 3 chiffres >
 < Mois > ::= < suite de 2 chiffres >
 < Jour > ::= < suite de 2 chiffres >

- < heure > ::= + H < heure > [< minute >]
 < heure > ::= < suite de 2 chiffres >
 < minute > ::= < suite de 2 chiffres >

Précisons que dans l'écriture de la date, si le jour existe, il en est de même pour le mois ; de même dans l'écriture de l'heure, si elle existe, elle est précédée de la date

- < nombre > ::= + < nombre entier > | + < nombre réel >
 < adjectif > ::= + < descripteur >

Un exemple d'écriture de dossier codé est donné en annexe 4.

RÉSUMÉ DE L'HOSPITALISATION

CONCLUSIONS THÉRAPEUTIQUES

- femme / 1 glucosant le matin
le régime à condition que poids \geq a 75 kg
3 mois cycle
- oculiste à choisir par le médecin
- voir probablement en grossesse avec Vichyminant

ÉTAT DU MALADE A LA SORTIE

CONDUITE ET EXAMENS A PRÉVOIR LORS DU PROCHAIN EXAMEN

CONSULTATION PRÉVUE

le | | | | | | | |

HOSPITALISATION PRÉVUE

le | | | | | | | |

MODE DE SURVEILLANCE DU DIABETE

inorganisé

- Par glycémie Tous les
- Par glycosurie Tous les
- Test de glycosurie par le malade Tous les fois par jour

- Surveillance médicale trimestrielle
- Pas de surveillance biologique
- Surveillance médicale annuelle

MOTIF DE L'ADMISSION

Date : 19/11/04/0131

- fatigue - tachycardie découvert diabète
- 7^e régime

le 23/8/77 glycémie 1.57 - C.T. 6.33 L.T. 7.20 NF 5050 000
30/3/77 — 1.57 9200

Régime glucide

8/10/77 sans régime à jeun 1.60 11 h 1.25 17 : 1.19

Trait. glucidant (1,01)

pté régime 2.69 16 h 1.64 glycémie null

HISTOIRE DE L'ÉPISODE ACTUEL

font E.C.G / tachycardie sinusale
Bloc focal droit p
Achilles = 320 mps / Trait. cardioythémique
Héparidém

ANNEXE 4 bis

DESCRIPTION DES TABLEAUX D'EXPLOITATION
=====

A chaque dossier médical correspondent cinq tableaux d'exploitation ; les quatre premiers NI1, NI2, NI3, NI4 sont relatifs aux descripteurs introducteurs de niveau, le cinquième AA contient les éléments d'informations complémentaires.

a) Tableaux NI1, NI2, NI3, NI4

Pour tous les tableaux, les deux premières colonnes ont la même signification :

- La première colonne contient les codes numériques des descripteurs du niveau correspondant ; dans le tableau NI3, il y a un réarrangement des lignes pour placer les codes numériques de cette colonne dans l'ordre croissant. Ce réarrangement a pour but de diminuer le temps de recherche d'un descripteur de troisième niveau en faisant par exemple une dichotomie.
- La deuxième colonne contient le numéro d'ordre dans le tableau AA (pointeur) du premier élément d'information complémentaire attaché au descripteur concerné par cette ligne.
Les tableaux NI1 et NI4 n'ont que ces deux colonnes.
- La troisième colonne des tableaux NI2 et NI3 contient un pointeur indiquant le rang dans les tableaux, respectivement NI1, NI2 du père du descripteur considéré.
Le tableau NI2 n'a que ces trois colonnes.
- Pour chaque ligne, la quatrième (cinquième) colonne de NI3 contient le numéro d'ordre dans le tableau NI4 du premier respectivement (dernier) fils du descripteur de troisième niveau considéré.
- Enfin, nous avons ajouté à ce tableau une sixième colonne indiquant, pour chaque ligne, son numéro d'ordre.

MODULES DU TRAITEMENT DES DOSSIERS
 =====

NIVEAU 4

MODULE INIT

UTILISE

- la chaîne de caractères que constitue un "dossier-carte" : cette chaîne est rangée à partir de l'adresse NT
- un pointeur indiquant l'endroit de reprise du traitement dans la chaîne : IPOINT
- quatre paramètres :
 - . METB (2) tableaux à deux éléments (double mot nécessaire pour ranger un descripteur : chaîne de 5 caractères
 - . NF : variable indiquant le type de l'information extraite de la chaîne : descripteur introducteur de niveau, non introducteur de niveau, date, heure, nombre réel ou entier
 - . NV . indique, lorsque l'information est un nombre réel, le nombre de chiffres après la virgule.

TRAITEMENT

- recherche le descripteur DOSSI en tête de la chaîne de caractères, initialise le pointeur IPOINT
- assure le passage des paramètres vers le programme principal.

MODULE CANSUIV

UTILISE

- la chaîne de caractères que constitue le "dossier-carte"
- le pointeur IPOINT
- les paramètres.

TRAITEMENT

- isole de la chaîne de caractères toute information comprise entre deux caractères blancs
- supprime les caractères placés en tête spécifiques du type, c'est-à-dire : +D et +H si l'information est respectivement une date ou une heure, "+" si c'est un autre élément d'information complémentaire
- range l'information dans le double mot METB (2) et positionne NF à une valeur reflétant le type de cette dernière si c'est un nombre réel alors NV= nombre de chiffres placés après la virgule
- **incrémenter** le pointeur IPOINT du nombre de caractères de l'information extraite.

