Conception d'un système automatisé de gestion de la scolarité de l'enseignement supérieur

Dominique Portal

THESE

présentée à

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE
INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

POUR OBTENIR LE GRADE DE
DOCTEUR INGENIEUR

Dominique PORTAL

Conception d'un système automatisé de gestion de la scolarité de l'enseignement supérieur.

Soutenue le 20 mars 1975 devant la Commission d'Examen

Président : N. GASTINEL

Examinateurs C. DELOBEL
J. KOULOUMDJIAN
J.R. ABRIAL

Invité : F. MONTAGNAT

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE		INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE
M. Michel SOUTIF	Présidents	M. Louis NEEL
M. Gabriel CAU	Vice-Présidents	MM. Lucien BONNETAIN Jean BENOIT

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BEAUDOING André	Clinique de Pédiatrie et Puériculture
	BERNARD Alain	Mathématiques Pures
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BEZES Henri	Pathologie chirurgicale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BOUCHERLE André	Chimie et toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARLIER Georges	Biologie végétale
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Clinique Oto-Rhino-Laryngologique
	CHATEAU Robert	Thérapeutique (Neurologie)
	CHIBON Pierre	Biologie animale
	COEUR André	Pharmacie chimique et chimie analytique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie pathologique
	CRAYA Antoine	Mécanique
Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBERMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumo-Phtisiologie
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée

MM.	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DRUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de dermatologie et syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques pures
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Mathématiques appliquées
	GAVEND Michel	Pharmacologie
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	KAHANE André	Physique Générale
	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LOUP Jean	Géographie
Mle	LUTZ Elisabeth	Mathématiques pures
	MALGRANGE Bernard	Mathématiques pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	MULLER Jean-Michel	Thérapeutique (néphrologie)
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-chirurgie
	SEIGNEURIN Raymond	Microbiologie et hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction mécanique
	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
MM.	VERAIN André	Physique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	CHEEKE John	Thermodynamique
	COPPENS Philip	Physique
	CORCOS Gilles	Mécanique
	CRABBE Pierre	CERMO
	GILLESPIE John	I.S.N.
	ROCKAFELLAR Ralph	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

Mlle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BERTRANDIAS Jean-Paul	Mathématiques pures
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
Mme	BONNIER Jane	Chimie générale
MM.	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique
	CONTE René	Physique
	DEPASSEL Roger	Mécanique des fluides
	GAUTHIER Yves	Sciences biologiques
	GAUTRON René	Chimie
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	GROULADE Joseph	Biochimie médicale
	HACQUES Gérard	Calcul numérique
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Méd. Préventive
	IDELMAN Simon	Physiologie animale
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
Mme	KAHANE Josette	Physique
MM.	KUHN Gérard	Physique
	LOISEAUX Jean	Physique nucléaire
	LUU-DUC-Cuong	Chimie organique
	MAYNARD Roger	Physique du solide
	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	PFISTER Jean-Claude	Physique du solide
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	Mathématiques appliquées
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	RICHARD Lucien	Biologie végétale
Mme	RINAUDO Marguerite	Chimie macromoléculaire
MM.	ROBERT André	Chimie papetière
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
MM.	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	AMBLARD Pierre	Dermatologie
	ARMAND Gilbert	Géographie
	ARMAND Yves	Chimie
	BARGE Michel	Neurochirurgie
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamique
M.	BOUCHARLAT Jacques	Psychiatrie adultes
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BRODEAU François	Mathématiques (IUT B)
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie papetière
	CHIAVERINA Jean	Biologie appliquée (EFP)
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CORDONNIER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	CYROT Michel	Physique du solide
	DELOBEL Claude	M.I.A.G.
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie végétale
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	FONTAINE Jean-Marc	Mathématiques pures
	GAUTIER Robert	Chirurgie générale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROS Yves	Physique (stag.)
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	KOLODIE Lucien	Hématologie
	KRAKOWIAK Sacha	Mathématiques appliquées
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LEROY Philippe	Mathématiques
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)
	MICHOULIER Jean	Physique (IUT A)
Mme	MINIER Colette	Physique
MM.	NEGRE Robert	Mécanique
	NEMOZ Alain	Thermodynamique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PERRET Jean	Neurologie
	PHELIP Xavier	Rhumatologie
	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RAMBAUD Pierre	Pédiatrie
Mme	RENAUDET Jacqueline	Bactériologie
MM.	ROBERT Jean-Bernard	Chimie-Physique

MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VROUSOS Constantin	Radiologie

MAITRES DE CONFERENCES ASSOCIES

MM.	COLE Antony	Sciences nucléaires
	FORELL César	Mécanique
	MOORSANI Kishin	Physique

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

MM.	BOST Michel	Pédiatrie
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	FAURE Gilbert	Urologie
	MALLION Jean-Michel	Médecine du travail
	ROCHAT Jacques	Hygiène et hydrologie

Fait à Saint Martin d'Hères, OCTOBRE 1974.

"MEMBRES DU CORPS ENSEIGNANT DE L'I.N.P.G."

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Radioélectricité
	BESSON Jean	ELectrochimie
	BONNETAIN Lucien	Chimie Minérale
	BONNIER Etienne	Electrochimie, Electrométallurgie
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	ELectronique
	COUMES André	Radioélectricité
	FELICI Noël	Electrostatique
	PAUTHENET René	Physique du solide
	PERRET René	Servomécanismes
	SANTON Lucien	Mécanique
	SILBER Robert	Mécanique des fluides

PROFESSEUR ASSOCIE

M.	BOUDOURIS Georges	Radioélectricité

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du solide et cristallographie
	COHEN Joseph	Electrotechnique
	DURAND Françis	Metallurgie
	MOREAU René	Mécanique
	POLOUJADOFF Michel	ELectrotechnique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM.	BOUVARD Maurice	Génie mécanique
	CHARTIER Germain	Electronique
	FOULARD Claude	Automatique
	GUYOT Pierre	Chimie minérale
	JOUBERT Jean Claude	Physique du solide
	LACOUME Jean Louis	Géophisique
	LANCIA Roland	Physique atomique
	LESPINARD Georges	Mécanique
	MORET Roger	Electrotechnique nucléaire
	ROBERT François	Analyse numérique
	SABONNADIERE Jean Claude	Informatique fondamentale et appliquée
Mme	SAUCIER Gabrièle	Informatique fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M.	LANDAU Ioan Doré	Automatique

CHARGE DE FONCTIONS DE MAITRES DE CONFERENCES

M.	ANCEAU François	Mathématiques appliquées

Fait à Saint Martin d'Hères JANVIER 1974

Je tiens à remercier,

Monsieur le Professeur Noël GASTINEL, Directeur du Centre Interuniversitaire de Calcul qui m'a fait l'honneur de présider le Jury de cette thèse.

Monsieur Claude DELOBEL, Maître de Conférences à l'Université Scientifique et Médicale de Grenoble qui m'a conseillé et guidé dans mon travail.

Monsieur Jacques KOULOUMDJIAN, Maître de Conférences à l'Université Claude Bernard de Lyon pour l'aide qu'il m'a apportée par ses conseils et ses critiques.

Monsieur Jean-Raymond ABRIAL, Ingénieur à l'I.R.I.A., qui en m'accueillant dans son équipe m'a initié à la recherche et m'a apporté la formation sans laquelle ce travail n'aurait pas été possible.

Monsieur François MONTAGNAT, Secrétaire Général de l'Université Scientifique et Médicale de Grenoble qui par sa conception éclairée de la gestion et la totale confiance qu'il m'a accordée m'a permis de mener à bien ce travail de recherche et son application.

Monsieur Michel WOLFF qui a réalisé l'analyse des filières d'études et a rendu possible le dialogue indispensable entre l'informaticien et les gestionnaires administratifs.

Monsieur Jean-François GRABOWIECKI, Mademoiselle Yolande FOITY, Madame Christiane TOUCHAIN qui, en assurant l'exploitation dans des conditions parfois difficiles, ont contribué en grande partie au succès de la réalisation.

Madame Micheline TREVISAN, Monsieur Samuel RASOLONJATOVO ainsi que Monsieur Daniel IGLESIAS et les membres de son service qui ont réalisé matériellement cet ouvrage.

Enfin, j'aimerais qu'il me soit permis de remercier le prototype SOCRATE qui a permis de réaliser et d'exploiter la base de données des enseignements et à travers lui tous ceux qui ont participé à sa conception : J. BAS, G. BEAUME, J.P. CAHEN, G. HENNERON, J.C. FAVRE, G. MAZARE, R. MORIN, A. STIERS et G. VIGLIANO.

Dominique Portal

L'I.R.I.A. a contribué matériellement, au titre du contrat CRI 73051, à la réalisation du système de gestion de scolarité qui fait l'objet de cette thèse.

A mon grand-père et à son fils.

1. INTRODUCTION

Les Universités de Grenoble utilisent pour la gestion de leurs services de scolarité un fichier exploité sur ordinateur et contenant des informations relatives aux étudiants et aux études qu'ils poursuivent dans chacune d'elles.

Notre travail a consisté à développer les possibilités d'exploitation de ce fichier ce qui nous a conduit à mettre en œuvre un fichier des enseignements organisés par ces Universités.

Cette thèse fait état de nos efforts pour essayer de définir rationnellement la structure de ce fichier et les procédures qui auront à l'exploiter.

Pour bien situer la nature du problème posé, il importe de présenter en premier lieu la gestion des étudiants telle qu'elle était réalisée en octobre 1972, date à laquelle notre étude a commencé.

Plutôt que de faire cette présentation de façon informelle, nous utiliserons les notations proposées dans [1] pour décrire la sémantique d'une base de données : elles nous conduisent à plus de rigueur dans l'exposé et seront de plus utiles dans les chapitres suivants (cf. chapitre 5).

1.1. Notations utilisées pour la description d'une application de gestion

Nous reprendrons donc les notations introduites dans [1].

Il s'agit de définir les objets pris en compte par les services gestionnaires (services de scolarité) et les relations qui existent entre ces objets.

A titre d'exemple, introduisons l'ensemble des personnes physiques, noté "individu", et l'ensemble des trois Universités grenobloises pour lesquelles la gestion est effectuée; appelons "établissement" cet ensemble :

individu = {DUPONT,DURAND,MARTIN,...}
établissement = {U1,U2,U3}

L'individu DUPONT peut faire ses études dans l'Université U1; nous dirons qu'il existe une relation entre individu et établissement.

Nous désignerons cette relation en nommant les deux fonctions d'accès qui permettent, connaissant un individu, d'atteindre le ou les établissements dans lesquels il fait ses études et inversement, connaissant un établissement, d'atteindre le ou les individus qui y sont étudiants.

Nous écrirons par exemple :

étudiant(U1) = {DUPONT,MARTIN,...}
étudiant(U2) = {DURAND,MARTIN,...}
études-à(DUPONT) = {U1}
études-à(MARTIN) = {U1,U2}

Il est clair que les fonctions d'accès peuvent être multivaluées, aussi préciserons-nous le cardinal maximum et le cardinal minimum des ensembles de valeurs qu'elles peuvent prendre. Un individu peut ne faire d'études dans aucune des Universités grenobloises ou au maximum dans les trois simultanément. Dans une université il y a au moins un étudiant; et comme il n'y a pas de contraintes sur le nombre maximum, nous le noterons par abus ∞.

Finalement, nous écrirons :

r1 = rel(individu,établissement,études-à=facc(0,3),
étudiant = facc(1,∞))

et nous utiliserons la représentation suivante :

individu ← (1,∞) étudiant   établissement

études-à (0,3) →

Remarque : Les deux fonctions d'accès étant inverses l'une de l'autre nous pourrons dans certains cas ne donner de nom qu'à l'une d'elles et faire usage alors de l'opérateur inv appliqué à celle qui est nommée pour désigner celle qui ne l'est pas.

Enfin, il nous faut retenir qu'une fonction d'accès peut être ou ne pas être enregistrée sur un support exploitable par le service gestionnaire (par l'intermédiaire d'un ordinateur ou de tout autre moyen). Ainsi dans l'exemple évoqué chaque Université disposant d'un fichier de ses étudiants la fonction d'accès "étudiant" est exploitable.

Par contre, un même individu étant identifié de façon indépendante sur chaque fichier une Université ne peut facilement savoir si l'un de ses étudiants poursuit des études dans une autre Université : la relation "études-à" n'est donc pas exploitable.

Nous avons désormais les éléments qui vont nous permettre de présenter globalement la gestion de la scolarité telle qu'elle fonctionnait en octobre 1972. Ce type de gestion n'est d'ailleurs pas particulier aux Universités de Grenoble, mais est à ce jour utilisé avec des moyens matériels variés par beaucoup d'Universités françaises.

1.2. Présentation de la gestion traditionnelle des étudiants

Grossièrement, on peut dire que la tâche des services de scolarité consiste en :

- l'identification et la description des étudiants,
- la description et la validation des cursus universitaires.

Cette tâche a sa place dans le cadre général que constitue : l'organisation pédagogique de l'Université.

1.2.1. Identification et description des étudiants

Lors de sa première inscription dans un établissement, un individu se voit attribuer un "numéro d'étudiant" qui sert à l'identifier. Ce numéro est repris lors des inscriptions ultérieures que l'individu pourra reprendre dans le même établissement.

Désignons par "no-étudiant" l'ensemble des numéros d'étudiants utilisables (ce peut-être l'ensemble des entiers de 0 à 99999 par exemple).

Par définition un "no-étudiant" n'identifie aucun individu s'il n'a pas encore été affecté, et en identifie au plus un s'il l'a été.

Si un individu s'inscrit simultanément ou successivement dans différentes Universités, chacune lui délivre un numéro différent dans une tranche qui lui est propre.

Donc un même individu peut avoir jusqu'à trois numéros d'étudiant différents qui lui servent d'identificateur.

On note donc :

r1 = <u>rel</u> (individu,no-étudiant,identificateur = <u>facc</u>(0,3),
identifié = <u>facc</u>(0,1))

L'établissement qui délivre le numéro d'étudiant en est l'unique assignateur, et il n'y a pas de contrainte sur le nombre de numéros délivrés qui chaque année s'accroît du nombre des nouveaux inscrits dans l'établissement.

On note donc :

r2 = <u>rel</u> (no-étudiant,établissement,assignateur = <u>facc</u>(0,1),
<u>facc</u>(1,∞))

Un individu peut n'être inscrit dans aucun des trois établissements. Rien se s'oppose à ce qu'un étudiant fréquente tour à tour les trois Universités de Grenoble.

Enfin, il n'y a pas de contrainte sur le nombre d'étudiants inscrits dans un établissement.

On note donc :

r3 = <u>rel</u> (individu,établissement,études-à = <u>facc</u>(0,3),
étudiant = facc(1,∞))

On a relevé que les fichiers d'étudiants de chacune des trois Universités sont indépendants, ce qui entraîne la possibilité pour un même individu d'avoir plusieurs numéros d'étudiant. Autrement dit la gestion porte sur les personnes étudiantes, c'est-à-dire enregistrées sous un numéro dans un établissement et non sur les personnes physiques.

On conçoit dans ces conditions la difficulté qu'il y a à gérer les cursus pluridisciplinaires (suivis simultanément dans plusieurs Universités).

Pour l'instant retenons que les fonctions d'accès "études-à" et "identificateur" ne sont pas exploitables.

Voyons maintenant les informations qui décrivent les étudiants, ce sont les suivantes :

nom, prénom, initiale du second prénom, nom du conjoint, jour de naissance, numéro d'INSEE,

ainsi qu'un certain nombre de renseignements utiles au Ministère de l'Education pour ses statistiques.

Nous ne noterons pas toutes les relations correspondantes, définissons cependant celles qui lient un individu à son numéro d'INSEE et à son nom de famille.

Désignons par "nombre" l'ensemble des entiers que l'on peut écrire avec treize chiffres. Un élément de "nombre" est le numéro d'INSEE d'au plus un individu et peut être celui d'aucun. Inversement, chaque individu, a au plus un numéro d'INSEE, et certains, étrangers notamment, n'en ont pas.

D'où :

r4 = <u>rel</u> (nombre,individu,<u>facc</u>(0,1),no-insee = <u>facc</u>(0,1))

Désignons par "nom" l'ensemble des chaînes de caractères; certaines ne représentent le nom d'aucun individu, d'autres représentent le nom d'un ou de plusieurs (cas d'homonymie) individus. Inversement chaque individu n'a qu'un nom (de famille).

Notons donc :

r5 = <u>rel</u>(nom,individu,<u>facc</u>(1,∞),nom-de-famille = <u>facc</u>(1,1))

On voit dans ces deux cas pourquoi les services administratifs sont obligés d'affecter aux individus pour les identifier un numéro d'étudiant. Aucun autre identificateur, propre à l'individu et sûrement discriminant, n'existe quand celui-ci se présente pour s'inscrire.

1.2.2. Description des cursus universitaires

Un étudiant inscrit dans une université y prépare des diplômes, pour cela il suit des enseignements.

Introduisons deux relations ternaires :

Soient x un numéro d'étudiant, y un diplôme, z un enseignement et t une année.

préparé(x,y,t) ↔ l'étudiant x prépare le diplôme y en l'année t

suivi(x,z,t) ↔ l'étudiant x suit l'enseignement z en l'année t

Pour définir ces deux relations avec les notations de [1], introduisons "diplôme" l'ensemble des diplômes, "enseignement" l'ensemble des enseignements, "préparé" l'ensemble des triplets {no-étudiant, diplôme, année} et "suivi" l'ensemble des triplets {no-étudiant, enseignement, année}. On note alors :

r6 = rel(préparé,diplôme,diplôme-préparé = facc(1,1),facc(0,∞))
r7 = rel(préparé,année,date-préparé = facc(1,1),facc(0,∞))
r8 = rel(préparé,no-étudiant,postulant = facc(1,1),facc(0,∞))
r9 = rel(suivi,enseignement,enseignement-suivi = facc(1,1),facc(0,∞))
r10 = rel(suivi,année,date-suivi = facc(1,1),facc(0,∞))
r11 = rel(suivi,no-étudiant,inscrit = facc(1,1),facc(0,∞))

Au terme des sessions d'examens on détermine les étudiants qui ont obtenu avec succès les enseignements qu'ils suivaient. Afin de ne pas alourdir l'exposé, nous ne faisons pas la différence entre les enseignements acquis en juin (première session d'examens) et ceux acquis en septembre (deuxième session d'examens). De la même façon nous ne tenons pas compte des mentions.

Définissons donc les deux nouvelles relations ternaires :

acquis(x,y,t) ↔ l'étudiant x a acquis l'enseignement y en l'année t
obtenu(x,z,t) ↔ l'étudiant x a obtenu le diplôme z en l'année t

introduisons les ensembles de triplets "acquis" = {no-étudiant, enseignement, année} et "obtenu" = {no-étudiant,diplôme,année}.

On note alors :

r12 = rel(acquis,enseignement,enseignement-acquis = facc(1,1),facc(0,∞))
r13 = rel(acquis,année,date-acquis = facc(1,1), facc(0,∞))
r14 = rel(acquis,no-étudiant,acquis-par = facc(1,1),facc(0,∞))
r15 = rel(obtenu,diplôme,diplôme-obtenu = facc(1,1),facc(0,∞))
r16 = rel(obtenu,année,date-diplôme = facc(1,1),facc(0,∞))
r17 = rel(obtenu,no-étudiant,diplômé = facc(1,1),facc(0,∞))

Toutes ces relations sont enregistrées sur le fichier étudiant, et mises à jour au terme de chaque session d'examens.

Bien évidemment, l'obtention des diplômes n'est pas indépendante de l'obtention des enseignements. Des réglements définissent quels enseignements il faut acquérir pour obtenir chaque diplôme, ils sont décrits dans le "livret de l'étudiant".

Dans certains cas, un jury délibère de la délivrance du diplôme , dans d'autres celui-ci constitue un agglomérat d'enseignements sanctionnés indépendamment, on peut alors appliquer automatiquement le règlement aux composants du diplôme.

On voit apparaître un axe de ce qui sera notre travail :

Est-il possible d'enregistrer les règlements de façon à effectuer automatiquement le "calcul" de la délivrance d'un diplôme à partir des enseignements qui le composent ?

1.2.3. Organisation des Universités

Chaque Université est composée d'Unités d'Enseignement et de Recherche qui organisent les enseignements et les diplômes.

Appelons "uer" l'ensemble de ces unités et notons la relation qui les relie à un établissement :

r18 = <u>rel</u>(établissement,uer,constituant = <u>facc</u>(1,∞),<u>facc</u>(1,1))

Chaque diplôme est organisé dans le cadre d'une UER.
On note donc :

r19 = <u>rel</u>(uer,diplôme,<u>facc</u>(1,∞),organisatrice = <u>facc</u>(1,1))

Les enseignements sont dispensés dans le cadre d'une UER, mais il se peut que, pour la composition d'un diplôme qu'elle organise, une UER utilise un enseignement dispensé par une autre.

Notons donc :

r20 = rel(uer,enseignement,facc(1,∞),dispensatrice = facc(1,1))

r21 = rel(uer,enseignement,facc(0,∞),utilisatrice = facc(0,∞))

Pour résumer ce qui précède, donnons-en le schéma suivant : (cf. figure 1.1).

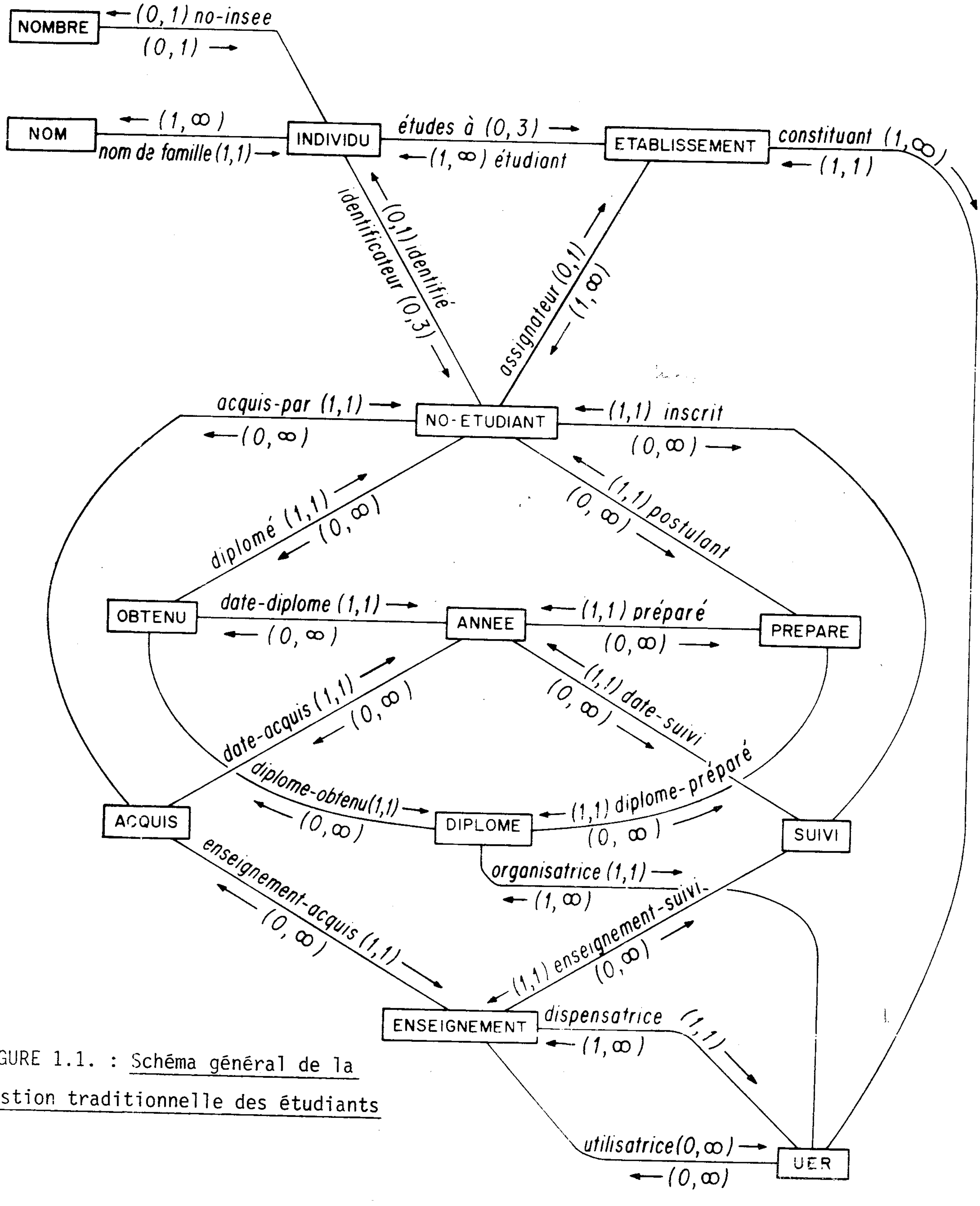


FIGURE 1.1. : Schéma général de la gestion traditionnelle des étudiants

1.3. Améliorations possibles

De la description précédente deux points susceptibles d'améliorations peuvent être dégagés. Le premier porte sur la gestion des individus poursuivant des études dans plusieurs établissements et relève plus de l'organisation globale de la gestion que de la technique informatique utilisée. Le second point porte sur la description de l'organisation des enseignements et diplômes qui nécessite pour être automatisée des outils informatiques nouveaux.

1.3.1. Intérêt d'une gestion commune aux trois Universités

La situation des trois Universités grenobloises sur un même site a naturellement favorisé les contacts entre elles, avec pour conséquence l'organisation d'enseignements pluridisciplinaires. Il est de plus vraisemblable que dans un avenir proche ce type d'enseignement aura à se développer.

Par ailleurs, chacune des Universités réalise sur le même matériel dans les installations du Centre Interuniversitaire de Calcul, ses travaux de gestion, qui, bien évidemment, répondent aux mêmes besoins. Il paraît illogique dans ces conditions de traiter indépendamment les fichiers d'étudiants de chacun des trois établissements.

Il est par contre souhaitable de constituer un fichier unique créant ainsi des conditions d'exploitation communes.

Ce regroupement doit être l'occasion de réaliser non plus la gestion des personnes étudiantes, c'est-à-dire enregistrées sous un numéro par un établissement, mais la gestion des personnes physiques. Pour cela il importe d'identifier chaque individu inscrit dans une des Universités par un code unique reconnu par les autres établissements.

On a vu que le numéro d'INSEE n'est malheureusement pas utilisable à cette fin, les étudiants étrangers ne l' ayant pas toujours.

La seule solution est donc que la première Université dans laquelle s'inscrit un individu lui attribue un numéro d'identification, et que toute inscription ultérieure dans cet établissement ou dans un autre soit faite sous ce même numéro.

Comme seul l'individu qui vient s'inscrire sait s'il a déjà pris une inscription à Grenoble, il faut l'inciter à le signaler lors des inscriptions ultérieures, par exemple en allégeant au maximum les procédures correspondantes. Ceci n'exclut pas de mettre en œuvre des moyens techniques pour détecter les individus qui essaieraient sciemment, et peut-être à des fins frauduleuses, de prendre plusieurs fois une première inscription.

Ceci nous conduit à redéfinir comme suit les relations entre "établissement", "individu" et "no-étudiant" :

r'2 = rel(individu,no-étudiant,identificateur=facc(0,1),identifié=facc
identifié = facc(0,1))

r'3 = rel(no-étudiant,établissement,assignateur= facc(0,1),facc(1,∞))

r"3 = rel(no-étudiant,établissement,réutilisateur = facc(0,2),facc(0,∞))

On a le schéma correspondant suivant (figure 1.2) :

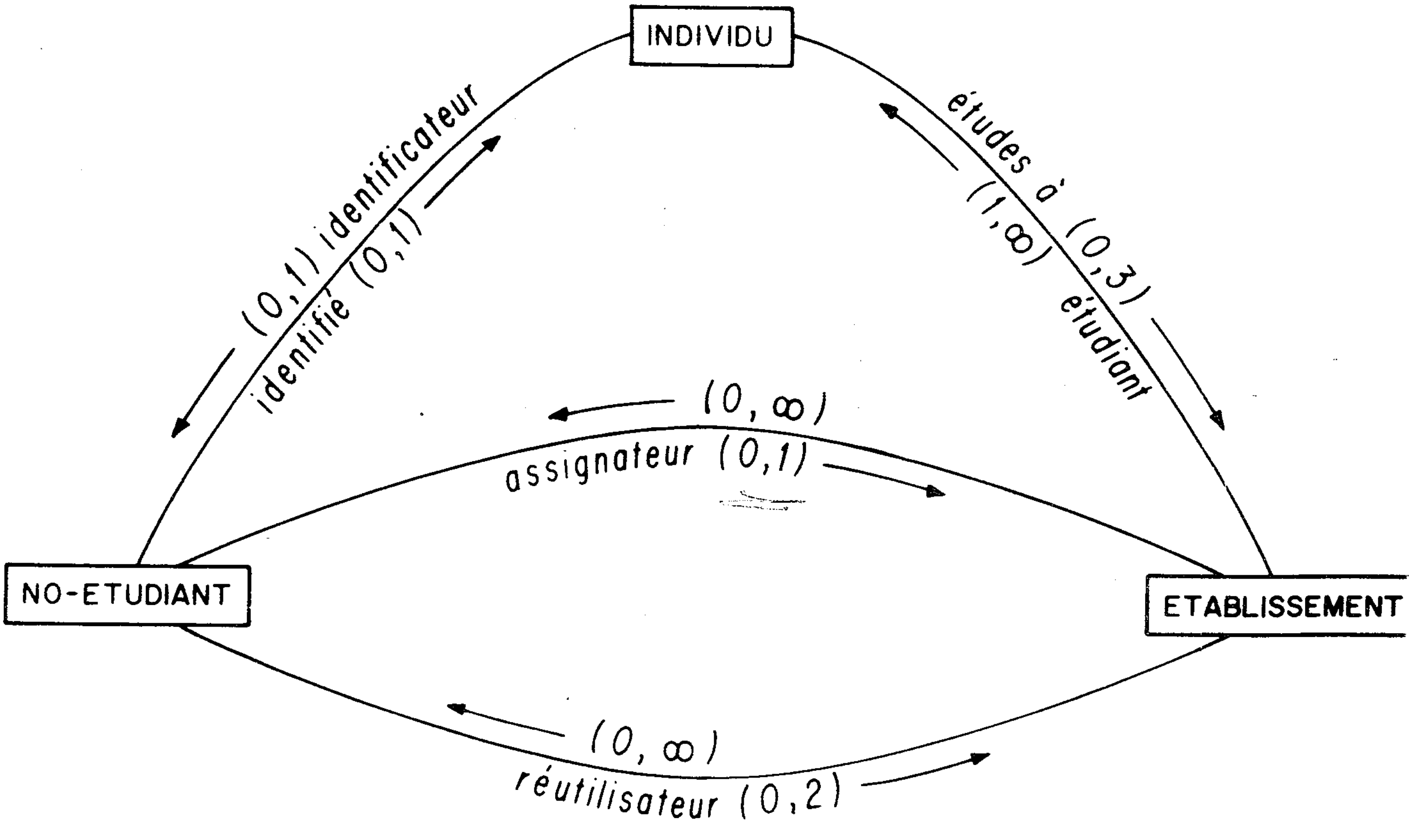


FIGURE 1.2. : Définition des relations entre "individu", "établissement" et "no-étudiant" dans l'optique d'une gestion commune aux trois Universités.

1.3.2. Nécessité d'une description rigoureuse de l'organisation des études

La mise en oeuvre d'un fichier unique des étudiants, regroupant plus de 25000 enregistrements, doit permettre outre les besoins statistiques de déterminer pour un étudiant donné :

- à quel point de son cursus il se trouve, c'est-à-dire quels sont les enseignements auxquels il a réussi ou échoué, quels sont les diplômes qu'il a acquis.
- à partir de ce point, quelles possibilités lui sont offertes.

Ceci implique de bien connaître les enseignements organisés et les filières d'études dans lesquelles ils interviennent Deux raisons font que ce n'est pas chose facile :

- Le découpage des enseignements en unités élémentaires de plus en plus petites, apparu avec la loi d'Orientation de l'Enseignement Supérieur en 1968, a augmenté le nombre d'enseignements dispensés dans chaque université. Pour les trois universités de Grenoble ce nombre dépasse 1500.

De pair avec cette augmentation quantitative, la complexité des règles de progression dans les différentes filières s'est accrue, ce qui est naturel puisque des combinaisons d'enseignements plus variées peuvent être offertes aux choix des étudiants.

- L'évolution rapide de l'organisation des études, due en grande partie aux réformes successives mises en place par le ministère de l'Education, fait que le (ou les) cursus d'un même étudiant peuvent se dérouler dans plusieurs organisations successivement ou même simultanément.

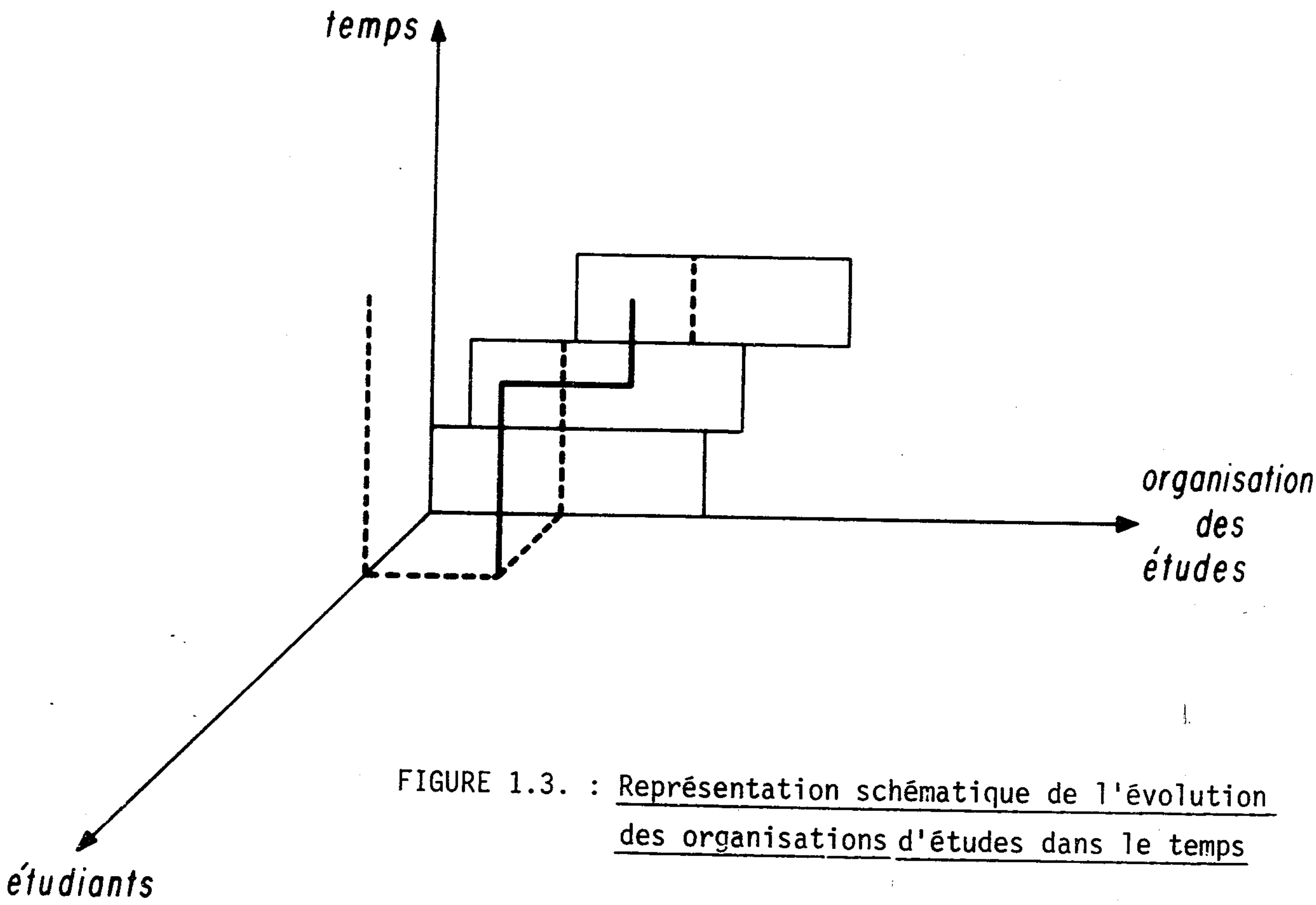


FIGURE 1.3. : Représentation schématique de l'évolution des organisations d'études dans le temps

Chaque pavé de la figure 1.3 représente un état de l'organisation des études dans le temps. S'il n'y avait pas d'évolution tous les pavés seraient superposés, mais du fait des changements on assiste dans le temps à la disparition d'un certain nombre de filières et à l'apparition de nouvelles. On a représenté pour un étudiant un cursus qui se déroule successivement dans deux types d'organisation.

