



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

THESE

présentée

à la Faculté des Sciences de l'Université de Metz

pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITE DE METZ

par

Michel SPADONI



ETUDE D'UN SYSTEME DE GENERATION AUTOMATIQUE DE GAMMES DE FABRICATION

Soutenue le 23 juin 1987 devant la Commission d'Examen :

MM. A. BOURJAULT

Rapporteur

C. COUET

Rapporteur

R. GESLOT

B. MUTEL

M. POTIER-FERRY

AVANT PROPOS

Cette étude a été réalisée dans le cadre des activités du Laboratoire d'Automatique et d'Electronique Industrielles (L.A.E.I.) de l'Université de Metz.

J'exprime ma reconnaissance à Monsieur le Professeur B. Mutel, Directeur du L.A.E.I., qu'il anime avec compétence et dynamisme, pour m'avoir accueilli au sein du laboratoire.

Qu'il me soit permis d'exprimer à Monsieur B. Mutel, le témoignage de ma gratitude pour la bienveillance et la confiance qu'il n'a cessé de m'accorder au cours de ces années d'études.

Nous sommes très reconnaissants à Monsieur A. Bourjault, Professeur à l'Ecole Nationale Supérieure de Micro Mécanique de Besançon (E.N.S.M.M.B.) d'avoir accepté de juger ce travail en tant que rapporteur et membre du jury.

Nous tenons à remercier Monsieur C. Couet, Professeur à l'Université de Reims, pour avoir accepté d'examiner ce travail comme rapporteur et pour sa participation au jury.

Nous sommes très honorés par la présence dans ce jury de Monsieur R. Geslot, Chef de département adjoint, département Usinage et Méthode du Centre Technique des Industries Mécaniques (C.E.T.I.M.), qui a accepté de juger ce travail.

Nous remercions vivement Monsieur M. Potier-Ferry, Professeur à l'Université de Metz, pour l'attention donnée à ce travail par sa participation au jury.

J'adresse mes remerciements amicaux à tous les camarades du laboratoire, pour les nombreux échanges constructifs et l'ambiance qu'ils entretiennent.

Que Mademoiselle F. Bitat qui s'est dévouée avec gentillesse et compétence pour la frappe de ce mémoire soit assurée de toute ma reconnaissance.

SOMMAIRE

	Pages
<u>CHAPITRE I</u> <u>INTRODUCTION</u>	8
1. Introduction	8
2. Etat de l'art	9
3. Types de gamme	15
3.1 Moyen terme : gamme de produit $g(p_i)$	16
3.2 Court terme : gamme opératoire $g(a_i)$	16
3.3 Très court terme : gamme détaillée $g^d(a_i)$	17
4. Approche	17
4.1 Niveau implantation	20
4.2 Niveau utilisation	20
4.3 Niveau apprentissage	21
5. Plan de la thèse	21

CHAPITRE II	REPRESENTATION DES GAMMES DE FABRICATION	24
1. Introduction		24
2. Choix d'une représentation		26
2.1 Automate à états finis		26
2.2 Réseaux de Pétri		28
2.3 Grafcet		30
2.4 Choix d'un modèle de représentation		32
3. Définition du Grafcet M		34
3.1 Description générale du Grafcet M		35
3.1.1 Actions virtuelles		35
3.1.2 Etape		35
3.1.3 Transition		36
3.1.4 Réceptivité		37
3.1.5 Arcs orientés		37
3.2 Règles de structuration du Grafcet M		38
3.2.1 Structure générale		38
3.2.2 Structure de base : séquence chaînée		39
3.2.3 Structure de base : sélection de séquences		39
3.2.4 Structure de base : séquences simultanées		40
3.3 Règles d'évolution du Grafcet M		40
3.3.1 Règle 1		40
3.3.2 Règle 2		41
3.3.3 Règle 3		41
3.3.4 Règle 4		41
3.3.5 Règle 5		41
3.3.6 Règle 6		41

4. Description formelle du Grafcet M	42
4.1 Notations	42
4.2 Formalisation des règles d'évolution	43
4.2.1 Règle 1	44
4.2.2 Règle 2	44
4.2.3 Règle 3	45
4.2.4 Règle 4	45
4.2.5 Règle 5	45
4.2.6 Règle 6	46
4.3 Formalisation des règles de structure	49
4.3.1 Séquence chaînée	49
4.3.2 Séquences sélectives	49
4.3.3 Séquences simultanées	50
4.4 Représentation d'une gamme de fabrication	50
 CHAPITRE III <u>ETUDE DU SYSTEME DE GENERATION DE GAMMES</u>	 52
1. Introduction	52
2. Formalisation de la connaissance	54
2.1 Connaissance déductive	54
2.2 Structure générale des règles de production	55
2.3 Expression des différents types de règles de production	57
3. Implantation de la connaissance	60
3.1 Prédicats	60
3.2 Fonction	61
3.3 Objet	62
3.4 Définition des opérateurs	63

3.5	Arguments	63
3.5.1	Types d'arguments pour un passage par valeur	63
3.5.2	Types d'arguments pour un passage par constante	64
3.5.3	Types d'arguments pour un passage par variable	65
3.6	Implantation en langage interne	68
3.7	Implantation en langage externe	69
4.	Exploitation de la connaissance	73
4.1	Mécanisme d'exploitation	74
4.2	Caractéristiques moteur I	77
4.2.1	Choix d'un état	78
4.2.1.1	Détermination du cycle de prise en compte d'une restauration d'état	78
4.2.1.2	Moyen utilisé pour la restauration d'état	79
4.2.1.3	Détermination du point de redémarrage	80
4.2.2	Choix d'une règle	81
4.2.2.1	Invocation des règles	81
4.2.2.2	Préselection des règles	82
4.2.2.3	Sélection des règles	83
4.2.3	Partie déclenchement	83
4.3	Caractéristiques moteur II	83
4.3.1	Choix des états	84
4.3.2	Choix des règles	85
4.4	Structure de l'arbre de recherche du mécanisme d'exploitation	85
5.	Exemple académique	86
5.1	Test	92

CHAPITRE IV APPLICATION DE SAGGA A LA FABRICATION MECANIQUE 99

1. Introduction	99
2. Présentation du système SAGGA	100
3. Présentation du système d'édition de gammes	102
4. Applications	104
4.1 Définition des références pièces	104
4.2 Définition de l'isostatisme	105
4.3 Système de codification	106
4.4 Exemples	107
4.4.1 Définition de la base de connaissances de la famille	108
a) Dessin de définition de l'article A	108
b) Dessin de définition de l'article B	111
4.4.2 Implantation de la base de règles	113
a) Définition du fichier règles	113
b) Fichier FicQ	115
c) Fichier FicQE	115
4.4.3 Exemple 1	115
4.4.4 Exemple 2	118
a) Dessin de définition de l'article C	118
b) Recherche de gammes	119
c) Modifications en langage externe	120
d) Etat de la base de règles après modification et avant mise à jour par le préparateur	122
4.4.5 Modification en langage interne de la base de règles	123
a) Fichier règles	123
b) Fichier FicQ	125
4.4.6 Exemple 3	126
a) Dessin de définition de l'article C	126
b) Recherche de gammes	127
4.4.7 Exemple 4	129

<u>CHAPITRE V</u>	<u>CONCLUSION</u>	132
-------------------	-------------------------	-----

ANNEXES

A.	Définition d'un vocabulaire de base.....	137
B.	Réseaux de Pétri : complément.....	142

<u>BIBLIOGRAPHIE</u>		144
----------------------	-------	-----

ANNEXES

C.	Structures informatiques.....	1
1.	Introduction	1
2.	Introduction des types d'arbres.....	2
2.1	Ensemble noeuds "contrôle" et "en avant".....	2
2.2	Ensemble noeuds "en arrière".....	3
3.	Contrôle de la connaissance	3
3.1	Définition de l'algorithme	4
4.	Procédure de validation des réceptivités.....	10
5.	Partie recherche	10
5.1	Algorithme moteur I.....	11
5.2	Algorithme moteur II.....	16
D.	Système d'édition de gammes détaillées.....	33
1.	Introduction.....	33
2.	Fichiers technologiques.....	34
3.	Programme Gestion F.....	35
4.	Programme Grille.....	39
5.	Programme Util.....	46
6.	Exemple d'utilisation.....	49

CHAPITRE I

Introduction

1. INTRODUCTION

Dans une entreprise, les besoins ainsi que les informations nécessaires à la fabrication d'un produit, diffèrent selon le type de production et le niveau d'automatisation des moyens de production. Il est donc important de bien préciser l'ensemble des caractéristiques utilisé dans un système de production. Aussi sous le vocable gamme de fabrication, correspond en réalité différentes informations et documents techniques selon le niveau d'organisation de la production auquel on se situe.

Après avoir rappelé les différentes approches utilisées [1, 49, 50, 51, 84] par les systèmes actuels de génération de gammes de fabrication, nous définissons de façon précise les différents types de gammes que l'on rencontre dans les systèmes de production. Puis nous présentons le contexte dans lequel se situe cette étude et justifions l'approche originale choisie. Enfin le dernier paragraphe de ce chapitre résume le plan de ce rapport.

2. ETAT DE L'ART

Chaque fois que l'on veut définir une classification de différents objets on peut utiliser plusieurs critères. Du point de vue de l'utilisateur, les systèmes de génération de gammes de fabrication peuvent être divisés en trois classes :

- a) Les éditeurs de textes spécialisés, qui proposent un cadre de rédaction des gammes par grilles d'affichage et de saisies, pouvant faire appel à des modules particuliers pour effectuer des calculs élémentaires (vitesse d'avance, vitesse de coupe, calcul des temps, ...). Ces systèmes sont des logiciels d'assistance à l'élaboration des gammes de fabrication (Autogam [15], Gam +).
- b) Les systèmes basés sur le concept technologie de groupe [52, 53] (figure 1.1) qui comparent, par l'utilisation d'informations numériques, un nouveau plan de pièce à réaliser à un ancien dont la gamme de fabrication existe déjà. Le système recherche dans une banque de données, la gamme de fabrication supposée la plus proche de cette nouvelle pièce. Le préparateur adapte si nécessaire cette proposition, à travers un éditeur de texte spécialisé. Cette classe est généralement appelée approche par variante (ICAPP [5, 6, 12], APPAS [85], AUTOCAP [81], MULTIPLAN, SAVEGAM [7]).

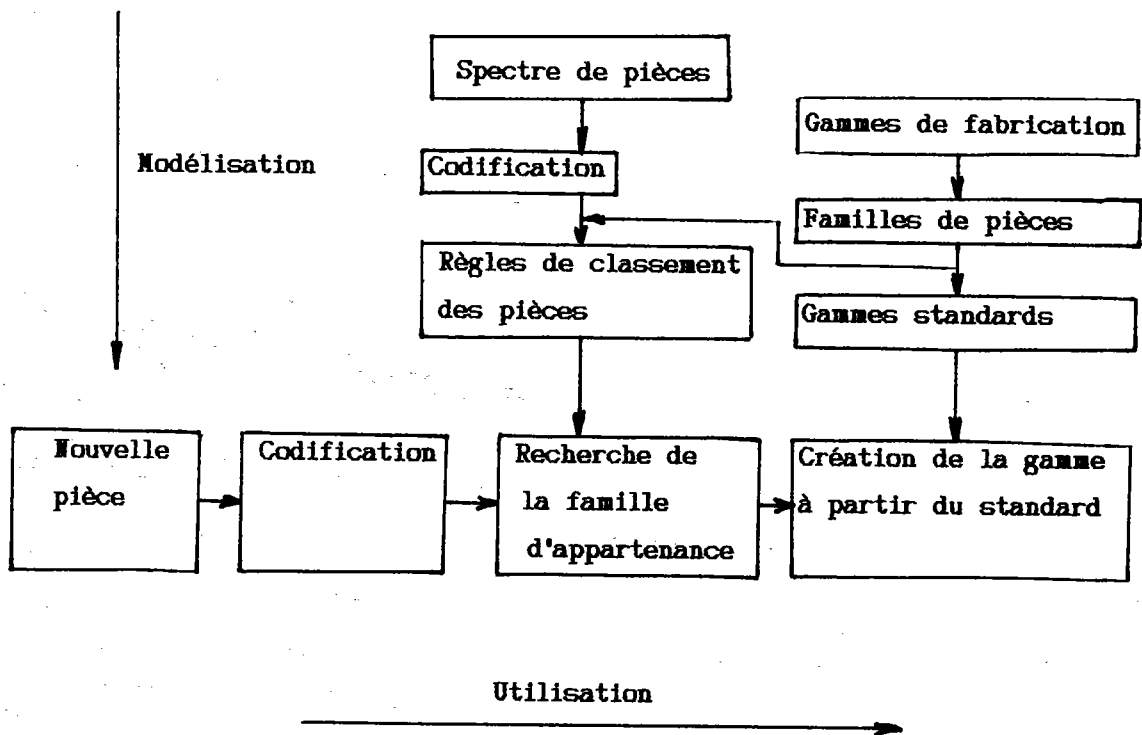


figure I.1 : Schématisation de la classe "Approche par variante" [3].