MODULE DESC

UTILISE

- les tableaux d'exploitation MI_i ($i = 1, 4$)
- l'information continue dans METB (2)
- le lexique (tableau ND)
- un sous-programme de recherche dichotomique dans ND (CHND)

TRAITEMENT

- recherche le descripteur introducteur de niveau (contenu dans METB(2)) dans le lexique ND (sous-programme CHND)
- si le descripteur n'existe pas alors sortir un message d'erreur
sinon
 - . vérifier que son niveau (reflété par son code numérique) est conforme à la structure arborescente
 - . ranger le code dans le tableau d'exploitation correspondant

si c'est un descripteur de niveau 4 alors

. vérifier qu'il est utilisé dans la famille à laquelle il appartient

si c'est le premier du succédant au descripteur de troisième niveau alors

. ranger le pointeur "premier fils" dans NI3

si c'est le dernier du succédant au descripteur de troisième niveau alors

. ranger le pointeur "dernier fils" dans NI3

si c'est un descripteur de niveau 2 ou 3 alors

. ranger les pointeurs "père" respectivement dans NI1, NI2

si l'élément est un nombre alors

si $NV \neq 0$ alors faire nombre = nombre/NV

ranger la valeur dans le tableau AA.

MODULE CLAS

UTILISE

- le tableau d'exploitation NI3

TRAITEMENT

- réorganise le tableau NI3 en le rangeant selon l'ordre croissant des codes numériques, afin de permettre une exploitation dichotomique de ce tableau.

La sixième colonne est sauvegardée lors de ce réarrangement.

ANNEXE 6

CONFIGURATION DU LEXIQUE

Ce lexique est logé dans un tableau à 810 lignes et est stocké sur bande magnétique.

Chaque ligne comporte les éléments suivants :

- la représentation externe d'un descripteur (chaîne de cinq caractères alphabétiques) sur deux mots
- la représentation interne (code numérique)
- si c'est un descripteur de 4ème niveau : un pointeur indiquant le rang dans le lexique du descripteur de 3ème niveau "père" de la famille à laquelle il appartient
- la signification en "clair" du descripteur (nous avons fixé un nombre maximal de 48 caractères utilisés pour cette traduction).

L'intérêt d'avoir en face de chaque descripteur un texte significatif apparaît dans les problèmes d'édition d'un dossier.

Les lignes sont rangées dans le tableau selon l'ordre alphabétique de la représentation externe du descripteur.

ANAMN	3001	0	L'INTERROGATOIRE DU MALADE MONTRE :
ANAST	675	600	ANASTOMOSE PORTOCAVE SPLENORENALE
ANEVR	450	294	ANEVRYSMES ARTERIELS ET ARTERIO VEINEUX
ANGAS	961	777	GASTRO EUNEROSTOMIE
ANGIO	922	510	ANGIOPATHIE DIABETIQUE
ANKYL	763	27	SPONDYLARTHRITE ANKYLOSANTE
ANNEE	510	380	SURVEILLANCE MEDICALE ANNUELLE SEULEMENT
ANUSS	1056	0	ANUS
APARU	474	201	DATE D'APPARITION DU DIABETE
APLAS	784	414	APLASTEMEDULLAIRE
APRES	2	0	APRES L'APPARITION DU DIABETE
APRØX	17	0	APPROXIMATIF
ARINF	1307	534	ARTHRITE AIGUE INFECTIEUSE
ARRET	1169	119	MEMØPAUSE NATURELLE
ARSUP	921	43	ARTERITE DES MEMBRES SUPERIEURS
ARTER	920	53	ARTERE
ARTIC	760	58	ARTICULATION
ARTIS	422	729	ARTISAN
ARTIT	540	529	ARTERITE DES MEMBRES INFÈRIEURS
ARTRØ	761	44	ARTHRØSE
ARYTH	528	786	ARYTHMIE COMPLETE LENTE
ASCAR	823	727	ASCARIS
ASCIT	672	372	ASCITE
ASTHM	631	214	ASTHME
ASYST	519	691	INSUFFISANCE CARDIAQUE
ATRØF	559	675	AMYØTROPHIE
ATYPI	889	0	EVØLUTION ATYPIQUE
AUDIO	1014	257	AUDIOGRAMME
AUTRE	482	251	DEPISTAGE EN D'AUTRES OCCASIONS
AUTRE	978	203	AUTRES MEDICAMENTS HYPØGLYCEMIANTS
AVANC	901	459	INJECTIONS AVANCEES PAR RAPPORT AU PETIT DEJEUNE
AVANT	54	0	AVANT L'APPARITION DU DIABETE
AVBRA	78	0	AVANT BRAS
AVDIN	982	210	GLYCEMIE AVANT LE DINER
AVØRT	915	496	FAUSSE COUCHE
AVREP	981	72	GLYCEMIE APRES LE REPAS DE MIDI
AZØTE	841	568	PRØTIDEMIE
BARAG	864	332	ELEVATION DU SEUIL RENAL
BARBI	830	588	INTØXICATION BARBITURIQUE
BASED	717	485	MALADIE DE BASEDØW
BASIN	85	0	BASSIN
BBØN	911	133	ENFANT DE PØIDS INFÈRIEUR A 4 KG A LA NAISSANCE
BBFØR	910	80	ENFANT DE PØIDS SUPERIEUR A 4 KG A LA NAISSANCE
BBSCH	1026	694	MALADIE DE BESNIER, BØECH, SCHAUMAN
BDBCØ	524	212	BLCØ DE BRANCHE COMPLET
BOBPA	523	83	BLCØ DE BRANCHE PARTIEL
BEVIT	1137	300	VITAMINØTHERAPIE DU GROUPE B
BIERM	775	144	ANEMIE HYPERCHROME
BILAN	41	0	BILAN ANNUEL
BILAT	33	0	BILATERAL
BILIA	941	788	VØIES BILIAIRES EXTRA HEPATIQUES
BILIR	1087	11	BILURUBINE TOTALE
BIØPS	72	0	BIØPSIE AVEC EXAMEN HISTØLOGIQUE
BKØCH	83	0	BACILLE DE KØCH
BLANC	862	17	GLØBULES BLANCS
BLENØ	1159	656	BALANITE URETHRITE
BØNNE	207	0	BØN, ADEQUAT
BØUCH	621	408	AFFECTION DE LA BØUCHE
BØUTØ	1186	359	FIEVRES ERUPTIVES VIRALES
BØUYØ	929	381	ENDOCARDITE RHUMATISMALE
BRASS	110	0	BRAS
BRØNC	629	602	BRØNCHE
BRØSP	1091	573	EPREUVE A LA BSP
BUCHE	427	439	BUCHERØN FØRGERØN DØCKER
BURAU	972	0	EMPLØYE DE BUREAU
BUTFØ	952	287	RUTFERMIN