On conçoit qu'alors il ne suffise plus d'enregistrer chaque année ce que fait chaque étudiant. Il importe de savoir contrôler l'enchaînement et l'évolution de ces données d'une année sur l'autre.

Une première solution peut servir à l'esprit : inclure dans les programmes de traitement du "fichier étudiant" les procédures correspondantes.

Cette solution n'est pas viable, d'une part ces procédures occuperaient un volume trop important, d'autre part toute évolution dans la structure des enseignements entraînerait de lourds efforts de reprogrammation dont on sait le peu d'efficacité.

D'où l'idée de constituer un catalogue non seulement des enseignements dispensés mais aussi des modalités de progressions dans les filières selon lesquelles ils interviennent.

Par rapprochement de ce catalogue et du fichier étudiant il doit être possible de satisfaire les deux points évoqués en tête de ce paragraphe.

Bien évidemment cela implique que ce catalogue, ou fichier des enseignements, et le fichier des étudiants sont les deux parties d'un tout conçu de façon homogène et qui pourrait être qualifié de système de gestion des scolarités. Telle a bien été la nature de notre démarche. Cependant, les problèmes soulevés par la conception du fichier des enseignements se sont avérés (à notre grande surprise, faut-il le dire ?) susceptibles d'être formalisés de façon extrêmement riche pour l'ensemble de l'application, aussi est-ce davantage sur ces problèmes que s'est orienté notre travail.

1.4. Plan de notre travail

Nous déterminerons d'abord un modèle général permettant de décrire les règles de progression dans les cursus universitaires.

Il s'appuie sur l'algèbre de Boole (cf. chapitre 3) et nous montrerons que ces règles peuvent être représentées par des fonctions booléennes croissantes.

Nous montrerons ensuite que les graphes ET/OU utilisés en intelligence artificielle pour décrire la décomposition d'un problème en sous-problèmes (cf. [2]) peuvent être utilisés dans notre cas pour représenter et pour évaluer ces fonctions booléennes (cf. chapitre 3).

Afin de fournir à l'utilisateur un moyen simple de décrire les règles de progression nous introduisons alors la notion de relation de dépendance. Cette notion, très analogue à celle de relation fonctionnelle (cf. [3]), nous donnera des éléments pour vérifier que ce qui est introduit dans le modèle est cohérent (cf. chapitre 4).

A ce stade le modèle utilisé aura donc été abordé sous trois angles :

- formalisation booléenne,
- représentation par des graphes ET/OU
- analyse des données à l'aide des relations de dépendances.

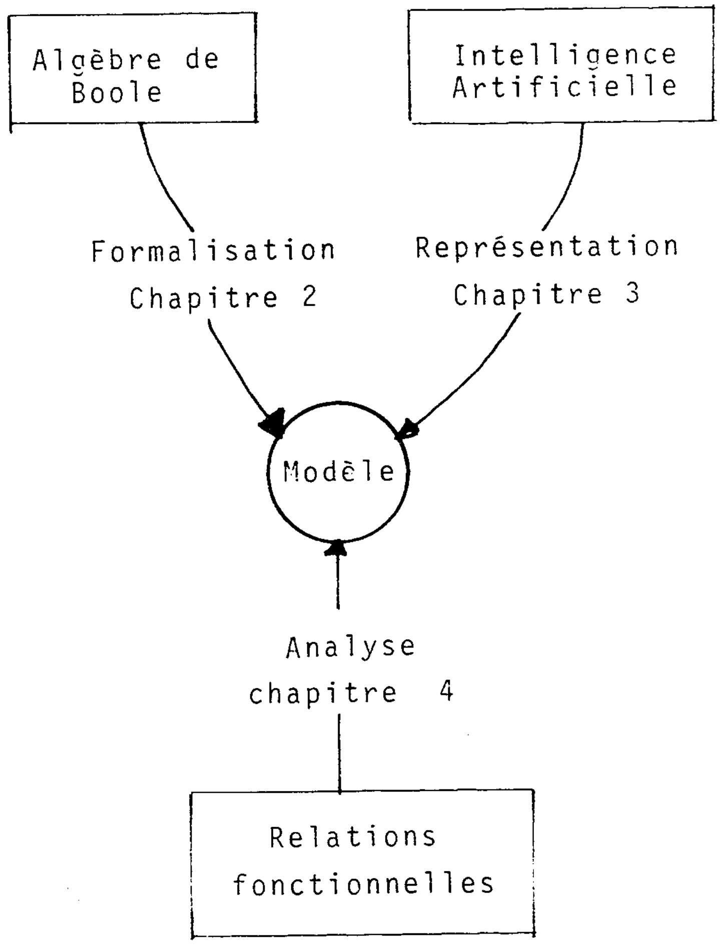


FIGURE 1.4. : Approche d'un modèle permettant de décrire les conditions de progression dans les cursus universitaires

Dans un cinquième chapitre, nous présenterons les traitements du fichier étudiant à l'aide du modèle précédemment défini. Nous effectuerons d'abord une approche globale en réutilisant les notations de [1]. Il apparaîtra alors que les algorithmes de validation dans les graphes ET/OU jouent un rôle important. Nous les décrirons dans le chapitre 6. Dans le chapitre 7 nous montrerons que sous certaines hypothèses il est possible d'optimiser le comportement d'un type d'algorithme.

Le huitième et dernier chapitre sera consacré à la réalisation du fichier des enseignements sous forme de base de données. Cette réalisation a été effectuée pour le compte des trois universités de Grenoble en s'appuyant sur les bases précédemment décrites.

2. LE MODELE BOOLEEN

Il s'agit ici de formaliser les relations qui existent entre enseignements et diplômes et qui déterminent les filières d'études organisées au sein d'une université.

2.1. Définitions

Commençons par donner un certain nombre de définitions.

2.1.1. Enseignement élémentaire

On désignera par enseignement élémentaire un ensemble de matières insécable vis-à-vis de la notion d'échec ou de réussite. La réussite à un enseignement élémentaire constitue pour un étudiant un résultat définitivement acquis dont il pourra toujours faire état.

A titre d'exemple une unité de valeur constitue un enseignement élémentaire, un certificat également, mais il arrive qu'il se subdivise en options dont l'obtention est sanctionnée indépendamment, ce sont alors ces options qui constituent des enseignements élémentaires. Une année d'étude peut constituer un enseignement élémentaire comme c'est le cas dans les Instituts Universitaire de Technologie et dans le deuxième cycle de médecine.

2.1.2. Niveau d'étude

Pour être autorisé à suivre des enseignements il faut pouvoir justifier d'un niveau d'étude suffisant. Pratiquement le niveau d'étude des étudiants est contrôlé chaque année au moment des inscriptions universitaires.

Un niveau d'étude peut être reconnu à un étudiant au vu d'un résultat acquis à l'extérieur de l'Université : c'est le cas du baccalauréat, mais aussi de l'ensemble des titres pour lesquels existe une équivalence dans le système universitaire. Il peut être reconnu à un étudiant au terme d'une décision exceptionnelle qui constitue une dérogation.

Enfin, et c'est le cas le plus fréquent, il est le résultat de l'acquisition d'enseignements élémentaires. Ainsi l'obtention de deux cerficats convenablement choisis permet de faire état d'une licence qui caractérise le niveau d'étude nécessaire pour suivre les certificats de la maîtrise correspondante.

Afin de rendre plus souples les modalités de progression dans le système d'étude, il est courant de voir définir, outre les conditions d'acquisition normale d'un niveau d'étude, des conditions dites conditionnelles qui permettent à un étudiant qui n'aurait pas satisfait entièrement aux conditions normales de s'inscrire malgré tout au niveau supérieur. Le niveau supérieur auquel accède alors l'étudiant ne peut être acquis définitivement que s'il régularise sa situation au niveau inférieur.

Ainsi, dans les maîtrises à quatre certificats, le premier niveau qui est celui de la licence quand elle existe est normalement acquis avec deux certificats; mais un étudiant qui n'a réussi qu'à l'un des deux peut cependant s'inscrire en deuxième année de maîtrise. Il est exclu cependant qu'il puisse obtenir la maîtrise sans avoir complété la licence, même si, par exemple, il obtenait trois certificats de deuxième année.

2.2. Représentation des mécanismes de progression dans un cursus universitaire.

De ce qui précède on retiendra qu'à un niveau d'étude sont liées d'une part, des conditions d'acquisition normales et éventuellement conditionnelles, d'autre part, des conditions d'inscription. Nous allons étudier successivement ces deux mécanismes, la terminologie et les résultats relatifs aux fonctions booléennes étant tirés de [4].

2.2.1. Mécanisme d'acquisition

A un enseignement élémentaire on associe une variable booléenne x_i. Pour un étudiant donné à un instant donné x_i égale 1 signifie que l'étudiant a obtenu cet enseignement qui lui est alors acquis définitivement (cf. 2.1.1.); x_i égale 0 signifie qu'il ne l'a pas obtenu.

Nous présenterons les conditions d'acquisition normale et conditionnelle d'un niveau d'étude par des fonctions booléennes Φ_j et Ψ_j des variables x_i.

Pour un étudiant donné Φ_j vaut 1 si les enseignements acquis permettent d'après les règles en vigueur d'obtenir ce niveau, Φ_j vaut 0 sinon.

De la même façon Ψ_j vaut 1 si les enseignements acquis permettent de s'inscrire au niveau supérieur, Ψ_j vaut 0 sinon

Si pour e_j les conditions d'acquisition conditionnelle ne sont pas définies on a $\Phi_j = \Psi_j$.

Enfin, dans tous les cas il est clair que $\Phi_j \Rightarrow \Psi_j$.

On remarquera qu'obtenir un enseignement élémentaire ne peut en aucun cas entraîner qu'un niveau d'étude déjà acquis de façon normale ou conditionnelle ne le soit plus; ceci implique que les variables x_i représentatives des enseignements n'interviennent jamais sous forme négative dans les fonctions Φ_j et Ψ_j, autrement dit que les fonctions Φ_j et Ψ_j sont des fonctions croissantes.

Par la suite on appellera étape un niveau d'étude pour lequel les fonctions Φ_j et Ψ_j sont définies; ainsi une licence, une maîtrise sont des étapes.

2.2.2. Mécanismes d'inscription

A un niveau d'étude on associe une variable booléenne e_j. Pour un étudiant donné "e_j égale un" signifie que l'étudiant a acquis ce niveau et "e_j égale 0" qu'il ne l'a pas acquis.

Si le niveau d'étude est reconnu par dérogation ou par équivalence, e_j constitue une donnée, sinon e_j est le résultat du calcul des fonctions Φ_j ou Ψ_j représentatives des conditions d'obtention normale ou conditionnelle du niveau.

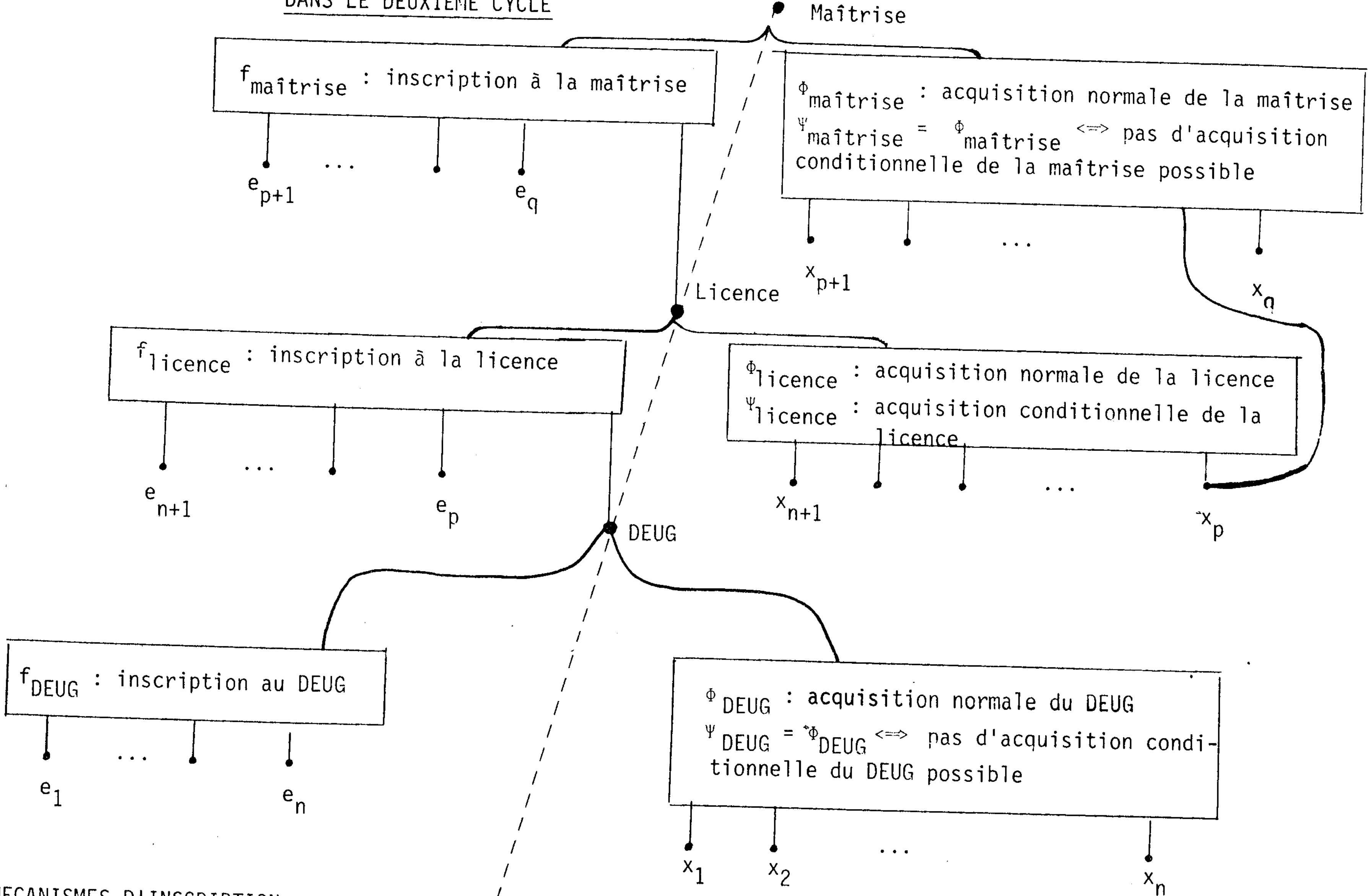
Avec ces conventions, les conditions d'inscription à un niveau d'étude donné, quand elles existent, seront représentées par une fonction booléenne f_j des autres niveaux d'étude. Pour un étudiant donné f_j vaut 1 si les niveaux acquis antérieurement de façon normale ou conditionnelle permettent de s'inscrire au niveau donné, f_j vaut 0 sinon.

Les fonctions f_j de même que les fonctions Φ_j et Ψ_j sont des fonctions booléennes croissantes car l'obtention d'un niveau d'étude ne peut faire perdre un droit à l'inscription déjà acquis; le lecteur retrouvera là une autre forme de la règle "qui peut le plus peut le moins".

<u>Remarque</u> : *On notera que l'inscription à un enseignement élémentaire n'est pas définie, parce qu'elle n'a de sens que par rapport au niveau d'étude dans l'obtention duquel intervient l'enseignement. Ainsi, une même unité de valeur pouvant être utilisée au niveau de la licence dans une filière et au niveau de la maîtrise dans une autre filière on ne saurait définir des conditions d'inscription à cette unité de valeur.*

La figure 2.1 montre dans le cadre du deuxième cycle comment s'articulent les mécanismes d'inscription et les mécanismes d'obtention.

FIGURE 2.1. MECANISMES D'INSCRIPTION ET D'ACQUISITION DANS LE DEUXIEME CYCLE



MECANISMES D'INSCRIPTION

(Portent sur les niveaux $e_1, e_2, \ldots$, DEUG, $e_{n+1}, \ldots$ Licence, $\ldots, e_q$, Maîtrise)

MECANISMES D'ACQUISITION

(Portent sur les enseignements élémentaires $x_1, x_2, \ldots, x_q$)

Remarque : x_p est un enseignement qui intervient dans les conditions d'acquisition de 2 niveaux différents ici Licence et Maîtrise.

2.3. Exemple

Prenons comme exemple la licence et la maîtrise d'enseignement de mathématiques organisées par l'Université Scientifique et Médicale de Grenoble. Cet exemple assez significatif de l'ensemble des problèmes qui se posent sera repris dans les paragraphes et chapitres suivants.

Les niveaux d'étude et les enseignements seront respectivement identifiés par les variables booléennes e_i et x_i.

Description de la licence (e_5)

- Pour s'inscrire à la licence (e_5) il faut possèder l'un des quatre D.U.E.S. (Diplôme Universitaire d'Etudes Scientifiques) soit MP2 (e_1), PC2 (e_2), CB2 (e_3), BG2 (e_4), soit un titre dérogatoire d'accès au deuxième cycle de science (e_7).

- La licence se compose de quatre unités d'enseignement appelées unité A (x_1), unité B (x_2), unité C (x_3), unité D (x_4). L'obtention de la licence implique l'obtention des quatre unités.

- Un étudiant qui a obtenu seulement deux des quatre unités précitées satisfait aux conditions d'obtention partielle de la licence et est autorisé à s'inscrire au niveau supérieur.

Formalisons ce qui précède en associant à e_5 le triplet de fonctions booléennes :

$f_5 \equiv e_1 + e_2 + e_3 + e_4$ représentative des conditions d'inscription

$\Phi_5 \equiv x_1\ x_2\ x_3\ x_4$ représentative des conditons d'obtention

$\Psi_5 \equiv x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$ représentative des conditions d'obtention partielle.

Description de la maîtrise (e_6)

- Pour s'inscrire à la maîtrise (e_6) il faut posséder la licence (e_5) au moins de façon partielle, on peut aussi faire état d'une dérogation (e_8).

- La maîtrise est composée de trois unités d'enseignement appelées unité A'(x_5), unité B'(x_6) et unité C'(x_7).

L'unité B' est obligatoire.

L'unité A' se divise en deux sous-unités $A'_1(x_8)$ et $A'_2(x_9)$

L'unité C' est divisée en quatre sous-unités $C'_A(x_{10})$, $C'_B(x_{11})$, $C'_C(x_{12})$, $C'_D(x_{13})$.

Si l'on possède l'unité A' complète, pour acquérir la maîtrise il faut en outre posséder l'une des quatre sous-unités C'_A, C'_B, C'_C, C'_D.

On peut enfin ne passer que la sous-unité $A'_1(x_8)$ de l'unité A' auquel cas il faut deux des sous-unités constituantes de C' pour acquérir la maîtrise.

- Enfin, il n'y a pas de conditionsd'obtention partielle de la maîtrise qui permettent de s'inscrire au niveau supérieur.

Le formalisme utilisé conduit à associer à e_6 les fonctions booléennes suivantes :

$f_6 \equiv e_5 + e_8$ condition d'inscription

$$\Phi_6 = \Psi_6 \equiv \underbrace{x_6 x_8 x_9 (x_{10}+x_{11}+x_{12}+x_{13})}_{\text{Condition d'obtention avec l'unité A' complète}} \ldots$$

$$+ \underbrace{x_6 x_8 (x_{10}x_{11}+x_{10}x_{12}+x_{10}x_{13}+x_{11}x_{12}+x_{11}x_{13}+x_{12}x_{13})}_{\text{Condition d'obtention avec l'unité A' incomplète}}$$

Une écriture équivalente aurait été :

$$\Phi_6 = \Psi_6 \equiv x_6 x_5 x_7 + x_6 x_8 \, (x_{10}x_{11}+x_{10}x_{12}+x_{10}x_{13}+x_{11}x_{12}$$

$$+ x_{11}x_{13}+x_{12}x_{13})$$

avec : $x_5 = x_8 \, x_9$

$x_7 = x_{10} + x_{11} + x_{12} + x_{13}$

2.4. Conclusion

Le tableau suivant (figure 2.2.) résume ce qui précède :

		niveau d'études	
	enseignement élémentaire	dérogation-équivalence	étape
obtention	constitue une donnée	constitue une donnée	fonction booléenne des enseignements élémentaires Φ_j
obtention partielle			fonction booléenne des enseignements élémentaires Ψ_j
inscription			fonction booléenne des niveaux d'études f_j

FIGURE 2.2. : On lira ce tableau comme suit :

L'obtention d'un enseignement élémentaire constitue une donnée
L'obtention partielle d'un enseignement élémentaire n'a pas de sens
L'inscription à un enseignement élémentaire n'a pas de sens.

.....

L'obtention d'une étape est définie comme le résultat du calcul de la fonction Φ_j des enseignements élémentaires.

...

Nous ferons les remarques suivantes sur l'adéquation à la réalité du formalisme ainsi défini :

- Il est évident que pour obtenir une étape il faut au préalable avoir pu s'y inscrire. Il est donc tentant de définir les conditions d'obtention et d'obtention partielle par les fonctions $A_j = \Phi_j f_j$ et $B_j = \Psi_j f_j$ respectivement. Deux raisons ont cependant conduit à effectuer la décomposition. La première est que les fonctions Φ_j, Ψ_j sont des fonctions des enseignements élémentaires alors que f_j est une fonction des niveaux d'étude, et il semblait peu souhaitable de définir une fonction unique portant sur des variables de natures si différentes.

La seconde raison, plus pragmatique est que les fonctions f_j peuvent être évaluées une fois pour toutes en début d'année, alors que les fonctions Φ_j et Ψ_j ne sont calculées qu'au terme des sessions d'examen.

- Si la distinction entre étape et enseignement élémentaire permet de lever certaines ambiguïtés, au niveau des conditions d'inscription notamment (cf. 2.2.2.), elle apparaît comme artificielle chaque fois qu'un seul enseignement élémentaire x_i intervient dans l'obtention d'une étape e_j, comme c'est le cas des études organisées par année. On a alors $\Phi_j = \Psi_j = x_i$ et on pourrait noter $e_j \equiv x_i$ et ne pas définir x_i, ceci aurait pour grave inconvénient de rompre l'homogénéïté du formalisme ici défini.

3. REPRESENTATION PAR LES GRAPHES ET/OU

Il résulte de ce que l'on a vu au chapitre précédent que pour se définir les conditions de progression dans l'enseignement supérieur il faut se donner :

- l'ensemble X des "enseignements élémentaires"
- l'ensemble N des "niveaux d'études"
- les triplets de fonctions booléennes associés à certains éléments de N (les étapes) :

$x \in \mathcal{P}(X) \xrightarrow{\Phi} \{0,1\}$ représentative des conditions d'obtention de l'étape

$x \in \mathcal{P}(X) \xrightarrow{\Psi} \{0,1\}$ représentative des conditions d'obtention partielle de l'étape

$x \in \mathcal{P}(N) \xrightarrow{f} \{0,1\}$ représentative des conditions d'inscription à l'étape

Dans ce chapitre, nous allons montrer que les graphes ET/OU, dont les applications premières sont du domaine de l'intelligence artificielle (cf. [2]) peuvent être utilisés pour représenter le formalisme précédemment défini. Enfin, nous essaierons de dégager les avantages de cette représentation. Pour la terminologie relative aux graphes nous nous référerons à [5].

3.1. Définition des graphes ET/OU

Commençons par donner la définition habituelle des graphes ET/OU.

Définition :

Un graphe ET/OU est un graphe orienté dans lequel les noeuds se répartissent en trois classes disjointes :

- les noeuds terminaux,
- les "ET-noeuds"
- les "OU-noeuds"

On définit sur un tel graphe le processus de validation suivant :

- les noeuds terminaux sont les données
- un ET-noeud est valide si tous ses successeurs le sont
- un OU-noeud est valide si au moins l'un de ses successeurs l'est.

Un graphe ET/OU peut servir à représenter la décomposition d'un problème en sous-problèmes élémentaires. On touvera plusieurs exemples de ce type d'application dans N.J. NILSON [2/chapitre 4].

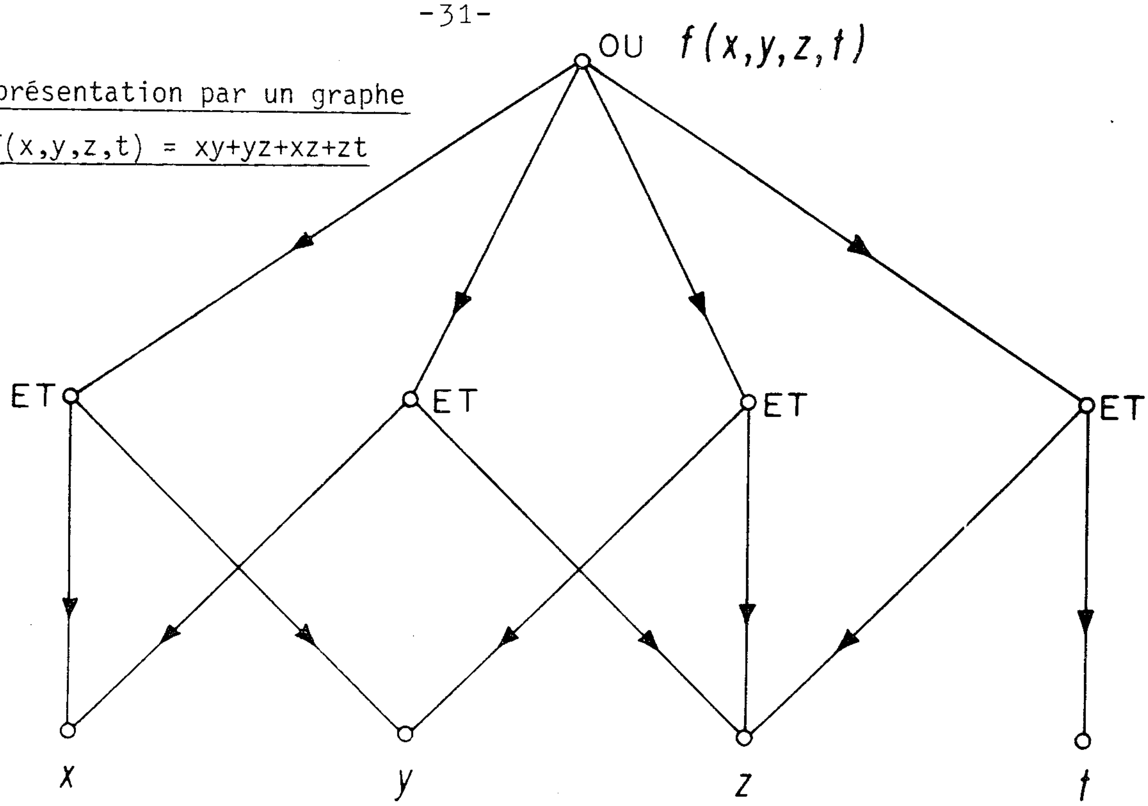
Nous associons à chaque noeud d'un graphe ET/OU une variable booléenne qui vaut 1 si le noeud est valide et 0 sinon. Avec cette convention et si le graphe est sans circuits le processus de validation défini ci-dessus est équivalent au calcul des fonctions booléennes croissantes. Les noeuds terminaux étant les variables, les ET-noeuds et les OU-noeuds permettent respectivement de représenter le produit et la somme booléenne.

On peut ainsi représenter toute fonction booléenne croissante par un graphe ET/OU sans circuit; donnons-en un exemple (figure 3.1) en représentant la fonction :

$$f(x,y,z,t) = xy + yz + xz + zt$$

NB. : On remarquera que cette fonction n'est pas représentable par un arbre ET/OU.

IGURE 3.1. : Représentation par un graphe ET/OU de $f(x,y,z,t) = xy+yz+xz+zt$



3.2. Une extension des graphes ET/OU

Nous allons maintenant donner une définition des graphes ET/OU mieux adaptée à l'objet de notre étude.

Définition :

Etant donné un graphe orienté sans circuit, à chaque noeud non terminal N_i de ce graphe on associe un entier positif λ_i tel que l'on puisse définir le processus de validation suivant :

- les noeuds terminaux sont les données,
- un noeud N_i est valide si et seulement si il existe au moins λ_i successeurs de N_i qui le sont.

Par la suite on appellera λ_i le poids attaché au noeud N_i.

Les cas particuliers pour lesquels λ_i est égal à un et λ_i est égal au nombre de successeurs de N_i permettent de définir des OU-noeuds et des ET-noeuds respectivement.

On peut donc représenter par un tel graphe toute fonction booléenne croissante. La valeur $\lambda_i = 0$ permet de définir des noeuds toujours valides analogues aux tautologies en logique booléenne.

Les valeurs $\lambda_i >$ nombre de successeurs de N_i permettent de définir des noeuds jamais valides analogues aux contradictions en logique booléenne.

Enfin, les valeurs de λ_i telles que :
$1 < \lambda_i <$ nombre de successeurs de N_i permettent de représenter les choix combinatoires de n éléments parmi p. L'un des avantages de l'extension ici définie est que ces choix combinatoires peuvent être représentés par p arêtes alors que leur représentation par un graphe ET/OU classique nécessite $(n+1)\ C_n^p$ arêtes (cf. figure 3.2).

FIGURE 3.2.a

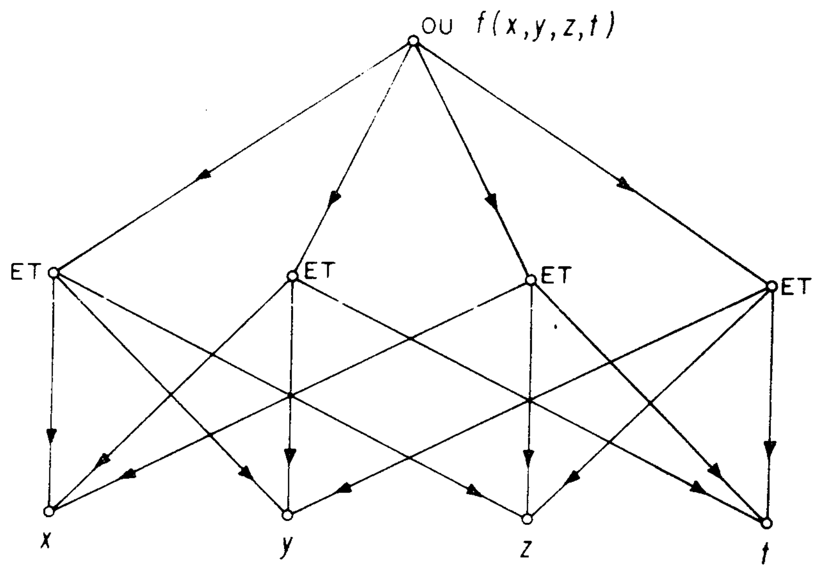


FIGURE 3.2.b

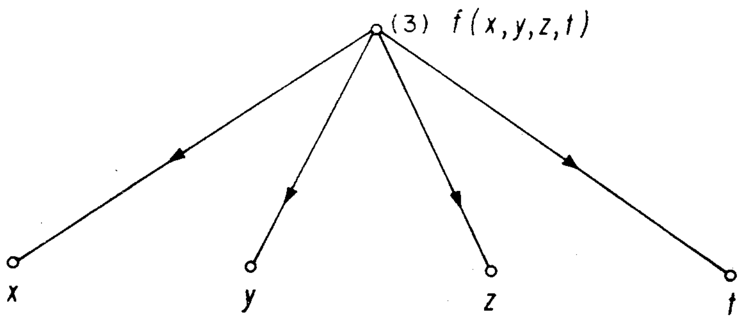


FIGURE.3.2. : <u>Représentation du choix de 3 éléments parmi 4 par un graphe ET/OU classique (2a) et par son extension (2b)</u>.(Le poids associé aux noeuds non terminaux est noté entre parenthèses).

On notera que la fonction booléenne associée est la fonction symétrique croissante : $f(x,y,z,t) = xyz + xyt+yzt+xzt$

Reprenons l'exemple donné en 3.1; avec l'extension des graphes ET/OU proposée on aura la représentation suivante de $f(x,y,z,t) = xy + yz + xz + zt$

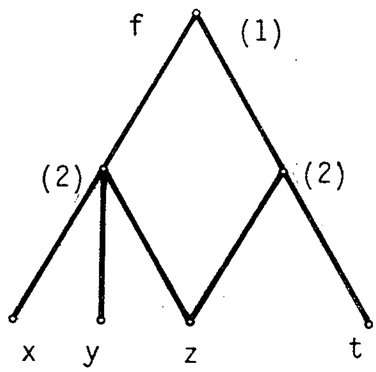
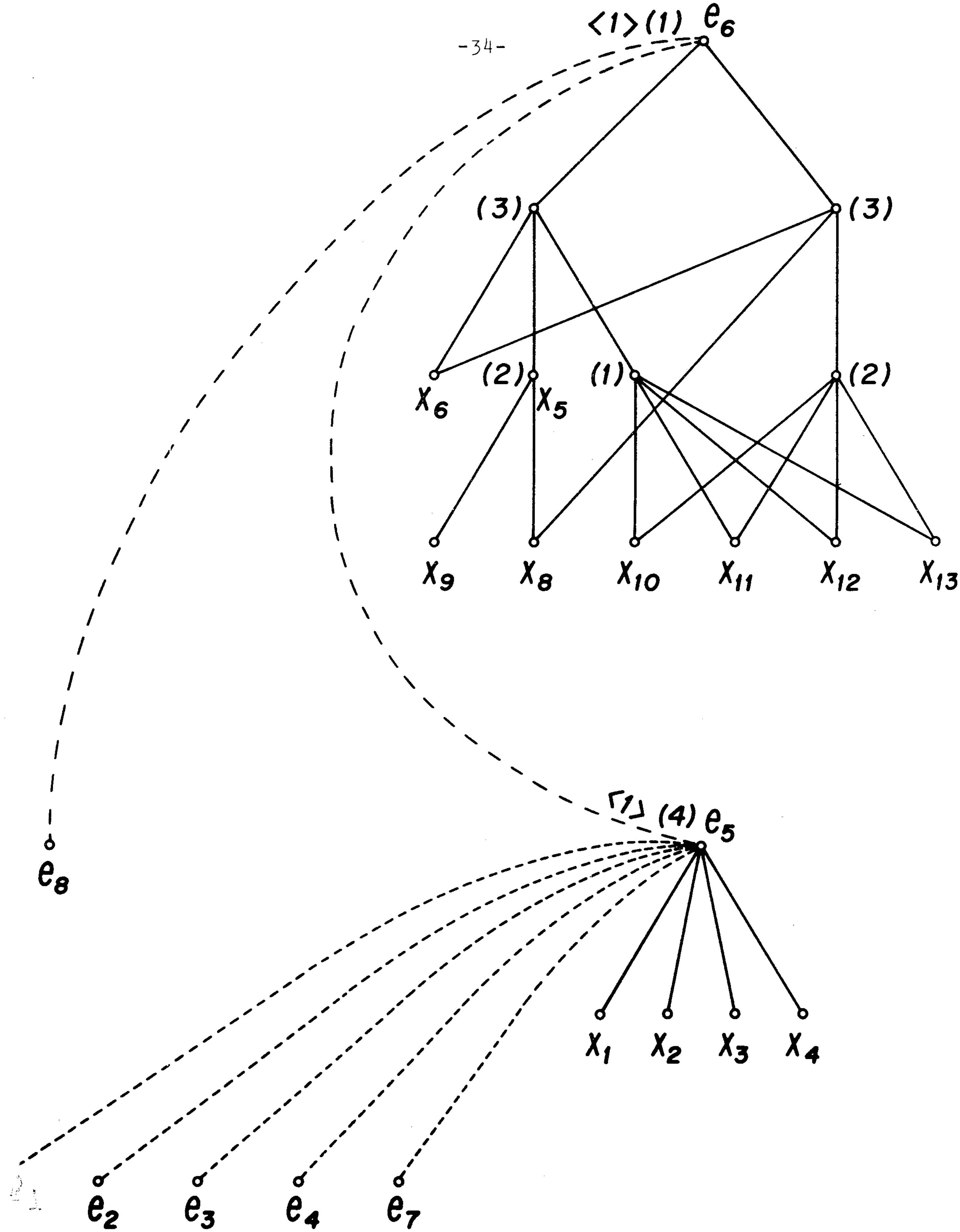


FIGURE 3.3. : Représentation de $f(x,y,z,t) = xy+xz+yz+zt$ par une extension de graphe ET/OU (comparer avec la figure 3.1.)