- c) La troisième classe est définie comme l'approche générative [2, 54, 56, 58] (figure I.2). Dans cette approche la forme de la pièce brute doit être définie soit :
- * manuellement, en utilisant des instructions géométriques et technologiques.
 - * directement, en interprétant la base de données géométriques du système C.A.O.

Les systèmes appartenant à cette classe, déterminent entièrement la gamme de fabrication : les séquences d'opérations, les caractéristiques d'usinage, Dans ce cas le travail du préparateur est réduit, aussi cette approche est la plus intéressante. De nombreux travaux et différents softwares ont été présentés à ce sujet (EXCAP [8], TOM, TAUPROG [55], GARI [10], AUTAP [58], TOSICAP [57], XPLANE [59], OPEX [60]).

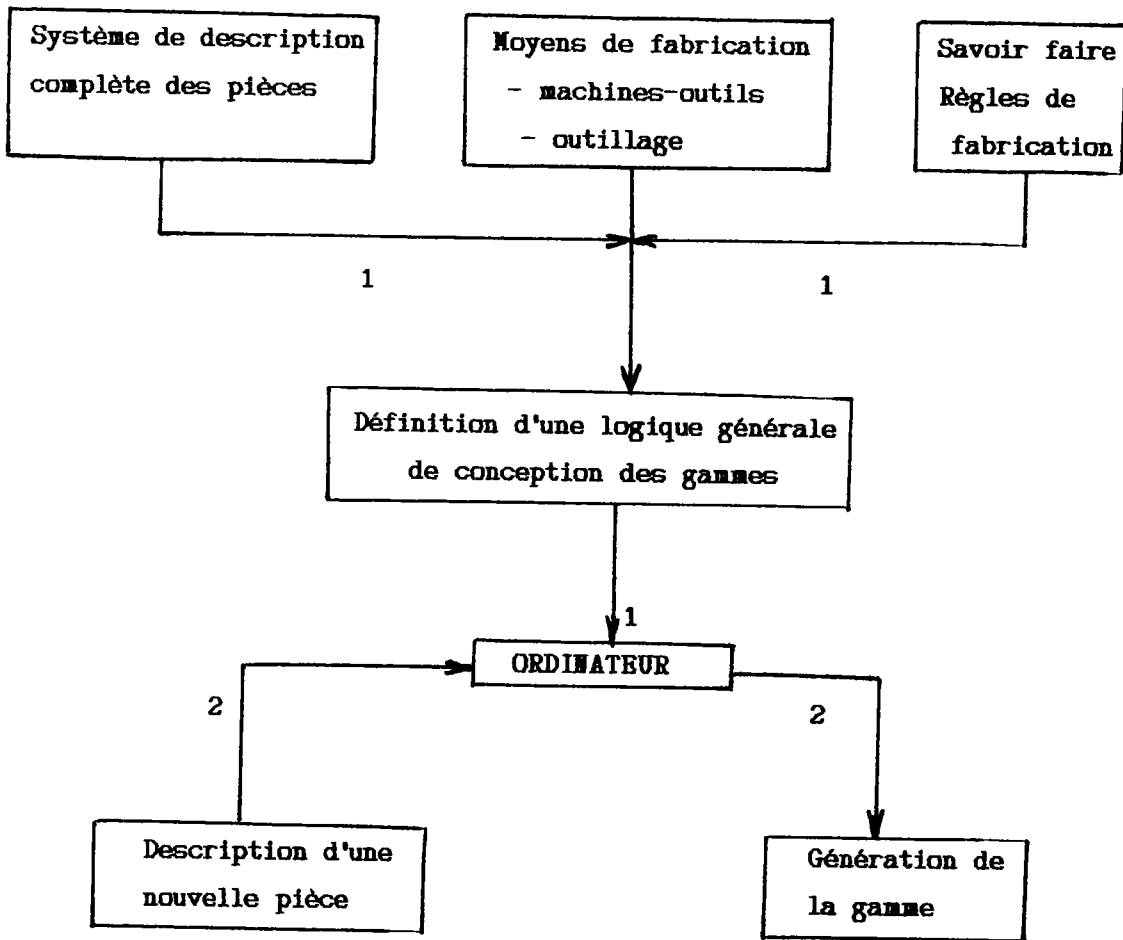


figure I.2 : Schématisation de la classe "Approche générative" [3].

Si maintenant nous désirons classer les systèmes de génération de gammes de fabrication du point de vue de leur algorithme, la classification suivante semble mieux adaptée :

- a) L'approche par variante, basée sur la technologie de groupe et ses extensions, tel que l'utilisation de tables de décisions et le codage automatique des pièces. Ces méthodes nécessitent la définition de familles de pièces et des gammes de fabrication associées. Elles ne sont pas véritablement capables de générer de nouvelles gammes. Le tableau figure I.3 résume les caractéristiques de cette approche.

Avantages	Inconvénients	Applications
- inclue les techniques de la T.G. - peut utiliser un grand nombre de familles - utilise un programme général	- spécifique à chaque entreprise - manque de flexibilité - problème de maintien de la cohérence de la base de données	- mono produit avec peu de variantes - production stable

figure I.3 : Tableau récapitulatif des caractéristiques de l'approche par variante.

De part ces caractéristiques, l'approche par variante est adaptée pour des fabrications stables, par lots de moyennes séries et nécessite des études coûteuses pour l'implantation dans chaque entreprise.

- b) Dans la méthode du modèle, un algorithme programmé représente la gamme de fabrication de la pièce la plus complexe d'une famille. Dans ce cas la gamme de fabrication de la nouvelle pièce est un sous-ensemble de ce modèle. Le logiciel (algorithme programmé) est spécifique à chaque famille, et à chaque entreprise donc une modification du modèle implique la modification d'une partie du logiciel. Le tableau de la figure I.4 représente l'ensemble des caractéristiques de cette méthode.

Avantages	Inconvénients	Application
- intégration dans un système complet CFAO - la gamme de fabrication est un sous-ensemble du modèle	- logiciel spécifique à l'entreprise et à chaque famille - non flexible	- mono produit avec variantes - pour les moyennes et grandes séries - production stable

figure I.4 : Tableau récapitulatif des caractéristiques d'une méthode par modèle.

De part ces caractéristiques, la méthode par modèle est utilisée pour les fabrications stables, par lots de moyennes et grandes séries, avec un nombre de familles restreint.

- c) Les méthodes basées sur l'intelligence artificielle [4, 9, 11, 79, 80] utilisent principalement les systèmes experts.

C'est une approche qui est très prometteuse, cependant à partir de l'expérience actuelle sur l'application des systèmes experts, on peut définir quelques limites [61] à cette approche :

- * Un système expert est efficace seulement s'il existe une très bonne expertise du domaine d'application. Ce n'est pas le cas de la fabrication mécanique.
- * L'expérience et le savoir faire d'une entreprise sont en général mémorisés dans un ordinateur par des tables de données et non des règles.

- * La base de connaissances doit suivre le progrès technique et être spécifique à chaque entreprise. En effet d'un point de vue commercial une entreprise est meilleure qu'une autre parce que son savoir faire, pour concevoir et réaliser ses produits, est meilleur. Aussi dans une économie de libre concurrence les bases de connaissances ne seront jamais communes. Ceci signifie des études coûteuses pour développer des bases de connaissances particulières.

Le tableau figure I.5 résume l'ensemble des caractéristiques de cette méthode.

Avantages	Inconvénients	Application
- flexibilité - génération de tous types de gammes - mémorisation du savoir faire	- pas de bonne expertise dans le domaine de la fabrication - utilisation coûteuse - non adaptable	- multi produit - grandes et moyennes séries - production variée

figure I.5 : Tableau récapitulatif des caractéristiques de la méthode basée sur l'Intelligence artificielle.

Une dernière remarque concernant les méthodes de génération de gammes de fabrication, qui ont donné lieu à de nombreux développements. La majorité de ces développements utilise des tables de décisions pour élargir le champ d'application du logiciel et diminuer son coût d'adaptation. Il nous semble excessif d'appeler "système d'intelligence artificielle" ou "système expert", un logiciel de génération de gammes de fabrication, qui utilise uniquement des tables de décisions définies ou non par des règles de production.

3. TYPES DE GAMME

La figure I.6 résume les principales étapes d'un système de gestion de production [17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28]. Le vocabulaire utilisé est précisé en annexe A.

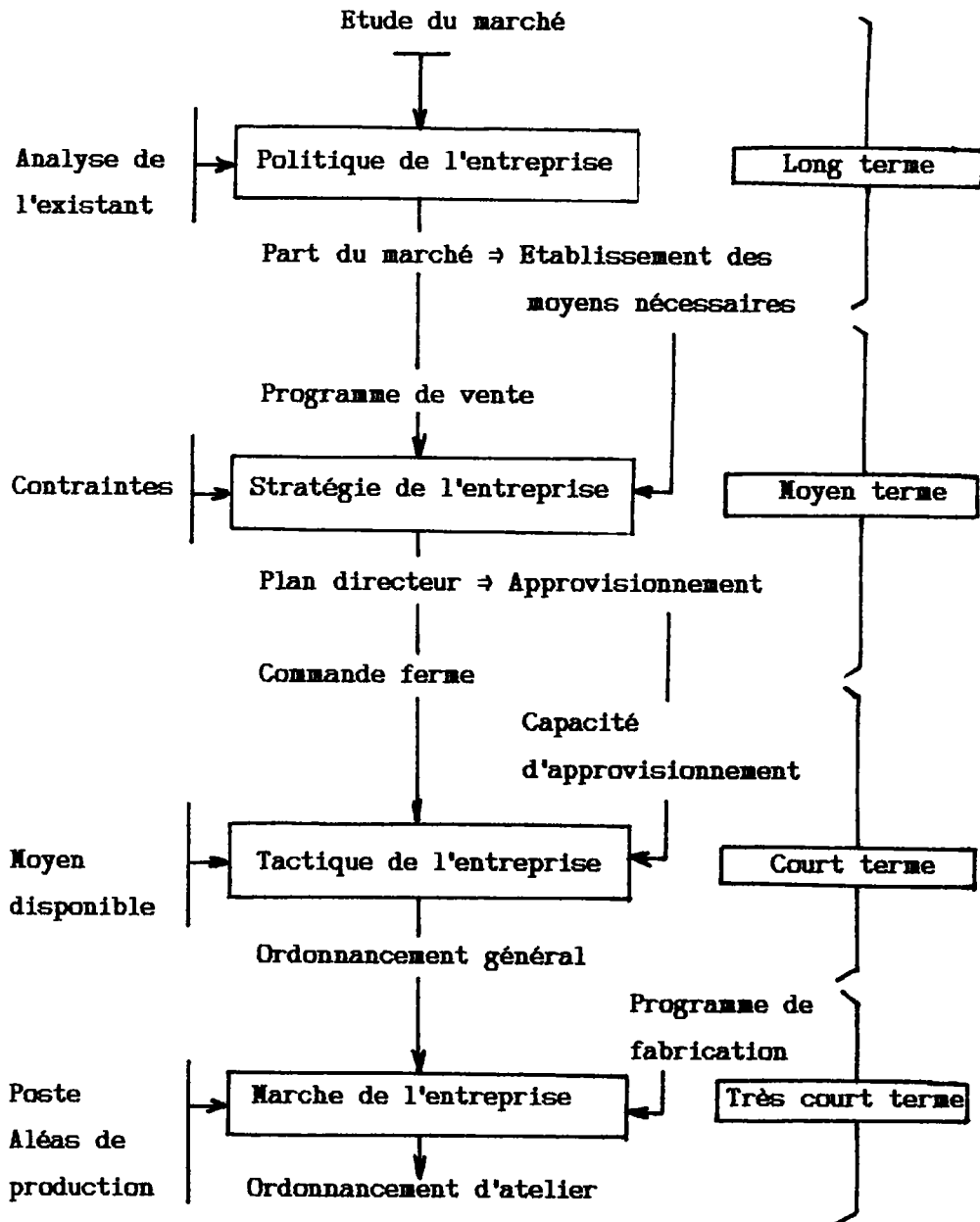


figure I.7 : Synoptique des tâches, incombant à un système de gestion de production.

Dans la pratique, on recherche des gammes de fabrication différentes, selon que le système de gestion de production considère ; le moyen terme, le court terme ou le très court terme.