EXEMPLE DE LISTING DE DOSSIER

DOSSI +822 +D9720125 +HOMME +D9250725 JOURN +D9711103 ETCIV FAMIL MARIE METIE +D
 950 +JUSGA +D971 MILIT ACTIV MODER PRØFE +SANSS ANAMN NAISS ENFAN +02 +AVANT BBB
 6N +02 +AVANT TRAVA +02 +AVANT HERED TAMAT +02 +ØBESE SØEUR +01 +ØBESE ETIØL APA
 RU +D9710823 SIGNE FØRME +D97107 PØIDS +85 TAILL +175 ALIME CALØR +2200 PRØTI +8
 0 LIPID +98 GLUCI +250 TRAIT CARBU +02 EXAME FØRME PØIDS +13 MINIM
 0 LIPID +98 GLUCI +250 TRAIT CARBU +02 EXAME FØRME PØIDS +79,8 TAILL +175 VAISS
 MAXIM +13 MINIM +07 ØTLAR PRØTE +PARTI GENIT IMPUI MUSCL PADUP DØSAG CØLES +2,33
 LIPTØ +7,2 RØUGE +505 BLANC +92 ACHIL +320 DIAGN MATURE VISIØ ACDRO +10 ACGAU +1
 0 MEGAT +80 INFRA +40 RENØR TERAP +D9711103 MANGE CALØR +2200 GLUCI +250 LIPID +
 98 PRØTI +80 TRAIT CARBU +02 VALEU AJEUN +1,4 DEJEU +2,25 AVREP +1,25 REPAS +1,4
 + AVDIN +1,19 GLYCO +00 EQUIL +MEDIØ DRØGE CØMPR +CØEUR CØMPR +HEPAT
 DDDD

 DOSSIER NUMERO 822 *****
 DOSSIER CODIFIE LE 25/ 1/ 1972 *****

LE MALADE EST DU SEXE MASCULIN
 IL EST NE LE 25/ 7/ 1925
 HOSPITALISATION DE JOUR

LE 3/ 11/ 1971

AU POINT DE VUE ETAT CIVIL LE MALADE EST

MARIE

DE SON METIER

DU 0/ 0/ 1950JUSQU'AU 0/ 0/ 1971

SØUS OFFICIER ASSIMILE ET SØLDAT

ACTIVITE NON PROFESSIONNELLE

ACTIVITE NON PROFESSIONNELLE MODEREE

PROFESSION DU CONJOINT DU MALADE

INCONNU ØU ABSENT

L'INTERROGATOIRE DU MALADE MONTRE :
 ENFANTS,MATERNITES,PROBLEMES ØBSTETRICAX

LE SUJET A SUBI UN TRAITEMENT

GLUCIDORAL

A L'EXAMEN ON A TROUVE
MORPHOTYPE

2.000

ETENAIG

LE POIDS EST DE
TAILLE EN CENTIMETRES

79.800
175.000

AU POINT DE VUE VASCULAIRE

TENSION ARTERIELLE MAXIMA
TENSION ARTERIELLE MINIMA

13.000
7.000

OTORHINOLARYNGOLOGIE

PROTHESE DENTAIRE

PARTIEL

APPAREIL GENITAL

IMPUISSANCE, HYPOFONCTIONNEMENT FRIGIDITE

MUSCLE, TENDON, CONJONCTIF

PAS DE DUPUYTREN

DOSAGE

CHOLESTEROL TOTAL
LIPIDES TOTAUX
GLOBULES ROUGES
GLOBULES BLANCS
ACHILLOGRAMME

2.330
7.200
505.000
92.000
320.000

DIAGNOSTIC DU DIABETE

DIABETE DE LA MATURITE

APPAREIL OCULAIRE

ACUITE VISUELLE DROITE AVEC CORRECTION
ACUITE VISUELLE GAUCHE AVEC CORRECTION
TENSION ARTERIELLE RETINIENNE MAXIMA
TENSION ARTERIELLE RETINIENNE MINIMA
EXAMEN DE LA RETINE NORMAL DES 2 COTES

10.000
10.000
80.000
40.000

ENFANTS NES VIVANTS
 AVANT L'APPARITION DU DIABETE 2
 ENFANT DE POIDS INFERIEUR A 4 KG A LA NAISSANCE 2
 AVANT L'APPARITION DU DIABETE 2
 ACCOUCHEMENT NORMAL A TERME
 AVANT L'APPARITION DU DIABETE 2
 HEREDITE DU DIABETE ET DE L'OBESITE

TANTES MATERNELLES
 OBESSE 2
 OBESSE SŒURS 1
 DEBUT DU DIABETE

DATE D'APPARITION DU DIABETE LE 29 / 8 / 1971
 LE DEPISTAGE EST DU AUX SIGNES CLINIQUES
 MORPHOTYPE LE 0 / 7 / 1971

***HABITUDES ALIMENTAIRES
 LE POIDS EST DE 85.000
 TAILLE EN CENTIMETRES 175.000

***LE SUJET A SUBI UN TRAITEMENT
 CALORIES NON ALCOOLIQUES 2200.000
 RATION PROTIDIQUE 80.000
 RATION LIPIDIQUE 98.000
 RATION GLUCIDIQUE 250.000

A L'EXAMEN ON A TROUVE GLUCIDORAL 2.000
 MORPHOTYPE

INF. COMPL. NON VALIDE77952576
 TENSION ARTERIELLE MINIMA
 DOSSIER MAL CONSTRUITGLUCI
 LE POIDS EST DE 13.000
 42414653

LISTE DES ERREURS D'ECRITURE
RENCONTREES LORS DE LA TRADUCTION D'UN TRAIN DE QUESTIONS

=====

- Erreur n° 1 : identificateur de variable ou de pile non autorisé
- Erreur n° 2 : pile déjà "annexée" par des piles de déclaration antérieure
- Erreur n° 3 : taille de pile trop grande
- Erreur n° 4 : débordement de pile incorrect
- Erreur n° 5 : type de partition incorrecte
- Erreur n° 6 : descripteur inexistant
- Erreur n° 7 : écriture de liste de bornes incorrectes
- Erreur n° 8 : expression booléenne mal écrite
- Erreur n° 9 : nom de dimension inconnu
- Erreur n° 10 : descripteur utilisé à un mauvais niveau
- Erreur n° 11 : identificateur de fonction de recherche incorrect
- Erreur n° 12 : écriture incorrecte de l'instruction ITER...FIT