3.3. Exemple d'application à la représentation des enseignements

Reprenons l'exemple de la licence et de la maîtrise d'enseignement de mathématiques tel qu'il a été présenté en 2.3.

Une représentation possible est la suivante (fig.3.4.)



<1> (1) e6
(3)
(3)
X6
(2)
X5
(1)
(2)
X9
X8
X10
X11
X12
X13
<1> (4) e5
e8
X1
X2
X3
X4
e2
e3
e4
e7

FIGURE 3.4.

3.4. Propriétés des graphes ET/OU appliqués à la représentation des fonctions booléennes croissantes

3.4.1. Différentes représentations d'une fonction booléenne

Une fonction booléenne croissante pouvant être représentée par différents graphes ET/OU, il est intéressant d'associer à chacune de ces représentations un coût qui peut être défini de diverses façons : par la suite, on choisira arbitrairement une fonction linéaire du nombre de noeuds et du nombre d'arêtes du graphe ET/OU.

Soit n le nombre de noeuds et p le nombre d'arêtes ; on utilisera la fonction $C(n,p) = \alpha n + \beta p$.

Si on considère le graphe ET/OU utilisé pour représenter la fonction $f(x,y,z,t) = xy + xz + yz + zt$ dans la figure 3.1., le coût en sera : $C = 9\alpha + 12\beta$.

De façon générale, la méthode des consensus assure que pour une fonction booléenne croissante la base irredondante de monômes premiers est unique (cf [4] chapitre IV). On peut donc toujours écrire de façon unique f sous la forme $f = \sum_{i=1}^{n} m_i$ où les m_i sont les monômes premiers de la base irredondante, et représenter f par un graphe ET/OU tel que :

- la racine du graphe est associée à f, son poids est 1.
- à chaque m_i est associé un noeud successeur de la racine et de poids égal au nombre de variables contenuesdans le monôme.
- les noeuds sans successeurs sont associés aux variables.

Cette représentation est unique; si on désigne par v_i le nombre de variables contenues dans m_i et par w le nombre des variables, le coût de cette représentation est :

$$C = (w+n+1)\alpha + (n+\sum_{i=1}^{n} v_i)\beta$$

Ce coût est maximal quand f est une fonction dont la base irredondante comporte $C_w^{[\frac{w}{2}]}$ monômes de $[\frac{w}{2}]$ variables.

(On montre en effet que le nombre maximal de monômes premiers d'une fonction de n variables est $C_n^{[\frac{n}{2}]}$, le nombre maximal de variables par monôme étant alors $[\frac{n}{2}]$).

On a donc une borne supérieure de C qui est :

$$C_{MAX} = (w + C_w^{[\frac{w}{2}]} + 1)\alpha + ([\frac{w}{2}] + 1)\, C_w^{[\frac{w}{2}]}\, \beta$$

La figure 3.5 donne les valeurs de C_{MAX} pour différentes valeurs de w.

w	C_{MAX}
1	$3\alpha + 2\beta$
2	$5\alpha + 4\beta$
3	$7\alpha + 9\beta$
4	$11\alpha + 18\beta$
5	$16\alpha + 40\beta$
6	$27\alpha + 80\beta$
7	$43\alpha + 175\beta$
8	$79\alpha + 350\beta$
9	$136\alpha + 756\beta$
10	$263\alpha + 1512\beta$
15	$6451\alpha + 57915\beta$
20	$184777\alpha + 2032316\beta$
25	$5200326\alpha + 72804200\beta$
30	$155117551\alpha + 2481880320\beta$

<u>Figure 3.5</u>

L'intérêt de l'extension des graphes ET/OU proposée en 3.2. est de réduire le coût des représentations en faisant apparaître des fonctions booléennes symétriques croissantes. On peut noter S_n^p ces fonctions qui sont les sommes de produits de n variables p à p, p = 1,...,n.

La somme et le produit booléen correspondant respectivement aux fonctions S_n^1 et S_n^n, on peut écrire toute fonction booléenne croissante uniquement à l'aide des fonctions S_n^p auxquelles on applique la loi de composition.

$$\text{Ainsi } f(x,y,z,t) = S_2^1(S_3^2(x,y,z),S_2^2(z,t)) \qquad (1)$$

$$= S_4^1(S_2^2(x,y),S_2^2(y,z),S_2^2(x,z),S_2^2(z,t)) \qquad (2)$$

etc...

A chaque écriture correspond une représentation par une extension de graphe ET/OU telle que :

- le nombre de noeuds est égal au nombre de variables de la fonction augmenté du nombre de fonctions S_p^n utilisées pour son écriture,
- le nombre d'arêtes est obtenu comme suit :
  . on compte 1 pour une variable booléenne
  . on compte 1 plus la somme des nombres d'arêtes nécessaires à la représentation des variables pour une fonction S_n^p.

De la sorte les coûts des écritures de f données plus haut sont : pour (1) : $C = 7\alpha + 7\beta$

pour (2) : $C = 9\alpha + 12\beta$

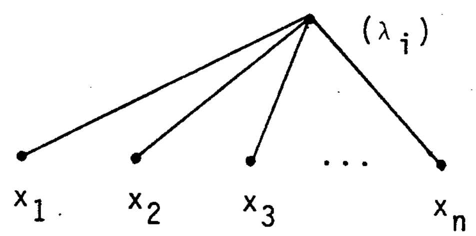
3.4.2. Propriété de dualité

Soit une extension de graphe ET/OU associée à une fonction booléenne f, soit λ_i le poids associé à chaque noeud N_i non terminal : montrons que si l'on associe à chaque noeud N_i le nouveau poids :

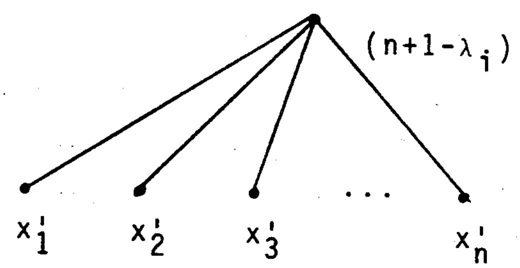
$$\mu_i = \text{nombre de successeurs de } N_i + 1 - \lambda_i$$

le graphe ainsi obtenu constitue une représentation de la duale de f.

Soit $f(x_1,x_2,\ldots,x_n)$ représentée par :



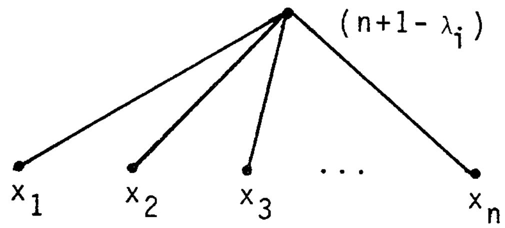
Par définition f est vraie si au moins λ_i des variables $x_1,\ldots,x_n$ le sont, par conséquent f est faux si $n+1-\lambda_i$ des variables $x_1,\ldots,x_n$ sont fausses. On représentera donc f' par :



Si on note f^* la duale de f on peut écrire :

$$f^*(x_1,x_2,\ldots,x_n) = f'(x'_1,x'_2,\ldots,x'_n)$$

par conséquent la représenter par :



Si les noeuds N_i ne sont pas des noeuds terminaux, c'est-à-dire s'ils sont des quantités booléennes à leur tour fonctions d'autres variables, on peut leur réappliquer ce qui précède

Le résultat annoncé en découle.

Reprenons l'exemple du paragraphe 3.2.; une représentation de la duale de f(x,y,z,t) = xy + xz + yz + zt est : (fig.3.5)

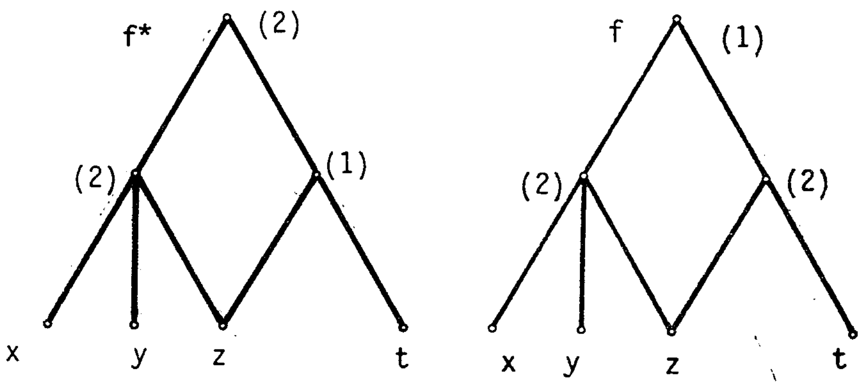


Figure 3.5 : Représentations comparées de f* et de f.

Si l'on fait intervenir les notions de coût développées dans le paragraphe précédent, il sera intéressant de comparer les différentes écritures de f et de f* afin de déterminer quelle représentation est la moins coûteuse. Si c'est une représentation de f^* qui est choisie le passage de f^* à f se fera conformément à la règle définie ci-dessus.

Le chapitre 4 (cf. 4.5) présentera le cadre général dans lequel ce qui précède est appliqué.

3.5. Conclusion

L'extension des graphes ET/OU proposée en 3,2 permet bien de représenter le formalisme booléen défini dans le paragraphe précédent, et offre de plus les avantages suivants :

- Les relations entre variables et fonctions apparaissent de façon symétrique ce qui signifie pour le problème qui nous intéresse que l'on pourra traiter également les deux questions suivantes :
    - Quels sont les enseignements qui interviennent dans l'obtention de tel diplôme ?
    - Quels sont les diplômes dans lesquels intervient tel enseignement.

- L'algorithme de calcul correspondant au processus de validation est simple à mettre en oeuvre car défini de façon homogène sur l'ensemble des noeuds non terminaux, ce qui n'est pas le cas pour les graphes ET/OU classiques dans lesquels on établit une distinction entre ET-noeuds et OU-noeuds.

- Les graphes obtenus constituent une représentation visuelle donc synthétique des modalités de progression dans les cursus universitaires.

4. ANALYSE PAR LES RELATIONS DE DEPENDANCES

Si nous reprenons le modèle défini au chapitre 2, nous savons qu'il répond à la logique du problème à traiter. Cependant, tel que nous l'avons présenté, il ne correspond pas à ce que perçoit l'utilisateur, que ce soit le gestionnaire d'un service de scolarité ou l'enseignant responsable de l'organisation d'un diplôme. Demander à l'un ou à l'autre de mettre les règlements sous forme de fonction booléenne est bien sûr impensable; il convient donc de définir une autre approche en accord avec ce que perçoit l'utilisateur. Cette approche doit être suffisamment générale pour tenir compte éventuellement des anomalies qui existent dans la réalité ou dans la perception de celle-ci par l'utilisateur et qui devront alors être détectées comme contraire à la logique du modèle.

4.1. Introduction des relations de dépendance

Tout ce qui suit se rapporte à la description des conditions d'obtention d'une étape mais peut facilement être étendu aux conditions d'obtention partielle et aux conditions d'inscription puisque fondamentalement la logique des différents mécanismes de progression est la même.

Il est courant dans la réalité d'exprimer les conditions d'obtention d'une étape à partir des enseignements élémentaires sous forme d'implication; ainsi dit-on "tel et tel certificat donnent la maîtrise".

Les relations de dépendance vont nous permettre de mieux formuler ce point de vue.

Considérons donc l'ensemble des étapes et des enseignements élémentaires noté $\mathcal{E}$ et introduisons une relation binaire entre les parties de $\mathcal{E}$ qui prendra le sens suivant :

- $e_1\ e_2 \ldots\ e_n \rightarrow e_p$

signifie que l'obtention de e_1 et l'obtention de e_2 ... et l'obtention de e_n impliquent/donnent l'obtention de e_p.

- $e_1 \rightarrow e_p$

  $e_2 \rightarrow e_p$

  ...

  $e_n \rightarrow e_p$

signifie que l'obtention de e_1 ou l'obtention de e_2 ... ou l'obtention de e_n implique l'obtention de e_p.

On nommera relations de dépendance les relations $\rightarrow$ et on se donnera, pour ces relations, le système d'axiomes suivant:

Réflexivité	$E_i \rightarrow E_i$	(1)
Transitivité	si $E_i \rightarrow E_j$ et $E_j \rightarrow E_k$ alors $E_i \rightarrow E_k$	(2)
Projection	si $E_i \subset E_j$ alors $E_j \rightarrow E_i$	(3)
Additivité	si $E_i \rightarrow E_j$ et $E_i \rightarrow E_k$ alors $E_i \rightarrow E_j\ E_k$	(4)
Augmentation	si $E_i \rightarrow E_j$ alors $\forall E_k$ $E_i\ E_k \rightarrow E_j$	(5)

<u>Remarque</u> : *De ces axiomes on déduit la propriété de pseudo-transitivité comme suit :*

Pseudo-transitivité : si $E_i \rightarrow E_j$ *et* $E_j\ E_k \rightarrow E_p$ *alors* $E_i\ E_k \rightarrow E_p$ (6)

en effet on a :

par augmentation $E_i\ E_k \rightarrow E_j$

par additivité $E_i\ E_k \rightarrow E_j\ E_k$

par transitivité $E_i\ E_k \rightarrow E_j\ E_k$ *et* $E_j\ E_k \rightarrow E_p$ *donne* $E_i\ E_k \rightarrow E_p$

la propriété annoncée en découle.

On vérifie aisément que ces axiomes correspondent à la réalité du problème que nous étudions.

Dans le but différent d'étudier les structures possibles pour une base de données on trouvera dans [3] un modèle analogue. Certains résultats étant transposables, nous les utiliserons.

4.2. Application à la description des conditions d'obtention d'une étape

On l'a vu, la notation sous forme de relations de dépendance est naturelle, puisqu'elle revient à énumérer les combinaisons d'enseignements élémentaires qui permettent d'obtenir une étape. Cette énumération pouvant dans certains cas être fort longue, il est commode d'introduire des variables intermédiaires non significatives de l'organisation des études.

L'analyse des conditions d'obtention d'une étape doit alors fournir un ensemble de relations de dépendance portant sur l'étape, les enseignements élémentaires qui interviennent dans son obtention et éventuellement les variables intermédiaires utilisées pour alléger l'écriture.

Exemple 1 : En reprenant la maîtrise d'enseignement de mathématiques qui a servi d'exemple en 2.3, et en gardant les mêmes notations, on pourra analyser les conditions comme suit :

$x_{10} \rightarrow y_1$	$x_{10}x_{11} \rightarrow y_2$
$x_{11} \rightarrow y_1$	$x_{10}x_{12} \rightarrow y_2$
$x_{12} \rightarrow y_1$	$x_{10}x_{13} \rightarrow y_2$
$x_{13} \rightarrow y_1$	$x_{11}x_{12} \rightarrow y_2$
$x_6x_8x_9y_1 \rightarrow e_6$	$x_{11}x_{13} \rightarrow y_2$
$x_6x_8y_2 \rightarrow e_6$	$x_{12}x_{13} \rightarrow y_2$

N.B.: On a utilisé deux variables intermédiaires y_1 et y_2.

Mais cette écriture est encore lourde dès qu'il s'agit d'exprimer que l'obtention d'une variable dépend de l'obtention de p variables parmi n. Or, ce type de condition intervenant souvent dans l'analyse on l'écrira sous la forme :

$$(x_1, x_2, \ldots, x_n)(p) \rightarrow x_0.$$

Avec cette notation l'exemple 1 devient :

Exemple 2 : $(x_{10}, x_{11}, x_{12}, x_{13})(1) \rightarrow y_1$

$(x_{10}, x_{11}, x_{12}, x_{13})(2) \rightarrow y_2$

$(x_6, x_5, x_9, y_1)(4) \rightarrow e_6$

$(x_6, x_8, y_2)(3) \rightarrow e_6$

4.3. Cohérence des résultats de l'analyse

On supposera désormais que le résultat de l'analyse d'un règlement se présente sous la forme d'un ensemble de relations de dépendance tel qu'on vient de le définir.

Nous allons montrer qu'à un tel ensemble on ne peut pas toujours associer un graphe ET/OU et que dans ce cas il y a une incohérence dans l'analyse.

Posons le problème dans l'autre sens : étant donné un graphe ET/OU on peut toujours lui associer un ensemble de relations de dépendance défini comme suit :

Soit N_0 un ET-noeud du graphe et $N_1, N_2, \ldots, N_n$ ses descendants; on associe à N_0 la relation :

$$N_1 \ N_2 \ \ldots \ N_n \rightarrow N_0$$

Soit N_0 un OU-noeud du graphe et $N_1, N_2, \ldots, N_n$ ses descendants; on associe à N_0 les relations :

$$N_1 \rightarrow N_0$$
$$N_2 \rightarrow N_0$$
$$\ldots$$
$$N_n \rightarrow N_0$$

Il est clair que les règles d'augmentation, de transitivité et de pseudo-transitivité appliquées aux relations ainsi obtenues permettent d'en générer de nouvelles. Dans tous les cas les relations ainsi générées sont telles que si les variables en partie gauche correspondent à des noeuds valides dans le graphe le processus de validation conduira à valider le noeud correspondant à la variable en partie droite.

Inversement à tout ensemble de relations de dépendance ne correspond pas toujours un graphe ET/OU. En effet, dans un tel graphe la relation de descendance constitue une relation d'ordre partiel entre les noeuds, tandis que la relation $\rightarrow$ ne constitue qu'une relation de préordre entre les variables (cf. [3] chap. 8).

A titre d'exemple donnons les ensembles de relations suivants :

Exemple 3 : $x_1 \rightarrow x_2$

$x_2 \rightarrow x_1$

Exemple 4 : $x_1x_2 \rightarrow x_3$

$x_1x_3 \rightarrow x_4$

$x_1x_4 \rightarrow x_2$

Dans de tels cas, il n'est pas possible de définir les conditions d'obtention de certaines variables, ce qui dénote une incohérence dans l'analyse.

Il importe donc de détecter ces incohérences, c'est-à-dire dans une première étape de s'assurer que la relation $\rightarrow$ constitue bien une relation d'ordre partiel entre les variables.

Nous ne donnerons pas ici le détail du traitement correspondant, mais on trouvera dans [6] page 258, un algorithme qui permet d'effectuer ce type d'opération. On remarquera que le temps d'exécution de cet algorithme est une fonction linéaire du nombre de relations données et du nombre de variables.

4.4. Irredondance des résultats de l'analyse

L'ensemble des relations de dépendance résultat de l'analyse n'est pas obligatoirement minimal, autrement dit, il existe peut-être un ensemble contenant moins de relations susceptible de décrire le même règlement.

Pour aborder le problème de la recherche de cet ensemble minimal, nous allons d'abord introduire deux notions inspirées de [3 chapitre 9], celles de fermeture et de couverture minimale.

4.4.1. Fermeture d'un ensemble de relations de dépendance

Définition : Une relation de dépendance $E_i \rightarrow E_j$ est dite élémentaire si pour toute partie E_k de E_i, $E_k \rightarrow E_j$ ne constitue pas une relation de dépendance.

On suppose que le résultat de l'analyse n'est constitué que de relations de dépendance élémentaires, car si ce n'est pas le cas il est aisé de s'y ramener par un traitement approprié.

Appelons alors fermeture d'un ensemble de relations de dépendance l'ensemble des relations élémentaires qu'il est possible d'obtenir à partir de l'ensemble de départ par application des règles de transitivité et de pseudo-transitivité; cet ensemble est unique.

Exemple 5 : Reprenons l'ensemble de relations donné en 4.2 (ex.1) la fermeture de cet ensemble est :

(1) $x_{10} \rightarrow y_1$

(2) $x_{11} \rightarrow y_1$

(3) $x_{12} \rightarrow y_1$

(4) $x_{13} \rightarrow y_1$

(5) $x_{10}x_{11} \rightarrow y_2$

(6) $x_{10}x_{12} \rightarrow y_2$

(7) $x_{10}x_{13} \rightarrow y_2$

(8) $x_{11}x_{12} \rightarrow y_2$

(9) $x_{11}x_{13} \rightarrow y_2$

(10) $x_{12}x_{13} \rightarrow y_2$

(11) $x_6 x_8 x_9 y_1 \rightarrow e_6$

(12) $x_6 x_8 y_2 \rightarrow e_6$

(13) $x_{10} x_6 x_8 x_9 \rightarrow e_6$

(14) $x_{11} x_6 x_8 x_9 \rightarrow e_6$

(15) $x_{12} x_6 x_8 x_9 \rightarrow e_6$

(16) $x_{13} x_6 x_8 x_9 \rightarrow e_6$

(17) $x_{10} x_{11} x_6 x_8 \rightarrow e_6$

(18) $x_{10} x_{12} x_6 x_8 \rightarrow e_6$

(19) $x_{11} x_{12} x_6 x_8 \rightarrow e_6$

(20) $x_{10} x_{13} x_6 x_8 \rightarrow e_6$

(21) $x_{11} x_{13} x_6 x_8 \rightarrow e_6$

(22) $x_{12} x_{13} x_6 x_8 \rightarrow e_6$

Dans le cadre du problème qui nous intéresse les relations de la fermeture telles qu'apparaissent en partie gauche uniquement des variables associées aux enseignements élémentaire et en partie droite, uniquement la variable associée à l'étape, constituent une liste exhaustive des combinaisons d'enseignements qui permettent d'obtenir une étape.

Exemple 6 : Reprenons la fermeture donnée dans l'exemple 5, les relations numérotées (13), (14), (15), (16), (17), (18), (19), (20), (21), (22) correspondent aux combinaisons : $x_6 x_8 x_9 x_{10}$, $x_6 x_8 x_9 x_{11}$, $x_6 x_8 x_9 x_{12}$, $x_6 x_8 x_9 x_{13}$, $x_6 x_8 x_{10} x_{11}$, $x_6 x_8 x_{10} x_{12}$, $x_6 x_8 x_{10} x_{13}$, $x_6 x_8 x_{11} x_{12}$, $x_6 x_8 x_{11} x_{13}$, $x_6 x_8 x_{12} x_{13}$.

Il y a donc dix façons différentes d'obtenir la maîtrise d'enseignement de mathématiques et dans certains cas on dénombre jusqu'à 231 façons d'obtenir un diplôme (Maîtrise de Mathématiques et Applications Fondamentales).

4.4.2. Couverture minimale d'un ensemble de relations de dépendance

Définition : **On appelle couverture minimale d'un ensemble de relations de dépendance, l'ensemble des relations de dépendance élémentaires dont la fermeture est égale à la fermeture de l'ensemble de départ et tel que cette propriété n'est plus vraie si l'on supprime une relation de dépendance de la couverture minimale.**

L'intérêt de cette notion est grand, puisque la couverture minimale constitue le plus petit ensemble de relations qu'il suffit d'implémenter pour pouvoir en déduire toutes les conditions d'obtention d'une étape.

En règle générale la couverture minimale d'un ensemble de relations de dépendance n'est pas unique, nous allons cependant montrer que le fait que la relation $\rightarrow$ soit une relation d'ordre partiel entre les variables implique l'unicité (cf.[7]).

La construction algorithmique de la fermeture de la couverture minimale s'appuie sur l'analogie qui existe entre un ensemble de relations de dépendance et une catégorie de fonctions booléennes.

On peut associer à l'ensemble des relations $R = \{E_i \rightarrow F_i\}$ la fonction booléenne $\rho = \sum_i e_i f_i'$.

Si S est un ensemble de relations de dépendance déduit de R par application des règles d'augmentation et de pseudo-transitivité, on notera $S \sim R$, le symbole $\sim$ signifiant que R peut être dérivé de S et inversement.

Soit ρ et σ les fonctions booléennes associées à R et S; on démontre (cf. [3] et [8]) le théorème suivant :

Théorème : $R \sim S \Longleftrightarrow \rho = \sigma$

Ce théorème découle du lemme suivant :

Lemme 1 : **Tous les monômes premiers de ρ peuvent être engendrés seulement par la contrepartie booléenne des propriétés de pseudo-transitivité et d'augmentation.**

Cette contrepartie peut s'interpréter comme suit :

- aux relations de dépendance :

$$\{E_i \to E_j, E_jE_k \to E_\ell\} \sim \{E_i \to E_j, E_jE_k \to E_\ell, E_iE_k \to E_\ell\}$$

correspondent les fonctions booléennes :

$$e_ie_j' + e_je_ke_\ell' = e_ie_j' + e_ie_ke_j'e_\ell' + e_je_ke_\ell' + e_je_ke_ie_\ell'$$

$$= e_ie_j' + e_je_ke_\ell' + e_ie_ke_\ell'$$

- aux relations de dépendance :

$$\{E_i \to E_j\} \sim \{E_i \to E_j, E_iE_k \to E_j\}$$

correspond de la même façon :

$$e_ie_j' = e_ie_j' + e_ie_ke_j'$$

Du théorème on déduit les deux corollaires suivants :

<u>Corollaire 1</u> : La fermeture d'un ensemble de relations de dépendance est l'image des monômes premiers de la fonction booléenne associée.

<u>Corollaire 2</u> : La couverture minimale d'un ensemble de relations de dépendance est l'image d'une base irredondante de la fonction booléenne associée.

Introduisons maintenant les notions de pseudo-chaîne et de pseudo-cycle :

<u>Définitions</u> : Un ensemble de relations de dépendance élémentaires constitue une <u>pseudo-chaîne</u> s'il peut s'écrire :

$$E_0 \to E_1,\ E_2 \to E_3,\ \ldots,\ E_{i-1} \to E_i,\ \ldots,\ E_{n-1} \to E_n$$

avec pour tout $i = 1,\ldots,n$ $E_{i-1} \subset E_i$ et $E_n \not\subset E_0$

De même un <u>pseudo-cycle</u> est une pseudo-chaîne telle que $E_n \subset E_0$.

On a vu en 4.3 que les ensembles de relations de dépendances que nous considérons ne comportent pas de pseudo-cycles.

Montrons alors qu'une pseudo-chaîne admet une couverture minimale unique.

Soit la pseudo-chaîne :

$$E_1 \to E_2,\ E_2E_3 \to E_4,\ldots,E_{2i}E_{2i+1} \to E_{2(i+1)},\ldots,E_{2k}E_{2k+1} \to E_{2(k+1)} \qquad (1)$$

et la fonction booléenne associée :

$$e_1e'_2 + e_2e_3e'_4 + \ldots + e_{2i}e_{2i+1}e'_{2i+2} + \ldots + e_{2k}e_{2k+1}e'_{2k+2} \qquad (2)$$

D'après le lemme 1, les monômes premiers de (2) sont engendrés par la contrepartie de la pseudo-transitivité et de l'augmentation.

On énumère les relations ainsi engendrées et les monômes correspondants :

$$e_1\ e_3\ \cdots\ e_{2i+1}\ e'_{2i+2} \qquad i = 1,2,\ldots,k$$

$$e_{2j}\ e_{2j+1}\ \cdots\ e_{2i+1}\ e'_{2i+2} \qquad \begin{cases} j = 1,2,\ldots,k-1 \\ i = 1,\ldots,k \end{cases}$$

Montrons que les monômes premiers ainsi générés sont interdits de première espèce, ce qui suffit à s'assurer de l'unicité de la base irredondante, donc de la couverture minimale (cf. [4] p. 115).

Considérons les coordonnées de l'hypercube représentées dans le tableau suivant :

e_1	$\cdots$	e_{2j}	e_{2j+1}	$\cdots$	e_{2i}	e_{2i+1}	e_{2i+2}	$\cdots$	e_{2k+2}
0	...	0	0	...	1	1	0	...	0
0	...	1	1	...	1	1	0	...	0

La première ligne exprime que la coordonnée correspondante est couverte par le monôme $e_{2i}\ e_{2i+1}\ e'_{2i+2}$. L'absence de pseudo-cycle assure que c'est le seul monôme qui couvre ce point.

On peut étendre cette remarque à tous les monômes initiaux associés à la pseudo-chaîne, qui sont donc tous premiers obligatoires.

Considérons alors le monôme $e_{2j}\ e_{2j+1}\ \cdots\ e_{2i+1}\ e'_{2i+2}$: il couvre la deuxième coordonnée du tableau, mais cette coordonnée est aussi couverte par d'autres monômes premiers et en particulier par $e_{2i}\ e_{2i+1}\ e'_{2i+2}, \ldots$

Tous les monômes du type $e_{2j}\ e_{2j+1},\dots,e'_{2i+2}$ sont donc couverts par un monôme obligatoire, ils sont donc non obligatoires interdits de première espèce, donc (2) admet une base irredondante unique et la couverture minimale de (1) est unique.

Ce qui vient d'être fait pour une chaîne est applicable à toutes les chaînes, le résultat annoncé en découle.

Ce résultat théorique a une conséquence pratique importante. En effet s'il est relativement facile pour un analyste unique d'exprimer sans redondances l'ensemble des relations qu'il désire voir implémenter en machine (cf.8.4.1) il est par contre peu probable que les résultats obtenus par des analystes travaillant de façon décentralisée (au niveau des U.E.R. par exemple) soient eux irredondants. En effet certaines règles de progression sont utilisées de façon commune par différentes U.E.R., et leur énoncé apparaîtra plusieurs fois sous des formes peut-être diverses.

La démonstration précédente montre qu'il sera possible dans ce cas de déterminer le minimum de relations nécessaires à la description et à l'implémentation des règlements qui ont été analysés.

4.5. Choix de la représentation graphique

Une fois assuré de la cohérence et de l'irredondance des résultats de l'analyse, on est amené à se poser le problème du choix d'une représentation par un graphe ET/OU.

Nous n'avons ni les critères, ni les moyens de déterminer automatiquement ce choix, cependant montrons comment on peut utiliser la propriété évoquée en 3.4 pour guider l'utilisateur dans ce choix.

Soit R l'ensemble de relations de dépendance résultat de l'analyse; on procède comme suit :

1) On vérifie qu'il n'y a pas d'incohérences
2) On construit la fermeture F_R de R
3) On retient les relations de F_R telles que la partie gauche contient uniquement des variables représentatives d'enseignements élémentaires, et la partie droite uniquement la variable représentative de l'étape; soit S l'ensemble ainsi construit.
4) On note σ la fonction booléenne dont les monômes sont les images des parties gauches des relations de S
5) On calcule la duale de σ, σ^* (cf. [13])
6) On évalue les coûts (cf. 3.4.1) comparés de σ et de σ^*
7) A ce stade on cherche à représenter la fonction sous sa forme la "moins coûteuse", compte-tenu du fait (cf. 3.4.2) que l'on passe facilement de la représentation de σ^* à la représentation de σ.

Exemple 7 : L'ensemble des relations de départ est donné dans l'exemple 1. La fermeture est donnée dans l'exemple 5.

L'ensemble S est donné dans l'exemple 6. On a alors :

$$\sigma = x_6x_8x_9x_{10} + x_6x_8x_9x_{11} + x_6x_8x_9x_{12} + x_6x_8x_9x_{13} + x_6x_8x_{10}x_{11} + x_6x_8x_{10}x_{12} + x_6x_8x_{10}x_{13} + x_6x_8x_{11}x_{12} + x_6x_8x_{11}x_{13} + x_6x_8x_{12}x_{13}$$

$$\sigma^* = x_6 + x_8 + x_9x_{10}x_{11}x_{12} + x_9x_{11}x_{12}x_{13} + x_9x_{10}x_{11}x_{13} + x_9x_{10}x_{12}x_{13} + x_{10}x_{11}x_{12}x_{13}$$

On sait alors qu'il existe une représentation de ces fonctions par un graphe ET/OU tel que :

Pour σ, le coût est $C = 18\alpha + 50\beta$

Pour σ^*, le coût est $C = 15\alpha + 36\beta$

De préférence on choisira donc de représenter σ^*.

L'extension de graphe ET/OU suivante est effectivement assez simple (figure 4.1) :

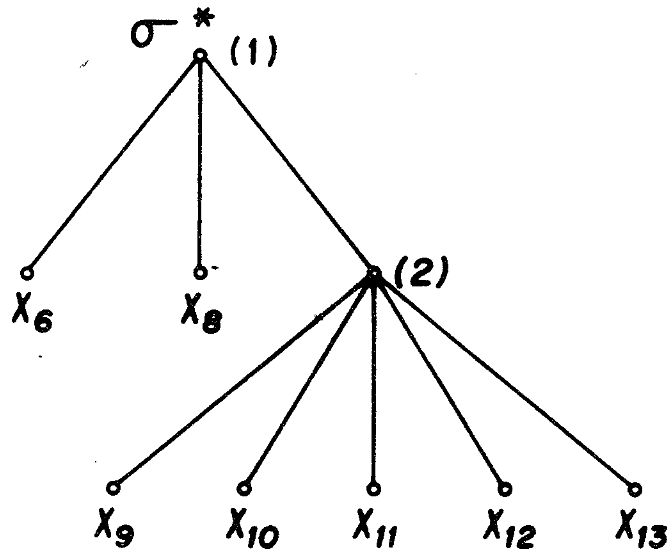


Figure 4.1 : <u>Représentation de σ^*</u>

Un programme opérationnel sous O.S. réalise cet ensemble d'opérations. On trouvera les sorties correspondantes en annexe (annexe A).

5. PRESENTATION GENERALE DES TRAITEMENTS

Ce chapitre reprend la présentation de la gestion des étudiants qui a été faite en 1.2, en y incorporant la représentation de l'organisation des enseignements que l'on a introduite dans le chapitre 3.

Il a semblé utile de compléter cette présentation par une description générale des traitements avant d'aborder de façon plus détaillée certains types d'algorithmes dans les chapitres suivants. Dans ce domaine les notations de [1] nous ont semblé bien adaptées car elles permettent de décrire la sémantique des traitements indépendamment de toute condition d'implémentation. De plus ces notations permettent d'exprimer commodément les opérations élémentaires correspondant au quantificateur universel (exemple : pour toute personne qui ...) et au quantificateur existentiel (exemple : si il existe une personne qui ...), ce que ne permettent pas les notations algorithmiques traditionnelles.