3.1 Moyen terme : gamme de produit $g(p_i)$

Ce type de gamme [28] est établi par produit et intéresse le système de gestion de production à moyen terme. Son rôle est de renseigner sur les différents types de sections de fabrication utilisés, ainsi que sur les temps moyens de passage sur chacun d'eux, ceci hors de tout contexte de production. Elle est donc utilisée pour l'établissement des coûts standards de chaque produit et pour l'établissement du plan directeur. Une gamme de produit est représentée par un ensemble de couples :

$$g(p_i) = (Tp_1, tm_1) \text{ --- } (Tp_n, tm_n)$$

avec Tp_n = Type du poste n

tm_n = temps moyen de passage sur le poste n (temps d'engagements d'atelier = temps d'usinage + temps de préparation

3.2 Court terme : gamme opératoire $g(a_i)$

Ce type de gamme est établi par type d'article. Son rôle est de fournir des renseignements concernant les différentes opérations standards à réaliser sur un type d'article, en précisant les types de moyens utilisés ainsi que les temps (approximatifs) nécessaires à la bonne marche de chaque opération. Une gamme opératoire est représentée par un ensemble de quintuplés :

$$g(a_i) = (A_1, A_2, A_3, A_4, A_5) \text{ ----- } (-----)$$

avec : A_1 = isostatisme de l'article, en fonction des faces principales.

A_2 = type de phase

- A_3 = opération standard à exécuter et face principale sur laquelle doit être réalisée l'opération.
- A_4 = temps moyen d'usinage.
- A_5 = temps moyen de préparation.

3.3 Très court terme : gamme détaillée $g^d(a_i)$

Ce type de gamme est établi également par article. Son rôle est de fournir des informations détaillées, utiles à la réalisation de l'article. Une gamme détaillée est représentée par un ensemble de quintuplets.

$$g^d(a_i) = (A_1(a_1), A_2(a_2), A_3(a_3), A_4(a_4), A_5(a_5))$$

avec : $a_1, \dots, a_4, \dots, a_5$ = l'ensemble des informations nécessaires à la réalisation proprement dite de l'article (Machine, outil, vitesse d'avance, ...).

En général à chacune de ces informations, il correspond une grille d'affichage et de saisies, qui permet l'introduction et l'affichage de données nécessaires à l'établissement de la gamme détaillée.

4. APPROCHE PROPOSEE POUR LA GENERATION DE GAMMES DE FABRICATION

L'approche proposée dans cette étude est le résultat d'une réflexion sur les méthodes précédentes. De cette réflexion, nous avons retiré les idées directrices, suivantes :

- * Garder le concept de la technologie de groupe, qui conduit à une bonne organisation du système de fabrication.
- * Garder l'aspect algorithmique quand il est bien défini et non ambigu.

- * Utiliser des algorithmes stables qui sont des composants importants du savoir faire en fabrication mécanique. Leurs utilisations augmentent l'efficacité du système.
- * Utiliser un système expert, pour sa modularité et sa flexibilité, mais faire attention à :
 - ne pas développer un système complexe qui deviendra rapidement incontrôlable.
 - concevoir une représentation du savoir faire qui inclura de la connaissance modulaire, par des règles de production et de la connaissance structurée par un graphe.
 - concevoir un système général ayant une fonction d'apprentissage qui permettra une évolution aisée de la base de connaissances, afin de diminuer les coûts d'implantation.

Ainsi le système automatique de génération de gammes par apprentissage SAGGA que nous présentons dans ce rapport se décompose en trois niveaux (voir figure I.7) :

- * Niveau implantation.
- * Niveau utilisation.
- * Niveau apprentissage.

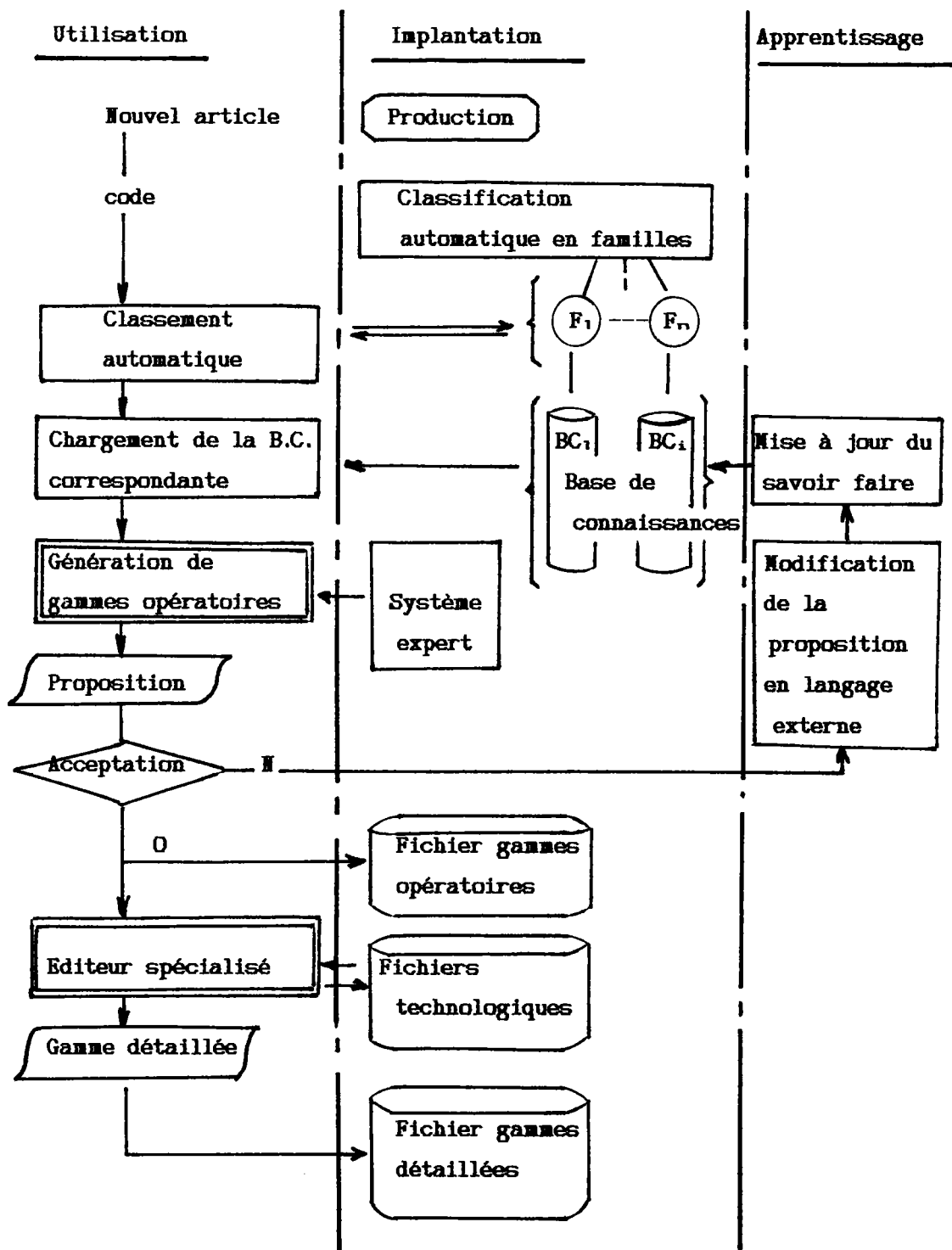


figure I.7 : Présentation générale du système SAGGA.

4.1 Niveau implantation

A partir d'un ensemble test de données de production sur une période déterminée, les familles d'articles sont définies par des techniques de classification automatique.

Les méthodes standards de tris telles que celles utilisées dans le logiciel Multigroup d'O.I.R. (Organization of Industrial Research) ne sont pas appropriées. Dans notre approche il est employé des méthodes de perception par ordinateur [17, 19] qui utilisent la notion de ressemblance entre articles.

Ces méthodes de classification automatique utilisées pour ce niveau ont été étudiées par ailleurs au L.A.E.I. [14, 16, 62, 63].

Lorsque les familles sont reconnues, nous définissons la base de connaissances associée. L'ensemble des connaissances est modélisé par un Grafcet modifié. Les étapes sont les opérations de fabrication et les transitions les conditions de fabrication.

A noter que ce réseau n'est pas l'arbre de recherche, il constitue seulement une représentation graphique de la base de connaissances.

4.2 Niveau utilisation

Soit un nouvel article défini par son dessin de définition. La première étape consiste à coder cet article selon un code utilisé en Technologie de Groupe [22]. Puis une fonction discriminante [86] basée sur une mesure de ressemblance détermine la base de connaissances correspondante. Alors un mécanisme d'inférence recherche des gammes de fabrication réalisables. Le système peut demander au préparateur des informations complémentaires pour préciser sa recherche. Dès lors qu'une gamme de fabrication est retenue, le préparateur fait appel à un éditeur de texte, qui l'assiste pour transformer la gamme opératoire proposée, en gamme détaillée.

4.3 Niveau apprentissage

Le niveau apprentissage a été défini à partir des deux réflexions suivantes :

- * Dans notre approche nous n'utilisons qu'une base de connaissances spécifique à une famille à la fois. Aussi nous n'avons pas à définir des règles génériques de fabrication.
- * Afin de mettre à jour aisément la base de connaissances, nous mettons à la disposition du préparateur un langage externe lui permettant de transmettre aisément son savoir faire. Ainsi, lors d'une proposition de gammes déficiente, le préparateur à partir de celle-ci et à l'aide du langage mis à sa disposition, introduit les différentes modifications conditionnelles, et le tout sera immédiatement traduit en règles de production dans la base de connaissances.

5. PLAN DE LA THESE

Notre étude, sur un système de génération de gamme de fabrication a porté sur :

- La représentation des connaissances de familles de gammes de fabrication.
- La définition et la réalisation d'un système expert associé à ce type de représentation de connaissances qui génère les gammes opératoires.
- L'étude et la réalisation d'un éditeur spécialisé pour écrire les gammes détaillées.
- L'étude et la réalisation du niveau d'apprentissage.
- Les essais de notre logiciel sur une application industrielle.

Dans le chapitre II, nous justifions le choix de la représentation des connaissances (gammes de fabrication d'une famille d'articles), par un Grafcet modifié. Puis nous présentons les propriétés de ce Grafcet modifié.

L'étude du système de génération de gammes de fabrication est présentée dans le chapitre III. Nous définissons en particulier l'implantation des connaissances et le mécanisme d'exploitation de celles-ci. Enfin nous présentons sur un exemple académique, les différentes caractéristiques du système.

Dans le chapitre IV nous présentons une application en fabrication mécanique du système SAGGA.

Les annexes C et D présentent l'aspect informatique, des différents algorithmes et procédures développés dans le système SAGGA et le système d'édition de gammes détaillées. Ces deux logiciels ont été écrits en langage PASCAL sous MS.DOS.

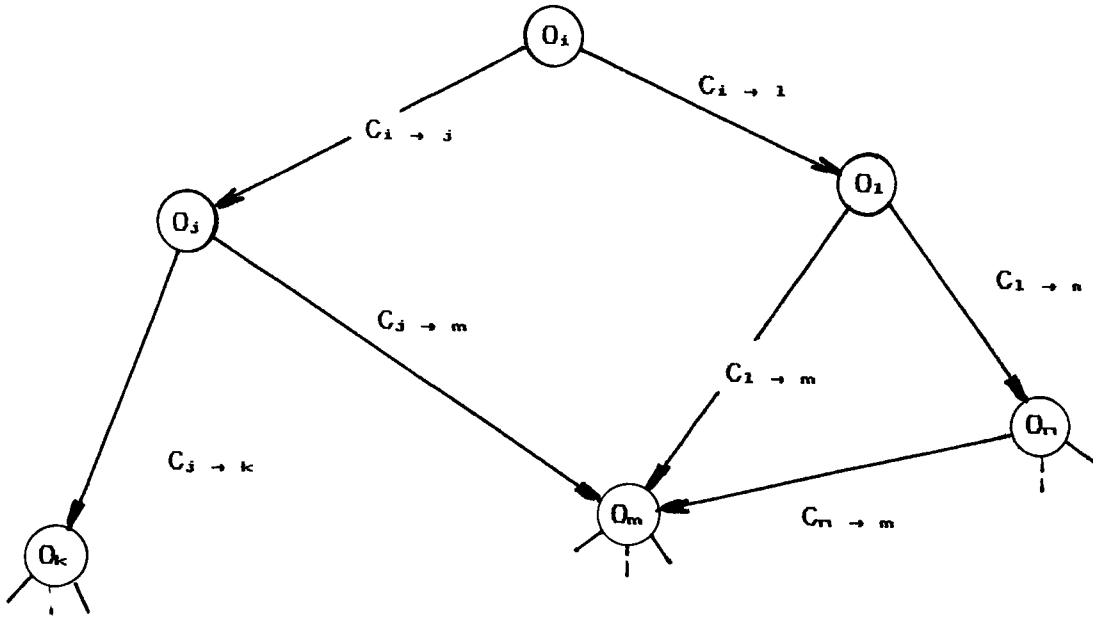
CHAPITRE II

Représentation des gammes de fabrication

1. INTRODUCTION

De part l'approche technologique de groupe choisie, une famille se compose d'un ensemble de gammes opératoires différentes. Chaque gamme est définie comme une liste ordonnée d'opérations dont le choix de chacune d'entre elles est fonction d'un ensemble de conditions de fabrication (poste de travail, outillage, nombre de pièces, ...) et de contraintes géométriques de la pièce.