SOUS PROGRAMME DE TRADUCTION D'IDENTIFICATEUR
 DE VARIABLE OU DE PILE RANG (NTYP, NRG, NAT, NERR)
 =====

Nous n'avons pas fait figurer, en paramètre formel, la variable contenant l'identificateur de variable ou de pile que l'on souhaite traduire. Les paramètres NTYP, NRG, NAT et NERR indiquent respectivement au module activant ce sous programme

- le type de l'identificateur de pile ou de variable (positionné à 1, 2, 3, 4 ou 5 si le type est réel, entier, code, date ou heure)
- le rang dans NV ou RV selon le type de l'identificateur de variable ou le rang dans NPIL pour un identificateur de pile
- le genre : identificateur de variable ou de pile (positionné à 1 ou 2)
- que l'identificateur n'est pas autorisé (NERR = 1, 0 sinon)

Ce sous programme :

UTILISE

- le rang dans NV des variables IC1, ID1, IE1 et IH1 ; le calcul, par exemple, du rang de la variable IC_i dans NV est réalisé de la façon suivante : rang de IC_i = rang de IC1 + i-1

Remarque : le calcul du rang d'une variable réelle dans RV est immédiat : rang Ri = i

- le rang dans NPIL des identificateurs IPC1, IPE1... IPH1 et RPI ; le calcul du rang de l'identificateur IPE_i est réalisé de la même façon : rang de IPE_i = rang de IPE1 + (i-1)

ANNEXE 10

SOUS PROGRAMME DE TRADUCTION
EN NOTATION POSTFIXEE D'UNE EXPRESSION BOOLEENNE
TEXTBOOL(TRAV, IPOINT, NERR)

=====

UTILISE une pile NPT dans laquelle sont rangés les opérateurs

TRAITEMENT Pour chaque élément de l'expression (opérande ou opérateur) faire

si c'est un opérande faire

- . traduire l'opérande et le ranger dans TRAV à partir du rang IPOINT ; incrémenter IPOINT

sinon si c'est un opérateur faire

si ce n'est pas une ")" faire

si l'élément du sommet de la pile est ET ou OU faire

- . ranger cet élément dans TRAV au rang IPOINT
- . incrémenter IPOINT
- . dépiler la pile des opérateurs NPT (h) : (h = h - 1)
- . ranger l'opérateur au sommet de la pile (h = h + 1 ; NPT (h) = opérateur) .

sinon faire

- . ranger l'élément du sommet de la pile dans TRAV au rang IPOINT ; incrémenter IPOINT
- . dépiler NPT (h) : h = h - 2 (on enlève aussi de la pile la ")")
- . vérifier lors de cette opération que NPT(h) ≠ \wedge (\wedge : indicateur de pile vide)

sinon faire NERR = 1 ; erreur n° 8

ANNEXE 11

MODULES DE TRADUCTION DES LISTES DE CARACTERES

ETAPE 4 MODULE TDIMBOOL (NERR, NØDIM)

UTILISE

- Les tableaux NDI et PAR
- Le sous programme TEXBOOL (TRAV, IPOINT, NERR)
traduisant, en notation post fixée, une expression booléenne écrite dans un tableau TRAV à partir du rang IPOINT et dont les opérandes sont des relations. Nous donnons son analyse en annexe 10.

TRAITEMENT NCOMPT = 0

Pour chaque expression booléenne faire

- . TEXBOOL (PAR, IPC, NERR)
- . NCOMPT = NCOMPT + 1
- . Positionner un séparateur d'expression dans PAR

Taille de dimension = NCOMPT

ETAPE 4 MODULE TDIMDES (NERR, NØDIM)

UTILISE

- Les tableaux NDI et NPAR
- Le sous programme CHND (NDES, NCØDE, NB)
réalisant une recherche dichotomique dans le lexique d'un descripteur contenu dans NDES et mémorisant son code dans NCØDE.
NB = 1 si recherche positive, 0 sinon

TRAITEMENT NCOMPT = 0

Pour chaque descripteur de la liste faire

```
    si ce n'est pas le descripteur RESTE alors  
        . recherche dichotomique dans le lexique du descripteur :  
          CHND (DES, NCØDE, NB)  
          si NB = 1 alors  
              - ranger dans NPAE au rang IPC NCØDE  
              - incrémenter IPC  
              - NCOMPT = NCOMPT + 1  
          sinon NERR = 1 ; erreur n° 6  
    sinon IREST = 1 ; NCOMPT = NCOMPT + 1
```

Taille de dimension = NCOMPT

Classer, par ordre numérique croissant, les codes numériques

si IREST = 1 alors

```
    - ranger le code 0 dans NPAR  
    - incrémenter IPC
```

Positionner dans NPAR un séparateur de liste.

ETAPE 4 MODULE TDIMNUM (NERR, MODIM, NTYP)

Précisons que le paramètre NTYP rappelle le type des bornes définissant la partition. Nous ne nous intéressons qu'aux deux cas suivants : NTYP = 1 (type réel) ou NTYP = code AGE ; la liste des bornes est alors rangée dans NPAR

NTYP ≠ 1 et NTYP ≠ code AGE ; la liste des bornes est alors rangée dans NPAR.