5.1. Description de la gestion des étudiants

Nous conservons donc les notations de [1] déjà introduites en 1.1. Avec elles, et compte-tenu des propositions de 1.3., les relations suivantes permettent de décrire l'identification d'un étudiant et son appartenance à un établissement (cf. 1.3.1.).

r1 = <u>rel</u>(individu,établissement,études-à = <u>facc</u>(0,3),étudiant=<u>facc</u>(1,∞))
r2 = <u>rel</u>(individu,no-étudiant,identificateur=<u>facc</u>(0,1),identifié=<u>facc</u>(0,1))
r3 = <u>rel</u>(no-étudiant,établissement,assignateur=<u>facc</u>(0,1),<u>facc</u>(1,∞))
r4 = <u>rel</u>(no-étudiant,établissement,réutilisateur=<u>facc</u>(0,2),<u>facc</u>(0,∞))

Du fait de la représentation choisie (cf. chapitre 3), on appelle "noeud" l'ensemble des noeuds des graphes obtention et inscription représentatifs des structures d'études.

On a vu (cf. chapitre 2) que ces noeuds peuvent être soit associés à des enseignements élémentaires, ou à des étapes, ou à des dérogations, soit nécessaires à la représentation par un graphe ET/OU des conditions de progression dans les différentes filières.

Pour décrire les graphes ET/OU représentatifs des conditions d'inscription, d'obtention partielle et d'obtention, on définit pour chacun la relation entre un noeud et ses successeurs ainsi que la relation qui associe à chaque noeud non terminal un entier.

On a donc :

r5=<u>rel</u>(noeud,noeud,successeur-inscription=<u>facc</u>(0,∞),prédécesseur-inscription=<u>facc</u>(0,∞))

r6=<u>rel</u>(noeud,noeud,successeur-obtention=<u>facc</u>(0,∞),prédécesseur-obtention=<u>facc</u>(0,∞))

r7=<u>rel</u>(noeud,noeud,successeur-obtention-partielle=<u>facc</u>(0,∞), prédécesseur-obtention-partielle=<u>facc</u>(0,∞))

r8=<u>rel</u>(noeud,entier,poids-inscription=<u>facc</u>(0,1),<u>facc</u>(0,∞))

r9=<u>rel</u>(noeud,entier,poids-obtention=<u>facc</u>(0,1),<u>facc</u>(0,∞))

r10=<u>rel</u>(noeud,entier,poids-obtention-partielle=<u>facc</u>(0,1),<u>facc</u>(0,∞))

Il n'est pas nécessaire de préciser la nature d'un noeud car :

- les noeuds associés à des étapes appartiennent à la fois au graphe inscription et au graphe obtention
- les noeuds associés à des enseignements élémentaires sont les noeuds terminaux du graphe obtention
- les noeuds associés à des dérogations sont des noeuds terminaux du graphe inscription.

Ces conditions sont suffisantes et nécessaires.

De façon analogue à ce qui a été fait en 1.2.2 on définit deux ensembles de triplets "suivi" et "acquis" chacun respectivement représentatif des relations ternaires.

suivi(x,y,t) <=> l'étudiant x suit le noeud y en l'année t
acquis(x,y,t)<=> l'étudiant x a acquis le noeud y en l'année t

Ce qui s'interprète par les relations suivantes :

r11=rel(suivi,noeud,inscrit-à = facc(1,1),facc(0,∞))
r12=rel(suivi,no-étudiant,inscrit = facc(1,1),facc(0,∞))
r13=rel(suivi,année,date-inscription = facc(1,1),facc(0,∞))
r14=rel(acquis,noeud,noeud-acquis = facc(1,1),facc(0,∞))
r15=rel(acquis,étudiant,acquis-par = facc(1,1),facc(0,∞))
r16=rel(acquis,année,date-acquis = facc(1,1),facc(0,∞))

Enfin, pour décrire l'organisation des établissements, reprenons les relations définies en 1.2.3 :

r17=rel(établissement,uer,constituant = facc(1,∞),facc(1,1))
f18=rel(uer,noeud,facc(1,∞),organisatrice = facc(0,1))
r19=rel(uer,noeud,facc(1,∞),utilisatrice = facc(0,∞))

L'ensemble de ces relations est schématisé page suivante (figure 5.1); on a rajouté dans ce schéma une relation entre un no-étudiant et un contexte:on verra (cf. 5.3) que cette relation permet de prendre en compte dans les traitements la situation d'un étudiant qui n'a pas encore acquis une étape mais a déjà acquis certains des enseignements élémentaires nécessaires à son obtention. On verra en 6.1 comment ce contexte est effectivement nécessaire à l'implémentation des algorithmes de validation.

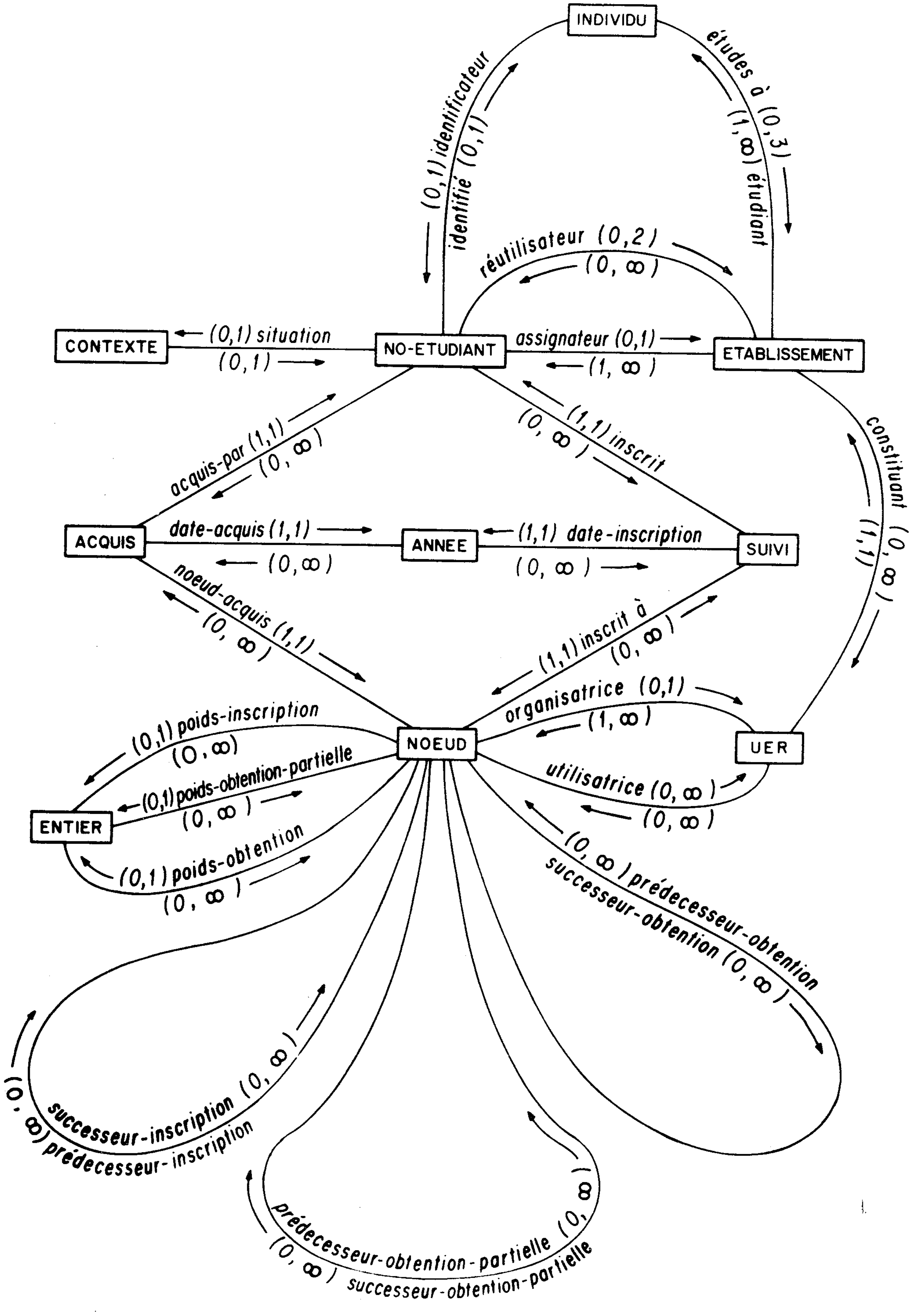


FIGURE 5.1. : Schéma pour la gestion des étudiants

5.2. Notations utilisées

Les concepteurs et utilisateurs de systèmes d'information n'ont pas su à ce jour définir un langage commun pour exprimer les opérations élémentaires de manipulation de données. Les langages de description algorithmique, de type pseudo-algol par exemple, s'avèrent difficilement utilisables car la formulation des opérations de création et de suppression ainsi que l'utilisation des quantificateurs universels et existentiels est extrêmement lourde.

Pour notre part, nous utilisons les notations proposées dans [1], en accord avec le choix fait en 1.1. On trouvera ci-dessous une énumération des opérateurs et instructions utilisés dans le paragraphe suivant.

a - $:\ni$ indique qu'une relation donnée existe entre deux objets
Exemple : acquis(n)$:\ni$ x — le noeud x est acquis par l'étudiant de numéro n.

b - $\leftarrow$ même sens que $:\ni$ mais réservé au cas où le cardinal maximum de la fonction d'accès est 1.
Exemple : identificateur(i) $\leftarrow$ n — l'individu i est identifié par le numéro d'étudiant n

c - $:\not\ni$ supprime une relation existante entre deux objets.
Exemple : successeur-inscription(x) $:\not\ni$ y — le noeud y n'est plus un successeur-inscription de x

d - l'évaluation d'une fonction d'accès se fait en écrivant le nom de la fonction suivi du nom d'un objet appartenant à son domaine.
Exemple : acquis(n) — ensemble des noeuds acquis par l'étudiant de numéro n.

e - est teste si un objet appartient à un ensemble ; le résultat du test est vrai ou faux.

Exemple : x est noeud — vrai si l'objet x est un noeud, faux sinon

f - ∈ teste si un objet appartient à l'ensemble résultant de l'évaluation d'une fonction d'accès.

Exemple : x ∈ acquis(n) — est vrai si x est un noeud acquis par l'étudiant de numéro n, faux sinon.

g - ∧ , ∨, ¬ opérateurs logiques correspondants à l'intersection, l'union, la négation et utilisés suivant les conventions habituelles.

h - ∃ quantificateur existentiel, a pour effet de bord d'affecter un nom à un objet délivré par une fonction d'accès.

Exemple : ∃ y ← organisatrice(x) est vrai s'il existe une UER organisatrice de x auquel cas cette UER est nommée y, faux sinon.

i - si ... alors ... fin
si ... alors ... sinon ... fin
instructions conditionnelles utilisées avec les conventions et la signification habituelles.

j - échec, succès noms externes des différentes valeurs d'une variable booléenne, ces valeurs résultent par effet de bord de l'application d'un opérateur à un opérande. Elles peuvent être testées et éventuellement affectées par l'utilisateur

Exemples: si succès alors ... fin
si x est noeud alors succès sinon échec fin

k - faire ... fin permet d'accéder et de traiter les différents objets délivrés par une fonction d'accès. refaire permet de "remonter" au niveau du faire, sortie permet de "forcer" la sortie de la boucle.
Des boucles faire ... fin pouvant être imbriquées, il faut paramétrer refaire et sortie afin d'indiquer le niveau d'imbrication auquel ils renvoient.

Exemple :

```
faire x ← inscrit(n)
      si (x ∈ acquis(n))
      alors sortie1
      sinon refaire1
      fin
fin
```

qui cherche parmi les noeuds x auxquels est inscrit l'étudiant de numéro n un noeud auquel il a réussi.

l- programme permet de nommer un ensemble d'instructions utilisant les opérateurs précédents; comme d'habitude les instructions qui constituent le programme suivent la définition des paramètres formels.

Quand un programme rend un résultat unique on utilise l'opérateur retourne.

Quand un programme délivre plusieurs résultats on utilise l'opérateur rendre.

Exemple :

```
descendant-obtention ← programme(x)
si ¬(x est noeud)
alors échec fin
faire y ← successeur-obtention(x)
      si ¬(y ∈ s)
      alors s :∋  y
            rendre(y)
            descendant-obtention(y)
      fin
      refaire1
fin
```

descendant-obtention est un programme qui calcule l'ensemble des noeuds descendant d'un noeud x dans le graphe obtention. L'ensemble s est nécessaire pour éviter d'énumérer plusieurs fois les noeuds appartenant à des cycles.

m - <u>accès</u>. L'exemple précédent montre un programme qui se comporte comme une fonction d'accès. Il est commode dans ces conditions de pouvoir le définir en tant que tel afin de pouvoir lui appliquer les opérateurs ∈, ←,...

On écrira donc :

<u>accès</u> (descendant-obtention) ← prog(x)
...

On pourra alors écrire :

<u>faire</u> x ← descendant-obtention(y)
    ...
    <u>refaire1</u>
<u>fin</u>

et tout aussi bien :

<u>si</u> x ∈ descendant-obtention(y) <u>alors</u> ... <u>fin</u>

n - <u>ctx</u>. Les valeurs d'un même identificateur sont différentes suivant le contexte dans lequel elles sont évaluées. Un contexte peut être lié à l'exécution d'un programme, mais ce n'est pas nécessaire. L'opérateur <u>ctx</u> permet de générer un nouveau contexte. Si c est un contexte une opération peut se faire <u>relativement à</u> ce contexte.

5.3. Description des traitements

Dans cette description nous ne tenons pas compte des conditions d'obtention partielle (cf. 2.2.1) les mécanismes d'obtention partielle étant analogues aux mécanismes d'obtention.

En premier lieu nous allons montrer que la nature des noeuds (étape , enseignement élémentaire ou dérogation) résulte uniquement de leur situation dans les graphes inscription et obtention, ce qui justifie qu'elle n'ait pas été décrite comme une caractéristique associée à chaque noeud.

Pour cela nous allons introduire trois fonctions d'accès "étape", "dérogation" et "enseignement" qui auront pour effet de calculer au vu des graphes inscription et obtention les ensembles des noeuds ayant la propriété correspondante.

On remarquera que ces fonctions d'accès n'ont pas de paramètres puisqu'elles délivrent un ensemble associé à une propriété implicitement nommée par la fonction elle-même.

Une étape est un noeud pour lequel sont définies à la fois les conditions d'inscription et d'obtention et qui lui-même n'intervient pas dans des conditions d'obtention. Naturellement une étape est organisée par une U.E.R., on décrit donc comme suit la fonction d'accès "étape" :

```
accès(étape) ← programme
faire x ← noeud
    si((∃y ← organisatrice(x))∧(∃z ← successeur-obtention(x))
     ∧ (∃t ← successeur-inscription(x))∧¬(∃u ← prédécesseur-obtention(x))
    alors rendre(x)
    fin
    refaire1
fin
```

Une dérogation est un noeud terminal du graphe inscription, c'est-à-dire qu'il conditionne certaines inscriptions sans être conditionné lui-même.

```
accès(dérogation) ←programme
faire x ← noeud
    si ((∃y←prédécesseur-inscription(x))∧¬(∃t ← successeur-inscription(x)))
    alors rendre(x)
    fin
    refaire 1
fin
```

Un enseignement élémentaire est un noeud terminal du graphe obtention, c'est-à-dire qu'il conditionne l'obtention de certaines étapes sans être décomposé lui-même en unités plus petites. Bien évidemment un enseignement est organisé par une UER :

```
accès(enseignement) ← programme
faire x ← noeud
    si((∃y ← organisatrice(x))∧(∃z ← prédécesseur-obtention(x)))
      ∧ ¬(∃t ← successeur-obtention(x))
    alors rendre(x)
    fin
    refaire1
fin
```

Enseignement est entendu ici au sens d'enseignement élémentaire; un enseignement est organisé par une U.E.R. et intervient dans des conditions d'obtention.

Décrivons maintenant les procédures d'inscription; nous avons vu qu'on distinguait la première inscription qui permettait d'identifier l'individu par un numéro étudiant des inscriptions suivantes où ce numéro était réutilisé.

Lors de la première inscription sont enregistrées les informations qui décrivent l'individu; nous ne décrirons pas ici ces opérations qui sont élémentaires.

On a vu que l'ensemble des numéros d'étudiant est considéré comme préexistant à toute opération d'identification, ainsi tous les numéros existent mais seuls certains identifient des individus. Une nouvelle fonction d'accès va nous permettre d'attribuer un numéro :

```
accès(identificateur) ← programme
faire x ← no-étudiant
  |   si (∃u ← identifié(x))
  |   alors retourne(x)
  |   sinon refaire 1
  |   fin
fin
```

Pour inscrire pour la première fois l'individu x il faut connaître l'étape t qu'il postule, la dérogation z qui lui permet de s'inscrire (c'est nécessairement une décision extérieure au système des enseignements), et l'année y.

Si z permet effectivement de s'inscrire à t, l'inscription est possible et la procédure rend le numéro d'étudiant affecté à x.

```
première-inscription ← programme(x,y,z,t)
si ¬((x est individu) ∧ (y est année) ∧ (z ∈ dérogation) ∧ (t ∈ étape)) alors échec fin
u ← identification          /u est un nouveau numéro d'étudiant/
situation(u) ← ctx          /création d'un contexte associé à u/
v ← générer(acquis)       acquis-par(v) ← u
noeud-acquis(v) ← z         /création d'un acquis correspondant à la dérogation dont
                             fait état l'étudiant/
si validinscr(u,t)= succès /vérification que z permet de s'inscrire à t/
alors w ← générer(suivi)   /inscription possible, on la crée/
|     inscrit(w) ← u   date-inscription(w) ← y     inscrit-à(w) ← t
|     identificateur(x) ← u  /u est le numéro d'étudiant de x/
|     assignateur(u) ← inv(constituant(organisatrice(t)))
|     étudiant(inv(constituant(organisatrice(t)))) ∋ x
|     /l'établissement assignateur de u est celui dans lequel se trouve l'U.E.R.
|      organisatrice de t, x devient étudiant de cet établissement/
|     retourne(u)
sinon tuer(v)                         /inscription impossible/
|     tuer(situation(u))
|     échec
fin
```

N.B. : *On note entre "/" les commentaires nécessaires à la bonne compréhension des programmes.*

La procédure de première inscription a pour effet de générer un contexte associé au numéro d'étudiant qui identifie x. C'est sous ce contexte qui reflète la situation scolaire de l'individu que seront évalués les effets de chaque nouvelle acquisition. (Observer à cet égard le comportement des procédures de validation).

Définissons la procédure de validation des conditions d'inscription. Cette procédure vérifie que l'étudiant u a le droit de s'inscrire à l'étape t.

```
validinscr ← programme(u,t)
si ¬((u est no-étudiant) ∧ (t est noeud))
alors échec fin
faire
|   faire z ← successeur-inscription(t)
|   |   si (z ∈ noeud-acquis(inv(acquis-par(u))) ∨ validinscr(u,z)=succès)
|   |   alors poids-inscription(t) ← poids-inscription(t)-1 relativement à situation(u)
|   |   |/le successeur z de t est valide soit que ce soit un acquis de u, soit que
|   |   | le mécanisme de validation l'ait déterminé comme tel. On décrémente le poids-
|   |   | inscription de t d'une unité, dans le contexte de la situation de u/
|   |   |   si (poids-inscription(t) = 0 relativement à situation(u))
|   |   |   alors  retourne (succès)
|   |   |   |      /la validation de t sous le contexte de u a réussi/
|   |   |   |      sortie 2
|   |   |   fin
|   |   fin
|   |   refaire 1        /on passe au successeur suivant/
|   fin
|   retourne (échec)     /on a testé tous les successeurs sans succès/
fin
```

Exemple : Donnons un exemple de déroulement de la procédure de première inscription.

L'étudiant GILBERT s'inscrit pour la première fois à Grenoble en 1973. Il veut passer la licence d'enseignement de mathématiques et possède un D.U.E.S. obtenu dans une autre université.

On utilise la description de la licence d'enseignement de mathématiques donnée en 2.3 et la représentation graphique correspondante (cf. 3.3).

première-inscription(GILBERT,1973,e_7,e_5)
65431 ← identification /65431 est un nouveau no-étudiant/
situation(65431) ← ctx / création d'un contexte associé à 65431/
v ← générer(acquis)
acquis-par(v) ← 65431
noeud-acquis(v) ← e_7
/création d'un acquis correspondant à la dérogation dont fait état l'étudiant/
validinscr(65431,e_5)
/appel de la procédure de validation "validinscr"/
z ← e_1 /premier successeur inscription de e_5/

e_1 ∉ noeud-acquis(<u>inv</u> acquis-par(65431))
/on rappelle validinscr sur e_1/
valindinscr(65431,e_1)
/e_1 n'a pas de successeurs/ retourne (échec)
refaire 1 /on passe au successeur suivant/
z ← e_2
/même situation que ci-dessus/
refaire 1 /on passe au successeur suivant/
z ← e_3
/même situation que ci-dessus/
refaire 1 /on passe au successeur suivant/
z ← e_4
/même situation que ci-dessus/
refaire 1 /on passe au successeur suivant/
z ← e_7
e_7 ∈ noeud-acquis(<u>inv</u> acquis-par(65431))
poids-inscription(e_5) ← poids-inscription(e_5)-1 <u>relativement-à</u> situation(65431)
/on a alors : poids-inscription(e_5)=0 <u>relativement-à</u> situation(65431)/
retourne(succès)
/on sort de "validinscr", e_7 permet bien de s'inscrire à e_5/

```
w ← générer(suivi)      /inscription possible on la crée/
inscrit(w) ← 65431
date-inscription(w) ← 1973
inscrit-à(w) ← e5
identificateur(GILBERT) ← 65431
assignateur(65431) ← inv(constituant(organisatrice(e5)))
étudiant(inv(constituant(organisatrice(e5)))) ← GILBERT
   /mise-à-jour des différentes relations/
retourne(65431)
/on renvoie à l'utilisateur le numéro d'étudiant qui a été
 affecté à GILBERT et qui permettra désormais de l'identifier/
```

Définissons maintenant la procédure d'inscription pour les inscriptions ultérieures.

On connaît le numéro d'étudiant u de l'individu qui s'inscrit à l'étape t pour l'année y.

```
inscription-ultérieure ← programme(u,y,t)
si ¬((u est no-étudiant)∧(y est année)∧(t∈étape)∧(∃x identifié(u)))
alors échec fin
si validinscr(u,t) = succès
alors     /u a le droit de s'inscrire à t/
   | v ← générer(suivi) inscrit(v) ← u  date-inscription(v) ← y
   | inscrit-à(v) ← t
   | e ← inv(constituant(organisatrice(u)))
   |     /désigne l'établissement qui dispense l'étape t/
   | si ¬((e = assignateur(u)) ∨ (e ∈ réutilisateur(u)))
   | alors    /u s'est inscrit dans un nouvel établissement/
   |    |   réutilisateur(u):∋ e
   |    |   étudiant(e) :∋ identifié(u)
   | fin
sinon échec    /inscription refusée/
fin
```

Il nous reste à définir une procédure qui permette d'enregistrer les résultats acquis par un étudiant. Si ces résultats concernent des enseignements, on en évalue immédiatement les conséquences grâce à la procédure validobt.

```
résultat ← programme(u,y,t)
si ¬(((u est no-étudiant)∧(y est année))∧((t∈dérogation)∨(t∈enseignement)))
alors échec fin
  v ← générer(acquis)        /on crée un nouvel acquis pour x/
  acquis-par(v) ← u   date-acquis(v) ← y   noeud-acquis(v) ← t
si (t ∈ enseignement)
alors validobt(u,y,t)   /évaluation des conséquences de l'obtention de t/
fin
```

La procédure validobt met à jour la situation d'un étudiant qui vient de réussir à un enseignement. Cette mise à jour implique d'évaluer dans le graphe obtention toutes les conséquences de la nouvelle acquisition.

```
validobt ← programme(u,y,t)
si ¬((u est no-étudiant)∧(t est noeud)∧(y est année)) alors échec fin
faire  z ← prédécesseur-obtention(t)
|     poids-obtention(z)←poids-obtention(z)-1 relativement-à situation(u)
|         /on modifie le poids des prédécesseurs de t/
|    si (poids-obtention(z)=0 relativement-à situation(u))
|    alors si (z ∈ étape) /l'étape z est obtenue on génère
|    |                      l'acquis correspondant/
|    |     alors v ← générer(acquis)
|    |     |     acquis-par(v) ← u   noeud-acquis(v) ← z
|    |     |     date-acquis(v) ← y
|    |     sinon validobt(u,y,z)  /z n'est qu'un noeud intermé-
|    |     |                      diaire, on évalue les consé-
|    |     |                      quences de sa validation/
|    |     fin
|    fin
|    refaire 1  / on évalue toutes les conséquences de l'obtention de y
|                 qui peut éventuellement intervenir dans plusieurs filières,
fin               donc entraîner l'obtention de plusieurs étapes/
```

6. LES ALGORITHMES DE VALIDATION

On a vu dans le chapitre précédent (cf. 5.3) le rôle important que jouent les algorithmes de validation dans l' application qui nous intéresse. La description qui en a été faite (cf. les procédures "validinscr" et "validobt") se voulait alors indépendante de toute condition d'implémentation; dans ce chapitre nous aborderons au contraire ces algorithmes en tenant compte de ces conditions et des possibilités qu'offrent les systèmes de base de données.

6.1. Les différents types d'algorithmes

Soit une extension de graphe ET/OU, les algorithmes de validations permettent de résoudre le problème suivant (cf. figure 6.1):

le noeud N_0 est-il valide étant donné que les noeuds terminaux $N_1, N_2, \ldots, N_n$ le sont ?

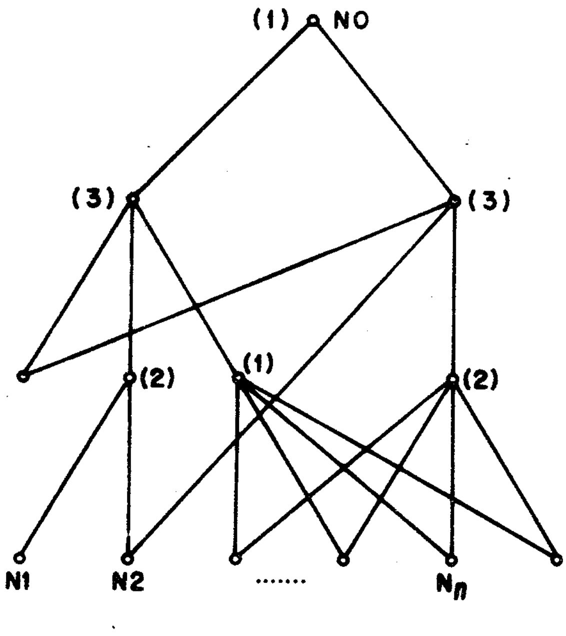


Figure 6.1. : N_0 est-il valide sachant que N_1, N_2, N_n le sont ?

Pour résoudre ce problème il suffit d'appliquer récursivement la propriété :

$$N_i \iff \exists \lambda_i \text{ successeurs de } N_i \text{ valides} \qquad (1)$$

où λ_i désigne le poids associé à chaque noeud (cf. 3.2)

On peut effectuer la validation soit en parcourant le graphe à partir des noeuds terminaux valides en cherchant les prédécesseurs valides jusqu'à atteindre N_0 on parlera d'<u>algorithmes ascendants</u>, soit à partir de N_0 en explorant ses descendants jusqu'à ce que l'on rencontre les noeuds $N_1, N_2, \ldots, N_n$ on parlera alors d'<u>algorithmes descendants</u>.

Le parcours du graphe peut se faire suivant deux types de stratégies (cf. [2]) soit en profondeur, soit en largeur.

Enfin on peut envisager de combiner une stratégie ascendante et une stratégie descendante, soit en les appliquant alternativement soit, si le matériel le permet, en les faisant fonctionner en parallèle.

Pour notre part, nous décrivons en détail les algorithmes descendant en profondeur (6.2), descendant en largeur (6.3), ascendant en profondeur (6.4) et ascendant en largeur (6.5). On trouve dans [2] une description des algorithmes descendant en profondeur et descendant en largeur dans le cas de validation dans un <u>arbre ET/OU</u>.

Dans notre cas, du fait que l'on travaille sur un graphe, il est nécessaire de marquer les chemins déjà explorés afin de ne pas les réemprunter; techniquement ceci peut être réalisé de deux façons différentes :

- La première solution consiste à associer à chaque noeud une variable informant de l'état du noeud dans la validation en cours; cet état est caractérisé par le nombre de successeurs qu'il reste à valider pour que le noeud soit valide.

Si l'on envisage que plusieurs utilisateurs puissent effectuer des validations simultanément, il convient alors d'associer à chaque noeud autant de variables qu'il y a d'utilisateurs, chaque variable caractérisant alors l'état du noeud dans la validation en cours pour un utilisateur donné.

- La deuxième solution consiste à mettre dans une pile les noeuds qui interviennent dans la validation ainsi que la variable d'état définie plus haut. C'est cette solution que nous retiendrons et pour chaque type d'algorithme nous donnerons un exemple d'activation accompagné des états successifs de la pile.

Dans le cas d'une utilisation par plusieurs utilisateurs simultanément il conviendrait de définir une pile par utilisateur.

Un des avantages de cette solution par rapport à la précédente tient au fait que l'on accède au graphe en lecture seulement. En cas de casse du système d'exploitation par exemple seules les piles associées à chaque utilisateur risquent d'être dans un état incohérent; il suffit de les réinitialiser correctement et de recommencer les validations interrompues. La solution précédente, elle, nécessiterait dans un cas analogue de réinitialiser les variables d'état associées à chaque noeud et cette opération serait du fait du nombre des noeuds beaucoup plus coûteuse.

On remarquera que les variables qui caractérisent l'état des noeuds en cours de validation constituent une description du contexte associé à l'étudiant pour le compte duquel la validation est effectuée (cf. 5.3).

Les algorithmes que nous donnons travaillent donc sur une pile. On désignera indistinctement par N_i un noeud du graphe et l'élément de pile qui le contient. Enfin on notera ETAT(N_i) la variable d'état associée au noeud N_i dans la pile. On rappelle que cette variable contient à un instant donné <u>le nombre de successeurs de N_i qu'il reste à valider</u> pour que N_i soit valide.

6.2. Algorithme descendant en profondeur

On initialise la pile en empilant dans l'ordre le noeud à valider N_0 et les noeuds terminaux valides $N_{i_1}, N_{i_2}, \ldots, N_{i_n}$.

On définit comme suit la procédure récursive EVALUE(N_i) :

Procédure récursive EVALUE(N_i)

P1. Pour tout successeur N_k de N_i tant que ETAT(N_i) > 0
- P1.1. Si N_k est dans la pile alors aller à P1.4./successeur déjà évalué/
- P1.2. Si N_k n'a pas de successeurs aller à P1.6./successeur non validé/
- P1.3. Mettre N_k dans la pile /évaluation du successeur/ EVALUE(N_k)
- P1.4. Si ETAT(N_k) $\neq$ 0 aller à P1.6./successeur non validé/
- P1.5. ETAT(N_i) $\leftarrow$ ETAT(N_i)-1
- P1.6. $\emptyset$

<u>N.B.</u> : *On note par $\emptyset$ l'instruction vide.*

La validation de N_o se fait en initialisant la pile comme indiqué ci-dessus et en appelant EVALUE(N_o).

(0)	(1)	(2)	(3)	(4)	(5)
N0 (1)	N0 (1)	N0 (1)	N0 (1)	N0 (1)	N0 (1)
N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)
N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)
N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)
N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)
	N1 (3)	N1 (2)	N1 (2)	N1 (2)	N1 (2)
			N3 (2)	N3 (1)	N3 (1)

(6)	(7)	(8)	(9)	(10)	(11)
N0 (1)	N0 (1)	N0 (1)	N0 (1)	N0 (1)	N0 (1)
N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)
N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)
N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)
N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)
N1 (2)	N1 (1)	N1 (1)	N1 (1)	N1 (1)	N1 (1)
N3 (1)	N3 (1)	N3 (1)	N3 (1)	N3 (1)	N3 (1)
N6 (1)	N6 (0)	N6 (0)	N6 (0)	N6 (0)	N6 (0)
		N11 (3)	N11 (2)	N11 (2)	N11 (1)
					N12 (2)

(12)	(13)	(14)
N0 (1)	N0 (1)	N0 (0)
N2 (0)	N2 (0)	N2 (0)
N4 (0)	N4 (0)	N4 (0)
N7 (0)	N7 (0)	N7 (0)
N9 (0)	N9 (0)	N9 (0)
N1 (1)	N1 (1)	N1 (1)
N3 (1)	N3 (1)	N3 (1)
N6 (0)	N6 (0)	N6 (0)
N11 (1)	N11 (1)	N11 (0)
N12 (1)	N12 (1)	N12 (0)

Figure 6.2. : Evolution de la pile (cf. 6.3)

L'état de chaque noeud est noté entre parenthèses à côté de son identificateur.

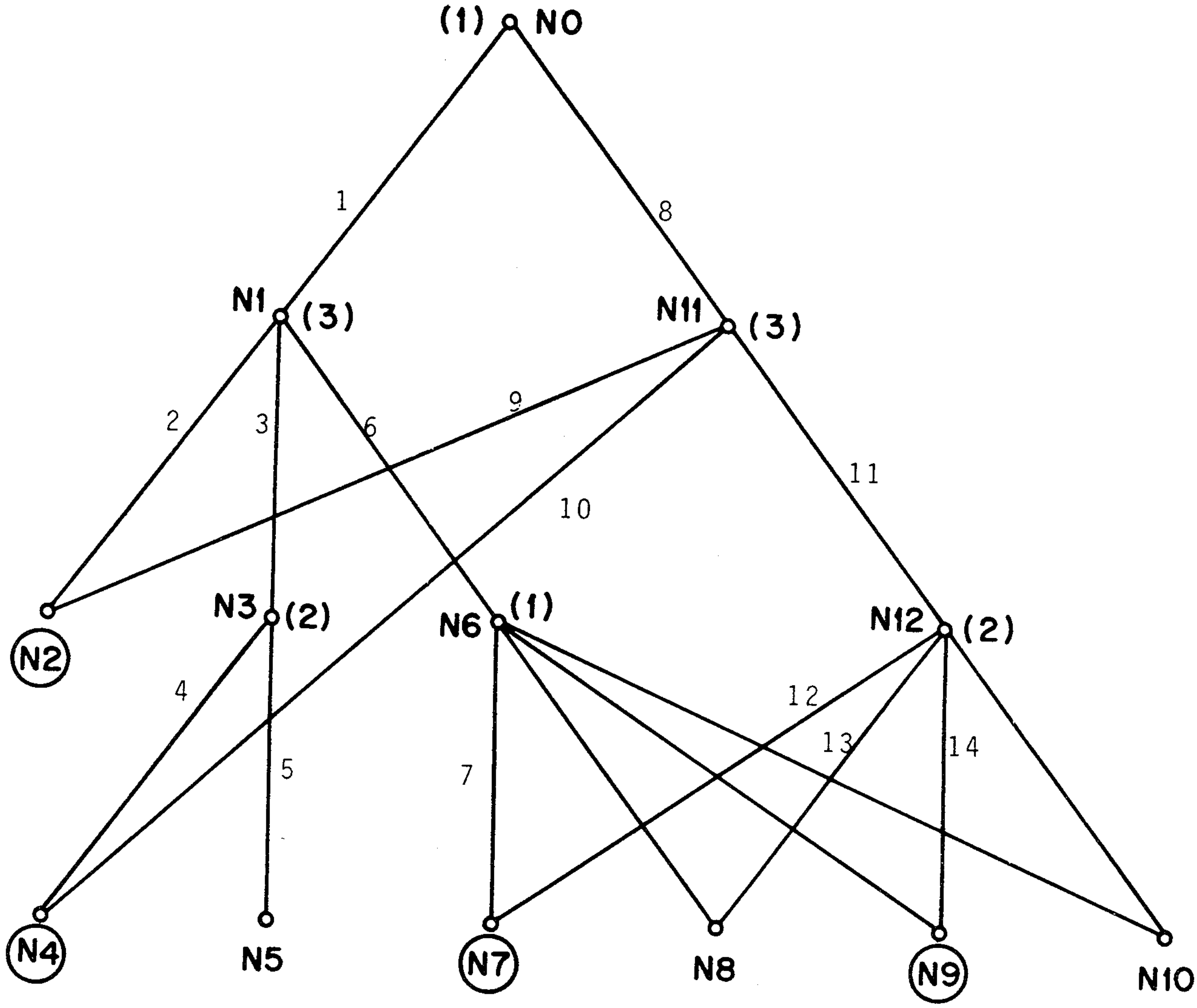


FIGURE 6.3. : Algorithme descendant en profondeur

Exemple de comportement de l'algorithme :

Les noeuds valides donnés sont N2,N4,N7,N9, dans un cercle sur la figure. Chaque phase de la progression dans le graphe est repérée par un numéro sur l'arête correspondante. Le contenu de la pile à ce stade est donné page précédente et identifié par le même numéro.