L'ensemble des gammes de fabrication appartenant à une même famille peut-être représenté par un graphe orienté (figure II.1) où les sommets sont les opérations (notées O_i) de la gamme, et dont à chaque arc est associé des conditions de réalisation de ces opérations.



$O = \{o_1, \dots, o_n\}$ ensemble des opérations
 $C = \{c_1 \rightarrow x, \dots, c_y \rightarrow z\}$ ensemble des conditions

figure II.1 : Graphe de représentation de l'ensemble des gammes de fabrication d'une famille.

Chaque opération O_i de la liste est considérée comme unique. Si plusieurs opérations identiques apparaissent plusieurs fois dans la liste, chacune d'entre elles sera différenciée à partir de son rang dans cette liste. La sélection d'une nouvelle opération de la liste, correspond à la progression d'un noeud dans le graphe figure II.1.

La recherche d'un chemin dans le graphe peut être interprétée comme la suite d'opérations à réaliser pour transformer un produit brut en produit fini. Cette interprétation permet d'assimiler le processus de recherche de la gamme à un système à évolution discrète.

2. CHOIX D'UNE REPRESENTATION

Il existe trois types de modèles de représentation de systèmes à évolution discrète :

- Automate à états finis
- Réseaux de Pétri
- Grafcet

Nous présentons succinctement ces trois représentations à partir de l'exemple de la figure II.2.

O_0 : état initial

$O_1 \dots O_4$: opérations

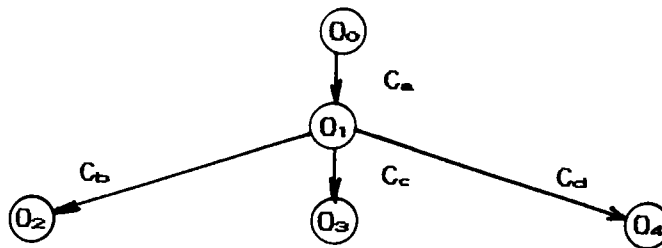


figure II.2 : Exemple de gamme de fabrication avec possibilité d'opérations en temps masqué (les événements C_b , C_c , C_d étant susceptibles d'arriver de façon totalement indépendante ou en parallèle).

2.1 Automate à états finis

La représentation par automate à états finis [30] utilise lors de la description du processus, la notion d'état.

Le processus est décrit par un graphe orienté (figure II.3) dont chaque sommet correspond à un état du processus (ou système) et chaque arc correspond à un ou plusieurs événements susceptibles d'arriver de façon totalement indépendante ou simultanée. De ce mode de représentation se dégagent quatre règles principales :

- * Règle 1 : La valeur vraie de l'évènement associé à un arc modifie l'état dans lequel se trouve le système.
- * Règle 2 : La progression à travers le graphe se fait avec un seul changement d'état à la fois.
- * Règle 3 : Si le système est à l'état E_i , le changement d'état n'intervient que si le ou les évènements liés à l'arc $E_i \rightarrow E_j \forall j$ sont vrais. Le système reste insensible à tout autre évènement.
- * Règle 4 : Il est convenu de représenter explicitement tous les liens ainsi que les maintiens, d'où l'existence de boucle élémentaire.

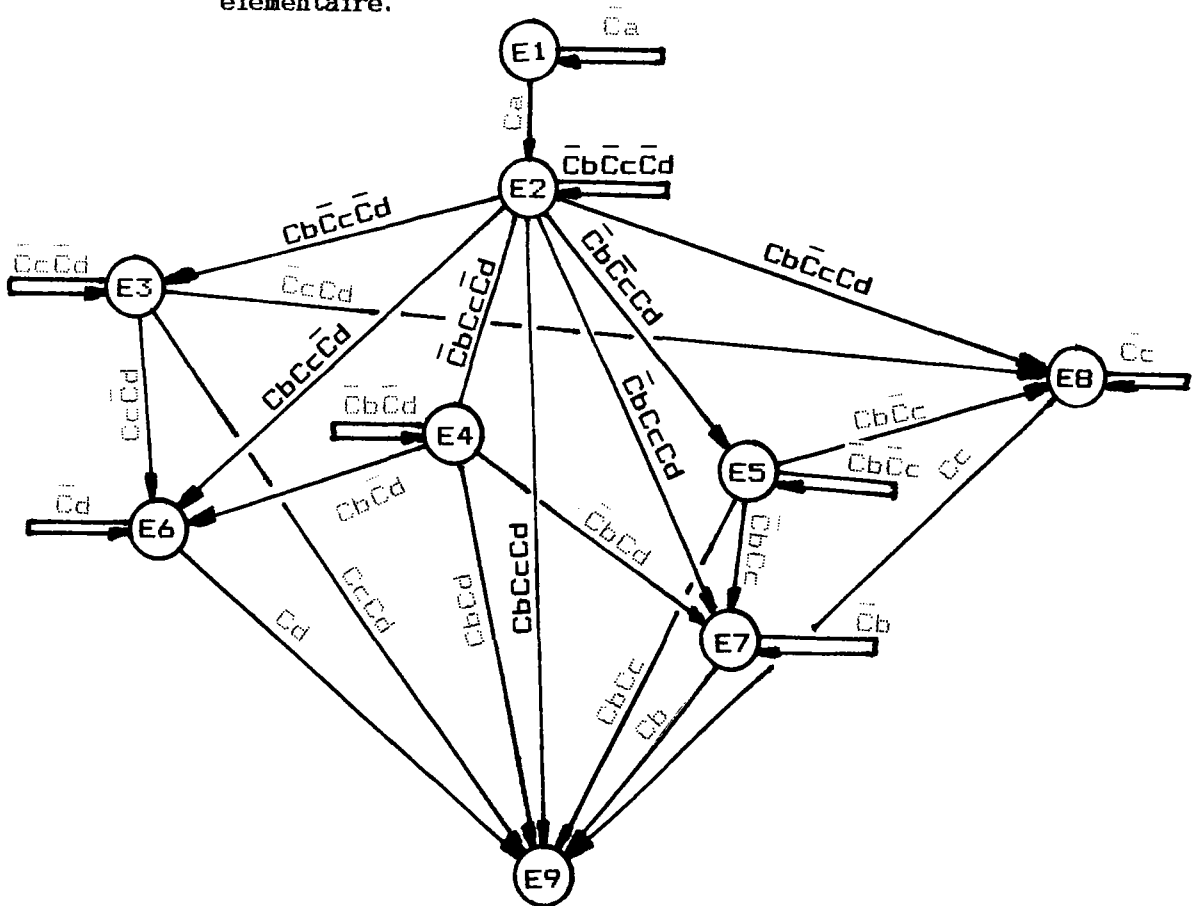


figure II.3 : Représentation par automate à états finis de la gamme figure II.2.

Une telle représentation engendre un accroissement exponentiel du nombre d'états, ce qui est un inconvénient majeur pour une représentation claire et précise d'un système. De plus l'ensemble des états représentatifs des combinaisons possibles sur les événements, n'est pas représentatif de l'ensemble des possibilités de gamme (sauf cas exceptionnel) car celles-ci sont très contraintes.

2.2 Réseaux de Pétri

La représentation par les réseaux de Pétri [35-36-37] utilise la notion d'étape associée à des jetons. Ce type de description emploie un graphe orienté avec deux types de sommet. :

- * Les sommets "étapes" qui représentent des états partiels du système (un état partiel est une situation où le système est réceptif à un certain nombre d'événements du monde extérieur).
- * Les sommets "transitions" représentent les conditions locales afin que le système soit réceptif à certains événements et permettent le passage d'un état à l'autre.

L'utilisation de la notion d'étape associée aux états partiels, implique que plusieurs sommets ou "étapes" peuvent être activés à la fois. Les jetons sont utilisés pour indiquer un état actif, la position de chacun d'entre eux dans le réseau, à l'instant t_i , indique l'état du système à cet instant. L'ensemble des positions évolue en même temps que le système.

L'évolution à travers un réseau de Pétri est soumise à quatre règles fondamentales :

- * Règle 1 : Une transition est dite sensibilisée ou validée, si toutes ses étapes précédentes possèdent au moins un jeton.

- * Règle 2 : Une transition est tirée ou franchie, lorsqu'elle est sensibilisée ou validée et que la condition logique qui peut lui être associée reste vraie. (La condition ne doit pas dépendre explicitement de la présence ou de l'absence d'un jeton dans une étape).
- * Règle 3 : Le tir d'une transition consiste à enlever un jeton à chacune des étapes précédant la transition et à en ajouter un pour celles suivant la transition.
- * Règle 4 : Le tir d'une transition est supposé instantané, donc deux transitions ne peuvent jamais être tirées ou franchies simultanément.

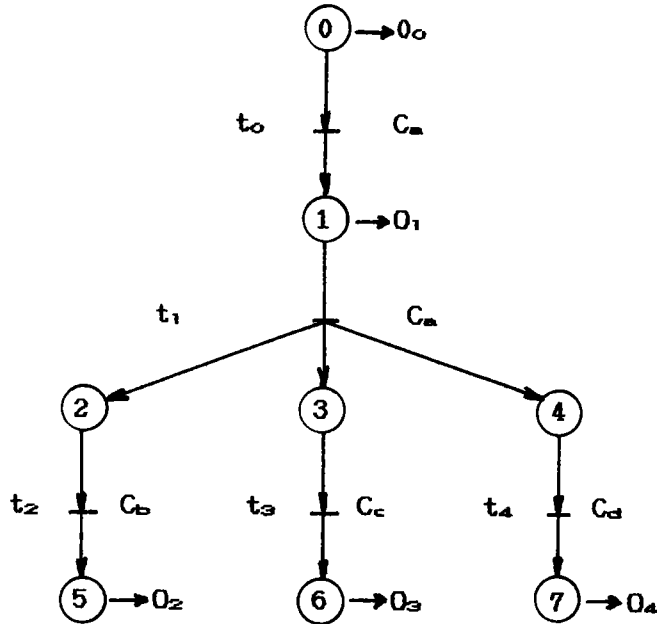


figure II.4A : Représentation de la gamme (fig 2.a) par un réseau de Pétri.

Lors de l'adaptation des réseaux de Pétri à la représentation de systèmes logiques, on ne considère que les réseaux dits "conformes" c'est à dire saufs, vivants et sans conflit (voir annexe B).

La vérification de ces propriétés entraîne une analyse rigoureuse du réseau et permet de détecter d'éventuelles anomalies du système logique [30-31-32-33-38].

La représentation d'une gamme par un réseau de Pétri, consiste à étiqueter celui-ci, c'est à dire associer à chaque sommet une signification (figure II.4A).

Il existe deux conventions de représentation :

* La première consiste à associer à chaque transition une opération. Les transitions sont ainsi munies d'étiquettes constituées chacune de conditions représentatives des événements et des opérations s'y rapportant. Le tir de transition équivaut à l'exécution d'opérations, puis à la mise d'un jeton dans toutes les étapes suivantes. (Machine de Mealy).

* La deuxième convention consiste à associer une opération à chaque étape. Le tir d'une transition consiste alors à marquer d'un jeton toutes les places suivantes, puis à exécuter les actions affectées à ces étapes (Machine de Moore).

Dans notre représentation (fig II.4A), nous avons utilisé la deuxième convention.

2.3 Grafcet

La représentation par GRAFCET [40-41] (Graphe de Commande Etape Transition) utilise la notion d'étape à laquelle on attribue un état actif ou inactif.

C'est un modèle moins général que les réseaux de Pétri. Dans cette représentation la notion d'étape représentant des états partiels a été conservée, d'où une représentation graphique très proche de celle des réseaux de Pétri. Par contre l'utilisation de la notion d'états actifs ou inactifs modifie les règles d'évolution.

- * Règle 1 : La phase d'initialisation précise les étapes activées au début du fonctionnement.
- * Règle 2 : Une transition est soit validée, soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives. Elle ne peut être franchie que lorsqu'elle est validée et que la réceptivité associée à la transition est vraie, la transition est alors obligatoirement franchie.
- * Règle 3 : Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.
- * Règle 4 : Plusieurs transitions simultanément franchissables sont simultanément franchies.
- * Règle 5 : Si au cours du fonctionnement, une même étape doit être désactivée et activée simultanément, elle reste active.

La représentation Grafcet se distingue sur deux points :

- La non utilisation de la notion de jeton permet de créer des liens implicites, notamment de décomposer un Grafcet en plusieurs graphes, tout en assurant de façon rigoureuse leurs interconnexions. Ces liens ne seront reconnus qu'au moment de l'interprétation du graphe, d'où une clarté accrue de la représentation. L'existence de ces liens implicites entraîne une analyse plus complexe du réseau.

- La règle 4 permet d'éviter toute évolution non déterministe, mais autorise la réalisation de parallélisme interprété qui entraîne une confusion dans la compréhension du graphe.

Cependant il ressort une plus grande clarté dans la représentation d'un processus. Cet avantage est mis en évidence par la figure II.4B.