TRAITEMENT

si type de partition = régulière alors

- ranger la borne inférieure, supérieure et le pas dans PAR ou NPAR à partir de IPC
- incrémenter IPC

si borne supérieure < borne inférieure
ou pas > (borne supérieure - borne inférieure) alors
NERR = 1 ; erreur n° 7

- taille de dimension = (partie entière (borne supérieure - borne inférieure)/pas) + 1

sinon faire

ICOMPT = 0

Pour chaque borne de la liste faire

- . la ranger dans NPAR ou PAR (selon le type) au rang IPC
- . incrémenter IPC et ICOMPT de un

Taille de dim = ICOMPT - 1

Positionner une marque de fin de liste dans NPAR ou PAR.

ANNEXE 12

MODULES DE TRADUCTION DE LA QUESTION ARBORESCENTE
=====

ETAPE 4 MODULE TRDES (NIV, NERR)

NIV indique, au module, le niveau que doit avoir le descripteur dont on traduit le code numérique.

UTILISE

- Les tableaux NL1, NL2, NL3 et NL4
- Un sous programme de recherche dichotomique CHND (NDES, NCØDE, NB). Ce sous programme recherche, dans le lexique, le code numérique mis dans NCØDE du descripteur contenu dans NDES. NB = 0 si le descripteur n'existe pas, 1 sinon. Nous appelons niv1 le niveau du descripteur recherché

TRAITEMENT

CHND (NDES, NCØDE, NB)

si NIV = 1 alors

si NB = 0 alors

si NDES = **INDIF** 1 alors

 NCØDE = 0 ; ranger la valeur dans NL1

sinon NERR = 1 ; erreur n° 6

sinon

si NIV ≠ niv1 alors

 NERR = 1 ; erreur n° 10

sinon ranger NCØDE dans NL1

sinon si NIV = 2 alors

si NB = 0 alors

si NDES = INDIF2 alors

NCØDE = 0 ; ranger la valeur dans NL2

sinon NERR = 1 ; erreur n° 6

sinon

si NIV ≠ niv2 alors NERR = 1 ; erreur n° 11

sinon ranger NCØDE dans NL2

positionner le pointeur père 2

sinon si NIV = 3 alors

si NB = 0 alors NERR = 1 ; erreur n° 6

si NIV ≠ niv3 alors erreur n° 10

sinon ranger NCØDE (ou - NCØDE, si le descripteur est précédé de NON) dans NL3

sinon

si NB = 0 alors NERR = 1 ; erreur n° 6

si NIV ≠ niv4 alors NERR = 1 ; erreur n° 10

sinon ranger NCØDE (ou - NCØDE, si le descripteur est précédé de NON) dans NL4

ETAPE 4 MODULE TREXGEN (NIV, NERR)

UTILISE

- Le module TRDES (NIV, NERR)
- Les tableaux NLi (i = 1, 4), NTITER, EXCRI et INST
- Une pile des opérateurs NPILEA

Rappelons que ce module traduit une expression booléenne générale en notation post fixée. Les opérandes peuvent être : soit un critère complémentaire, soit une liste d'instructions de rangement, soit un descripteur de niveau 3 suivi éventuellement d'une expression de critères complémentaires et d'une expression booléenne de descripteurs de

niveau 4. Dans ce dernier cas, le module complète la traduction de l'instruction ITER...FIT.

Nous avons ajouté, aux opérateurs courants ET, OU, "(", les opérateurs "[": début d'expression booléenne de descripteurs de niveau 4 et AVEC: début d'expression booléenne de critères complémentaires. Nous pouvons remarquer que l'opérateur "[" peut aussi indiquer le début d'une expression booléenne de niveau 3; nous avons levé cette ambiguïté en activant ce module, lorsqu'il s'agit de traduire une telle expression, lors de la rencontre du premier d3.

Nous utilisons un mot d'état MET indiquant, à chaque instant, le type d'expression booléenne traduite (expression booléenne de d3 ou de d4 ou de critères complémentaires).

TRAITEMENT Pour chaque élément de l'expression booléenne faire

si c'est un opérande alors

si c'est un d3 (éventuellement précédé de non) alors

NIV = 3; TRDES (NIV, NERR)

si c'est le premier d3 de l'expression booléenne alors

II = ITR (rang de la dernière ligne de NTITER)

Pour IIT = II-1 tant que NTITER (IIT, 1) = -1 faire

remplacer la marque d'existence de ITER par le rang dans NL3 du premier d3 placé derrière ce symbole, c'est-à-dire:

NTITER (IIT, 1) = rang du code numérique du d3 dans NL3

sinon si c'est un d4 (éventuellement précédé de non) alors

NIV = 4; TRDES (NIV, NERR)

si c'est le premier d4 de l'expression booléenne alors

positionner le pointeur p fils dans NL3

sinon si c'est un critère complémentaire alors

- traduire le critère (s'il contient un adjectif, nous réalisons une recherche dichotomique dans le lexique, pour obtenir son code numérique (sous programme CHND)
- mettre à jour le pointeur pexi (i = 1, 2, 3 ou 4)

sinon si c'est une liste d'instructions de rangement alors

- pour chaque instruction de rangement faire
 - . traduire l'instruction
 - . mettre à jour le pointeur prgi (i = 1, 2, 3 ou 4)
- positionner un séparateur de fin de liste

si c'est un opérateur alors

si c'est le début d'une expression booléenne de d4 ("[") faire

- mettre dans MET : "début expression booléenne d4"

sinon si c'est la fin d'une expression booléenne d4 ("[") faire

- mettre dans MET : "début expression booléenne d3"
- positionner, dans NL3, le pointeur "dfiis" (rang dans ML4 du code numérique du dernier d4)

sinon si c'est le début d'une expression booléenne de critères complémentaires (avec) faire