6.3. Algorithme descendant en largeur

Contrairement à l'exploration en profondeur, l'exploration en largeur implique d'utiliser une pile même si l'on travaille sur des arbres uniquement. Il faut en effet parcourir tous les successeurs d'un même noeud avant d'entreprendre le parcours des graphes issus de ces successeurs.

Enfin, nous avons besoin d'une procédure récursive VALIDE(N_i) dont le paramètre d'appel est un noeud valide contenu dans la pile ; cette procédure essaie de valider les prédécesseurs de N_i contenus dans la pile :

Procédure récursive : VALIDE (N_i ≡ un noeud valide dans la pile)

P1. Pour tout prédécesseur N_k de N_i contenu dans la pile
  P1.1. Si ETAT(N_k) = 0 aller à P1.4 (prédécesseur précédemment validé)
  P1.2. ETAT(N_k) ← ETAT(N_k) - 1
  P1.3. Si ETAT(N_k) = 0 alors VALIDE(N_k)
  P1.4. Ø

La pile est initialisée en empilant dans l'ordre les noeuds terminaux valides $N_{i_1}, N_{i_2}, \ldots, N_{i_n}$ et le noeud à valider N_0; un index STINDEX désigne le noeud de la pile en cours de validation. STINDEX est initialisé avec la valeur n+1. On définit comme suit la procédure de validation :

P1. STINDEX ← n
P2. Si ETAT(N_0) = 0 alors aller à P7 /validation effectuée/
P3. STINDEX ← STINDEX+1
P4. Si pas d'élément de rang STINDEX dans la pile aller à P7 /validation impossible/
P5. Soit N_i le noeud de rang STINDEX dans la pile
  Pour tout successeur N_k de N_i tant que ETAT(N_i) > 0

P5.1 Si N_k n'est pas dans la pile alors /successeur pas encore évalué/mettre N_k dans la pile. Aller à P5.5.

P5.2 Si ETAT(N_k) $\neq$ 0 aller à P5.5 /successeur non validé/

P5.3 ETAT(N_k) $\leftarrow$ ETAT(N_k)-1 /successeur validé/

P5.4 Si ETAT(N_k) = 0 alors VALIDE(N_i)

P5.5 Ø

P6. Aller à P2

P7. Ø.

(0)	(1)	(2)	(3)	(4)	(5)
N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)
N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)
N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)
N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)
N0 (1)	→ N0 (1)	→ N0 (1)	N0 (1)	N0 (1)	N0 (1)
	N1 (3)	N1 (3)	→ N1 (2)	→ N1 (2)	→ N1 (2)
		N11 (3)	N11 (2)	N11 (2)	N11 (2)
				N3 (2)	N3 (2)
					N6 (1)

(6)	(7)	(8)	(9)	(10)	(11)
N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)	N2 (0)
N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)	N4 (0)
N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)	N7 (0)
N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)	N9 (0)
N0 (1)	N0 (1)	N0 (1)	N0 (1)	N0 (1)	N0 (1)
N1 (2)	N1 (2)	N1 (2)	N1 (2)	N1 (2)	N1 (2)
→ N11 (2)	→ N11 (1)	→ N11 (1)	N11 (1)	N11 (1)	N11 (1)
N3 (2)	N3 (2)	N3 (2)	→ N3 (1)	→ N3 (1)	N3 (1)
N6 (1)	N6 (1)	N6 (1)	N6 (1)	N6 (1)	→ N6 (0)
		N12 (2)	N12 (2)	N12 (2)	N12 (2)
				N5 (Ø)	N5 (Ø)

(12)	(13)	(14)
N2 (0)	N2 (0)	N2 (0)
N4 (0)	N4 (0)	N4 (0)
N7 (0)	N7 (0)	N7 (0)
N9 (0)	N9 (0)	N9 (0)
N0 (1)	N0 (1)	N0 (0)
N1 (1)	N1 (1)	N1 (1)
N11 (1)	N11 (1)	N11 (0)
N3 (1)	N3 (1)	N3 (1)
N6 (0)	N6 (0)	N6 (0)
→ N12 (1)	→ N12 (1)	→ N12 (0)
N5 (Ø)	N5 (Ø)	N5 (Ø)
	N8 (Ø)	N8 (Ø)

Figure 6.4. : Evolution de la pile (cf. 6.5)

- Les noeuds terminaux non valides ont un état indéfini noté Ø; pour les autres noeuds l'état est noté entre parenthèses.
- La flèche à gauche de la pile désigne l'élément de rang STINDEX
- Les flèches à droite de la pile indiquent les noeuds validés récursivement par VALIDE.

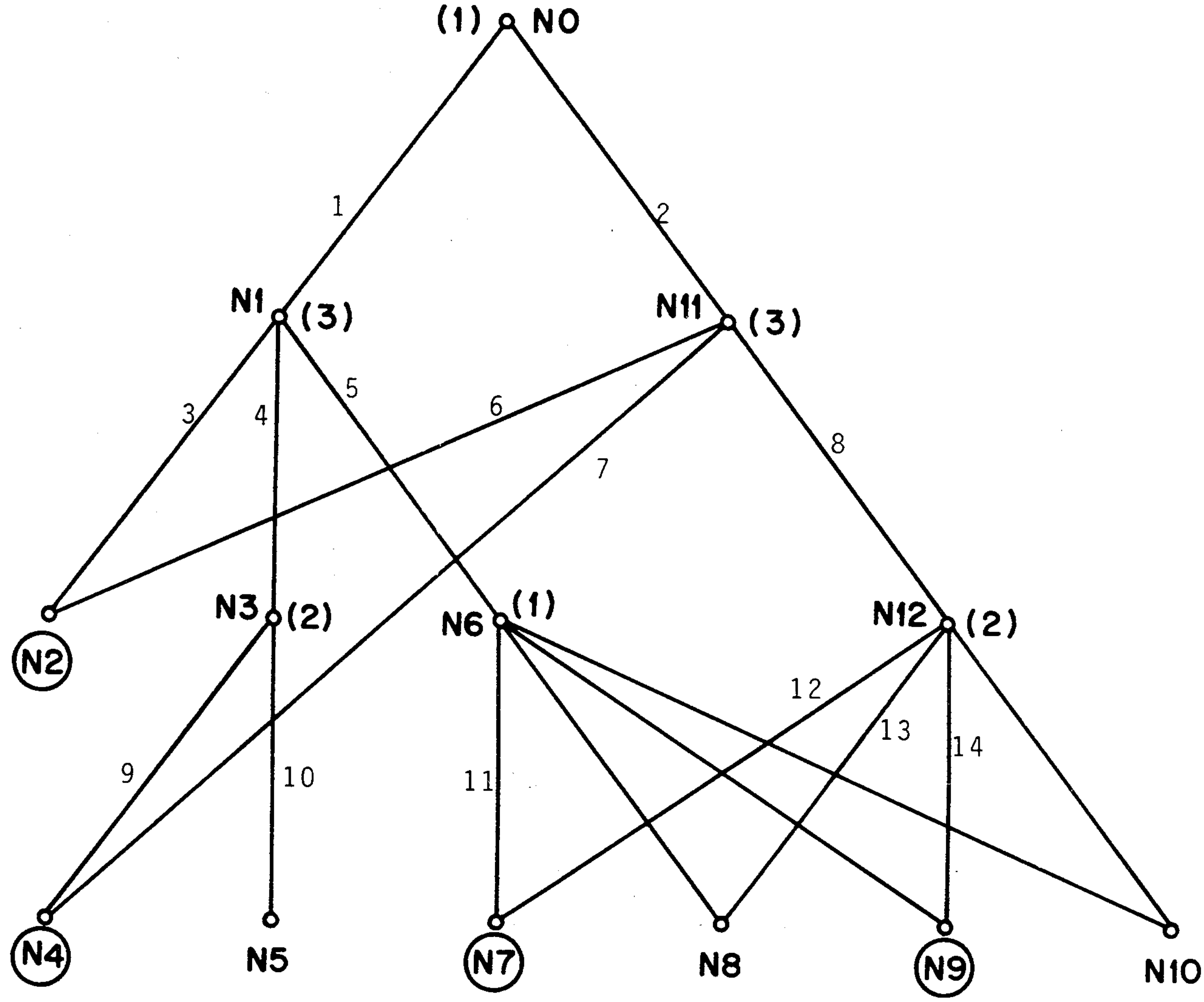


FIGURE 6.5. : Algorithme descendant en largeur

Comme pour la figure 6.2 les noeuds valides donnés sont notés dans un cercle. Chaque phase de la progression dans le graphe est repérée par un numéro sur l'arête correspondante. Le contenu de la pile à ce stade est donné page précédente et identifié par le même numéro.

6.4. Algorithme ascendant en profondeur

A partir des noeuds terminaux valides donnés les algorithmes ascendants déterminent les prédécesseurs valides, ils opèrent récursivement jusqu'à ce que la validation de N_0 soit effectuée, ou jusqu'à ce que l'impossibilité de valider N_0 soit établie.

On définit la procédure récursive VALIDE ($N_i \equiv$ un noeud dans la pile) comme suit :

Procédure récursive VALIDE(N_i)

P1. Pour tout prédécesseur N_k de N_i

P1.1. Si N_k est dans la pile aller à P1.3

P1.2. Mettre N_k dans la pile /prédécesseur non contenu dans la pile/

Aller à P1.5

P1.3. Si ETAT(N_k) = 0 aller à P1.5 /prédécesseur déjà validé/

P1.4. ETAT(N_k) ← ETAT(N_k)-1

Si ETAT(N_k) = 0 alors VALIDE(N_k)

P1.5. Ø

L'initialisation de la pile et l'activation de la procédure se font suivant le schéma suivant :

P1. /Initialisation de la pile/
Mettre N_0 dans la pile
Mettre $Ni_1, Ni_2, \ldots, Ni_n$ dans la pile
P2. STINDEX ← 2
P3. Si ETAT(N_0) = 0 alors aller à P7 /Validation effectuée/
P4. Soit N_i le noeud correspondant à l'élément de rang STINDEX dans la pile, VALIDE(N_i)
P5. Si STINDEX = n+2 alors aller à P7 /validation impossible/
P6. STINDEX ← STINDEX+1 Aller à P3 /Itération/
P7. Ø

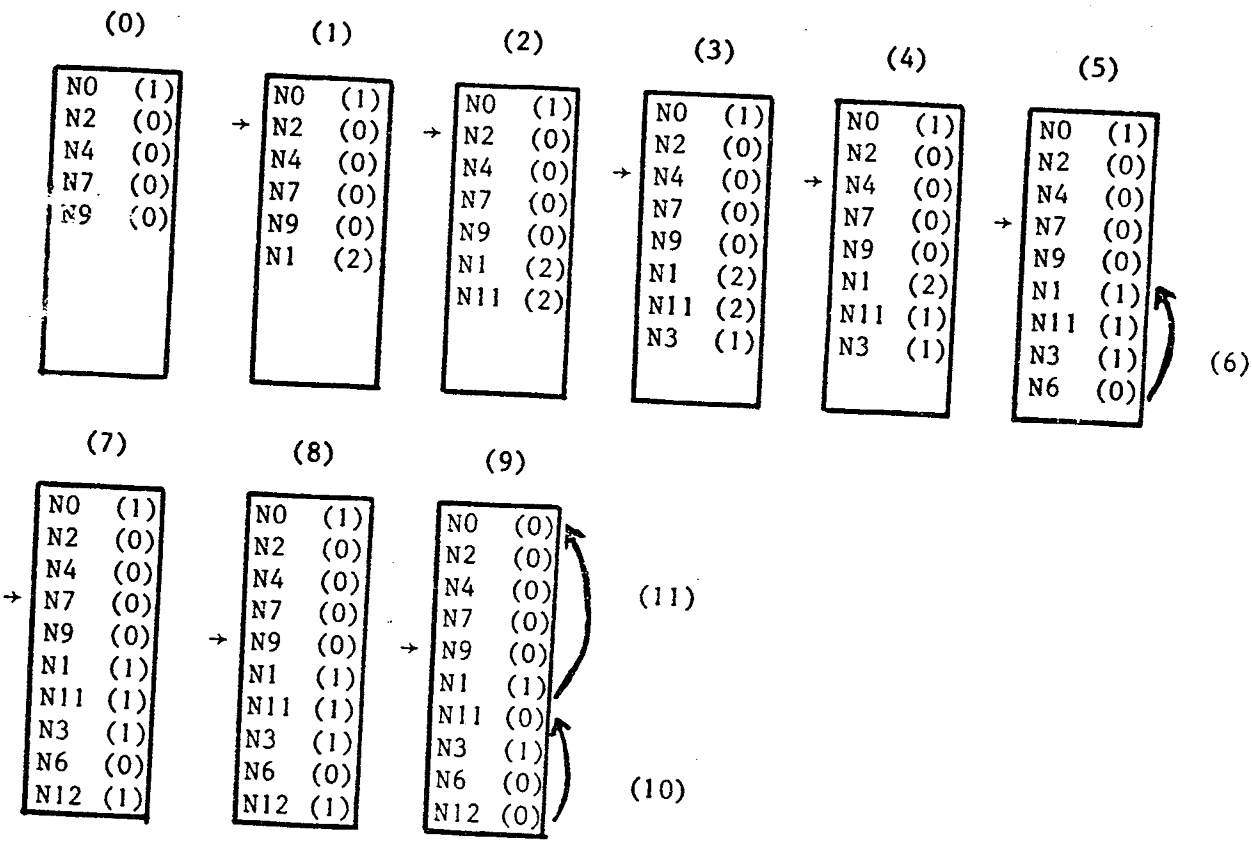


Figure 6.5. Evolution de la pile (cf. 6.7)

L'état de chaque noeud est noté entre parenthèses.
La flèche à gauche de la pile désigne l'élément de rang STINDEX
Les flèches à droite de la pile indiquent les noeuds validés récursivement par VALIDE.

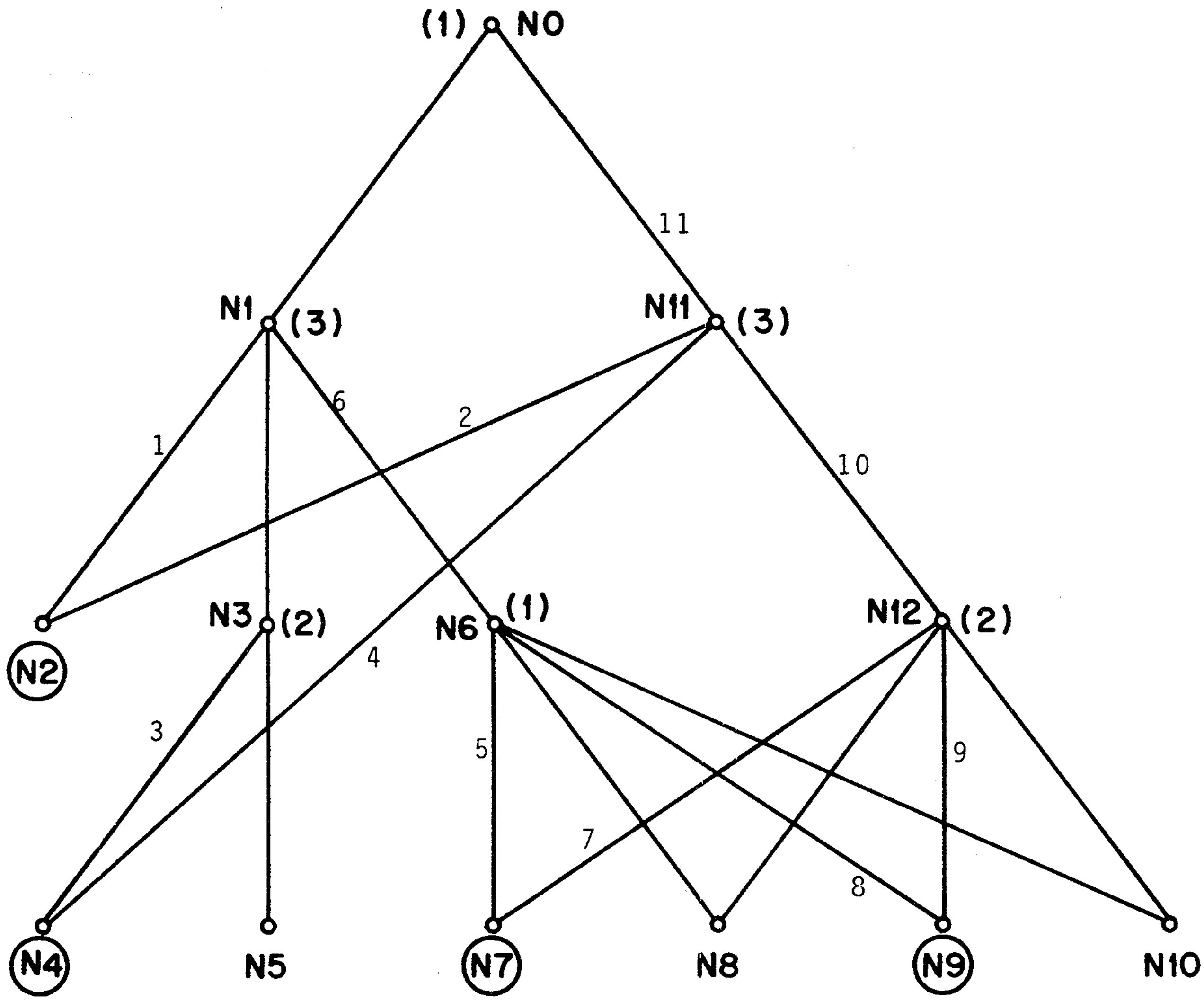


FIGURE 6.7. : Algorithme ascendant en profondeur

Comme pour la figure 6.2 les noeuds valides donnés sont notés dans un cercle. Chaque phase de la progression dans le graphe est repérée par un numéro sur l'arête correspondante. Le contenu de la pile à ce stade est donné page précédente, identifié par le même numéro.

6.5. Algorithme ascendant en largeur

On utilisera cette fois deux piles, l'une contenant les noeuds déjà validés que l'on appellera STVAL, l'autre contenant les noeuds dont la validation est en cours, que l'on appellera STENC. On utilisera l'algorithme suivant :

P1. Mettre N_0 dans la pile STENC
Mettre Ni_1, Ni_2,..., Ni_n dans la pile STVAL /Initialisation/
P2. STINDEX $\leftarrow$ 1
P3. Si ETAT(N_0) = 0 alors aller à P7 /Validation effectuée/
P4. S'il existe un élément de rang STINDEX dans la pile STVAL
alors soit N_i le noeud correspondant sinon aller à P7
/Validation impossible/
P5. Pour tout prédécesseur N_k de N_i
P5.1. Si N_k est dans la pile STVAL aller à P5.5
/Prédécesseur déjà validé/
P5.2. Si N_k n'est pas dans la pile STENC mettre N_k dans STENC
P5.3. ETAT(N_k) $\leftarrow$ ETAT(N_k)-1
P5.4. Si ETAT(N_k) = 0 alors mettre N_k dans STVAL
P5.5. Ø
P6. STINDEX $\leftarrow$ STINDEX+1
Aller à P3
P7. Ø

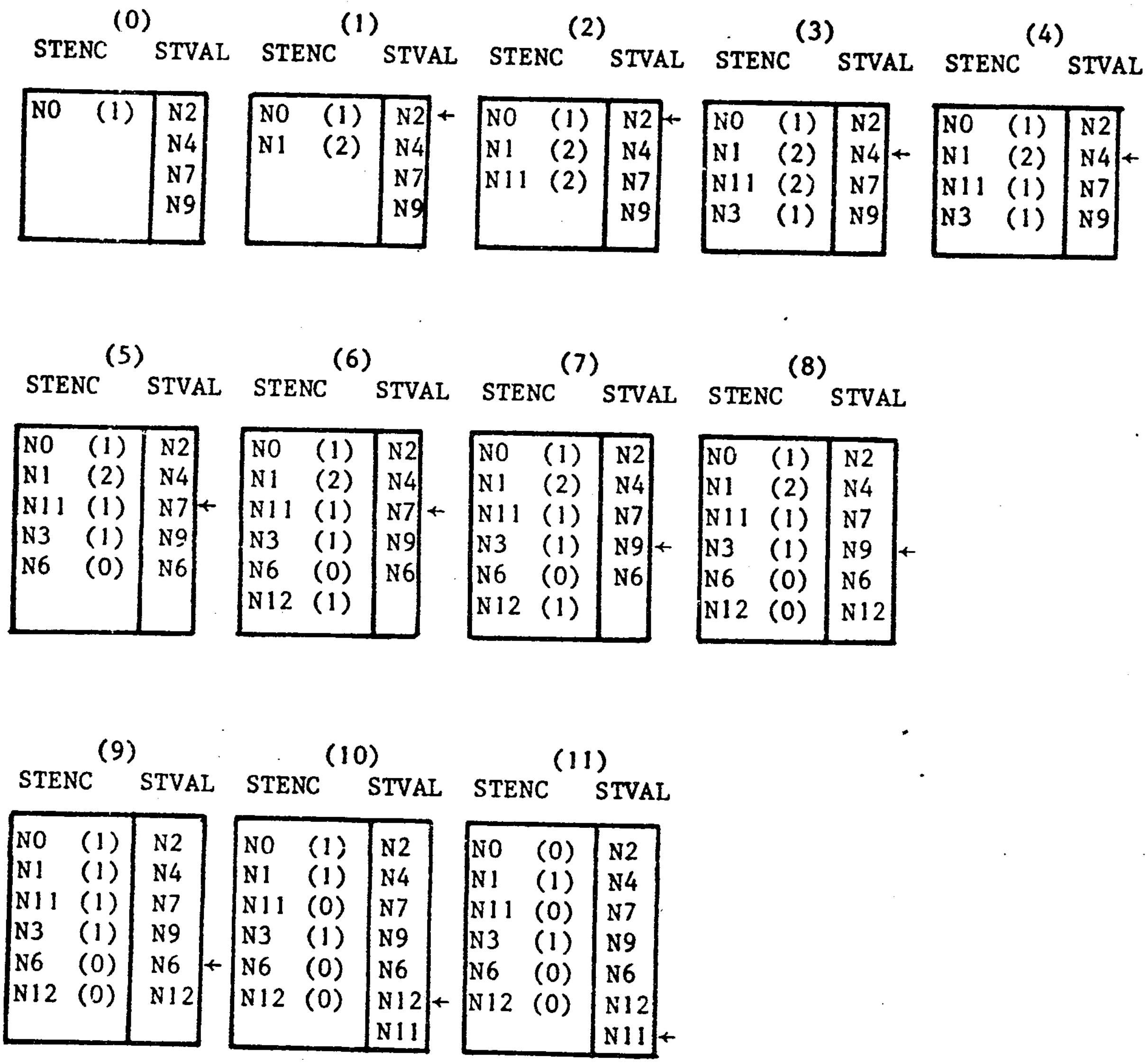


Figure 6.8. : Evolution des piles (cf. 6.9)

On donne simultanément les deux piles STENC et STVAL.

L'état de chaque noeud dans STENC est noté entre parenthèses.

La flèche à droite de STVAL désigne le noeud de rang STINDEX.

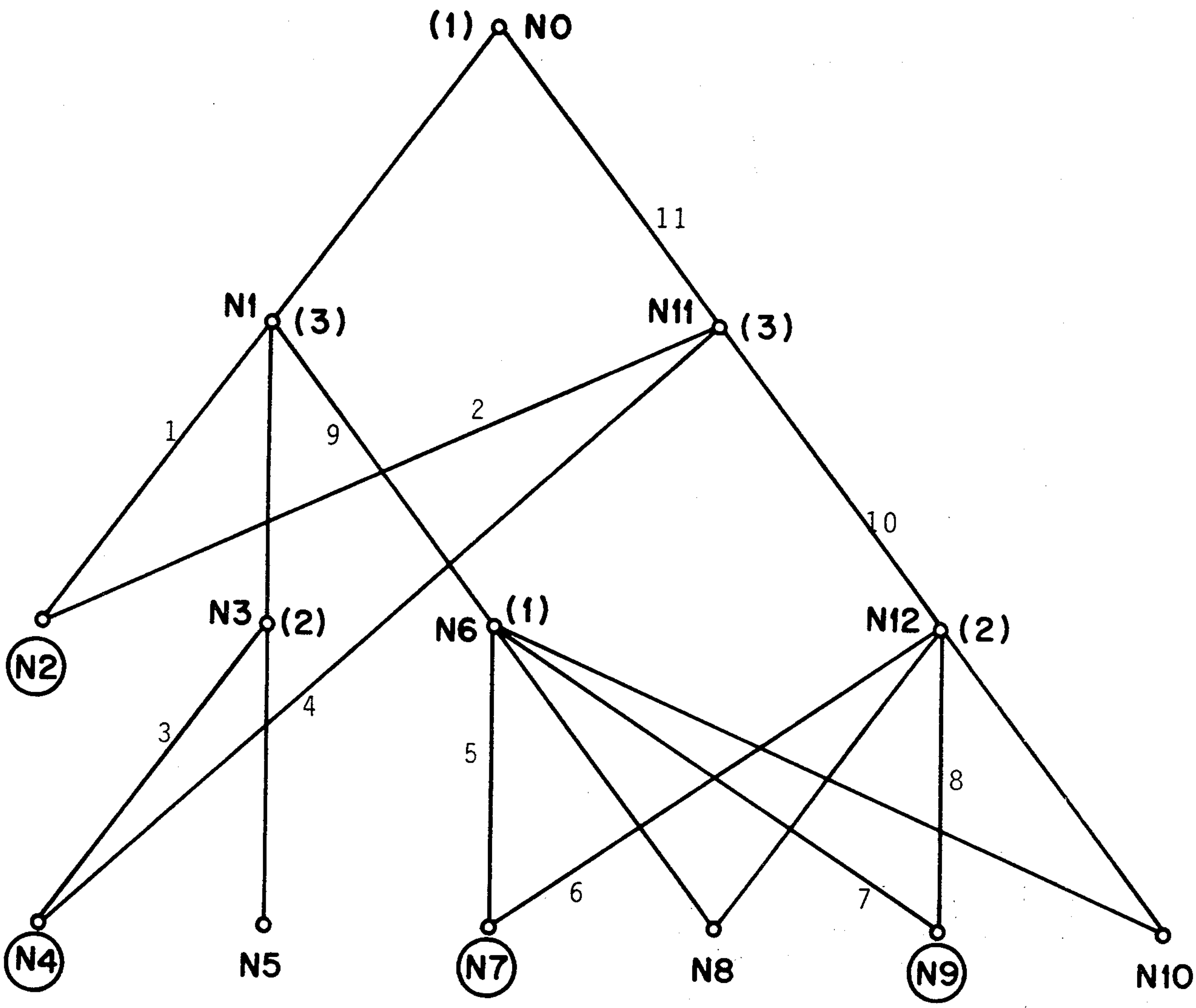


FIGURE 6.9. : Algorithme ascendant en largeur

Comme pour les figures précédentes, les noeuds valides donnés sont notés dans un cercle. Chaque phase de la progression dans le graphe est repérée par un numéro sur l'arête correspondante, le contenu de la pile à ce stade est donné page précédente, identifié par le même numéro.

7. OPTIMISATION ET CHOIX D'UN ALGORITHME

Le chapitre précédent nous a permis de décrire quatre types d'algorithmes de validation, bien évidemment la question qui se pose est celle du choix de l'un de ces quatre algorithmes.

7.1. Considérations générales sur les algorithmes de validation

Certaines caractéristiques peuvent déjà être dégagées :

- Les algorithmes qui travaillent en profondeur sont fondamentalement récursifs; on a noté que si l'on avait à effectuer des validations uniquement sur des arbres il serait possible d'écrire un algorithme descendant en profondeur qui n'utilise pas de pile mais qui rende à l'appelant l'état de noeud pour la validation duquel il a été appelé.

- Les algorithmes qui travaillent en largeur ne sont pas fondamentalement récursifs; l'algorithme ascendant en largeur (cf. 6.5) a été écrit sans récursivité aucune. Par contre ces algorithmes nécessitent l'utilisation d'une pile même si les validations à effectuer ne portent que sur des arbres.

- Le comportement des algorithmes ascendants est fonction de l'ordre dans lequel les noeuds terminaux valides sont donnés, donnons-en un exemple (cf. figure 7.1) à partir de N_2,N_4,N_7,N_9 (à gauche) et de N_7,N_4,N_9,N_2 (à droite);l'ordre dans lequel les arêtes sont parcourues est noté au dessus de chacune d'elles.

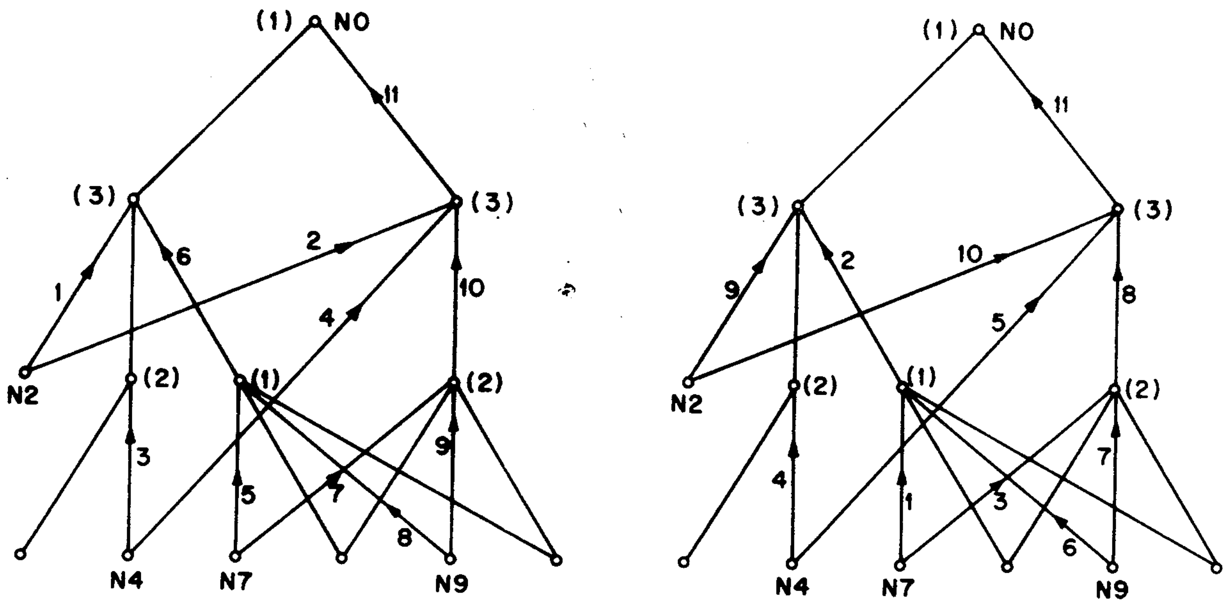


FIGURE 7.1. : Comportement de l'algorithme ascendant en profondeur dans la validation de N_0 à partir de N_2, N_4, N_7, N_9 (à gauche) et de N_7, N_4, N_9, N_2 (à droite), l'ordre dans lequel les arêtes sont parcourues est noté au dessus de chacune d'elle.

Or dans l'application qui nous intéresse un noeud valide donné correspond à un résultat positif à un examen et nous ne sommes pas maîtresde l'ordre dans lequel ces résultats nous sont donnés. Indirectement cela signifie que nous ne pouvons contrôler pleinement le déroulement des algorithmes ascendants.

N.B. : *Il est évident que le comportement des algorithmes descendantsest lui, indépendant de l'ordre dans lequel les noeuds terminaux valides sont donnés.*

- Des mesures de temps montrent que le comportement des différents algorithmes est très influencé par la forme générale du graphe (graphe très large, graphe très profond, ...) et la nature des données à valider.

Ceci nous amène à nous poser la question suivante :

En supposant que les données à valider évoluent peu dans le temps, ce qui revient à supposer que d'une année sur l'autre le comportement de l'ensemble des étudiants reste sensiblement le même (1), peut-on organiser le graphe de façon à optimiser le comportement d'un algorithme ?

Nous allons voir dans le paragraphe suivant quelle réponse apporter à cette question.

7.2. Possibilité d'optimisation

L'idée évoquée précédemment implique que l'on connaisse l'utilisation qui est faite du graphe au cours des validations.

On associe donc à chaque arête du graphe un compteur qui est utilisé de la façon suivante :

Lors de la validation avec succès du noeud N_0 à partir des noeuds $N_1, N_2, \ldots, N_j$, on incrémente d'une unité les compteurs associés aux arêtes telles que :

- l'origine et l'extrémité de ces arêtes sont validées par le processus de validation de N_0 à partir de $N_1 \ldots N_j$.

- si l'origine de cette arête n'était pas validée par le processus de validation, le noeud N_0 ne pourrait pas l'être.

Un tel ensemble d'arêtes est appelé "graphe solution".

---

(1) Cette hypothèse dite du "comportement constant de l'étudiant" est généralement admise par les experts en matière d'éducation (cf. [14]: Mesure de la réussite des étudiants : Une analyse statistique de données universitaires).

Exemple : figure : 7.2. Graphe de solution de la validation de e_6 à partir de x_6, x_8, x_{10}, x_{12} (en trait gras).