Le tableau de la figure II.5 met en évidence la multiplicité des noeuds en fonction de la méthode utilisée. La méthode par automate à états finis nécessite neuf noeuds pour la représentation du processus, la méthode par réseaux de Pétri sept noeuds, tandis que la méthode Grafcet nécessite seulement quatre noeuds pour cet exemple.

Dans la détermination du choix d'une représentation nous considérons que les gammes de fabrication sont souvent linéaires, ne comportent que très peu de bouclage et n'utilisent pas la notion de synchronisme.

La méthode par automate à états finis est à rejeter, car elle représente l'ensemble des états finis, qui lors d'une gamme complexe peut aboutir à une explosion combinatoire. Par contre sur le plan conceptuel rien ne s'oppose pour la représentation de gamme, à l'utilisation des deux méthodes restantes [43-44]. En pratique, en ce qui concerne l'aspect description, l'utilisation du Grafcet est la plus intéressante.

Nous avons donc choisi de représenter les gammes de fabrication par un modèle Grafcet dit modifié, c'est à dire adapté à la représentation des connaissances sur les gammes de fabrication.

Les principales caractéristiques du Grafcet M seront :

- * Etapes non réactivables.
- * Pas de réceptivité interne.
- * Pas de parallélisme interprété (tous les liens entre étapes seront représentés explicitement).
- * Liaisons orientées mais réversibles.

3. DEFINITION DU GRAFCET M

Un système de génération de gammes de fabrication peut être intégré dans un schéma de principe permettant la liaison : Bureau des méthodes-Atelier.

Aussi il apparaît (figure II.6) :

- La partie commande représentée par le bureau des méthodes.
- La partie opérative représentée par l'atelier.

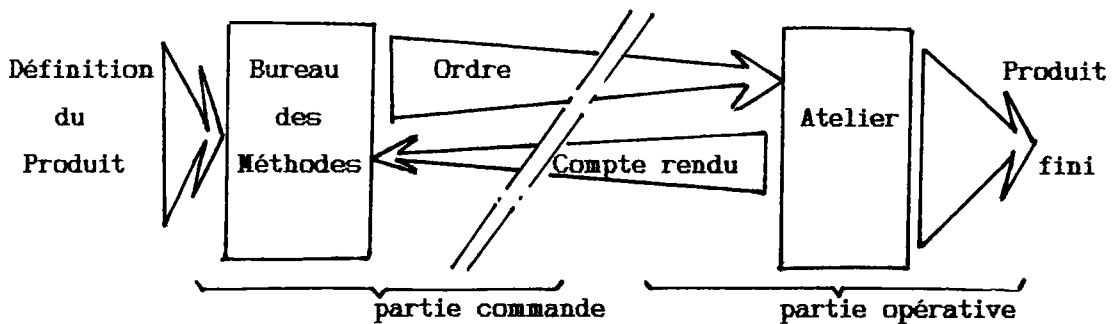


figure II.6 : Liaison Bureau des méthodes - Ateliers

Dans le cas d'un système intégré de fabrication et de suivi en temps réel des opérations de fabrication, à chaque étape du Grafcet correspond réellement un ordre d'exécution d'opérations. Ces opérations exécutées, la transition suivante peut alors être franchie.

Dans le cas d'une préparation du travail "off line", cas le plus courant, nous considérons la partie opérative comme virtuelle.

3.1 Description générale du Grafcet M

Par définition le Grafcet M est un graphe orienté comprenant :

- * Des étapes auxquelles sont associées un ensemble d'actions virtuelles.
- * Des transitions auxquelles sont associées des réceptivités.
- * Des arcs orientés type "Etape-Transition".
- * Des arcs orientés type "Transition-Etape".

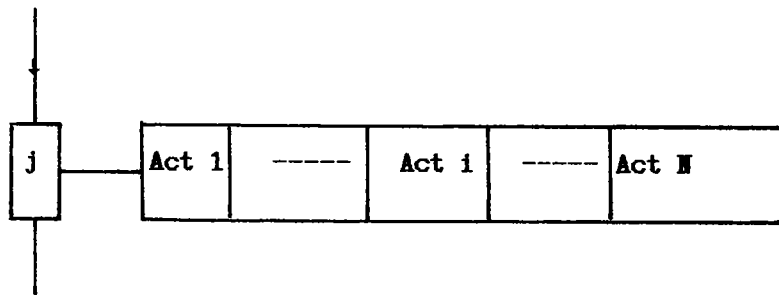
3.1.1 Actions virtuelles

A chaque étape nous associons un ensemble M d'actions virtuelles. Une action peut être vide.

3.1.2 Etape

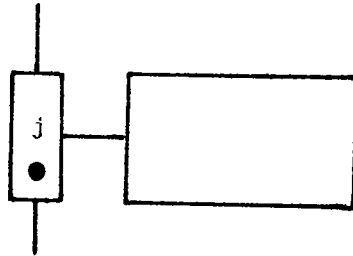
A chaque étape nous associons un et uniquement un ensemble d'actions.

Représentation graphique :

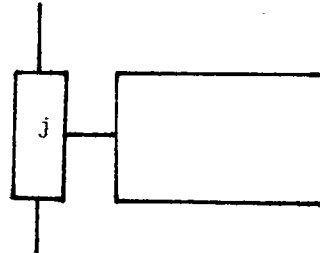


A chaque étape correspond deux états :

- * L'état actif d'une étape indique que l'ensemble des actions associées est mis en exécution.
- * L'état inactif d'une étape indique que l'ensemble des actions associées n'est pas concerné.



état actif (noté par un point)



état inactif

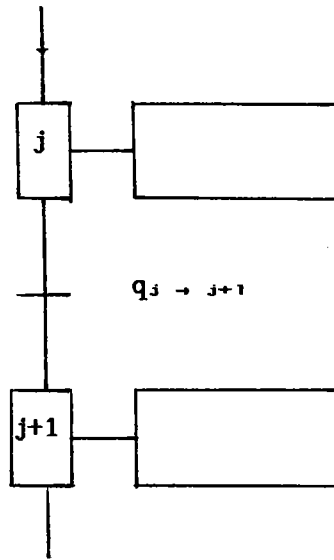
Dès l'instant qu'une étape a été mise dans un état actif, elle conserve cet état tant que le système n'a pas entrepris d'activer une étape immédiatement suivante.

A un instant donné, l'état du système est entièrement défini par l'ensemble des étapes ayant des états actifs.

Une étape dont l'état est actif, correspond à une situation dans laquelle le comportement local du système de commande associé (dans notre cas celui-ci est virtuel) reste invariant.

3.1.3 Transition

Une transition indique la possibilité d'évolution entre deux étapes. A chaque transition est associée une réceptivité.



$q_j \rightarrow j+1$ = transition indiquant la possibilité d'évolution de l'étape e_j à l'étape e_{j+1}

3.1.4 Réceptivité

Une réceptivité est représentée sous la forme d'une fonction logique. Cette fonction logique peut être composée de variables portant sur l'état du système présent, passé ou futur et l'état d'informations externes au système (plan de pièces, disponibilité machines, ...).

3.1.5 Arcs orientés

Nous définissons deux types d'arcs orientés, reliant des noeuds différents :

- Etape → Transition
- Transition → Etape

L'orientation des arcs définit le sens de l'évolution du processus. En ce qui concerne le Grafcet M, le fait que les différents arcs soient orientés n'exclut pas la réversibilité pendant le parcours de l'arbre. Cela est possible car l'on ne considère que des actions virtuelles.

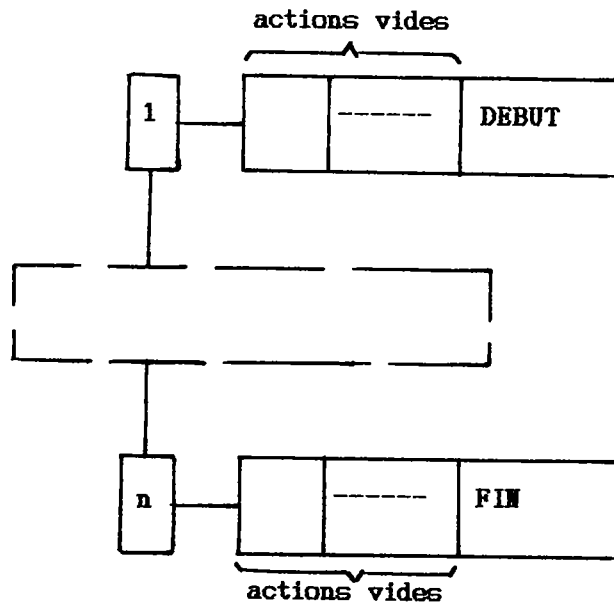
3.2 Règles de structuration du Grafcet M

Dans la représentation d'une gamme de fabrication, nous incluons les opérations en temps masqué ainsi que la possibilité de choix entre diverses séquences d'opérations. La structure graphique d'un ensemble de choix de gammes est composée d'une structure générale et d'un ensemble de structures de base définis ci-après.

3.2.1 Structure générale

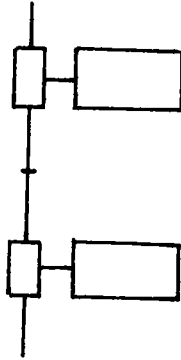
Le début du graphe est indiqué en utilisant la partie action, avec "DEBUT" sur la dernière action de la première étape.

La fin du graphe est indiquée avec "FIN" sur la dernière action de la dernière étape.



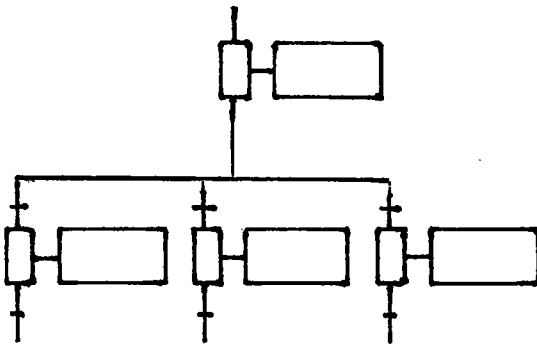
3.2.2 Structure de base : séquence chaînée

Une séquence chaînée est composée d'une suite d'étapes pouvant être activées les unes après les autres. Chaque étape n'est suivie que par une seule transition et chaque transition n'est validée que par une seule étape.

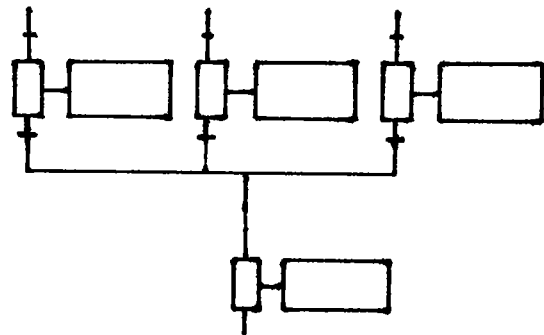


3.2.3 Structure de base : sélection de séquences

Une sélection entre plusieurs séquences se représente à partir d'une étape, suivie par autant de transitions qu'il y a d'évolutions possibles. C'est une divergence de "ou". La fin d'une sélection de séquences est indiquée par une convergence de "ou".



Divergence "ou"



Convergence "ou"

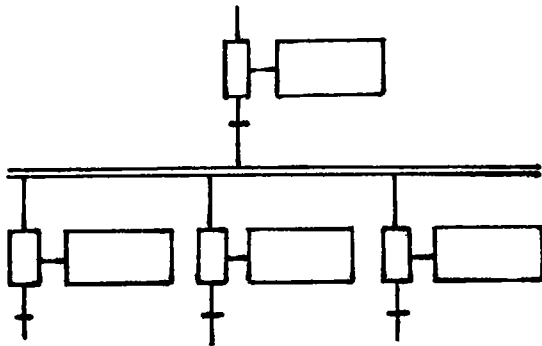
Les réceptivités de chaque transition composant une sélection de séquences, ne doivent en aucun cas être identiques, mais peuvent être incluses les unes aux autres.

3.2.4 Structure de base : séquences simultanées

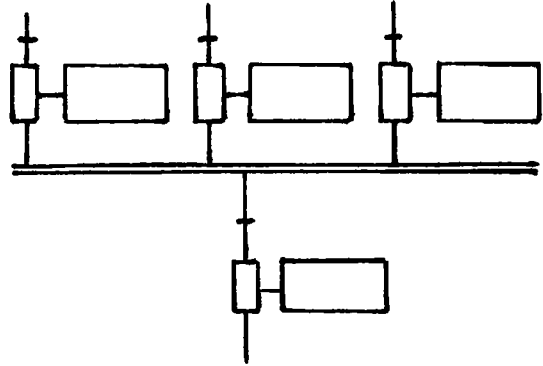
Lorsque le franchissement d'une transition conduit à activer plusieurs étapes, les séquences issues de ces étapes sont dites simultanées. Chaque évolution ultérieure de l'ensemble des séquences est alors totalement indépendante.

Le début d'un ensemble de séquences simultanées est représenté par une divergence de "ET".

La fin d'un ensemble de séquences simultanées est représentée par une convergence de "ET".