- mettre "début expression booléenne CRIT" dans NPILEX (h) ;
h = h + 1
- sauvegarder le contenu de MET
- ranger dans MET : "début expression booléenne CRIT"

sinon si c'est l'opérateur et, ou, "(" faire

- si l'élément du sommet de la pile est et ou ou faire
 - si MET = début expression booléenne d3
ranger cet élément dans NL3
 - sinon si MET = début expression booléenne d4
ranger cet élément dans NL4
 - sinon ranger cet élément dans EXCRI
- dépiler la pile des opérateurs NPILEX (h) ; h = h - 1
- ranger l'opérateur au sommet de la pile
NPILEX (h) = opérateur ; h = h + 1

sinon si c'est l'opérateur ")" faire

- si MET = début expression booléenne d3
ranger NPILEX (h) dans NL3

sinon si MET = début expression booléenne d4
ranger NPILEX (h) dans NL4

sinon ranger NPILEX (h) dans EXCRI

- dépiler NPILEX (h) ; $h = h - 1$

si l'opérateur AVEC est au sommet de la pile faire

. restaurer MET

. dépiler NPILEX (h) ; $h = h - 1$

. mettre un séparateur dans EXCRI

ANNEXE 13

SOUS PROGRAMME EVAL (NTRAV, IPOINT, INAT, NRES)
=====

Ce sous programme évalue, soit :

- 1) une relation booléenne auquel cas, NTRAV est remplacé lors de l'appel par EXCRI : tableau contenant l'expression booléenne de relation traduite en post fixée.
IPOINT indique, dans NTRAV, le début de la relation booléenne ;
INAT indique le type d'expression évalué ; ici, NAT = 1
- 2) un critère complémentaire. Le paramètre NTRAV est remplacé par EXCRI ; il a, ainsi que IPOINT, la même signification que précédemment NAT = 2
- 3) une expression de descripteur de troisième niveau, auquel cas NTRAV est remplacé lors de l'appel par NL3 ; IPOINT indique le début dans NL3 de l'expression booléenne de d3 NAT = 3
- 5) une expression booléenne de descripteur de quatrième niveau. Le paramètre NTRAV est remplacé lors de l'appel par NL4, IPOINT indique le début dans NL4 de l'expression booléenne de d4 NAT = 4.
Dans tous les cas NRES = 1 si évaluation correcte, 0 sinon

UTILISE - les tableaux NI3, NI1, NI2, NI4 et NTRAV
- le sous programme EVALBOOL (NTRAV, IPOINT, INAT, NRES)
- le sous programme ACTDES (cf. annexe n°14)
- le sous programme VALAGE (date) qui calcule l'âge d'un individu à la date indiquée en paramètre
- des variables transmises vers le programme principal en
COMMON : NDFI3, NPI3 : adresse dans NI4 du premier fils et de dernier fils d'un descripteur de troisième niveau.

TRAITEMENT si INAT = 1 alors

si NTRAV(IPOINT) = code de la fonction AGE alors

 - rechercher dans NV au rang NTRAV(IPOINT+1) la valeur de la date

 - VALAGE (date) ; aller a éti

sinon rechercher dans IV ou RV au rang NTRAV(IPOINT+1) la valeur de la variable (le type est indiqué en NTRAV(IPOINT))

éti . décodé le code de l'opérateur de relation (placé dans NTRAV(IPOINT+2))

 . rechercher dans NV ou RV au rang NTRAV(IPOINT+4) la valeur du second membre (type indiqué dans NTRAV(IPOINT+3))

 IPOINT = IPOINT + 5

 . évaluer la relation

 . si elle est vraie alors NRES=1 aller a fin sous programme

sinon NRES = 0 aller a éta

si INAT = 2 alors

 . appel de la fonction de recherche dont le code est dans NTRAV(IPOINT)

 . si l'information recherchée est absente alors NRES = 0 aller a éta

sinon

 . décodé le code de l'opérateur de relation (placé dans NTRAV(IPOINT+1))

 . rechercher dans IV ou RV au rang NTRAV(IPOINT+3) la valeur du second membre (type indiqué dans NTRAV(IPOINT+2))

 . IPOINT = IPOINT + 4

 . évaluer la relation

 . si la relation est vraie alors NRES=1 aller a fin sous-programme sinon NRES = 0 aller a éta

si INAT = 3 alors

- NCØDE = NTRAV(IPOINT)

- NIV = 3 ; ACTDES(NIV, NCØDE, NRANG, NRES)

si NRES=0 aller a éta

sinon

vérification NDFI3=NI3(NRANG,4) ; NPF13=NI3(NRANG,5) ;
du NRANG=NI3(NRANG,3) ; NR2 =NTRAV(IPOINT + 3)

lien vertical NCØDE=NL2(NR2, 1) ; NIV =2 ;

d2 → *d3* ACTDES(NIV, NCØDE, NRANG, NRES)

si NRES = 0 aller a éta

sinon

vérification NRANG=NI2(NRANG,3) ; NR1=NL2(NR2,4) ;
du NCØDE=NL1(NR1,1) ; NIV=1 ;

lien vertical ACTDES(NIV, NCØDE, NRANG, NRES)

d1 → *d2* si NRES = 0 aller a éta

sinon

vérification NR4=NTRAV(IPOINT+4) ; NIV=4
du EVALBOOL(NL4, NR4,4, NRES)

lien vertical si NRES ≠ 0 aller a lab

d3 → *famille*

de d4

si INAT = 4 alors

. NCØDE=NTRAV(IPOINT) ; NIV = 4

. ACTDES(NIV, NCØDE, NRANG, NRES)

si NRES = 0 aller a éta

sinon aller a fin sous programme

éta : Pour chaque pile faire Plm=Pls ;

Pgm=Pgs aller a fin sous programme

lab : Pour chaque pile faire Pls=Plm ;

Pgs=Pgm fin sous programme

ANNEXE 14

SOUS PROGRAMME ACTDES(NIV, NCØDE, NRANG, NRES)
=====

Ce sous programme recherche, pour chaque sommet de niveau 3 (x3) de la question, un sommet du dossier (x3d) de même étiquetage. Il vérifie également que leurs prédécesseurs respectifs se correspondent et ont même étiquetage et que tout successeur de x3q appartient à la famille des successeurs de x3d. En tout sommet, il évalue l'expression booléenne de critères complémentaires et les instructions de rangement.