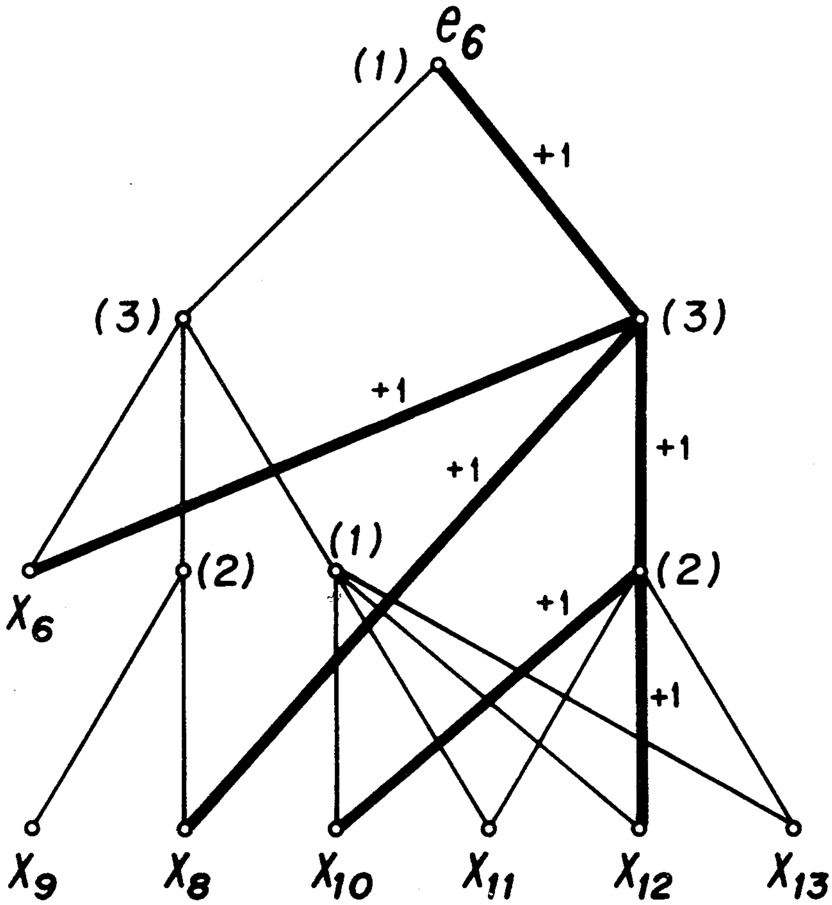


FIGURE 7.2.

Ces compteurs mesurent l'utilisation qui est faite du graphe, c'est-à-dire l'usage que font les étudiants des règles de progression dans le système d'étude.

Sous l'hypothèse que cet usage est stable dans le temps, on peut se demander, étant donné un type d'algorithme, dans quel ordre ranger les arêtes issues d'une même origine afin de minimiser le nombre total d'arêtes parcourues.

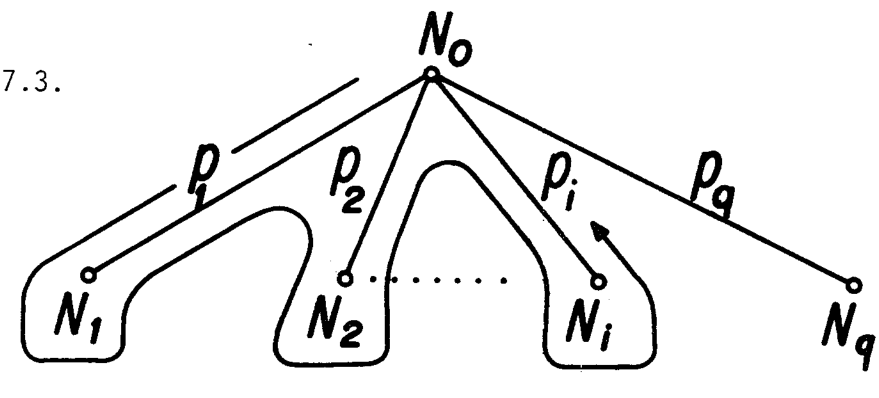
Nous ne savons pas résoudre ce problème de façon générale aussi allons nous faire l'hypothèse restrictive que les validations portent sur des arbres, et nous allons sous cette hypothèse résoudre le problème pour l'algorithme descendant en profondeur.

Raisonnons par récurrence sur la profondeur de l'arbre :

- considérons d'abord un arbre de profondeur 1. Soit un noeud N_0 et ses descendants $N_1, N_2, \dots, N_q$. On suppose que l'arête N_0N_i a été utilisée p_i fois avec succès (cf. figure 7.3).

La stratégie utilisée, validation en profondeur vers le bas, implique que les p_i fois où on est passé avec succès par l'arête N_0N_i on a en fait parcouru ip_i arêtes.

FIGURE 7.3.



Le nombre total d'arêtes parcourues est donc :

$$TOT = \Sigma ip_i$$

La permutation d'indice pour laquelle TOT est minimal est définie par :

$$\boxed{p_1 \geqslant p_2 \geqslant \dots p_i \geqslant p_{i+1} \dots \geqslant p_q} \qquad (I)$$

On remarquera que l'on ne tient pas compte des arêtes parcourues dans les validations de N_0 qui ont échoué parce qu'en cas d'échec il faut de toute façon parcourir toutes les arêtes; dans ces conditions il est normal de ne réorganiser l'arbre qu'en fonction des arêtes parcourues avec succès.

On sait donc minimiser le nombre d'arêtes parcourues dans un arbre de profondeur 1. Supposons qu'on sache le faire pour les arbres de profondeur 2,3,...,n.

Considérons un arbre de profondeur n+1. Soit N_0 sa racine, et $N_1, N_2, \ldots, N_q$ les descendants de N_0; comme précédemment p_i désigne le nombre de fois où l'arête $N_0\ N_i$ a été empruntée. Le noeud N_i est la racine d'un arbre de profondeur $\leq$ n dans lequel on parcourt R_i arêtes, R_i étant par hypothèse minimal. On désignera par a_i le nombre d'arêtes de l'arbre de racine N_i (cf. figure 7.4).

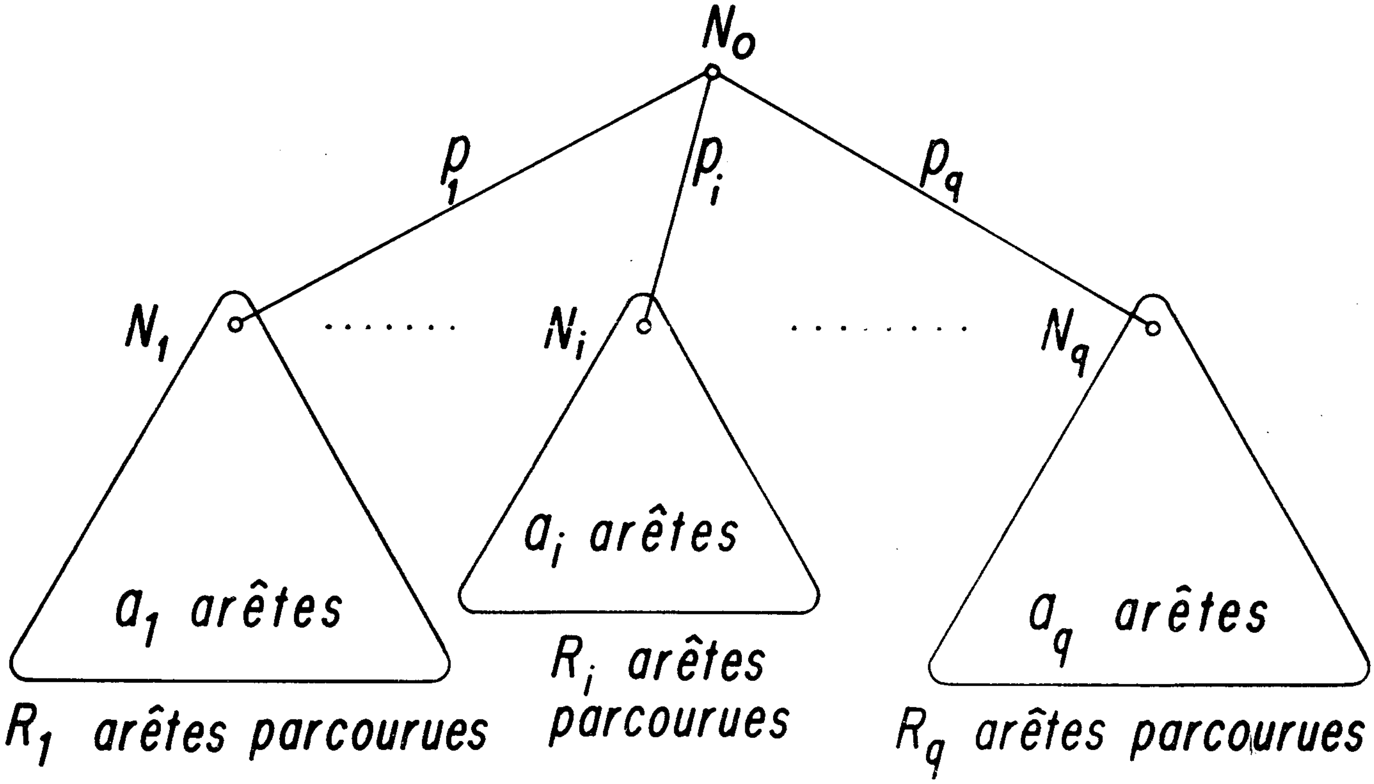


FIGURE 7.4.

Le nombre total d'arêtes parcourues est :

$$TOT = p_1 + R_1 + p_2((a_1+1) + 1) + R_2 + \dots + p_j \left(\sum_{k=1}^{j-1} (a_k+1)+1 \right) + R_j + \dots$$

$$TOT = p_1 + p_2 \, (a_1+1) + \dots + p_j \left(\sum_{k=1}^{j-1} (a_k+1)+1 \right) + \dots + p_q$$

$$\left(\sum_{k=1}^{q-1} (a_k+1)+1 \right) + \sum_{i=1}^{q} R_i$$

La quantité $\sum_{i=1}^{q} R_i$ étant par hypothèse minimale, il faut minimiser la quantité :

$$A = TOT - \sum_{i=1}^{q} R_i = p_1 + p_2(a_1+1) + \dots + p_q \left(\sum_{k=1}^{q-1} (a_k+1) +1 \right)$$

Ajoutons à A la quantité $B = \sum_{i=1}^{q} a_i \, p_i$

$$A + B = p_1(a_1+1) + p_2((a_1+1) + (a_2+1)) + \dots + p_q \sum_{k=1}^{q} (a_k+1)$$

Si une permutation de $\{1,\dots,q\}$ minimise la quantité A+B, elle minimise aussi la quantité A puisque la quantité B reste invariante.

Or on démontre aisément (1) que la quantité A+B est minimale pour la permutation d'indice définie par :

(1) On trouvera la démonstration dans D.E. KNUTH [10] p. 400 qui considère la quantité $\sum_{i=1}^{q} p_i \left(\sum_{j=1}^{i} L_j \right)$ comme le temps moyen de recherche séquentielle d'enregistrement de longueur variable, L_i désignant la longueur de l'enregistrement R_i et P_i la probabilité d'accès à R_i.

$$\boxed{\frac{P_1}{a_1+1} \geqslant \cdots \geqslant \frac{P_i}{a_i+1} \geqslant \frac{P_{i+1}}{a_{i+1}+1} \geqslant \cdots \geqslant \frac{P_q}{a_q+1}} \qquad \text{(II)}$$

La formule (I) apparaît alors naturellement comme un cas particulier de (II).

On a ainsi un critère qui permet de réorganiser les arêtes d'un arbre exploré par un algorithme en profondeur en tenant compte des explorations antérieures.

La figure 7.5 donne un exemple de réorganisation.

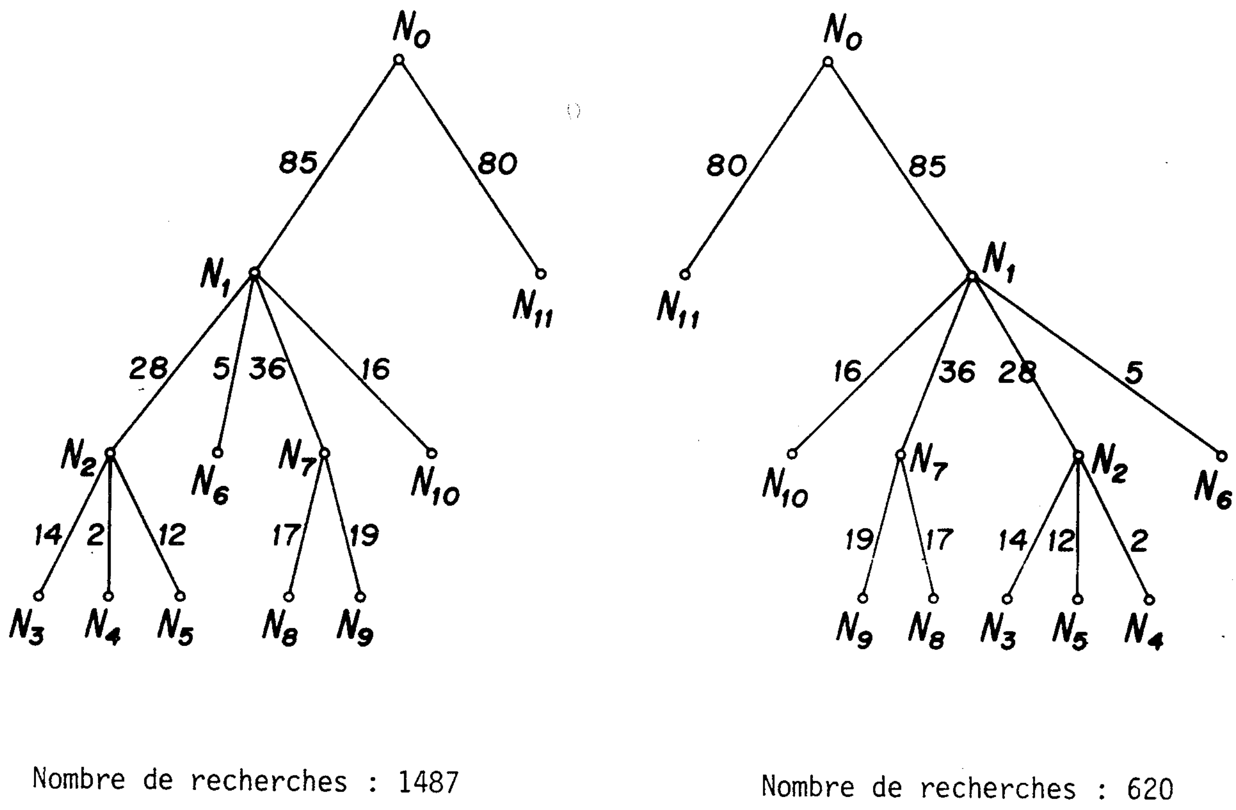


FIGURE 7.5. : <u>Réorganisation d'un arbre sur lequel opère un algorithme de validation en profondeur</u>

Remarques :

- Ce type de réorganisation ne fait intervenir que la structure logique de l'arbre, mais il serait intéressant de faire intervenir la position sur le support physique de chaque noeud ceci dans le but d'optimiser les accès disques (cf. [9]).

- On notera que l'on peut aussi optimiser les algorithmes descendants en abandonnant la validation d'un noeud N_i ayant n_i successeurs dès que l'on a trouvé un nombre de successeurs non valides supérieur à $(n_i - \lambda_i)$. Cette démarche est fructueuse quand on tente de valider un ET-noeud puisque dans ce cas $n_i = \lambda_i$.

7.3. Application

Dans la pratique les conditions d'inscription sont actuellement toujours représentées par des arbres, le résultat précédent leur est donc toujours applicable. Les conditions d'obtention d'une étape sont dans 60 % des cas représentées par un arbre qui peut donc être réorganisé; les 40 % des cas restants ne peuvent être traités de façon automatique, on peut cependant déterminer des portions de graphes réductibles à des arbres que l'on peut donc réorganiser.

A titre d'exemple, nous donnons page suivante les résultats obtenus en réorganisant le graphe inscription au vu des données résultant des inscriptions universitaires d'octobre 1973.

Le résultat [II] peut être appliqué à l'ensemble du graphe inscription qui comporte 2689 arêtes.

Sur ce graphe tel qu'il était organisé le nombre d'arêtes balayées est de 90433. Après réorganisation de l'ensemble du graphe et en supposant que les données à valider restent les mêmes ce nombre est de 50337.

En fait, compte-tenu du prix de la réorganisation d'un noeud, il n'est pas souhaitable de réorganiser tout le graphe mais seulement les noeuds les plus mal organisés.

On se donne donc un critère qui est le nombre de recherches d'arêtes que l'on peut gagner en réorganisant un noeud, et on ne réorganise que les noeuds qui permettent effectivement d'économiser un nombre de recherches supérieur à ce critère.

Ainsi, on peut estimer qu'il n'est pas rentable de réorganiser les noeuds pour lesquels le nombre de recherches inutiles est inférieur à 50 parce que le coût de la réorganisation est alors supérieur au coût des 50 recherches d'arêtes inutiles.

Le tableau 7.6 donne, en fonction de ce critère :

- le nombre de noeuds réorganisés,
- le nombre d'arêtes réorganisées, c'est-à-dire le nombre d'arêtes ayant pour origine les noeuds réorganisés (significatif du coût de la réorganisation),
- le nombre de recherches d'arêtes que l'on gagne (significatif du gain de performance des validations),
- le nombre de recherches gagnées par arête optimisée.

Critère	Nbre de noeuds réorganisés	Nbre d'arêtes réorganisées	Nbre de recherches gagnées	Recherches gagnées par arête réorganisée
1 000	3	51	23 850	467,6
500	9	104	28 233	271,4
300	15	181	30 492	168,5
200	26	223	33 152	148,6
100	48	329	36 196	110,2
80	56	372	36 918	99,2
50	76	482	38 217	79,8

TABLEAU 7.6.

La figure 7.7. donne le pourcentage de recherches d'arête gagnées en fonction du pourcentage d'arêtes réorganisées : il apparaît que réorganiser un peu plus de 12 % des arêtes permet de gagner 90 % des recherches inutiles.

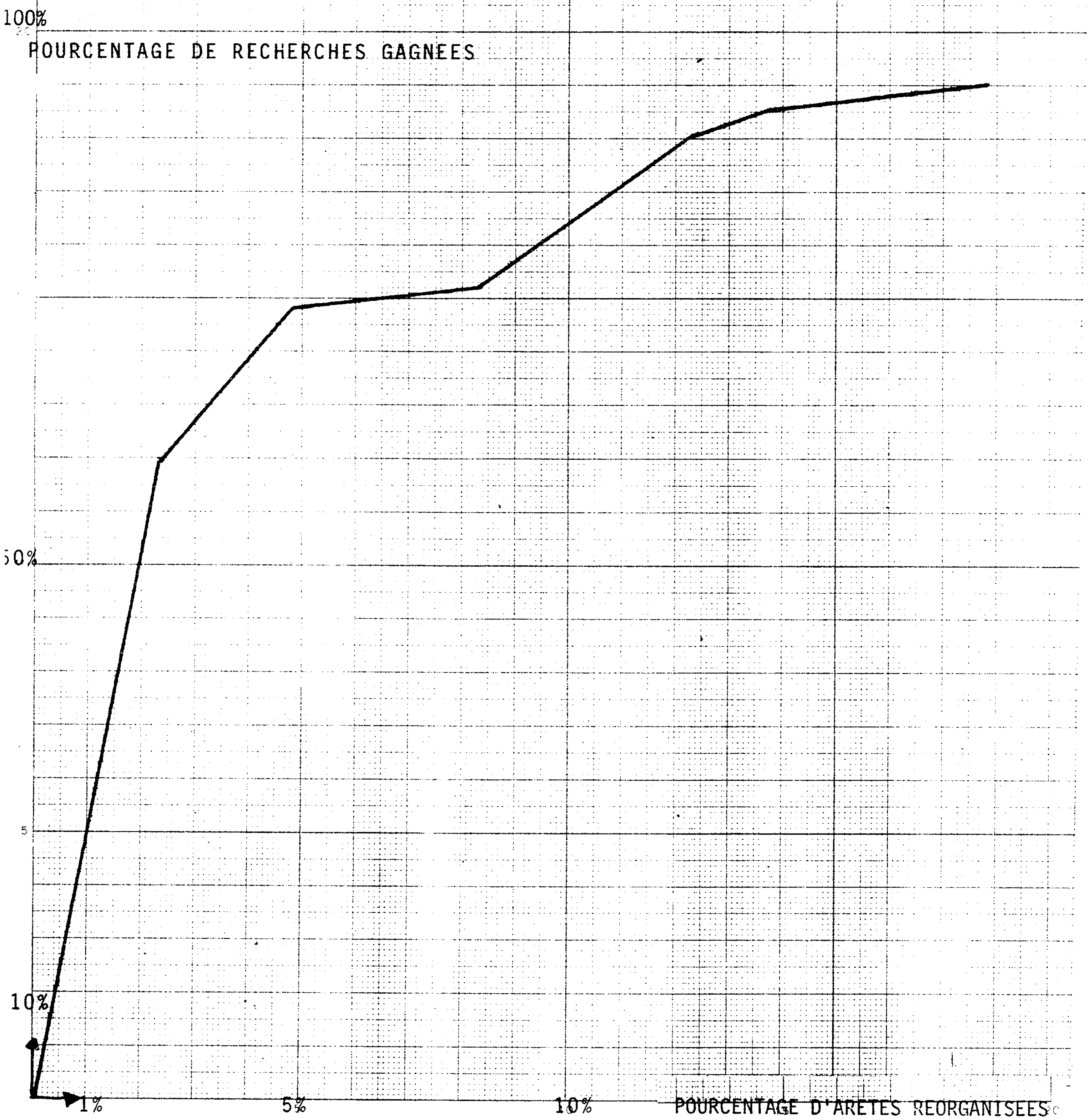


FIGURE 7.7.

7.4. Conclusion

Les résultats précédents qui découlent de la mesure des chemins effectivement empruntés dans l'exploitation des graphes inscription et obtention conduisent à choisir l'algorithme descendant en profondeur pour effectuer les validations.

Mais la mesure de ces chemins a un autre intérêt lié lui à la nature de l'application.

On a mentionné en effet que les compteurs associés à chaque arête mesurent en fait l'usage fait par les étudiants des règles de progression dans le système d'étude.

Ils constituent ainsi un précieux outil de mesure à la fois pour les organisateurs des différentes filières d'études et pour les utilisateurs de ces filières que sont les étudiants.

8. LA REALISATION

Le travail de conception d'un système de gestion des enseignements évoqué dans les chapitres précédents a naturellement conduit à la réalisation d'un produit opérationnel, qui a été menée dans le cadre des trois universités grenobloises :

- Université Scientifique et Médicale,
- Université des Sciences Sociales,
- Université de Langues et Lettres.

Dans ce chapitre, nous évoquerons tout d'abord les besoins des utilisateurs à satisfaire, puis nous nous proposerons de montrer de façon précise la constitution de la base de données des enseignements et son exploitation.

Le lecteur ne s'étonnera pas de voir que certains des aspects théoriques envisagés dans les chapitres précédents n'ont été que partiellement mis en oeuvre :

- d'une part, il est évident que les exigences d'un produit opérationnel (notamment dans le domaine des coûts, donc des performances) ne permettent pas de généraliser l'usage de solutions très élaborées dont l'efficacité n'est réelle que pour une minorité de situations très complexes ;
- d'autre part, en ce qui nous concerne, la réalisation matérielle et le travail de réflexion ont été en partie conduits simultanément et certains des aspects théoriques dégagés notamment en ce qui concerne l'analyse, ne l'ont pas été assez tôt pour être utilisés dans la réalisation du produit.

Ainsi, le programme qui permet d'évaluer et de comparer le coût de la représentation graphique d'une fonction booléenne et de sa duale (cf.4.5) n'a pas été intégré dans la réalisation, il constitue plus tôt un outil marginal dont on s'est servi dans quelques cas particulièrement complexes.

8.1. Les besoins des utilisateurs

Les besoins des administrations des universités grenobloises étaient clairement définis dans les deux domaines suivants :

- aide à la gestion courante des services centraux de scolarité.
- traitement du "fichier étudiant" en tenant compte de l'organisation des études et de l'évolution de cette organisation.

Si l'outil réalisé devait satisfaire en priorité ces deux besoins, il était souhaitable d'envisager plus globalement les services que pourraient rendre la base de données des enseignements au corps enseignant d'une part et aux étudiants d'autre part, respectivement responsables et utilisateurs des études organisées par les universités.

8.1.1. Aide à la gestion courante des services de scolarité

La base de données des enseignements doit constituer un catalogue de l'ensemble des enseignements dispensés par les universités et de leur organisation.

Des éditions différentes de ce catalogue vont former le support matériel nécessaire aux tâches quotidiennes du service.

On distingue deux catégories d'éditions suivant que la structure des études est prise en compte ou non.

- Editions ne tenant pas compte de la structure des études

  A titre d'exemple citons :
  - liste des diplômes délivrés par une université,
  - liste des enseignements organisés par une U.E.R.,
  - liste des enseignements organisés par une université et donnant lieu à l'édition de procès-verbal,
  - etc...

- Editions tenant compte de la structure des études

On retiendra les deux types suivants :

1) Pour un diplôme donné : - liste des titres qui permettent de s'y inscrire,
   - liste des enseignements qui interviennent dans son obtention.

2) Pour un diplôme donné : - graphes représentatifs des conditions d'inscription et des conditions d'obtention.

8.1.2. Traitements du fichier étudiant

Ils doivent permettre de confronter les renseignements sur les acquis d'un étudiant contenus dans le fichier étudiant et la structure des études telle qu'elle est décrite dans la base de données des enseignements, dans le but :

- de vérifier que l'étudiant s'inscrit à juste titre,
- de déterminer, au vu des résultats obtenus aux enseignements élémentaires, si l'étudiant a acquis, acquis partiellement ou n'a pas acquis un diplôme.

Il est à noter que l'absence de moyens automatisés adéquats impliquait que ces tâches soient réalisées manuellement, ce qui, vu la charge des services, ne pouvait être fait qu'à la demande des intéressés : ainsi pour les années 1970-1971-1972-1973, il n'a pas été possible de déterminer le nombre d'étudiants ayant réussi à une maîtrise scientifique.

8.1.3. Services offerts au corps enseignant et aux étudiants

Dans ce domaine, les besoins n'ont pas été clairement formulés : il s'est agit donc de déterminer quelle pouvait être l'utilité de la base de données des enseignements pour les enseignants et les étudiants. Dans cette optique nous avons retenu les deux points suivants :

- fournir des informations sur l'utilisation faite par les étudiants des règles de progression dans le système d'étude. On a vu (cf. 7.2) que cette information pouvait être obtenue en associant aux arêtes du graphe un compteur incrémenté par l'algorithme de validation chaque fois que l'arête est utilisée. Il a semblé intéressant de conserver cette information pour chaque année universitaire afin d'établir des tableaux comparatifs, année par année, d'utilisation du système d'étude.
- Informer les étudiants précisément des possibilités qui leur sont offertes en fonction de leur acquis.

  Une difficulté réelle se pose ici du fait que la base de données des enseignements ne rend compte que des règles de progression normale : or de nombreuses dérogations étant toujours possibles, on ne pourra pas indiquer à un étudiant les possibilités offertes grâce aux éventuelles dérogations auxquelles il peut prétendre.

8.2. Choix du système de gestion de données - Modalités d'exploitation

On a choisi d'utiliser le système de gestion de données SOCRATE pour constituer la base de données des enseignements.

La version prototype de ce système, réalisée à l'Institut de Mathématiques Appliquées de Grenoble, sous la direction de J.R. ABRIAL (cf. [11]), a permis de faire les premiers essais et de mener à bien la réalisation de la base qui a pu être utilisée dès la rentrée universitaire 73/74 (octobre 73).

Le choix de SOCRATE a été déterminé en grande partie par le fait que des versions industrielles de ce logiciel ont été développées sur des matériels variés (IRIS 45 et 50, IRIS 80 et 10070, IBM 360, ...), ce qui ouvre à l'application présentée ici des perspectives d'utilisations dans d'autres universités que celles de Grenoble.

A Grenoble même l'utilisation de la version prototype de SOCRATE est une solution provisoire adoptée tant qu'une version industrielle satisfaisante n'est pas exploitable.

Techniquement, il était nécessaire de décrire facilement des graphes, ce que le système permet par l'utilisation des références. Il était aussi souhaitable de pouvoir écrire des programmes récursifs afin de mettre en oeuvre les algorithmes de validation, ce qui est également possible sous SOCRATE.

L'exploitation du prototype est à l'heure actuelle réalisée sous le système générateur de machines virtuelles CP/67. La machine virtuelle utilisée est une machine virtuelle standard (cf. [12]) de 36 cyclindres et 512 K-octets de mémoire centrale.

8.3. Structure du fichier des enseignements

Le choix de la structure du fichier des enseignements a été guidé d'une part par les besoins des utilisateurs, d'autre part par la nécessité de prendre en compte trois graphes ET/OU, l'un représentatif des conditions d'inscription, l'autre représentatif des conditions d'obtention, le troisième représentatif des conditions d'obtention partielle.

Compte-tenu du fait que les conditions d'obtention partielle d'une étape ne sont pas toujours définies, et que, quand elles le sont on a la relation : obtention $\Rightarrow$ obtention partielle, on a représenté par un seul graphe ET/OU les conditions d'obtention partielle et d'obtention. Dans ce graphe, que nous désignerons par graphe obtention, on associe à chaque noeud non plus un mais deuxentiers qui permettent de définir, conformément aux règles vues en 1.2.2., l'un les conditons d'obtention partielle, l'autre les conditions d'obtention.

8.3.1. Description des graphes

Les graphes sont constitués de noeuds doublement chaînés entre eux par des références. A chaque noeud sont associées un certain nombre de caractéristiques qui les décrivent.

Ces caractéristiques sont les suivantes :

- CODE valeur numérique discriminante qui identifie le noeud pour l'utilisateur et constitue un nom historique (cf. [1]). En ce sens un code est attribué une fois pour toutes à un noeud et n'est jamais réutilisé même si ce noeud n'a plus de signification dans l'organisation des études.

Une tranche de code est réservée aux noeuds non significatifs (CODE > 10000). Pour les noeuds significatifs associés à des enseignements élémentaires, étapes, dérogations, c'est ce code qui est enregistré sur le fichier étudiant pour spécifier les renseignements correspondants.

- TITRE libellé de 60 caractères qui permet à l'utilisateur de qualifier clairement un noeud.

L'exemple 8.1 donne pour chaque noeud utilisé pour la représentation de la maîtrise d'enseignement de mathématiques le CODE et le TITRE correspondant (cf. 2.3 et 3.3).

```
00418   SC MAITRISE ENSEIGNEMENT MATHEMATIQUES
10115   SC PARTIE * CURSUS MAIT ENS MATH AVEC A' COMPLET
10116   SC PARTIE * CURSUS MAIT ENS MATH AVEC A' INCOMPLET
00305   SC UNITE B' MAITRISE ENSEIGNEMENT MATHS
00326   SC UNITE A' MAITRISE ENSEIGNEMENT MATHEMATIQUES
00327   SC UNITE C' A 1 UNITE MAITRISE ENSEIGNEMENT MATHEMATIQUES
10117
00303   SC UNITE A'1 MAITRISE ENSEIGNEMENT MATHS
00304   SC UNITE A'2 MAITRISE ENSEIGNEMENT MATHS
00306   SC UNITE C'A MAITRISE ENSEIGNEMENT MATHEMATIQUES
00307   SC UNITE C'B MAITRISE ENSEIGNEMENT MATHEMATIQUES
00308   SC UNITE C'C MAITRISE ENSEIGNEMENT MATHEMATIQUES
02094   SC UNITE C'D MAITRISE ENSEIGNEMENT MATHEMATIQUES
```

- FONCTION Spécifie la nature du noeud (étape, dérogation, enseignement élémentaire, noeud non significatif). On a vu en 5.3 que de façon rigoureuse cette caractéristique est redondante puisque la nature d'un noeud résulte entièrement de sa situation dans les graphes, il était cependant difficile de s'en passer dans la réalisation.

L'exemple 8.2 donne pour chaque noeud utilisé pour la représentation de la maîtrise d'enseignement de mathématiques la fonction correspondante.

```
00418   E (E: ETAPE)
10115   Z (Z: NOEUD NON SIGNIFICATIF)
10116   Z (Z: NOEUD NON SIGNIFICATIF)
00305   F (F: ENSEIGNEMENT ELEMENTAIRE)
00326   M (M: ENSEIGNEMENT NON ELEMENTAIRE )
00327   M (M: ENSEIGNEMENT NON ELEMENTAIRE )
10117   Z (Z: NOEUD NON SIGNIFICATIF)
00303   F (F: ENSEIGNEMENT ELEMENTAIRE)
00304   F (F: ENSEIGNEMENT ELEMENTAIRE)
00306   F (F: ENSEIGNEMENT ELEMENTAIRE)
00307   F (F: ENSEIGNEMENT ELEMENTAIRE)
00308   F (F: ENSEIGNEMENT ELEMENTAIRE)
02094   F (F: ENSEIGNEMENT ELEMENTAIRE)
```

- DATE de CREATION   Indique à quelle date le noeud intervient effectivement dans l'organisation des études.
  Exemple : Les noeuds utilisés pour la représentation des premières années de D.E.U.G. (Diplôme d'Etudes Universitaires Générales) sont intervenus dans l'organisation des études à dater d'octobre 1973; pour ces noeuds DATE de CREATION égale 73/10.

- DATE de SUPPRESSION   Indique à partir de quelle date le noeud n'intervient plus dans l'organisation des études. Le noeud est alors "archivé" et il n'est plus possible à un étudiant de l'obtenir; mais un étudiant qui l'avait obtenu antérieurement peut toujours en faire état.

- CYCLE   Spécifie dans quel cycle d'étude intervient un noeud (uniquement pour les noeuds significatifs).
  Exemple : Les noeuds utilisés pour la représentation de la maîtrise d'enseignement de mathématiques interviennent dans le 2ème cycle d'études supérieures.

- SORTIE   Spécifie à quels types d'édition peut donner lieu un noeud; on distingue l'absence d'édition, l'édition de procès-verbal, l'édition d'attestation ainsi que l'édition de diplôme.

- UNIVERSITE   Désigne l'université organisatrice du noeud.

- UER : Désigne l'U.E.R. organisatrice du noeud.

- [UNIVERSITE, UER] Information répétitive.
  Chaque occurence indique une université et une U.E.R. qui utilisent le noeud sans en être les organisatrices.

- CODE-MINISTERE   Indique s'il y a lieu le code attribué par le ministère au diplôme national représenté par le noeud.

- PI (Poids Inscription) Entier qui spécifie les conditions d'inscription du noeud (nombre de successeurs dans le graphe inscription qui doivent être valides pour que le noeud le soit).

  N.B. *Si PI n'est pas défini c'est que les conditions d'inscription à ce noeud ne le sont pas, soit qu'il n'appartienne pas au graphe inscription, soit que ce soit un noeud terminal de ce graphe.*

- [I-FILS] Information répétitive ; chaque occurrence est associée à un successeur du noeud dans le graphe et contient les informations suivantes :
  - VAL pointeur sur le successeur
  - [FLUX] information répétitive associée à une arête; chaque occurrence est associée à une année d'exploitation et contient :
    - VF nombre de fois dans l'année où l'arête a été utilisée pour valider une inscription.

- [I-PERE] Information répétitive; chaque occurrence est associée à un prédécesseur du noeud dans le graphe inscription et contient l'information suivante :
  - VAL pointeur sur le prédécesseur.

- PO Entier qui spécifie les conditions d'obtention du noeud (nombre de successeurs dans le graphe obtention qui doivent être valides pour que l'obtention du noeud soit acquise).

- PM Entier qui spécifie les conditions d'obtention partielle du noeud (nombre de successeurs dans le graphe obtention qui doivent être valides pour que l'obtention partielle du noeud soit acquise).

  N.B. *Si PO et PM ne sont pas définis c'est que les conditions d'obtention et d'obtention partielle du noeud ne sont pas définies, soit que le noeud n'appartienne pas au graphe obtention, soit qu'il soit un noeud terminal dans ce graphe.*

Enfin si les conditions d'obtention partielle seules ne sont pas définies on a PO = PM.