Divergence "ET"



Convergence "ET"

3.3 Règles d'évolution du Grafcet M

L'évolution du Grafcet M est soumise aux règles suivantes :

3.3.1 Règle 1 :

L'étape associée à l'action DEBUT est automatiquement activée lors de l'initialisation.

3.3.2 Règle 2 :

Une transition est soit validée, soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives. Elle ne peut être franchie que lorsqu'elle est validée et que la partie réceptivité, associée à la transition, est vraie. Alors la transition est obligatoirement franchie.

3.3.3 Règle 3 :

Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes, et met à l'état inactif toutes les étapes immédiatement précédentes.

3.3.4 Règle 4 :

Lors de l'apparition d'une divergence de "ou" si plusieurs réceptivités sont simultanément vraies, alors la sélection de la transition dépend du poids affecté aux réceptivités. Si les poids sont identiques alors on choisit la première réceptivité rencontrée.

3.3.5 Règle 5 :

L'activation de l'étape associée à l'action FIN entraîne automatiquement l'affichage du résultat, c'est à dire la suite des étapes franchies.

3.3.6 Règle 6 :

Lors du franchissement d'une transition de divergence "ET", toutes les étapes immédiatement suivantes, correspondant au début de chaque branche, sont activées. Mais l'évolution à l'intérieur du bloc "ET" est effectuée branche par branche.

4. DESCRIPTION FORMELLE DU GRAFCET M

Le Grafcet M est défini sur la base d'un ensemble de notations que nous introduisons ci-après. Les notations nous permettent de formaliser les règles de structure et d'évolution.

4.1 Notations

- Un Grafcet M est défini comme un quadruplet que l'on note :

$$G_r = (E, Q, L^+, L^-)$$

avec :

- E = ensemble fini d'étapes avec $e_i \in E$
- Q = ensemble fini de transitions avec $q_i \in Q$
- L^+ = ensemble fini d'arcs Etape $\rightarrow$ Transition avec $l^+_i \in L^+$
- L^- = ensemble fini d'arcs Transition $\rightarrow$ Etape avec $l^-_i \in L^-$

- L'état de l'étape e_j à l'instant t_i est noté $x^{t_i}_{(e_j)}$:

$$\forall i, x^{t_i}_{(e_j)} \in \{0, 1\}$$

0 équivaut à un état inactif

1 équivaut à un état actif

- L'état du système à l'instant t_i est représenté par une matrice colonne que l'on note :

$$M_i = \begin{pmatrix} x^{t_i}_{(e_1)} \\ \vdots \\ x^{t_i}_{(e_j)} \\ \vdots \\ x^{t_i}_{(e_{\text{card}(E)})} \end{pmatrix}$$

- L'ensemble des étapes immédiatement précédentes, d'une transition q_i est noté :

$$E_i^- = \{ e_j \in E / \overline{(e_j, q_i)} \in L^+ \}$$

- L'ensemble des étapes immédiatement suivantes d'une transition q_i est noté :

$$E_i^+ = \{ e_j \in E / \widehat{(q_i, e_j)} \in L^- \}$$

- L'ensemble des transitions immédiatement précédentes d'une étape e_i est noté :

$$Q_i^- = \{ q_j \in Q / \widehat{(q_j, e_i)} \in L^- \}$$

- L'ensemble des transitions immédiatement suivantes d'une étape e_i est noté :

$$Q_i^+ = \{ q_j \in Q / \widehat{(e_i, q_j)} \in L^+ \}$$

- Soit CF_i les conditions de franchissement de la transition q_i
Soit S une séquence finie de transitions

$$S = q_i, \dots, q_{i+k}$$

On dit que S est une séquence de transitions franchissable à partir de l'état M_i , s'il existe des états du système

$$M_i, M_{i+1}, \dots, M_{i+k+1}$$

$$\begin{array}{ccc} & CF_i & \\ \text{tels que } M_i & \longrightarrow & M_{i+1} \end{array} \quad \begin{array}{ccc} & CF_{i+k} & \\ M_{i+k} & \longrightarrow & M_{i+k+1} \end{array}$$

$$\begin{array}{ccc} & S & \\ \text{Soit } M_i & \longrightarrow & M_{i+k+1} \end{array}$$

4.2 Formalisation des règles d'évolution

Les règles d'évolution régissent le cheminement à travers le Grafcet M . L'évolution dans le Grafcet M se fait par franchissement successif des transitions, après avoir satisfait les conditions de franchissement de chacune d'entre elles. Ces conditions sont regroupées dans une fonction logique [46] notée :

$$CF_i = \prod x^{t_j}_{e_k} \cdot R_i$$

CF_i = condition de franchissement de la transition q_i

$\prod x^{t_j}_{e_k}$ = produit logique de l'ensemble des états des étapes $e \in E_i$

R_i = Réceptivité associée à la transition q_i :

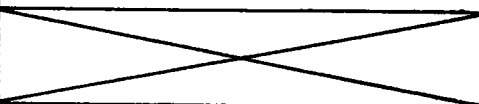
4.2.1 Règle 1 :

L'état initial du système est noté :

$$Mo = \begin{vmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{vmatrix} \quad x^0(e_1) = 1 \text{ à l'initialisation}$$

4.2.2 Règle 2 :

Le franchissement des transitions est régi par la table de vérité ci-dessous :

$\prod x_{e_k}$	R_i	CF_i	Etat de la transition
0	0	0	
0	1	0	Réceptrice
1	0	0	Validée
1	1	1	Franchissable

Ce tableau résume l'état d'une transition q_i en fonction du produit logique $\prod x_{e_k}$ et de la valeur de la réceptivité R_i associée.

4.2.3 Règle 3 :

Le franchissement d'une transition est considéré comme une opération qui, à un état M_i du système fait correspondre un état M_{i+1} ,

$$CF_i = 1$$

$$M_i \longrightarrow M_{i+1}$$

$$\text{avec : } \forall e_j \in E \quad \left. \begin{array}{l} \text{Si } e_j \in E_i^- \\ CF_i = 1 \end{array} \right| \quad \text{alors } x_{\langle e_j \rangle} = 0$$

$$\forall e_j \in E \quad \left. \begin{array}{l} \text{Si } e_j \in E_i^+ \\ CF_i = 1 \end{array} \right| \quad \text{alors } x_{\langle e_j \rangle} = 1$$

4.2.4 Règle 4 :

Soit p_k le poids associé à la transition q_k

Soit $x_{\langle e_i \rangle} = 1$

$$\left. \begin{array}{l} \text{Si } q_k \in Q_i^+ \\ q_1 \in Q_i^+ \\ p_k > p_1 \end{array} \right|$$

alors q_k est sélectionnée en priorité

$$\left. \begin{array}{l} \text{Si } q_k \in Q_i^+ \\ q_1 \in Q_i^+ \\ p_k = p_1 \\ \text{et l'ordre d'apparition de } q_k \text{ antérieur à l'ordre} \\ \text{d'apparition de } q_1 \end{array} \right|$$

alors q_k est sélectionnée en priorité.

4.2.5 Règle 5 :

L'état final noté :

$$M_F = \begin{array}{|c} 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{array} \quad \begin{array}{l} x_{\langle e_j \rangle} = 1 \text{ entraîne l'affichage de} \\ \text{la proposition de gamme} \\ \text{avec } e_j = \text{dernière étape activée} \end{array}$$

4.2.6 Règle 6 :

Soit S_t un ensemble de séquences simultanées

Soit e_i l'étape immédiatement précédente de S

Soit e_j l'étape immédiatement suivante de S

Soit q_a la transition immédiatement suivante de e_i

Soit q_b la transition immédiatement précédente de e_j

Soit $n = \text{Card } |S_t|$

Avec : $e_{i1}, \dots, e_{in} \in E_a^+$

$e_{j1}, \dots, e_{jn} \in E_b^-$

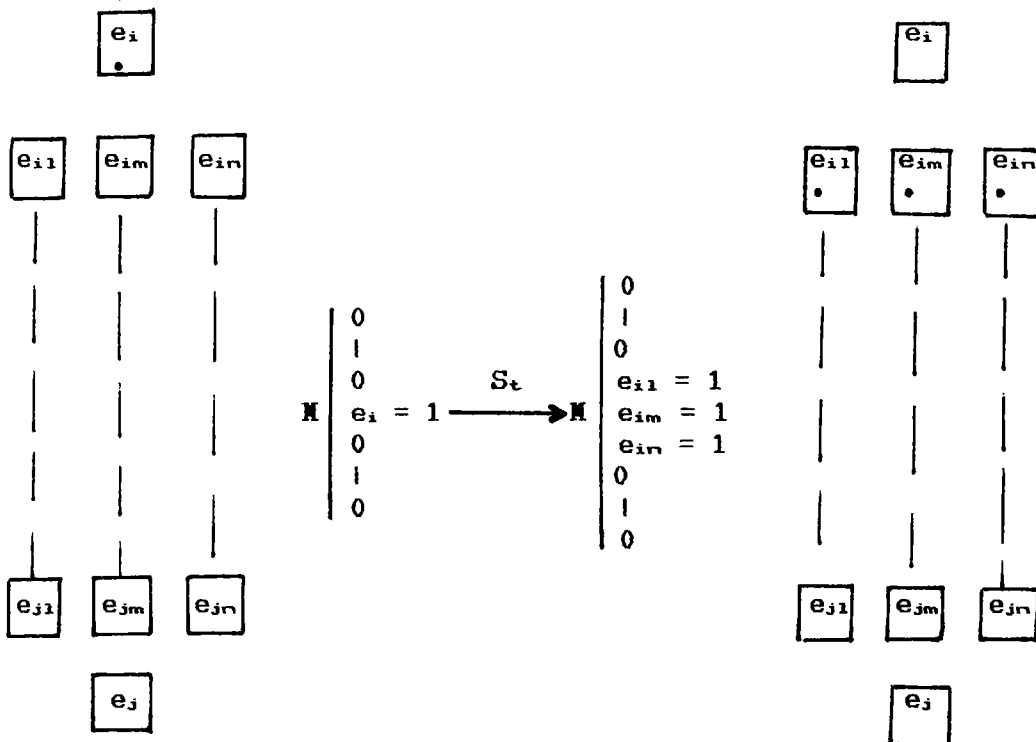
$S_1, S_m, S_n \in S$

Alors S_t est telle que

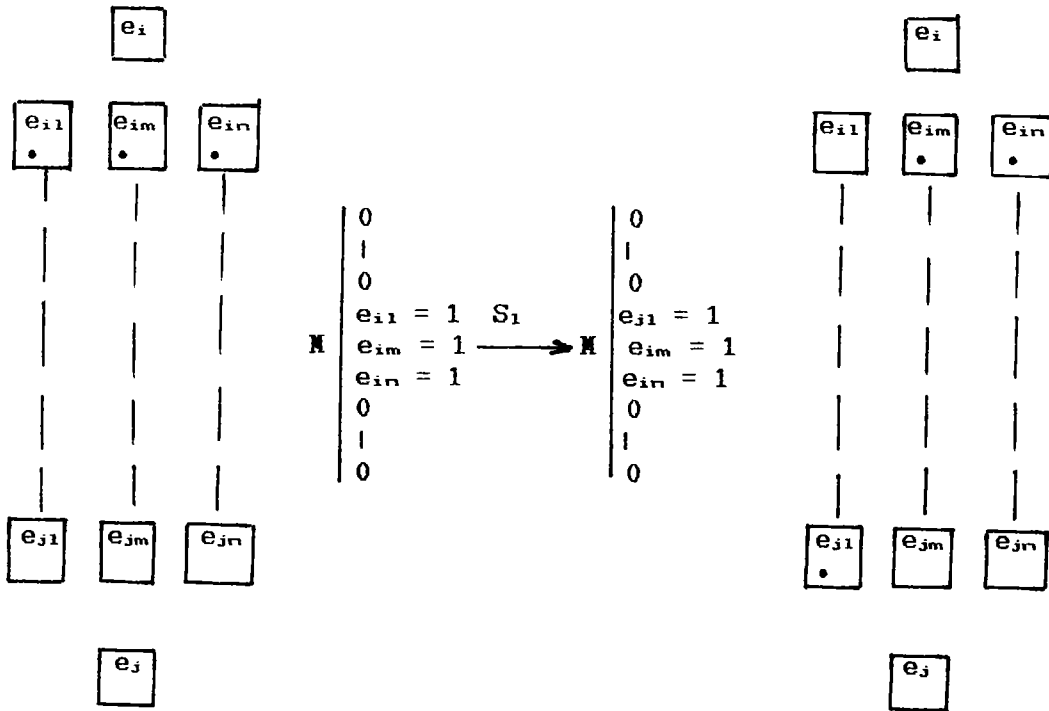
$$M \begin{array}{c} 0 \\ | \\ e_i = 1 \\ 0 \\ | \\ 0 \end{array} \xrightarrow{S_t} M \begin{array}{c} 0 \\ | \\ e_j = 1 \\ 0 \\ | \\ 0 \end{array}$$

Décomposition de S_t :