PARAMETRES FORMELS

NIV indique le niveau du descripteur recherché (donné)
NCØDE indique au sous programme le code du descripteur
NRANG paramètre de sortie, contient le rang dans NIi du descripteur (dossier)
décode NCØDE et de niveau i
NRES=1 si l'examen du dossier est pertinent, 0 sinon

Ce sous programme

UTILISE

1) Les tableaux NI1, NI2, NI3, NI4

Les paramètres suivants, transmis en COMMON :

- . INI3 : indique le rang dans NI3 à partir duquel la recherche d'un code numérique est effectuée
- . NNEI3 et HPEI3 (voir sous programme EVAL)
- . ISUP3 = sup (des rangs des codes numériques de niveau 3 dans NI3 avant le réarrangement)
- . ISUP3=INI3 avant tout examen ou toute nouvelle itération d'examen de dossier
- . INI3=ISUP3 après tout examen ou toute nouvelle itération d'examen de dossier

- . NPEX : pointeur indiquant le début dans EXCRI de l'expression booléenne de critères complémentaires
- . NPINST pointeur indiquant le début dans INST de la liste d'instructions de rangement

2) - Le sous programme EVALBCOL(NTRAV, IPOINT, INAT, NRES)

- Les sous programmes ACTINS(INST, NPINST) évaluant la liste des instructions de rangement éventuellement associée à un sommet et RECDIC (NI3, NCØDE, NRANG, IRG3, INI3, NB) où

- . NCØDE indique, au sous programme le code numérique dont on recherche le rang
- . NRANG contient le rang dans NI3 du code numérique
- . IRG3 contient le rang dans NI3 du code numérique avant le réarrangement du tableau
- . NB=0 si le code n'existe pas dans NI3, sinon 1

```

TRAITEMENT   si NIV=3      alors
                RECDIC(NI3, NCØDE, NRANG, IRG3, INI3, NB)
                si NB = 0 et NCØDE < 0 alors aller a éta
                si NB = 1 et NCØDE > 0 alors aller a éta
                sinon NRES = 0 aller a fin sous programme
                éta : . si IRG3 > ISUP3 alors ISUP3 = IRG3
                    . aller a étii
  
```

```

                *
                si NIV=2 (où respectivement 1) alors
                si NCØDE = 0 ou NCØDE = NI2(NRANG, 1)
                (respectivement NCØDE = NI1 (NRANG, 1)) alors aller a étii
                sinon NRES = 0 aller a fin sous programme
  
```

* nous avons condensé l'écriture pour éviter une description trop lourde.

si* NIV = 4 alors

si NCØDE < 0 et NCØDE ≠ NI4(i,1)

(i=MPFI3 pas 1 jusqa MDFI3) aller a éti 1

si NCØDE > 0 et NCØDE ∈ NI4(i,1)

(i=MPFI3 pas 1 jusqa MDFI3) aller a éti 1

sinon NRES = 0 ; fin sous programme

éti 1 : .si le pointeur NPEX (début dans EXCRI de l'ex-
pression booléenne de critère complémentaire ≠ 0

alors

EVALBOOL(EXCRI, NPEX, 2, NRES)

. si NRES = 0 aller a fin sous programme

.sinon si le pointeur NPINST (début dans INST de
la liste d'instructions de rangement ≠ 0 alors

ACTINS(INST, NPINST) fin sous programme

* (voir page précédente)

ANNEXE 15

SOUS PROGRAMME ACINS (INST, NPINST)
=====

Ce sous-programme effectue la liste des instructions de rangement éventuellement accolée à un sommet.

INST est le tableau dans lequel sont rangées les listes d'instructions de rangement

NPINST pointe vers le début d'une liste d'instructions de rangement.

UTILISE mes tableaux NV, RV et NPIL

TRAITEMENT Pour chaque instruction de rangement de la liste faire

- appel de la fonction de recherche dont le code est dans INST (NPINST)
- si INST (NPINST + 2) > 0 (indicateur de variable) alors
 - . ranger l'information recherchée dans NV ou RV au rang INST (NPINST + 1) (le type est indiqué dans INST(NPINST+1))
- sinon NR1 = - INST (NPINST + 2)
 - . recherche de l'adresse dans NV ou RV du premier élément de la pile : NR2 = NPIL (NR, 2)
(le type est indiqué dans INST (NPINST + 1))
 - . ranger l'information au rang NV (NR2) ou RV (NR2)
 - . incrémenter d'un NV (NR2) ou RV(NR2)

SOUS PROGRAMME RECELEM (NRDIM, NELEM, NBR)

Ce sous programme détermine, pour une dimension, rangée dans NRDIM au rang NRDIM, les caractères pertinents. Il transmet au programme l'activant le nombre de ces caractères (NBR) et leurs numéros (tableau NELEM).

- UTILISE
- Les tableaux NRDIM, NPAR, PAR, NPIL, NV et RV et CRIT ;
 - Un pointeur IPC indiquant l'adresse dans NPAR ou PAR du début de la définition des partitions ;
 - Trois tableaux de travail : un de type entier ICAR, l'autre de type réel RCAR destiné à recevoir la valeur de la variable ou les valeurs des éléments de la pile associée à la caractéristique et un troisième CRIT utilisé lors de l'évaluation des expressions booléennes constituant les caractères booléens (nous précisons l'intérêt de ce tableau au cours de la phase traitement) ;
 - Une pile de travail NPT (h) permettant l'évaluation des expressions booléennes ;
 - Le sous programme VALAGE (date) (âge à une date donnée).