- [O-FILS] Information répétitive; chaque occurrence est associée à un successeur du noeud dans le graphe obtention et contient les informations suivantes :
  - VAL Pointeur sur le successeur
  - [FLUX] Information répétitive associée à une arête du graphe obtention; chaque occurrence est associée à une arête d'exploitation et contient :
    - VF nombre de fois dans l'années où l'arête a été utilisée pour valider une obtention.

- [O-PERE] Information répétitive; chaque occurrence est associée à un prédécesseur du noeud dans le graphe obtention et contient l'information suivante :
  - VAL Pointeur sur le prédécesseur.

Nous donnons en annexe (cf. B 1) la description en langage SOCRATE de la structure de données définie ci-dessus, ainsi que (cf. B 2) l'édition des caractéristiques associées aux noeuds 418 et 305 utilisés pour la représentation de la maîtrise d'enseignement de mathématiques.

La figure 8.3 donne une représentation synthétique de la structure d'un noeud.

8.3.2. Description des fichiers inverses

On rappelle (cf. [11]) qu'un fichier inverse constitue une redondance d'information qui permet d'accéder aux éléments d'un ensemble qui vérifient une propriété P (appelée critère d'inversion) de façon plus rapide qu'avec la méthode d'accès courante.

Dans notre cas nous avons retenu les critères d'inversion suivants :

- le noeud X est utilisé par l'université U
- le noeud X a été utilisé par l'université U et est archivé
- le noeud X est utilisé par l'unité d'enseignement et de recherche V
- la fonction du noeud X est F.

Les fichiers inverses ainsi constitués sont organisés en chaîne de bits, c'est-à-dire qu'à une propriété P des noeuds est associée une chaîne de bits comportant autant de bits qu'il y a de noeuds. Dans cette chaîne le i-ème bit est à 1 si la i-ème réalisation de l'entité noeud vérifie la propriété P.

L'intérêt de ce type d'organisation réside dans les deux caractéristiques suivantes :

- encombrement minime du fichier inverse,
- possibilité de réaliser sur les chaînes de bits des opérations booléennes qui permettent de répondre aux questions du type :

  Ensemble de noeuds ayant la propriété P_1 et la propriété P_2 ?

  Ensemble de noeuds ayant la propriété P_1 ou la propriété P_2 où P_1 et P_2 désignent deux critères d'inversion.

Le schéma 8.3 (page suivante) résume ce qui vient d'être dit sur la structure des noeuds et sur les fichiers inverses.

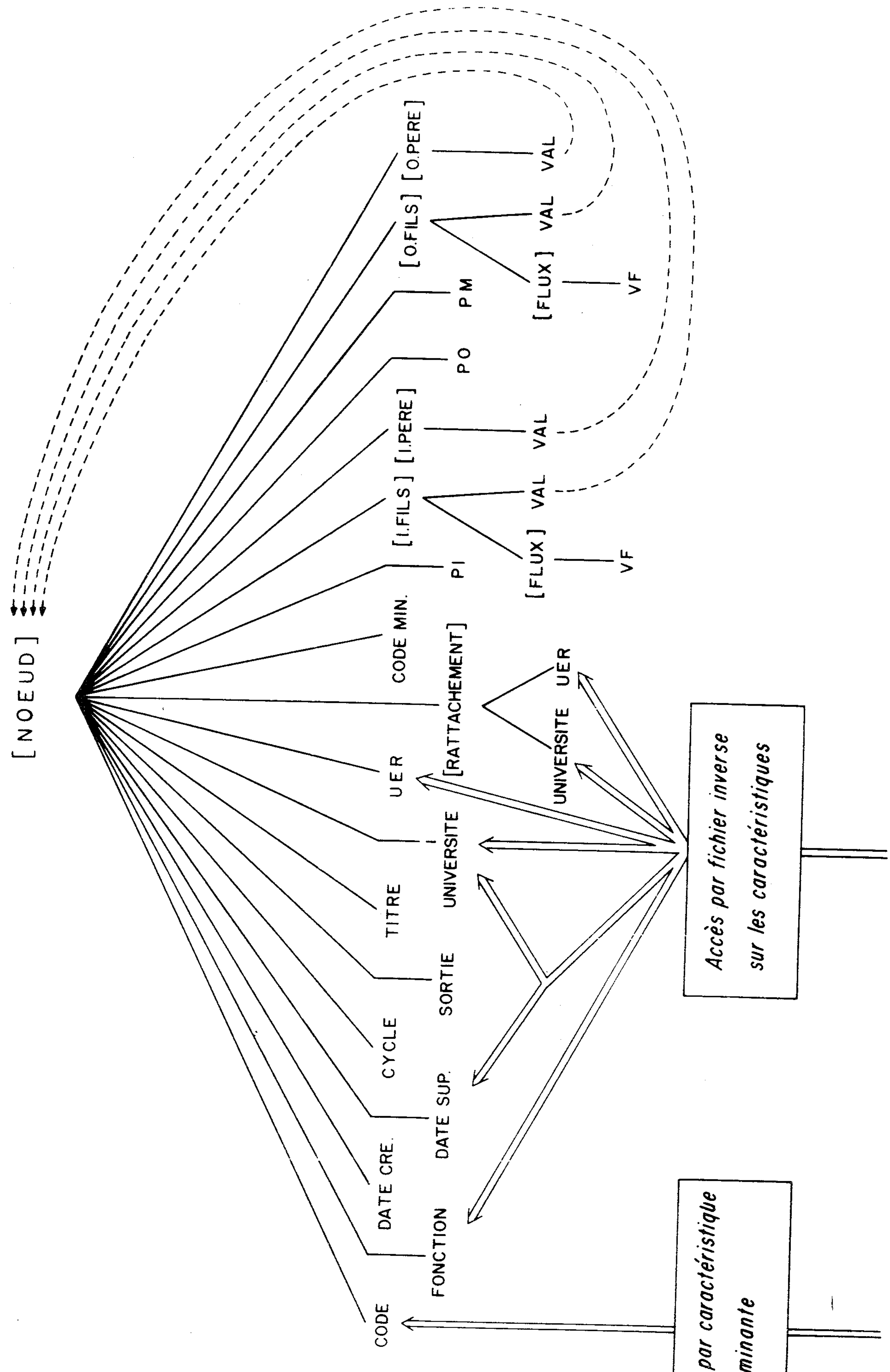


FIGURE 8.3. : Schéma général de la structure des données et des méthodes d'accès privilégiées.

8.4. Création du fichier des enseignements

8.4.1. Analyse des structures d'étude

L'analyse des enseignements pour les trois universités de Grenoble a été réalisée par une seule personne Monsieur M. WOLFF et a nécessité 5 mois de travail. Les aspects théoriques dégagés au chapitre 2 n'avaient pas encore été pressentis et n'ont donc pas pu être utilisés.

Pour effectuer son travail l'analyste s'est appuyé sur les règlements tels qu'ils sont définis par les universités et par le ministère de l'éduction nationale et généralement regroupés dans un document appelé "livret de l'étudiant". Le résultat de l'analyse est reporté sur des bordereaux de saisie[(cf. figure 8.4] qui permettent de définir :

- les caractéristiques élémentaires associées aux enseignements, titres et étapes.
- les liens entre étapes et titres (liaisons inscription) et les liens entre étapes et enseignements (liaisons obtention).

Ces liens sont décrits par des expressions du type : (fils1, fils2,...,filsn) (nombre de fils nécessaires pour que le père soit valide).

Ainsi l'analyste décrit-il entièrement les graphes. Mais on a vu dans le chapitre 4 qu'une telle méthode n'est pas souhaitable car elle ne permet de contrôler ni la cohérence ni l'irredondance des relations ainsi définies. Dans notre cas ces deux écueils ont été en grande partie évités par le fait que l'analyste était une personne unique : le problème aurait sans doute été plus grave si l'analyse avait été réalisée au niveau de chaque service de scolarité.

La principale difficulté rencontrée a porté sur les enseignements utilisés simultanément dans des universités différentes : très souvent ils portaient un nom différent dans chaque université et il était difficile de savoir par exemple, s'il s'agissait d'un même enseignement appelé ici X et là Y, ou bien de deux enseignements distincts X et Y.

Quantitativement l'analyse a porté sur 1482 enseignements élémentaires et sur 675 étapes organisés par les trois universités de Grenoble.

USMG - SYGESCO - Bordereau de perforation n° 1 OBLIGATOIRE

Code CARTE	1 | 1 |
Code ENSEIGNEMENT	2 | 0 | 0 | 4 | 1 | 8 |
CYCLE	7 | 2 |
UNIVERSITE	8 | 1 |
SORTIE — O code 0; P code 1; A code 2; D code 3; P,A code 4; P,A,D code 5	9 | 3 |
DATE DE CREATION AN - MOIS	10 | | | | |
DATE SUPPRESSION AN - MOIS	14 | | | | |
FONCTION E, X, P, M, F, (ALPH.)	18 | E |

TITRE (ALPH.NUM.)

19 |S|C| |M|A|I|T|R|I|S|E| |E|N|S|

34 |E|I|G|N|E|M|E|N|T| |M|A|T|H|E|

49 |M|A|T|I|Q|U|E|S| | | | | | | |

64 | | | | | | | | | | | | | | | |

CONTROLE NOMBRE DE CARTES |3|
79

Figure 8.4. : Bordereau de perforation utilisé pour coder le noeud 418

USMG - SYGESCO - Bordereau de perforation n° 6 FACULTATIF

CODE CARTE 1 | 6 |

CODE ENSEIGNEMENT REPETE 2 | 0 | 0 | 4 | 1 | 8 |

LIAISON AMONT
INSCRIPTION
(Alpha numér.)

7 | (| 2 | 4 | 9 | , | 2 | 7 | 8 | , | 3 | 0 | 0 |) | (| 1 |

22 |) |

37

52

67

USMG - SYGESCO - Bordereau de perforation n° 7 FACULTATIF

CODE CARTE 1 | 7 |

CODE ENSEIGNEMENT REPETE 2 | 0 | 0 | 4 | 1 | 8 |

LIAISON AMONT
ORIENTION
(Alpha numér.)

7 | (| 1 | 0 | 1 | 1 | 5 | , | 1 | 0 | 1 | 1 | 6 |) | (| 1 |

22 |) |

37

52

67

Figure 8.4. (suite)

8.4.2. Création du fichier

Les données saisies sur bordereau ont donné lieu à perforation et les cartes obtenues ont été traitées par un programme PL/1 qui a constitué une bande magnétique à partir de laquelle la création effective de la base a pu être faite.

La base des enseignements occupe 2 millions d'octets ; elle est à l'heure actuelle pleine à 65 %, ce qui, compte-tenu des ajouts qui interviennent chaque année, devrait lui permettre d'être opérationnelle jusqu'en 1977. Il devrait être possible avant cette date d'utiliser une version industrielle du produit SOCRATE.

Depuis sa création, des changement sont déjà intervenus (dus à l'application des DEUG* en particulier). A l'heure actuelle, les graphes contenus dans la base comptent 3251 noeuds et comportent 2689 arêtes pour le graphe inscription et 3884 arêtes pour le graphe obtention.

Le tableau suivant (figure 8.5) donne pour chaque université le nombre de noeuds significatifs ayant une fonction donnée (étape, enseignement élémentaire, dérogation ou équivalence).

---

* Diplômes d'Etudes Universitaires Générales : ils ont remplacé les Diplômes Universitaires d'Etudes Scientifiques, Littéraires, ... à la rentrée universitaire 73/74.

E t a b l i s s e m e n t s	Etapes	Enseignements élémentaires	Dérogations ou équivalences
Université Scientifique et Médicale	301	526	74
Université de Langues et Lettres	171	521	100
Université de Sciences Sociales	225	628	166
Ensemble du fichier (1)	697	1572	283

Figure 8.5. : Dénombrement des étapes, enseignements élémentaires, dérogations, organisés ou utilisés par chaque université.

(1) N.B. : *Comme certaines fonctions sont utilisées simultanément par plusieurs universités, les chiffres donnés pour l'ensemble du fichier ne correspondent donc pas toujours à la somme de ceux donnés pour chaque université.*

8.5. Exploitation du fichier des enseignements

Dans ce paragraphe nous détaillerons les modalités d'exploitation de la base de données. On distinguera d'une part l'utilisation du fichier des enseignements effectuée par les services de scolarité et qui n'implique pas d'accès au fichier étudiant, d'autre part, les traitements réalisés à partir du fichier des enseignements sur le fichier étudiant.

8.5.1. Utilisation par les services de scolarité

Un certain nombre de fonctions sont mises à la disposition des services de scolarité. Elles permettent de modifier la base de données et de l'interroger, elles sont accessibles à partir d'un terminal et opèrent de façon conversationnelle.

Dans chaque service de scolarité on distingue deux types d'utilisateurs :

- les opérateurs privilégiés qui ont le droit de modifier le contenu de la base dans la mesure où ces modifications n'affectent que des enseignements organisés par l'université qu'ils représentent ; cette responsabilité relève du chef de service par exemple.
- les opérateurs non privilégiés qui ont le droit d'interroger la base de données uniquement.

Nous donnons ci-dessous une liste des différentes fonctions mises à la disposition des utilisateurs :

- AJOUTER (opération privilégiée)

  Permet de créer un nouveau noeud dans la base et de définir toutes ses caractéristiques autres que celles qui définissent les graphes inscription et obtention.

- LIER (opération privilégiée)

  Permet de relier entre eux des noeuds déjà définis dans la base ainsi que les poids (PI, PO, PM) qui fixent les conditions d'inscription ou d'obtention.

- MODIFIER (opération privilégiée)

  Permet de modifier les caractéristiques suivantes d'un ou de plusieurs noeuds : FONCTION, TITRE, SORTIE, UNIVERSITE, UER, RATTACHEMENT.

- ARCHIVER (opération privilégiée)

  Permet d'archiver un noeud dans la base ; cette opération consiste en :
  - mise à jour de la date de suppression de ce noeud : à partir de cette date ce noeud n'apparaîtra plus sur les différentes listes ; si c'est une étape un étudiant ne pourra plus s'y inscrire, si c'est un enseignement un étudiant ne pourra plus l'obtenir.
  - suppression des arêtes ayant ce noeud pour origine (à noter qu'on ne supprime pas les arêtes ayant ce noeud pour extrémité, car on continue à reconnaître les droits de ceux qui ont acquis ce noeud avant son archivage).
  - édition d'un état qui constitue une trace matérielle de l'opération réalisée.

- DELIER (opération privilégiée)

  Permet de supprimer un lien. Cette opération n'est permise qu'au responsable de l'université organisatrice du noeud origine du lien.

- AFFICHER (opération non privilégiée)

  Permet d'obtenir au terminal les caractéristiques les plus importantes d'un noeud (FONCTION, TITRE, UNIVERSITE, UER, PI, PO, PM, DATESUP éventuellement), ainsi que les codes et titres de ses prédécesseurs et de ses successeurs dans les graphes inscription et obtention.

- EDITER (opération non privilégiée)

Permet d'obtenir différents types de listes (éditées sur l'imprimante); ces listes sont faites à la demande soit par UER, soit par Université. Les différents types prévus sont :

- liste générale,
- liste des étapes,
- liste des enseignements,
- liste pour codage (contient les informations nécessaires pour réaliser le codage des informations saisies en début d'année pour la constitution du fichier étudiant),
- liste des fractions de cursus (description pour chaque étape des titres d'accès et des enseignements qui interviennent dans son obtention).

- DESSINER (opération non privilégiée)

Permet d'obtenir les données nécessaires pour tracer une partie du graphe obtention ou du graphe inscription. Ces données sont ensuite exploitées par un programme PL/1 opérant sous OS et le dessin est réalisé par un traceur BENSON.

A titre d'exemple nous donnons page suivante (cf. fig. 8.6) la suite d'opérations qui permet de créer un nouveau noeud (procédure AJOUTER), de la rattacher à un noeud déjà existant (procédure LIER), d'éditer le noeud créé (procédure AFFICHER) et enfin de créer les données nécessaires au tracé d'un graphe sur BENSON (procédure DESSINER).

On trouvera d'autre part en annexe (Annexe C) différents types de listes (Annexe C1 - liste de fraction de cursus; annexe C2 - liste pour codage; annexe C3 - liste générale) ainsi que des exemples de graphe obtention (annexes C4, C5, C6).

```
sygesco
VERSION 4.D __SEPT. 74                ← accès au fichier des enseignements
NOM DE L'APPLICATION ?
_*PROFILE:
FONCTION(B,SORTIE)
B
ajouter                               ← appel de fonction
CODE
4507
CYCLE
2
UNIVERSITE
1
UER
608
ENTREE
u
OUT
1
FONCTION
f
DATECRE
7410
TITRE
sc unite a'3 maitrise enseignement de mathematiques
CREATION DE RATTACHEMENT
REPONSE
non
CODE
u
AJOUTER TERMINE
B
lier                                  ← appel de fonction
MODE:
fp                                    ← spécifie qu'on va lier un noeud à
CIRCUIT                                 différents prédécesseurs (fp=fils-père)
obtention
FILS:
CODE
4507
PERES:
CODE
526                                   ← modification des poids des prédécesseurs
POIDS:
oui
PO,PM     03   03
PO
4
PM
4
CODE
0
FILS:
CODE
0
CIRCUIT
u
LIER TERMINE
B
```

```
afficher                                              ← appel de fonction

CODE
4507

04507 (F)  SC UNITE A'3 MAITRISE ENSEIGNEMENT DE MATHEMATIQUES
UNIVERSITE: 1     UFR: 08      PI,PO,PM: 00 00 00

OBTENTION PERE
00326        SC UNITE A' MAITRISE ENSEIGNEMENT MATHEMATIQUES

CODE
326

00326 (M)  SC UNITE A' MAITRISE ENSEIGNEMENT MATHEMATIQUES
UNIVERSITE: 1     UER: 08      PI,PO,PM: 00 04 04

OBTENTION FILS
00303 0070 SC UNITE A'1 MAITRISE ENSEIGNEMENT MATHS
00304 0078 SC UNITE A'2 MAITRISE ENSEIGNEMENT MATHS
04507 0000 SC UNITE A'3 MAITRISE ENSEIGNEMENT DE MATHEMATIQUES

OBTENTION PERE
00366        SC PARTIE * CURSUS MAIT ENS MATH AVEC A' COMPLET

CODE
0
AFFICHER TERMINE
B
dessiner                                              ← appel de fonction
FDC DESSINEE ?
CODE
418
CIRCUIT
obtention
LONGUEUR
380
NB-NIVEAU
5
GRAPHE NO 080
FDC DESSINEE ?
CODE
0
DESSINER TERMINE
B
non#sortie                                            ← sortie du système
FUNCTION( ,SORTIE)
---FICHIER CATALOGUE---
R; T=4.94/8.51 17:52:24
```

8.5.2. Traitements du fichier étudiant

Le fichier étudiant est à l'heure actuelle sur bande magnétique ; il est exploité par des programmes COBOL. Sur ce fichier nous avons vu (cf. 1.2.1) que l'on trouve deux types d'informations :

- Les informations qui identifient un étudiant et le décrivent en tant que personne physique (numéro d'INSEE, état-civil, situation militaire, dernier établissement fréquenté, ...)
- Les informations qui décrivent les études qu'il poursuit.

Seules ces dernières informations nous intéressent; elles comportent pour chaque année universitaire et pour chaque filière suivie :

* le millésime de l'année
* l'étape postulée et le résultat afférant
* le ou les titres qui permettent de s'y inscrire (à l'heure actuelle, dans la grande majorité des cas, ce titre est unique ; une exception apparaît cependant pour les maîtrises de sciences et techniques où, pour s'inscrire, il faut posséder un D.E.U.G. et un certificat préparatoire).
* le ou les enseignements suivis pour obtenir l'étape postulée ainsi que les résultats acquis à ces enseignements.

(cf. ANNEXE D - Dessin de bande du fichier étudiant).

1ère fraction de cursus	Année		73/74
	Etape postulée	3030 :	DEUG Lettres et Arts section A 1ère année (acquis)
	Titre	0003 :	Bac série A2
	Enseignement 1	0736 :	Langue et Littérature latines niveau 1 (acquis)
	Enseignement 2	0750 :	Langue française (acquis)
	Enseignement 3	0741 :	Initiation aux lettres (acquis)
	Enseignement 4	3008 :	Littérature générale et comparée (acquis)
	Enseignement 5	0795 :	Histoire générale de la musique (acquis)

2ème fraction de cursus	Année		74/75
	Etape postulée	3031 :	DEUG Lettres et Arts section A (acquis)
	Titre	3030 :	DEUG Lettres et Arts section A 1ère année
	Enseignement 1	0815 :	Littérature française (acquis)
	Enseignement 2	0760 :	Langue et Littérature allemandes (acquis)
	Enseignement 3	0798 :	Langue et Littérature latines niveau 2 (acquis)
	Enseignement 4	3020 :	Linguistique générale et française en 1er cycle (acquis)
	Enseignement 5	0771 :	Merveilleux et Fantastique (acquis)

3ème fraction de cursus	Année		75/76
	Etape postulée	1097 :	Licence de lettres modernes (acquis partiellement)
	Titre	3031 :	DEUG Lettres et Arts section A
	Enseignement 1	1007 :	Certificat L Lettres modernes (acquis)
	Enseignement 2	1016 :	Etudes latines (échec)

4ème fraction de cursus	Année		76/77 (en cours)
	Etape postulée	1097 :	Licence de lettres modernes
	Titre	3031 :	DEUG Lettres et Arts section A
	Enseignement 1	1016 :	Etudes latines

5ème fraction de cursus	Année		76/77 (en cours)
	Etape postulée	1225 :	Maîtrise de lettres modernes
	Titre	1097 :	Licence de lettres modernes (titre conditionnel)
	Enseignement 1	1164 :	TER lettres modernes
	Enseignement 2	1157 :	Statistique linguistique

N.B. : *Remarquer les articulations Etape obtenue ≡ Titre permettant de s'inscrire au niveau supérieur.*

FIGURE 8.7. : Description des études poursuivies par un étudiant de lettres modernes.

Le fichier étudiant s'accroît chaque année des informations relevées au cours des inscriptions universitaires. Une fois ces informations saisies et le fichier constitué, on effectue un premier traitement qui permet de vérifier la validité des inscriptions prises.

Ensuite, au terme de chaque session d'examen (juin et septembre), on enregistre sur le fichier étudiant les résultats acquis aux enseignements et on effectue un traitement qui permet de calculer le résultat acquis à l'étape en fonction des résultats acquis aux enseignements.

Décrivons en détail ces deux types de traitements :

VALIDATION DES CONDITIONS D'INSCRIPTION

Pour chaque fraction de cursus auquel s'inscrit l'étudiant on vérifie que le ou les titres avancés ont bien été acquis dans le passé. Si le titre d'accès est une dérogation cette vérification n'a pas lieu mais l'étudiant doit avoir fourni l'attestation correspondante au guichet où il s'est inscrit.

On procède alors à la validation des conditions d'inscription en appliquant l'algorithme descendant en profondeur décrit en 6.2 pour valider dans le graphe inscription le noeud correspondant à l'étape postulée sachant que le ou les noeuds correspondant aux titres d'accès sont des noeuds valides.

Enfin, on vérifie que les enseignements auxquels s'inscrit l'étudiant sont bien définis dans le fichier des enseignements et qu'en outre ils interviennent bien dans les conditions d'obtention de l'étape postulée; autrement dit il s'agit de s'assurer qu'il existe bien un chemin entre le noeud associé à l'enseignement et le noeud associé à l'étape postulée dans le graphe obtention.

On trouvera ci-dessous une description précise du traitement effectué; chaque fois qu'une erreur est détectée on appelle une procédure de traitement d'erreur notée ERREUR, (on a noté entre parenthèses une brève description de la cause de l'erreur rencontrée).

P1. Si l'étape existe dans le fichier des enseignements aller à P3.

P2. ERREUR (étape inexistante) aller à P9.

P3. Si le titre d'accès est une dérogation ou équivalence aller à P6.

P4. Si le ou les titres d'accès ont été acquis dans le passé aller à P6.

P5. ERREUR (l'un au moins des titres avancés n'a pas été acquis dans le passé) aller à P9.

P6. Procéder à la validation des conditions d'inscription à l'étape postulée à partir du ou des titres donnés.
Si le résultat est positif aller à P8.

P7. ERREUR (le ou les titres donnés ne permettent pas de s'inscrire à l'étape postulée) aller à P9.

P8. Pour tout enseignement auquel s'est inscrit l'étudiant:

- P8.1. L'enseignement existe dans le fichier des enseignements aller à P8.3.
- P8.2. ERREUR (enseignement inexistant) aller à P8.5.
- P8.3. L'enseignement intervient effectivement dans les conditions d'obtention de l'étape postulée aller à P8.5.
- P8.4. ERREUR (enseignement hors filière)
- P8.5. Ø

P9. Ø

<u>N.B.</u> : *Le test fait en P4. revient à s'assurer que le titre présenté est une étape acquise au moins de façon partielle dans le passé.*

VALIDATION DES CONDITIONS D'OBTENTION

Les résultats acquis par un étudiant aux enseignements élémentaires sont enregistrés sur le fichier étudiant au terme de chaque session d'examen. Une fois ces résultats recueillis on peut effectuer dans le graphe obtention la validation des conditions d'obtention de chaque étape postulée par l'étudiant.

Pour valider l'obtention d'une étape, il n'est pas toujours suffisant de prendre en compte les résultats acquis aux enseignements élémentaires de la fraction de cursus correspondant au cours de l'année considérée. En effet des résultats acquis les années antérieures dans cette même fraction de cursus peuvent exister et ont alors toute chance d'intervenir.

En toute rigueur il peut même arriver que des enseignements élémentaires acquis dans d'autres fractions de cursus interviennent dans le mécanisme d'obtention d'une étape. Ces cas sont rares et les traiter automatiquement est utopique car cela reviendrait à faire intervenir dans la validation d'une étape l'ensemble des enseignements élémentaires acquis par un étudiant (ou du moins avec l'organisation actuelle des cycles d'études, l'ensemble des enseignements acquis par un étudiant dans le cycle auquel appartient l'étape considérée).

Nous adopterons donc la position suivante :
la validation de l'obtention d'une étape portera uniquement sur l'ensemble des enseignements élémentaires acquis par l'étudiant dans la fraction de cursus correspondante, éventuellement au cours de plusieurs années.

Les résultats de cette validation peuvent être les suivants :

R1 - L'étape est acquise
R2 - L'étape est acquise partiellement, l'étudiant peut donc s'inscrire au niveau supérieur
R3 - L'étape est acquise mais le titre d'inscription est un titre conditionnel : l'acquisition effective est donc conditionnée par l'obtention du niveau inférieur
R4 - L'étape est acquise partiellement mais le titre d'inscription est conditionnel ; l'étudiant peut cependant s'inscrire au niveau supérieur bien qu'en attente de validation des deux niveaux inférieurs (ce cas est rare).
R5 - L'étape n'est même pas acquise partiellement, il y a échec.

On notera qu'il faut répercuter les résultats de l'obtention définitive d'une étape sur les fractions de cursus où cette même étape, acquise antérieurement de façon partielle, constituait un titre d'accès conditionnel.

Pour toutes ces fractions de cursus le résultat à l'étape est transformé comme suit :

R3 devient R1
R4 devient R2.

Le schéma suivant décrit les traitements effectués :

P1. Si des résultats positifs aux enseignements élémentaires ont été enregistrés aller à P3.
P2. Aller à P13. /aucun acquis : validation inutile/
P3. Procéder à la validation des conditions d'obtention de l'étape postulée à partir des enseignements acquis dans la fraction de cursus correspondante, au cours éventuellement de plusieurs années.
P4. Si le résultat n'est pas positif aller à P9
P5. Si le titre d'inscription est conditionnel alors résultat (étape) ← R3 /succès sous réserve de validation du niveau inférieur/ aller à P13.

P6. Résultat (étape) ← R1 /succès définitif/

P7. Pour toute fraction de cursus où l'étape sert de titre conditionnel

P7.1. Si résultat (étape) = R3 alors résultat (étape) ← R1

P7.2. Si résultat (étape) = R4 alors résultat (étape) ← R2

P8. Aller à P13

P9. Procéder à la validation des conditions d'obtention partielle de l'étape à partir du même ensemble d'enseignements qu'en P3.

P10. Si le résultat n'est pas positif alors résultat (étape) ← R5 aller à P13. /échec/

P11. Si le titre d'inscription est conditionnel alors résultat (étape) ← R4 /succès partiel sous réserve de validation du niveau inférieur/ aller à P13.

P12. Résultat(étape) ← R3 /succès partiel/

P13. Ø

A titre d'exemple nous donnons ci-après les résultats des mesures faites lors des validations d'obtention de la session de juin 1974.

Pour chaque université nous donnons le nombre de fractions de cursus traitées, le temps moyen de validation par fraction de cursus, le coût correspondant, le pourcentage de réussite (R1 et R3), de réussite partielle (R2 et R4), d'échec (R5), de cas où il manque un résultat à un enseignement (procès-verbal non parvenu, erreur de saisie, ...).

N.B. : *Le coût est établi sur la base de 1800 frs l'heure unité centrale, il tient compte de la validation proprement dite et des opérations annexes nécessaires.*

Universités	Nombre de f.d.c. traitées (1)	Temps en millisecondes	Coût en centimes	Réussite	Réussite partielle	Echec	Absence de résultats
Université Scientifique et Médicale	2908 (2)	146,3	8,53	35,4 %	7,8 %	55,6 %	1,2 %
Université de Sciences Sociales	5929	162,0	9,29	45,8 %	2,4 %	34,4 %	17,4 %
Université de Langues et Lettres	2868	398,4	21,42	20,0 %	11,8 %	66,7 %	1,5 %

(1) On ne traite que les étudiants qui ont obtenu effectivement un résultat (négatif ou positif) à un examen de la session de juin, soit environ la moitié des étudiants inscrits en début d'année.

(2) Les étudiants de médecine ne sont pas compris dans ces chiffres.

9. CONCLUSION

9.1. Les moyens d'évaluation de la progression des étudiants dans l'enseignement supérieur

On peut distinguer essentiellement trois systèmes de mesure destinés à évaluer la progression des étudiants dans l'enseignement supérieur :

- Dans le premier, les résultats obtenus par un individu dépendent des résultats obtenus par les autres, on parle alors de concours ou de "numerus clausus". Ce système est utilisé pour régler l'accès aux grandes écoles, l'accès en deuxième année des études médicales, ainsi que l'accès aux CAPES et aux agrégations.

Dans ce système il importe peu d'avoir de bons résultats, ce qui compte c'est d'avoir des résultats meilleurs que ceux des autres candidats.

- Dans le deuxième système, les résultats obtenus par un individu sont indépendants des résultats obtenus par les autres, mais les différents résultats obtenus par un même individu dépendent les uns des autres. Dans ce système on attribue à chaque individu un ensemble de notes, des formules arithmétiques déterminent les conditions que doivent satisfaire ces notes.

- Dans le troisième et dernier système, les résultats obtenus par un individu sont indépendants des résultats obtenus par les autres., et les différents résultats obtenus par un même individu sont indépendants les uns des autres. C'est ce système qui a fait l'objet de notre étude, on a vu que les résultats élémentaires pouvaient alors être représentés par des variables booléennes, une expression booléenne détermine quels ensembles de résultats élémentaires doivent être acquis.

Bien évidemment ces trois systèmes de mesure peuvent être utilisés conjointement :

- Dans les concours, on utilise le système de notation arithmétique qui a l'avantage de fournir des résultats comparables, c'est ce qui permet de classer les candidats.

- Le système de notation booléen (appelons ainsi le dernier système mentionné, même si le terme n'est pas très heureux) est pratiquement toujours associé à un système de notation arithmétique. Ainsi l'obtention d'un certificat ou d'une unité de valeur est le résultat d'un calcul de note portant sur les différentes épreuves organisées dans le cadre de ce certificat ou de cette unité de valeur. Cette obtention intervient alors en tant que variable booléenne dans les règles d'obtention d'une licence ou d'une maîtrise par exemple.

On remarque le parallélisme qui existe entre le système de notation arithmétique et le système de notation booléen. Dans le premier cas interviennent des fonctions arithmétiques croissantes, dans le deuxième cas on a vu que les fonctions utilisées sont des fonction booléennes croissantes. La différence essentielle entre ces deux systèmes est que le premier fournit des mesures comparables entre elles, ce qui n'est pas le cas du second (c'est la raison pour laquelle les diplômes nationaux DEUG, Licences, Maîtrises, ... qui relèvent de ce système sont délivrés sans mentions).

La Loi d'Orientation de l'enseignement supérieur élaborée en 1968 a provoqué une évolution dans les système de mesures utilisés. D'un système à tendance plutôt arithmétique et peu booléen qui caractérisait l'ancien régime d'études, on est passé de par l'introduction des unités de valeurs à un système un peu moins arithmétique et davantage booléen.

Curieusement, parallèlement à cette tendance apparaît un phénomène de nature différente qui tend à faire passer d'un système de mesure arithmétique au système utilisant le "numerus clausus". Ce phénomène apparaît dans les études médicales et para-médicales (pharmacie).

La figure 9.1 schématise ces deux tendances, qui ne sont pas propres à la France mais existent aussi dans d'autres pays industrialisés tels que l'Allemagne, l'Autriche, la Suisse, ... (cf. [14]).

Cette évocation rapide des systèmes de mesure destinés à évaluer la progression des étudiants dans l'enseignement supérieur nous a semblé utile car elle permet de dégager la raison profonde pour laquelle le système de gestion des étudiants mis en place à Grenoble en 1961 s'est avéré inadapté après la réforme de 1968.

En effet cette réforme en introduisant de façon implicite un glissement vers le système de mesures booléen a rendu inefficaces les traitements antérieurs qui ne tenaient pas compte de ce système.

On peut alors situer <u>notre travail</u> en disant <u>qu'il a introduit les outils nécessaires pour intégrer le système de mesure booléen dans les traitements utilisés pour la gestion des scolarités de l'enseignement supérieur</u>.

Système de mesures	"numerus clausus"	arithmétique	booléen
Exemples d'utilisation	Accès aux grandes écoles premier cycles des études médicales	I.U.T. (1) licences et maîtrises M.I.A.G. (2) organisées par Droit (Grenoble II) certificats	licences et maîtrises organisées par unités de valeur
Tendances			

Figure 9.1. : Les différents systèmes de mesures destinés à évaluer la progression des étudiants dans l'enseignement supérieur.

(1) - Institut Universitaire de Technologie

(2) - Maîtrise d'Informatique Appliquée à la Gestion

9.2. Evaluation critique des solutions retenues

Le paragraphe précédent a permis de situer dans un cadre très général notre travail, nous aimerions dans ce dernier paragraphe revenir sur les solutions techniques retenues afin d'en résumer les qualités et aussi les défauts.

Le modèle booléen utilisé (cf. chapitre 2) s'accorde bien avec la réalité puiqu'il a pu s'appliquer sans difficulté aux différentes filières d'études organisées par les trois universités de Grenoble. Sa généralité ne fait pas de doutes puisqu'il repose sur deux hypothèses toujours vérifiées :

H1 - **L'acquisition d'un enseignement élémentaire ou d'un niveau d'étude par un étudiant constitue un résultat définitif qui ne peut être remis en cause par d'autres échecs ou réussites.**

H2 - **"Qui peut le plus peut le moins" autrement dit les fonctions de progression utilisées sont toujours croissantes.**

La dissociation des mécanismes d'acquisition et des mécanismes de progression peut sembler artificielle, on rappelle que ce sont des raisons pragmatiques (cf. 2.4.) qui ont conduit à effectuer cette dissociation.