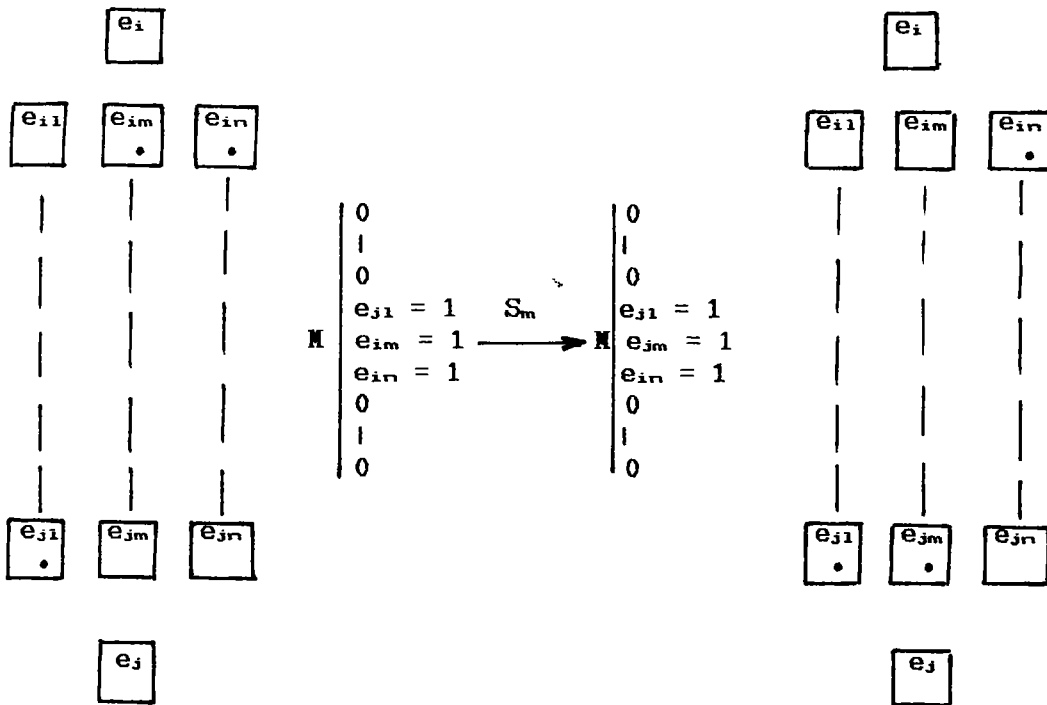
Déclenchement de la séquence S_t :



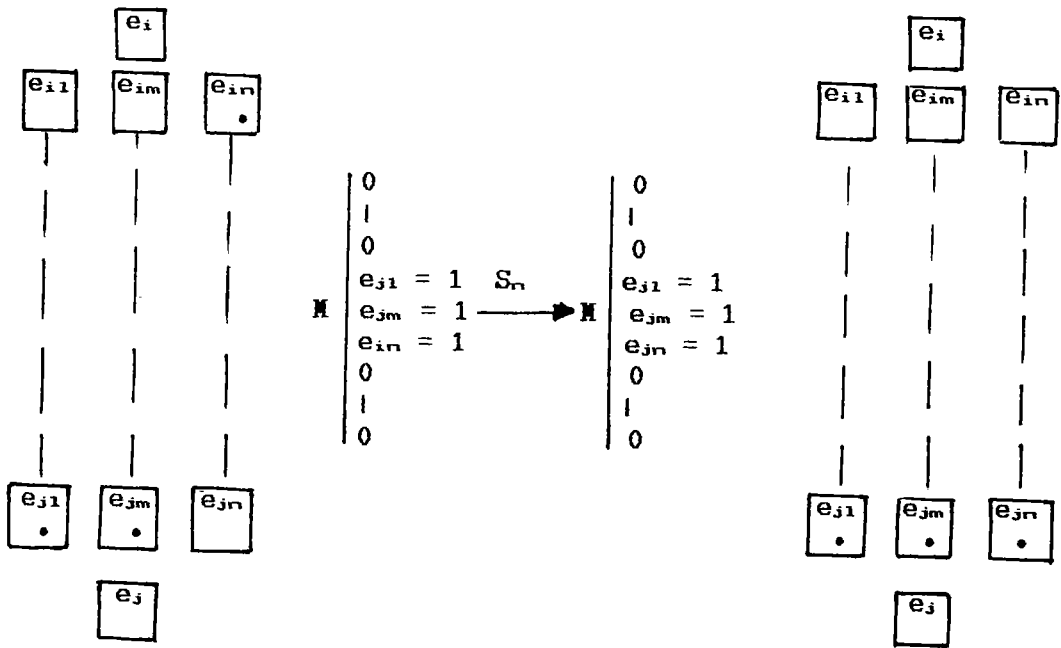
Déclenchement de la séquence S_1 :



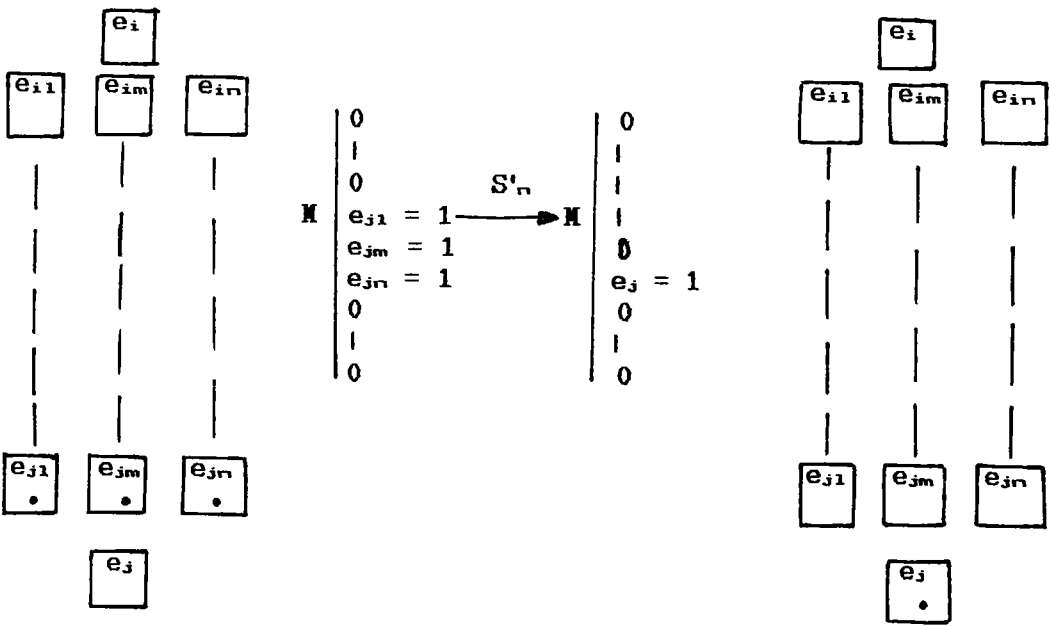
Déclenchement de la séquence S_m :



Déclenchement de la séquence S_n :



Déclenchement de la séquence S'_n :



4.3 Formalisation des règles de structure

Les règles de structure régissent la construction du grafcet M.
L'ensemble des règles de structure est soumis à diverses contraintes
spécifiques à chaque transition. (voir Chapitre II-3.2)

4.3.1 Séquence chaînée

Une transition q_i relie deux étapes chaînées si :

$$\text{Card } (E_i^+) = 1$$

$$\text{Card } (E_i^-) = 1$$

4.3.2 Séquences sélectives

Un ensemble de transitions Q forme un début de séquences
sélectives si :

avec $Q = q_j, \dots, q_v$

$$E_j^- = \text{-----} = E_v^- = e_i$$

$$CF_j = x_{(e_i)} \cdot R_j$$

$$E_j^+ \neq \text{-----} \neq E_v^+$$

$$R_j \neq \text{-----} \neq R_v$$

$$CF_v = x_{(e_i)} \cdot R_v$$

Un ensemble de transitions Q forme une fin de séquences
sélectives si :

$$E_j^- \neq \text{-----} \neq E_v^-$$

$$CF_j = x_{(e_j)} \cdot R_j$$

$$E_j^+ = \text{-----} = E_v^+ = e_k$$

$$CF_v = x_{(e_v)} \cdot R_v$$

4.3.3 Séquences simultanées

Une transition q_i forme un début de séquences simultanées si :

$$\text{Card } (E_i^-) = 1$$

$$\text{Card } (E_i^+) > 1$$

Une transition q_k forme une fin de séquence simultanée si :

$$\text{Card } (E_k^-) > 1$$

$$\text{Card } (E_k^+) = 1$$

4.4 Représentation d'une gamme de fabrication

Une gamme de fabrication est obtenue par la somme logique des états M_i appartenant à la séquence de franchissement S telle que :

$$M_0 - S \longrightarrow MF$$

$$\text{ainsi : } g = \sum_{i=0}^F M_i$$

CHAPITRE III

Etude du système de génération de gammes

1. INTRODUCTION

Dans ce chapitre nous présentons les caractéristiques générales du système de génération de gammes SAGGA. La figure III.1 montre la structure du système étudié.

Le paragraphe 2 concerne la formulation de la connaissance. Il nous permet de définir la syntaxe des règles de production en langage interne utilisées pour la représentation interne des connaissances. Les différents éléments manipulés par les règles de production, utilisées tant en langage interne qu'en langage externe sont définis dans le paragraphe 3 : implantation de la connaissance. Le mécanisme d'exploitation des connaissances est présenté dans le paragraphe 4. Après avoir justifié l'utilisation d'un mécanisme à chaînage avant et chaînage arrière, nous définissons les caractéristiques des deux moteurs d'inférence correspondants.

Par ailleurs le contrôle de la cohérence de la base de connaissance, effectué après compilation de la base de règles sélectionnée est examiné dans l'annexe : Structure informatique.

Enfin le paragraphe 5 présente les possibilités du système SAGGA à partir d'un exemple académique.

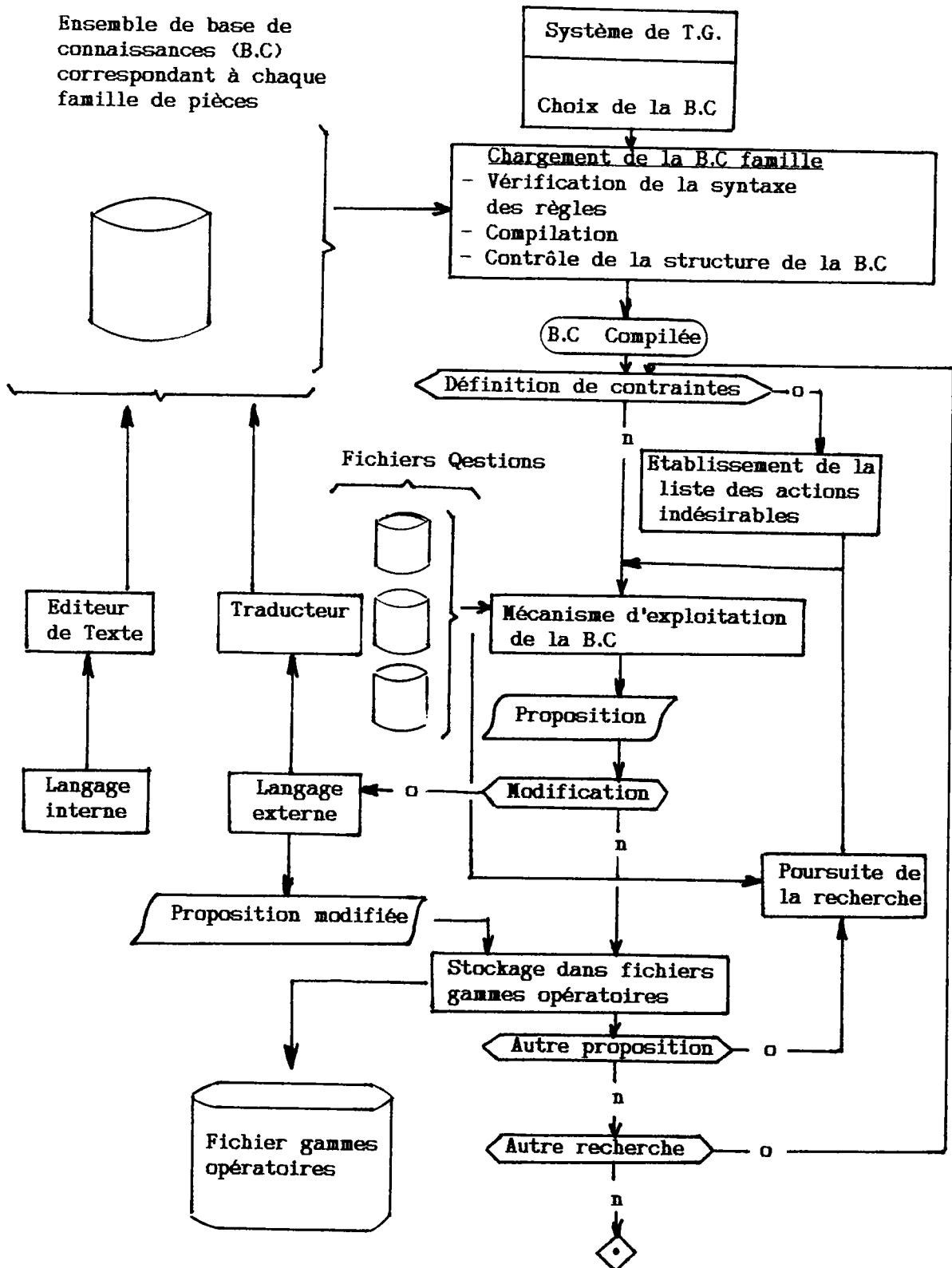


Figure III.1 : Description générale du fonctionnement.