- TRAITEMENT
- si la caractéristique est représentée par la fonction AGE (paramètre) alors
 - si le paramètre est une variable date alors
 RCAR (1) = VALAGE (date)
 - sinon pour chaque élément de la pile paramètre faire
 RCAR (i) = VALAGE (élément de pile)
 - sinon si la caractéristique est représentée par une variable ou une pile de type réel alors
 ranger la variable dans RCAR(1) ou ranger chaque élément de la pile dans RCAR(i)
 - sinon ranger la variable dans ICAR(1) ou ranger chaque élément de la pile dans ICAR(i)

si le type de partition est numérique régulière
(NDIM (NRDIM, 3) = 1) alors

pour chaque élément ICAR(i) ou RCAR (i) (selon le type de l'identificateur) faire

NELEM(i) = (((ICAR(i) - NPAR(IPC))/(NPAR(IPC + 2)) + 1

ou

NELEM(i) = partie entière (((RCAR(i) - PAR(IPC))/
(PAR(IPC + 2)) + 1)

NBR = NBR + 1 ; aller a fin sous programme

si le type de partition est numérique irrégulière
(NDIM (NRDIM, 3) = 2) alors

NTAIL = NDIM (NRDIM, 2)

pour chaque élément ICAR(i) (resp. RCAR(i)) faire

pour k = IPC pas 1 jusqu'à IPC + NTAIL - 2 faire

si ICAR(i) (resp. RCAR(i)) \geq NPAR(h) (resp. PAR(h)

et ICAR(i) (resp. RCAR(i)) < NPAR(h+1)

(resp. PAR (h + 1) alors

NELEM(i) = k aller a éti

éti : NBR = NBR + 1

aller a fin sous programme

si le type de partition est lexicographique
(NDIM(NRDIM, 3) = 3) alors

NTAIL = NDIM (NRDIM, 2)

si la caractéristique est représentée par la fonction SEXE alors

si valeur de SEXE = NPAR (IPC) alors

NELEM (i) = 1 sinon NELEM (i) = 2

NBR = 1 ; aller a fin sous programme

sinon si NTAIL ≤ 5 alors

pour chaque élément ICAR(i) faire

- . recherche séquentielle dans NPAR (du rang IPC au rang IPC + NTAIL - 1)
- . si l'élément existe dans NPAR alors
 NELEM (i) = k (rang de l'élément dans la liste)
- . sinon NELEM (i) = NTAIL + 1

aller a fin sous programme

sinon

pour chaque élément ICAR(i) faire

- . recherche dichotomique dans NPAR (du rang rang IPC + NTAIL - 1)
- . si l'élément existe dans NPAR alors
 NELEM (i) = k (rang de l'élément dans la liste)
- . sinon NELEM (i) = NTAIL + 1

aller a fin sous programme

si le type de partition est booléen

(NDIM (NRDIM, 3) = 4) alors

Commentaire : nous avons vu (chapitre IV) que chaque caractère booléen est une expression booléenne de critères numériques ou lexicographiques.

Un premier membre d'un critère peut être, soit un identificateur de variable, soit un identificateur de pile, soit un nom de fonction.

Pour ne pas avoir à réévaluer des critères constants (c'est-à-dire ceux dont le premier membre est un identificateur de variable ou AGE (date)), nous procédons, lors de l'évaluation du caractère, de la façon suivante :

- 1) Lors de chaque évaluation de caractère, nous recopions la traduction post fixée de l'expression booléenne dans un tableau CRIT, en remplaçant chaque "critère constant" par sa valeur (1 si vraie, 0 si fausse)
- 2) S'il existe, dans chaque caractère, au moins un critère non constant (le premier membre est un identificateur de pile), nous évaluons les expressions booléennes à partir du tableau CRIT, autant de fois que les piles (nécessairement parallèles) ont d'éléments

pour chaque caractère booléen de la liste faire

pour chaque élément de l'expression booléenne

(caractère) faire

si c'est un opérande (critère) alors

si le 1er membre est une variable alors

- . évaluer le critère ;
- . ranger sa valeur dans NCRIT ;
NPCR = IPCR + 1 ;
- . ranger le résultat de l'évaluation (0 si faux, 1 si vrai) sur la pile NPT(h) ($h = h + 1$)

sinon. évaluer le critère en prenant comme premier membre le premier élément de la pile ;

- . recopier le critère dans NCRIT ;
IPCR = IPCR + 5
- . ranger le résultat de l'évaluation sur la pile NPT (h) ;
 $h = h + 1$

si c'est un opérateur alors

. dépiler NPT (h) (h = h - 1)
ranger le résultat dans MOT 1

. dépiler NPT (h) (h = h - 1)
ranger le résultat dans MOT 2

si c'est l'opérateur ou alors

NPT(h) = MOT 1 + MOT 2 (h=h+1)

sinon NPT (h) = MOT 1 × MOT 2
(h=h+1)

sinon si marque de fin de caractère alors

si NPT (h) ≠ 0 alors

NELEM (i) = k (rang du caractère
dans la liste) ;

NBR = 1 aller a éti

éti : fin pour chaque

pour chaque élément des piles apparaissant dans les

critères faire

pour chaque caractère booléen de la liste NCRIT

faire

. évaluer le caractère

s'il est pertinent alors

NELEM (i) = k (rang du caractère) ;

aller a éti

NBR = NBR + 1

éti : fin pour chaque

fin sous programme

B I B L I O G R A P H I E

- (1) DERNIAME J.C. et son équipe
Séminaires 1970-1971 de l'I.U.C.A. sur le projet CIVA
- (2) GIRARD T. (Datar)
Le système d'accumulation des données
- (3) BOUSSINEAU, CHARNIAUX, TARDIEU (C.E.T.E.)
Le système MIIS FIIT
- (4) FISKUS (D.R.M.E.)
Système informatique pour banque de donnée
- (5) MARTIN J.M.
Un mode d'exploitation du dossier médical : structure arborescente, système modulaire d'interrogation (thèse de troisième cycle)
- (6) DUFOURD J.F.
Traitement automatique de la gestion scolaire d'une université, réalisation et extensions prévues
- (7) ROY B.
Algèbre moderne et théorie des graphes (Dunod, 1970)
- (8) CHANG C.L. et SLOGLE
An admissible and optimal algorithm for searching AND/OR graphs

NOM DE L'ETUDIANT : GERMAIN Patrick

Nature de la thèse : Doctorat de Spécialité en Mathématiques Appliquées



Vu, Approuvé

et permis d'imprimer

NANCY, le 28-11-1972

Le Président du Conseil de l'Université de NANCY I

A handwritten signature in black ink, which appears to be "J.R. HELLUY", is written over a circular stamp. The stamp contains the text "UNIVERSITE DE NANCY I" and "FACULTE DE SCIENCES".

J.R. HELLUY