La représentation des fonctions booléennes croissantes par des graphes ET/OU (cf. chapitre 3) est une solution originale à notre connaissance. Elle a l'avantage d'être visualisable, les graphes obtenus étant "parlants" pour l'utilisateur. Elle permet en outre facilement de répondre aux questions "dans quelles fonctions intervient telle variable ?" et "quelles variables utilisent telles fonctions", question dont la valeur sémantique est importante pour notre application.

Cette représentation n'est cependant pas forcément la plus économique en place (on aurait pu envisager par exemple d'associer à chaque monôme des fonctions booléennes utilisées des chaînes de bits représentatives des variables présentes dans le monôme).

Enfin et c'est sans doute là son plus grand défaut, il existe plusieurs graphes ET/OU représentatifs d'une même fonction booléenne croissante, il n'y a donc pas unicité de la représentation le problème du choix demeurant difficile et les solutions proposées dans ce domaine restent de portée limitée (cf. 3.4.2 et 4.5).

Les relations de dépendances utilisées comme outils d'analyse (cf. chapitre 4) se sont avérées être commodes pour l'utilisateur. Les résultats théoriques dégagés à savoir possibilité de contrôler la cohérence et d'assurer l'unicité d'une solution minimale n'ont d'intérêt que dans les cas rares où l'organisation d'études analysée est extrêmement complexe (moins de 3 % des cas étudiés ont nécessité d'appliquer ces résultats). Ces cas bien que rares n'en sont pas pour autant négligeables, en effet il importe de pouvoir garantir une très grande fiabilité du fichier des enseignements puisque lui-même sert à valider des informations contenues dans le fichier étudiant. En ce sens les résultats dégagés sont précieux puisqu'ils permettent d'éviter des erreurs dans les situations où l'analyse est difficile.

La mise en oeuvre des algorithmes de validation (cf. chapitre 6) était indispensable pour l'application. L'effort fait pour déterminer l'algorithme le mieux adapté et les possibilités de l'optimiser (cf. chapitre 7) a permis d'obtenir des temps de traitement donc des coûts raisonnables. Les résultats obtenus dans le domaine de l'optimisation montrent combien il est important de se donner les moyens d'évaluer le comportement d'un algorithme quand ce comportement est fonction de données dont on n'est pas maître. Il nous a semblé significatif que ces moyens, en l'occurrence les compteurs d'utilisation des arêtes des graphes, aient une valeur pertinente pour l'utilisateur, en l'occurrence mesure de l'utilisation par les étudiants de l'organisation des études.

Au niveau de la réalisation (cf. chapitres 5 et 8), la dissociation des deux fichiers enseignements et étudiants nous semble méthodologiquement importante et c'est sur ce point que nous conclurons.

Le fichier des enseignements (2 millions de caractères) se comporte comme un descripteur de la logique du fichier des étudiants (40 millions de caractères). Nous avons ainsi opéré une hiérarchisation des données qui permet de distinguer les trois types d'opérations suivantes :

1) Opérations faisant intervenir uniquement le fichier des enseignements (cf. 8.5.1)
2) Opérations faisant intervenir uniquement le fichier des étudiants.
3) Opérations faisant intervenir simultanément les deux fichiers (cf. 8.5.2).

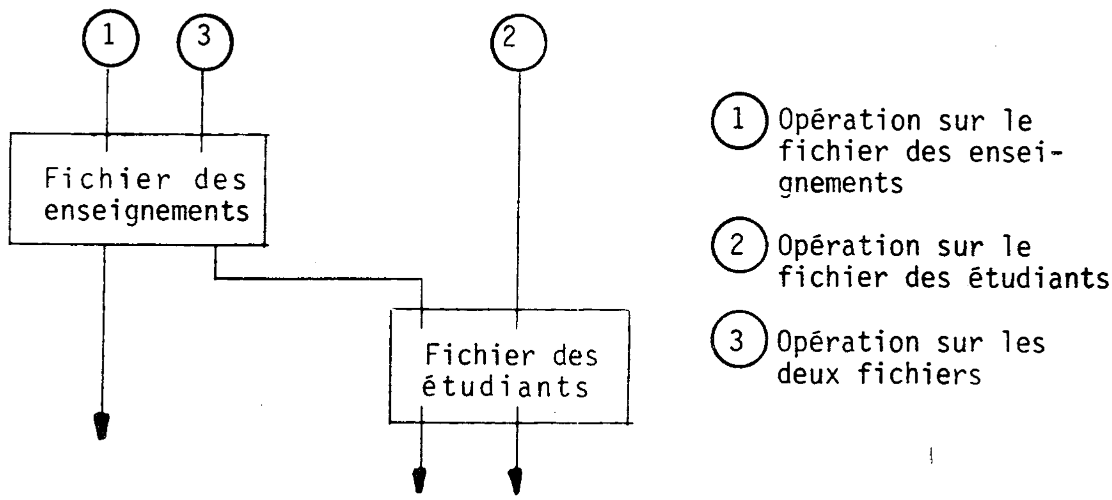


Figure 9.2. : Opérations sur les fichiers des étudiants et des enseignements

Le fichier des enseignements constituant un descripteur de la logique du fichier des étudiants les opérations de type 3 supposent que les informations de ce fichier sont faibles et permettent de détecter des anomalies sur le fichier des étudiants, c'est ce qui nous permet de parler de hiérarchisation des données.

L'intérêt de cette approche est d'augmenter automatiquement la fiabilité des informations sur un fichier de taille importante au prix d'un effort portant sur la fiabilité d'un fichier de taille beaucoup plus petite.

ANNEXE A : Calcul de la fermeture d'un ensemble de relations de dépendance

Enumération des combinaisons d'enseignements élémentaires qui donnent un diplôme - Choix de la représentation graphique (A 4)

RELATIONS DE DEPART

```
***1  00366  --> 00418
***2  00370  --> 00418
***3  00305 00326 00327  --> C0366
***4  00303 00304  --> 00326
***5  00306  --> 00327
***6  00307  --> 00327
***7  00308  --> 00327
***8  02094  --> 00327
***9  00305 00303 10117  --> C0370
**10  00306 00307  --> 10117
**11  00306 00308  --> 10117
**12  00307 00308  --> 10117
**13  00305 02094  --> 10117
**14  00307 02094  --> 10117
**15  00308 02094  --> 10117
```

FERMETURE:

```
***1  00366  --> 00418
***2  00370  --> 00418
***3  00305 00326 00327  --> C0366
***4  00303 00304  --> C0326
***5  00306  --> 00327
***6  00307  --> 00327
***7  00308  --> 00327
***8  02094  --> 00327
***9  00305 00303 10117  --> CC370
**10  00306 00307  --> 10117
**11  00306 00308  --> 10117
**12  00307 00308  --> 10117
**13  00306 02094  --> 10117
```

**14 00307 02094 --> 10117

**15 00308 02094 --> 10117

**16 00305 00326 00327 --> 00418

**17 00305 00327 00303 00304 --> 00366

**18 00305 00327 00303 00304 --> 00418

**19 00305 00326 00306 --> 00366

**20 00305 00326 00306 --> 00418

**21 00305 00303 00304 00306 --> 00366

**22 00305 00303 00304 00306 --> 00418

**23 00305 00326 00307 --> 00366

**24 00305 00326 00307 --> 00418

**25 00305 00303 00304 00307 --> 00366

**26 00305 00303 00304 00307 --> 00418

**27 00305 00326 00308 --> 00366

**28 00305 00326 00308 --> 00418

**29 00305 00303 00304 00308 --> 00366

**30 00305 00303 00304 00308 --> 00418

**31 00305 00326 02094 --> 00366

**32 00305 00326 02094 --> 00418

**33 00305 00303 00304 02094 --> 00366

**34 00305 00303 00304 02094 --> 00418

**35 00305 00303 10117 --> 00418

**36 00305 00303 00306 00307 --> 00370

**37 00305 00303 00306 00307 --> 00418

**38 00305 00303 00306 00308 --> 00370

**39 00305 00303 00306 00308 --> 00418

**40 00305 00303 00307 00308 --> 00370

**41 00305 00303 00307 00308 --> 00418

**42 00305 00303 00306 02094 --> 00370

**43 00305 00303 00306 02094 --> 00418

**44 00305 00303 00307 02C94 --> 00370

**45 00305 00303 00307 02C94 --> 00418

**46 00305 00303 00308 02C94 --> 00370

**47 00305 00303 00308 02C94 --> 00418

OBTENTION DU DIPLOME

***1 00305 00303 00304 00306 --> 00418

***2 00305 00303 00304 00307 --> 00418

***3 00305 00303 00304 00308 --> 00418

***4 00305 00303 00304 02C94 --> 00418

***5 00305 00303 00306 00307 --> 00418

***6 00305 00303 00306 00308 --> 00418

***7 00305 00303 00307 00308 --> 00418

***8 00305 00303 00306 02C94 --> 00418

***9 00305 00303 00307 02C94 --> 00418

**10 00305 00303 00308 02C94 --> 00418

ANNEXE A4

```
00305 00303 00304 00306 + 00305 00303 00304 00307 + 00305 00303 00304 00308 + 00305 00303 00304 02094 + 00305 00303
00306 00307 + 00305 00303 00306 00308 + 00305 00303 00307 00308 + 00305 00303 00306 02094 + 00305 00303 00307 02094
+ 00305 00303 00308 02094

    FORME DUALE:          21 NOEUDS       29 ARETES

00305 + 00303 + 00304 00306 00307 00308 + 00304 00307 00308 02094 + 00304 00306 00308 02094 + 00304 00306 00307 02094
+ 00306 00307 00308 02094
```

ANNEXE B - B1

```
ENTITE 8000 NOEUD
DEBUT
      CODE DE 0 A 99999 DISCR
      CYCLE DE 1 A 3
      SORTIE DE 0 A 9
      DATECRE DE 0 A 9999
      DATESUP DE 0 A 9999
      FONCTION ( E P M F R C Z X )
      TITRE TEXTE 1
      UNIVERSITE DE 0 A 99
      UER DE 0 A 999
      ENTITE 15 RATTACHEMENT
      DEBUT
            UNIVERSITE DE 0 A 999
            UER DE 0 A 999
      FIN
      CODE-MINISTERE DE 0 A 999999
      PO DE 0 A 99999
      PM DE 0 A 99999
      PI DE 0 A 99999
      ENTITE 100 0-FILS
      DEBUT
            VAL REFERENCE UN NOEUD
            ENTITE 15 FLUX
            DEBUT
                  VF DE 0 A 99999
            FIN
      FIN
      ENTITE 100 0-PERE
      DEBUT
            VAL REFERENCE UN NOEUD
      FIN
      ENTITE 100 1-FILS
      DEBUT
            VAL REFERENCE UN NOEUD
            ENTITE 15 FLUX
            DEBUT
                  VF DE 0 A 99999
            FIN
      FIN
      ENTITE 100 1-PERE
      DEBUT
            VAL REFERENCE UN NOEUD
      FIN
FIN
```

Annexe B1 : Structure SOCRATE pour la description du fichier des des enseignements

ANNEXE B - B2

```
NOEUD
 CODE                  : 0418
 CYCLE                 : 02
 SORTIE                : 03 (CODE SORTIE POUR DIPLOME)
 FONCTION              : E (CODE FONCTION POUR ETAPE)
 TITRE                 : SC MAITRISE ENSEIGNEMENT MATHEMATIQUES
 UNIVERSITE            : 01
 UER                   : 08
 CODE-MINISTERE        : 020022
 PO                    : 02
 PM                    : 02
 PI                    : 02
 0-FILS                (POINTEUR SUR LE NOEUD 10115)
  FLUX
   VF                    : 034
 0-FILS                (POINTEUR SUR LE NOEUD 10116)
  FLUX
   VF                    : 03
 1-FILS                (POINTEUR SUR LE NOEUD 249)
  FLUX
   VF                    : 0128
 1-FILS                (POINTEUR SUR LE NOEUD 278)
  FLUX
   VF                    : 048
 1-FILS                (POINTEUR SUR LE NOEUD 300)
  FLUX
   VF                    : 01
 1-PERE                (POINTEUR SUR LE NOEUD 1252)


NOEUD
 CODE                  : 0305
 SORTIE                : 01 (CODE SORTIE POUR PROCES-VERBAL)
 FONCTION              : F (CODE FONCTION POUR ENSEIGNEMENT ELEMENTAIRE)
 TITRE                 : SC UNITE B' MAITRISE ENSEIGNEMENT MATHS
 UNIVERSITE            : 01
 UER                   : 08
 0-PERE                (POINTEUR SUR LE NOEUD 10115)
 0-PERE                (POINTEUR SUR LE NOEUD 10116)
```

Annexe B2 : Editions des différentes caractéristiques qui décrivent les noeuds 418 et 305 utilisés pour la représentation de la maîtrise d'enseignement de mathématiques

(Les commentaires rajoutés à la main sont entre parenthèses)

ANNEXE C - C1

-05- GEOGRAPHIE GENERALE ET ALPINE   CYCLE: 2

APE: 00150   GEO MAITRISE * GEOGRAPHIE HUMAINE

SCRIPTION SUR PASSAGE A MINIMA POSSIBLE AVEC:
    00109   GEO MAITRISE GEOGRAPHIE HUMAINE TITRE CONDITIONNEL

RELATION(S) OU EQUIVALENCE(S):
    03325   GEO TITRE DEROGATOIRE MAITRISES * GEOGRAPHIE

APE(S) EN ENTREE:
    00104   GEO MAITRISE * GEOGRAPHIE HUMAINE 1ERE ANNEE

ISEIGNEMENT(S):
    00115 (M)   GEO CES SOCIOLOGIE GENERALE
    00116 (M)   PS C3 SOCIOLOGIE * BEAUX ARTS & * OEUVRES * LA CULTURE
    00117 (M)   GEO CES HISTOIRE * PHILOSOPHIE ANCIENNE 4103
    00118 (M)   GEO CES HISTOIRE * PENSEE MEDIEV * EXPRES. FRANCAISE & ROMAN
    00119 (M)   GEO CES HISTOIRE * PHILOSOPHIE MODERNE 4105
    00120 (M)   GEO CES HISTOIRE * PHILOSOPHIE CONTEMPORAINE 4106
    00121 (M)   GEO CES PSYCHOSOCIOLOGIE 4107
    00122 (M)   GEO CES ECONOMIE POLITIQUE & SOCIALE 4163
    00123 (M)   GEO INTRODUCTION A LA SCIENCE POLITIQUE
    00124 (M)   GEO CES METHODES * ANALYSE * STRUCTURES DEMOGRAPHIQUES 4420
    00125 (M)   GEO CES METHODES * ETUDE*L'ORGANISATION*L'ESPACE URBAIN 4421
    00126 (M)   GEO CES METHODES * RECHERCHE EN GEOGRAPHIE AGRAIRE 4447

Annexe C1 : Description de la fraction de cursus correspondant à la maîtrise de géographie humaine

DICTIONNAIRE GENERAL DES ENSEIGNEMENTS

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE - GRENOBLE 1

02746 DU SKI DIPLOME

02742 DU SKI ENSEIGNEMENT 1ERE ANNEE

02745 DU SKI ENSEIGNEMENT 2EME ANNEE

02743 DU SKI EXAMEN FIN 1ERE ANNEE

02741 DU SKI TITRE 1ERE ANNEE

02744 DU SKI TITRE 2EME ANNEE

00049 DUES ANGLAIS FACULTATIF PC2 CB2

00039 DUES ANGLAIS OPTION FAIBLE

00040 DUES ANGLAIS OPTION FORT

00044 DUES ANGLAIS PC2 CB2 EPREUVE FACULTATIVE

00072 DUES BG2

00066 DUES BG2 2EME ANNEE TEMPS COMPLET

00067 DUES BG2 2EME ANNEE 1ERE MI-TEMPS

00068 DUES BG2 2EME ANNEE 2EME MI-TEMPS

00028 DUES CB-BG TITRE SPECIALISE

00043 DUES CB-BG 1ERE ANNEE

00036 DUES CB-BG 1ERE ANNEE TEMPS COMPLET

00037 DUES CB-BG 1ERE ANNEE 1ERE MI-TEMPS

00038 DUES CB-BG 1ERE ANNEE 2EME MI-TEMPS

00071 DUES CB2

00050 DUES CB2 BG2 TITRE SPECIALISE

00056 DUES CB2 OPTION MATHEMATIQUES

00063 DUES CB2 2EME ANNEE TEMPS COMPLET

00064 DUES CB2 2EME ANNEE 1ERE MI-TEMPS

00065 DUES CB2 2EME ANNEE 2EME MI-TEMPS

00041 DUES MP1

00026 DUES MP1 TITRE SPECIALISE

00030 DUES MP1 1ERE ANNEE TEMPS COMPLET

00031 DUES MP1 1ERE MI-TEMPS

Annexe C2 : Liste générale des diplômes et enseignements organisés par l'U.S.M.G.

ANNEXE C - C3

~~00206~~	~~1~~	~~07~~	~~M~~	~~SC INSTITUT INFORMATIQUE CHIMIQUE 1ERE ANNEE * SCOLARITE~~
00207	1	07	M	SC 1ERE ANNEE * ETUDES MAIT INFORMATIQUE APPLI A LA GESTION
~~00240~~	~~1~~	~~07~~	~~E~~	~~SC PROGRAMMEUR * ETUDES DIPLOME~~
00241	1	07	E	SC INSTITUT INFORMATIQUE CHIMIQUE 1ERE ANNEE
~~00242~~	~~1~~	~~07~~	~~E~~	~~SC MAITRISE INFORMATIQUE APPLIQUEE A LA GESTION 1ERE ANNEE~~
00246	1	07	E	SC MAITRISE EEA 1ERE ANNEE
~~00247~~	~~1~~	~~07~~	~~E~~	~~SC MAITRISE MECANIQUE 1ERE ANNEE~~
00248	1	07	E	SC MAITRISE GEOLOGIE 1ERE ANNEE
~~00260~~	~~1~~	~~07~~	~~X~~	~~SC PROGRAMMEUR EXPERT TITRE EX PROG=ETUDES OU C2 INFOR 6020~~
00261	1	07	M	SC PROGRAMMEUR * ETUDES EQUIV STAGE = PROGRAMMEUR EXPERT
~~00262~~	~~1~~	~~07~~	~~R~~	~~SC INSTITUT INFORMATIQUE CHIMIQUE 2E ANNEE TITRE SPECIALISE~~
00263	1	07	R	SC TITRE SPECIALISE 2E ANNEE MAIT INF APPLIQUEE A LA GESTION
~~00270~~	~~1~~	~~07~~	~~R~~	~~SC TITRE SPECIALISE 2E ANNEE MAITRISE EEA~~
00271	1	07	C	SC TITRE CONDITIONNEL 2E ANNEE MAITRISE EEA
~~00272~~	~~1~~	~~07~~	~~R~~	~~SC TITRE SPECIALISE 2E ANNEE MAITRISE MECANIQUE~~
00273	1	07	C	SC TITRE CONDITIONNEL 2E ANNEE MAITRISE MECANIQUE
~~00274~~	~~1~~	~~07~~	~~R~~	~~SC TITRE SPECIALISE 2E ANNEE MAITRISE GEOLOGIE~~
00275	1	07	C	SC TITRE CONDITIONNEL 2E ANNEE MAITRISE GEOLOGIE
~~00301~~	~~1~~	~~07~~	~~M~~	~~SC PROGRAMMEUR EXPERT & PRATIQUE~~
00302	1	07	M	SC PROGRAMMEUR EXPERT STAGE
~~00311~~	~~1~~	~~07~~	~~M~~	~~SC INSTITUT INFORMATIQUE CHIMIQUE 2EME ANNEE SCOLARITE~~
00312	1	07	M	SC 2EME ANNEE * ETUDES MAIT INFORMATIQUE APPLI A LA GESTION
~~00319~~	~~1~~	~~07~~	~~M~~	~~SC CES 6014 ELECTRONIQUE C3 EEA~~
00320	1	07	M	SC CES 6015 ELECTROTECHNIQUE C3 EEA
~~00322~~	~~1~~	~~07~~	~~M~~	~~SC CES 6026 MECANIQUE * MILIEUX DEFORMABLES C3 MECANIQUE~~
00323	1	07	M	PARTAGE CEC-SC CES 6017 GEOLOGIE APPLIQUEE
~~00324~~	~~1~~	~~07~~	~~M~~	~~PARTAGE CEC-SC CES 6019 GEOLOGIE STRUCTURALE~~
00342	1	07	R	SC MST SC*MATERIAUX ANALYSE MESURE CONTROLE TITRE SPEC
~~00343~~	~~1~~	~~07~~	~~M~~	~~SC MST SC * MATERIAUX 1ERE ANNEE ENS~~
00344	1	07	E	SC MST SC * MATERIAUX 1ERE ANNEE FIN
~~00345~~	~~1~~	~~07~~	~~M~~	~~SC CES 6005 AUTOMATIQUE~~
00350	1	07	X	DEUG SC*NATURE&*VIE 1E ANNEE TITRE DEROG
~~00353~~	~~1~~	~~07~~	~~X~~	~~DEUG SC*STRUCTURES&*MATIERE OPTION MP 1E ANNEE TITRE DEROG~~
00354	1	07	X	DEUG SC*STRUCTURES&*MATIERE OPTION PC 1E ANNEE TITRE DEROG
~~00357~~	~~1~~	~~07~~	~~M~~	~~DEUG SC*STRUCTURES&*MATIERE OPTION PC 1E AN ENS T.C.~~
00358	1	07	M	SC CES 6018 GEOLOGIE HISTORIQUE
~~00363~~	~~1~~	~~07~~	~~M~~	~~SC CES 6023 MATHEMATIQUES C4~~
00367	1	07	M	SC CES 6027 ONDES & MATIERES C4
~~00368~~	~~1~~	~~07~~	~~M~~	~~PARTAGE CEC-SC CES 6028 PETROLOGIE~~
00371	1	07	E	SC MST ANALYSE MESURE CONTROLE 1E ANNEE FIN
~~00372~~	~~1~~	~~07~~	~~M~~	~~SC CES 6032 PHYSIQUE & METROLOGIE C4~~
00373	1	07	M	SC CES 6033 SIGNAUX & SYSTEMES C4
~~00387~~	~~1~~	~~07~~	~~M~~	PARTAGE CEC-SC CES 6409 HYDROGEOLOGIE
00389	1	07	F	SC CES 6491 MECANIQUE APPROFONDIE OPT ECHANGES THERMIQUES
~~00390~~	1	07	F	SC CES 6492 MECANIQUE APPROFONDIE OPT ECOULEMENT * FLUIDES
00392	1	07	F	SC CES 6494 MECANIQUE APPROFONDIE OPT HYDRAULIQUE
~~00393~~	1	07	F	SC CES 6495 MECANIQUE APPROFONDIE OPT MECANIQUE SPATIALE
00394	1	07	M	SC CES 6411 MECANIQUE APPROFONDIE C4
00397	1	07	M	PARTAGE CEC-SC CES 6414 MINERALOGIE
00398	1	07	M	SC CES 6420 INFORMATIQUE * GESTION C4
00407	1	07	E	SC PROGRAMMEUR EXPERT DIPLOME
00408	1	07	E	SC INSTITUT INFORMATIQUE CHIMIQUE DIPLOME
00409	1	07	E	SC MAITRISE INFORMATIQUE APPLIQUEE A LA GESTION
00415	1	07	E	SC MAITRISE EEA
00418	1	07	E	SC MAITRISE MECANIQUE

CODE UER FONCTION TITRE
UNIVERSITE

Annexe C3 : <u>Liste générale des noeuds triée par ordre numérique par université et par UER</u>

(ici USMG, UER Formation des cadres techniques).

ANNEXE C- C4

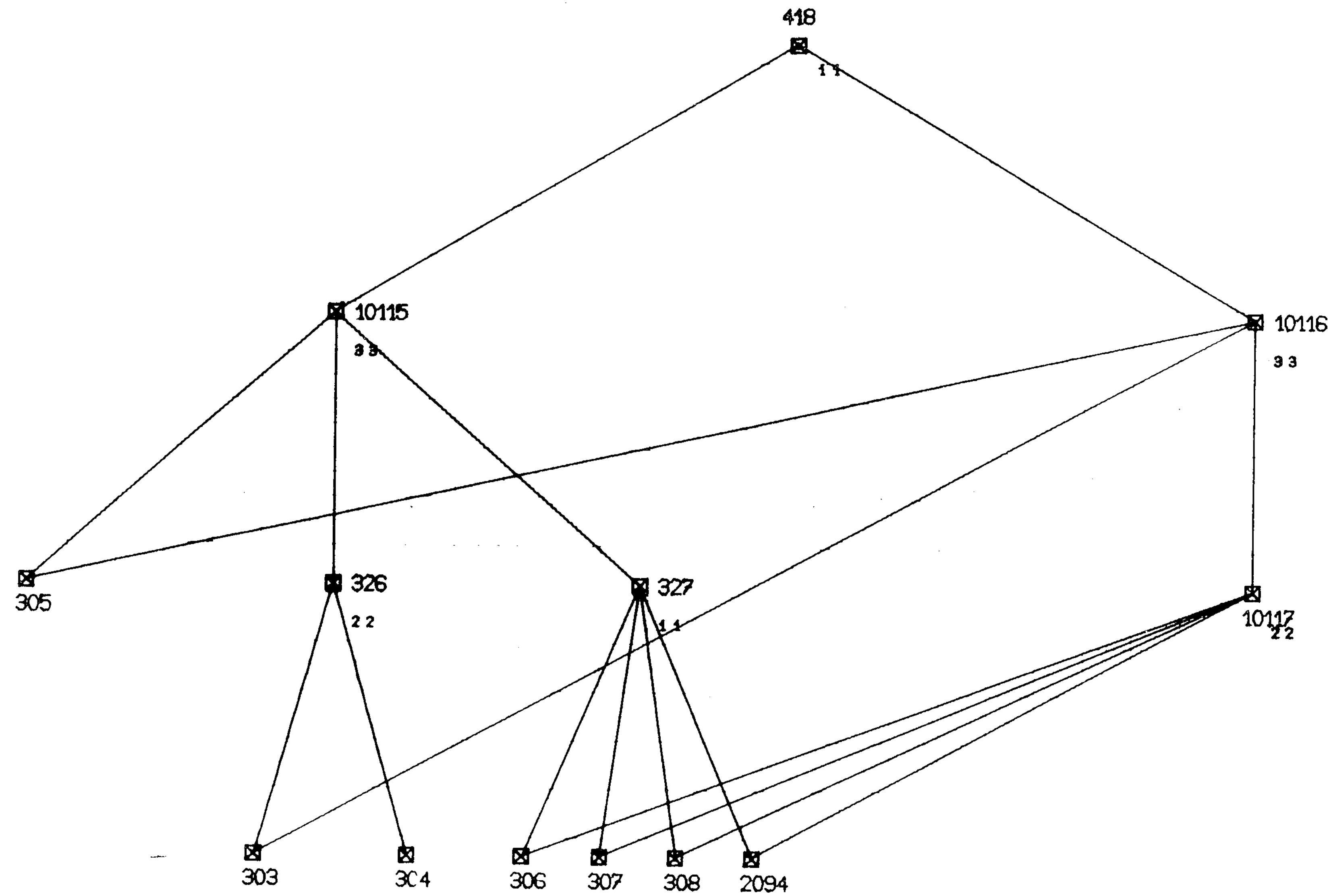
GRAPHES REPRESENTATIFS DES CONDITIONS D'OBTENTION

DE DIFFERENTS DIPLOMES

SC MAITRISE ENSEIGNEMENT MATHEMATIQUES

SYGESCO

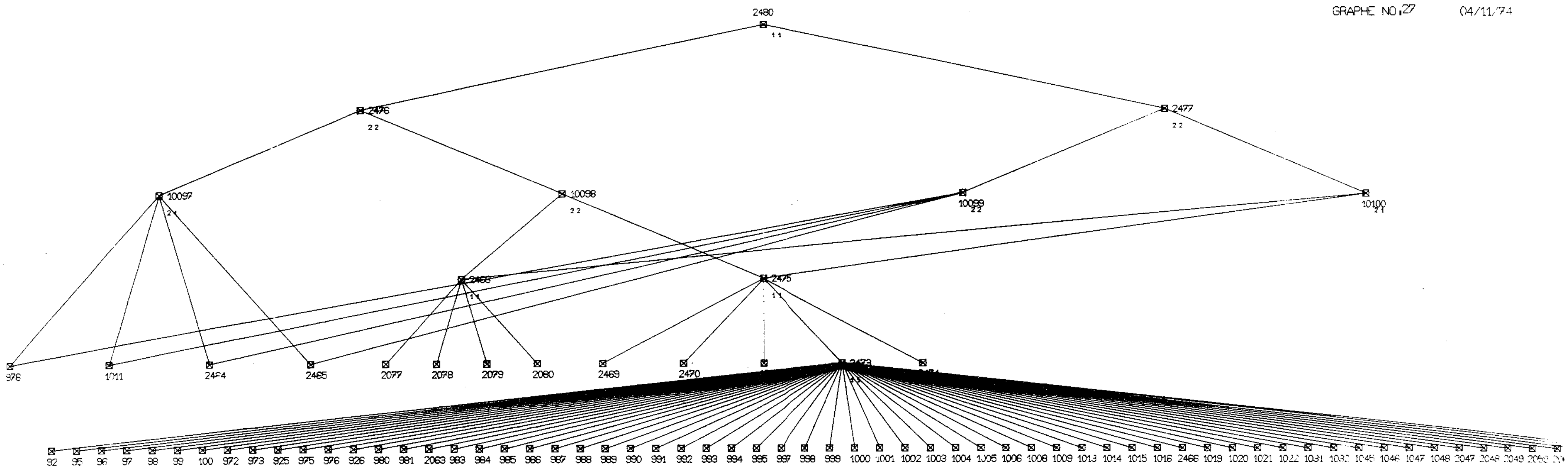
GRAPHE NO,79 04/11/74



HIS LICENCE D'HISTOIRE

SYGESCO

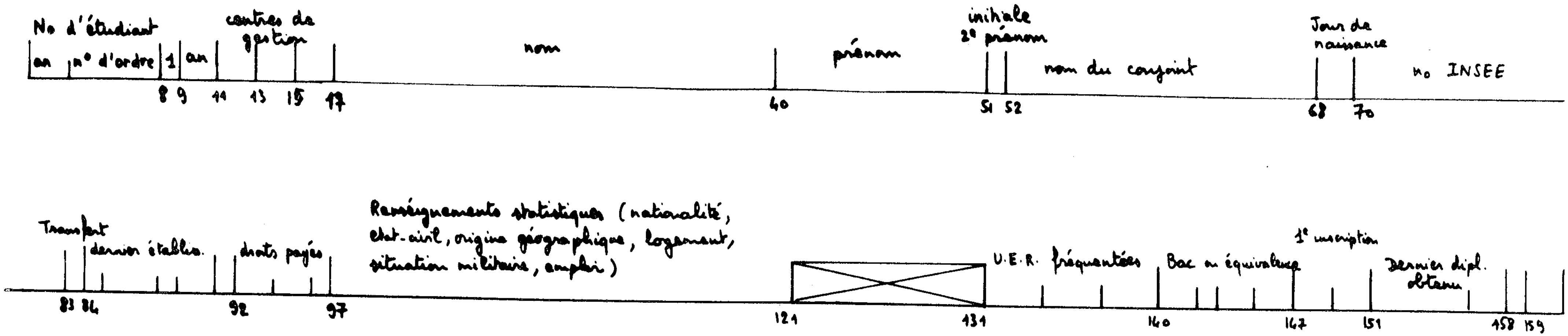
GRAPHE NO.27 04/11/74



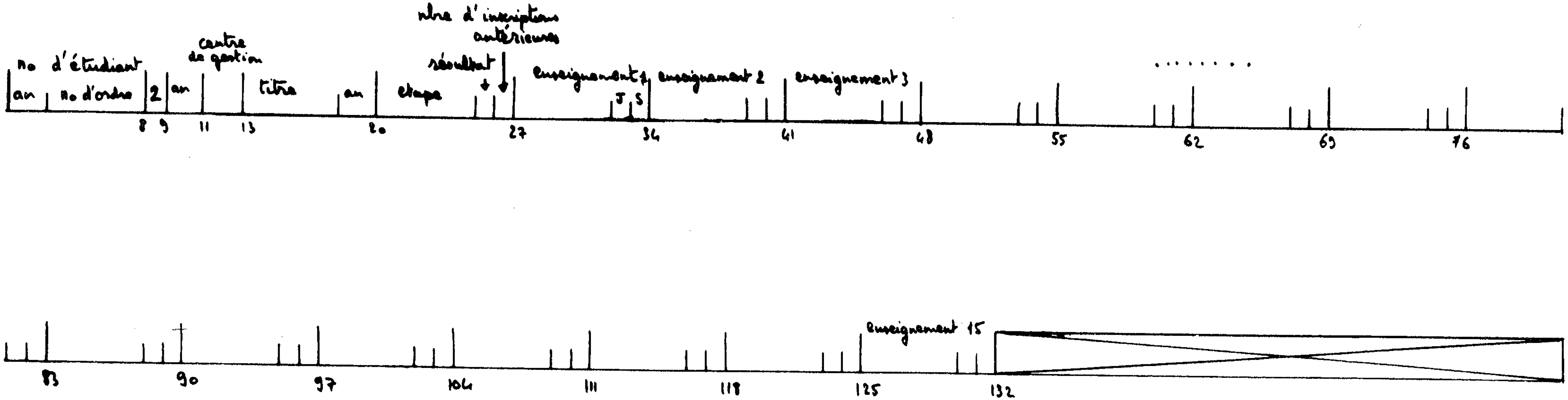
DESSIN DU FICHIER ETUDIANT

ANNEXE D

IDENTIFICATION (partie fixe)



ETUDES SUIVIES (partie répétée n fois)



B I B L I O G R A P H I E

[1] J.R. ABRIAL - Projet SOCRATE - Course given in European Countries
Seminar on Computer System Architecture
Alpe d'Huez, décembre 1973

[2] N.J. NILSON - Problem Solving Methods in Artificial Intelligence
Mac Graw Hill, 1971

[3] C. DELOBEL - Contribution théorique à la conception et l'évaluation d'un système d'information appliqué à la gestion.
Thèse, Université de Grenoble, 1973

[4] J. KUNTZMANN - Algèbre de Boole
Dunod, 1965

[5] C. BERGE - Graphes et Hypergraphes
Dunod, 1970

[6] D.E. KNUTH - The Art of Computer Programming
Fundamental Algorithms
Addison Wesley, 1968

[7] C. DELOBEL et D. PORTAL - Méthodologie pour l'analyse des structures d'enseignement d'une Université
(I.F.I.P., à paraître)

[8] C. DELOBEL et R.G. CASEY - I.B.M. Journal of Research and Development
Vol. 17, n°5, 1973

[9] A. FLORY - Optimisation de l'implantation des fichiers situés sur disques par des méthodes de classification automatique
Thèse, Université Claude Bernard, Paris, 1973

[10] D.E. KNUTH - The Art of Computer Programming
Sorting and Searching
Addison Wesley, 1973

[11] Projet SOCRATE - Nouvelles spécifications
Institut de Mathématiques Appliquées, Grenoble, 1972

[12] Systèmes de programmation générateurs de machines virtuelles
C.I.C.G., 1973

[13] A. GROSSI - Algorithme à séparation de variables pour la dualisation d'une fonction booléenne.
Revue française d'Automatique, Informatique et Recherche Opérationnelle, février 1974

[14] O.C.D.E. Programme sur la gestion des établissements d'enseignement supérieur. Deuxième conférence générale des institutions membres (20-22 Janvier 1975)

TABLE DES MATIERES