2. FORMALISATION DE LA CONNAISSANCE

Le choix d'une formalisation de la connaissance dépend essentiellement du type de connaissances [70, 71, 72, 73, 77, 78] à manipuler et des objectifs fixés. Le modèle Grafcet M défini au chapitre précédent, utilise une représentation en arbre, dans laquelle apparaît la notion de **CONDITION** (attachée aux transitions) et la notion d'**ACTION** (attachée aux étapes). Ces notions nous permettent d'envisager la connaissance sous une forme déductive.

2.1 Connaissance déductive

La connaissance déductive est formée par un ensemble d'opérateurs de changement d'états [67, 68]. Ceux-ci sont formalisés en terme de règles dites "Règles de production" de la forme :

$$MG \longrightarrow MD$$

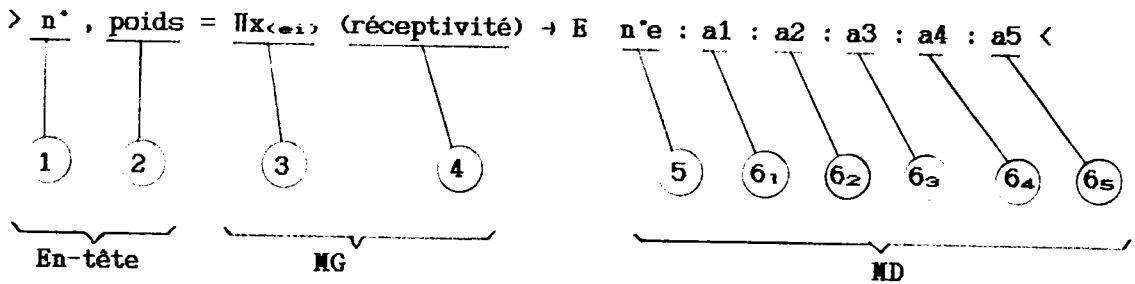
Le membre de gauche (MG) est représentatif d'une certaine situation. Lorsque celle-ci est décelée la règle est déclenchée ce qui provoque la mise en action du membre de droite (MD), qui appliqué à la situation présente, engendre un nouvel état. Les règles de production appliquées au modèle Grafcet M , permettent une évolution de celui-ci d'un état M_i à un état M_{i+1} , par application d'un opérateur noté :

$$M_i \xrightarrow[\text{opérateur}]{\text{transition}} M_{i+1}$$

La progression à travers le Grafcet M est effectuée par l'application successive d'un ensemble adéquate d'opérateurs de changement d'états.

2.2 Structure générale des règles de production

La syntaxe d'écriture des règles de production a été choisie dans le but de faciliter l'écriture et la compréhension de celles-ci. La structure se compose de six éléments séparés par des caractères spéciaux :



* Composition de l'EN-TETE

• Elément 1 :

L'élément 1 représente la référence de la règle dans la base de connaissances. C'est un entier.

• Elément 2 :

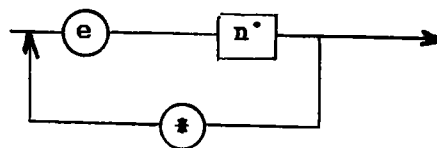
L'élément 2 représente le poids de la règle. Celui-ci est utilisé lorsqu'il y a plusieurs règles sélectionnées qui peuvent être déclenchées, ainsi il faut effectuer un choix. La règle ayant le plus grand poids sera retenue en priorité. C'est un entier.

* Composition du MEMBRE DE GAUCHE

• Elément 3 :

L'élément 3 correspond à la partie validation d'une transition ($\Pi x_{\langle e_i \rangle}$ avec $e_i \in E_1^-$).

Représentation :

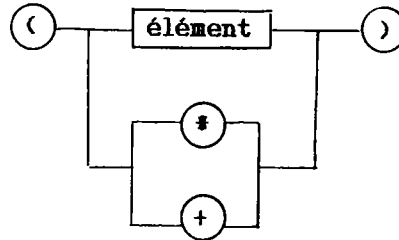


Nota : Pour la description du langage utilisé, on fait appel aux diagrammes syntaxiques de Conway.

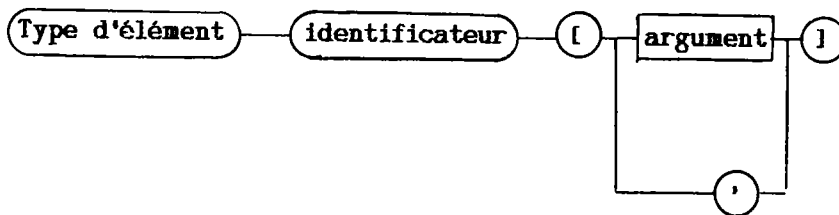
• Elément 4 :

L'élément 4 correspond à la partie réceptivité d'une transition.

Représentation :



L'écriture de cette partie est obtenue par l'emploi d'un ensemble d'éléments ayant comme structure :



* Composition du MEMBRE DE DROITE

• Elément 5 :

L'élément 5 représente le numéro de l'étape à activer en cas de déclenchement de la règle.

Représentation :



• Elément 6 :

L'élément 6 définit l'action à exécuter lors du déclenchement de la règle. Une action représente un élément A_i du quintuplet défini au chapitre I-3.2.

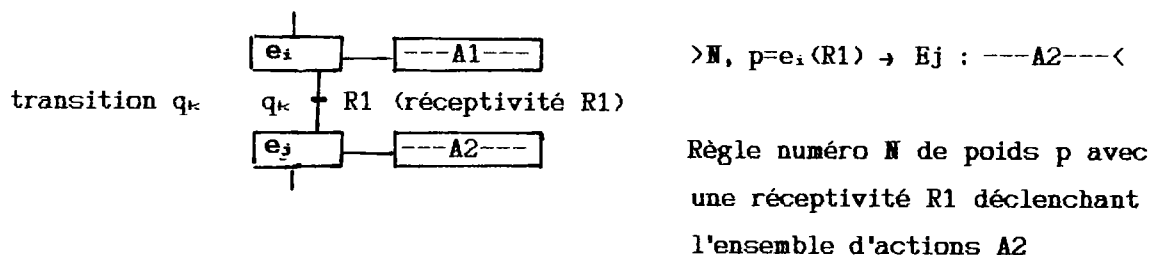
Représentation :



2.3 Expression des différents types de règles de production

A chaque type de transition du Grafcet M est associé un type de règle.

Règle associée à une transition de type "séquence" :



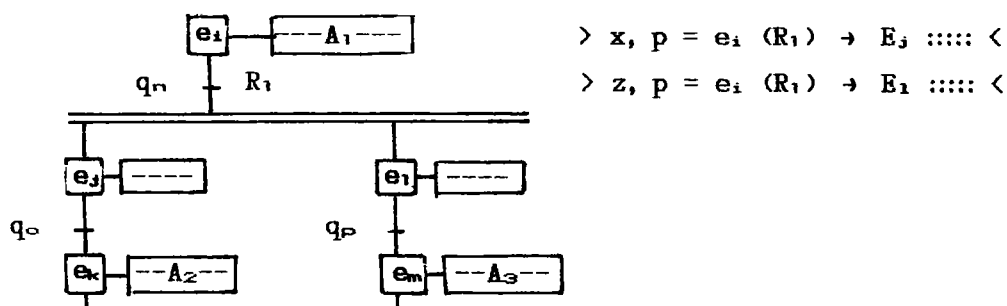
Transition "séquence"

Règle correspondante

La partie validation de la règle est composée d'un seul élément e_i . Il représente le numéro de l'étape qui précède immédiatement la transition q_k . L'élément e_i n'intervient que dans une seule règle (il est unique), ce qui nous donne :

$$\text{Card}(E_k^-) = 1, \text{Card}(E_k^+) = 1$$

Règles associées à une transition de type "divergence ET" :



Transition "divergence ET"

Règles numéro X et Z
correspondant à la transition
 q_n de réceptivité R_1

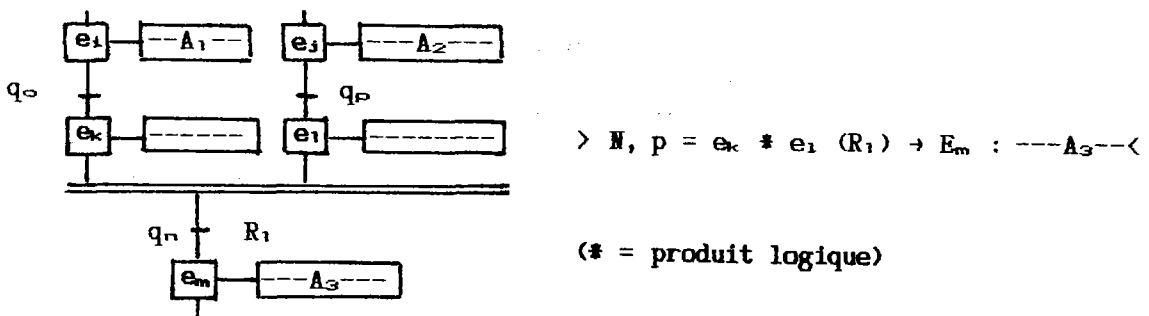
Une divergence de "ET" est réalisée avec un ensemble de règles (autant de règles que de séquences simultanées) ayant :

- un seul et identique élément dans la partie validation, qui représente le numéro de l'étape précédant la transition q_n .
- une partie réceptivité identique
- le numéro de l'étape à exécuter, différent.

Ce qui nous donne :

$$\text{Card } (E_n^-) = 1 \quad , \quad \text{Card } (E_n^+) > 1$$

Règle associée à une transition de type "convergence de ET" :



Transition type "convergence ET"

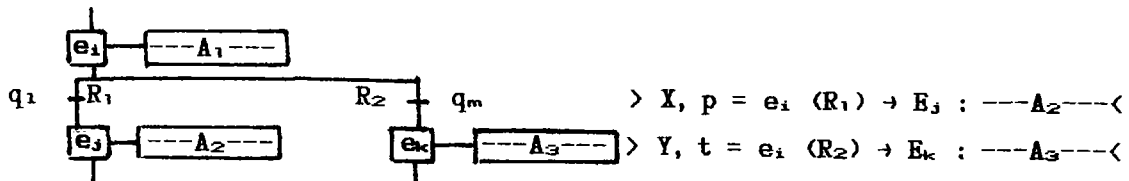
Règle correspondante

Une convergence de "ET" est obtenue par une règle dont la partie validation est composée d'éléments représentant la totalité des étapes immédiatement précédentes de la transition q_n .

On obtient :

$$\text{Card } (E_n^-) > 1 \quad , \quad \text{Card } (E_n^+) = 1$$

Règle associée à une transition de type "divergence de OU" :



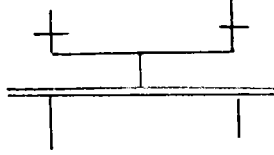
Transition type "divergence de OU"

Règles correspondantes

$$\text{Card } (E_n^+) = \text{Card } (E_o^+) = \text{Card } (E_p^+) = 1$$

$$E_n^- \neq E_o^- \neq E_p^-$$

Cette définition interdit le cas de figure suivant :



3. IMPLANTATION DE LA CONNAISSANCE

L'implantation de la connaissance peut s'effectuer soit en langage interne, soit en langage externe. Le langage interne respecte la syntaxe des règles de production définie au paragraphe précédent. Le langage externe a pour but de définir les éléments de connaissance à partir de la proposition de gammes.

La différence entre ces deux niveaux d'introduction de connaissances concerne uniquement la syntaxe des règles et non les éléments de définition de l'environnement. Ceux-ci sont identiques quelque soit le niveau d'implantation de la connaissance.

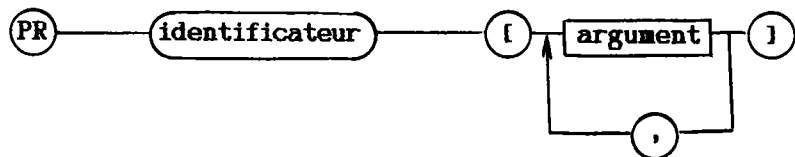
On utilise trois types d'éléments :

- type prédicat
- type fonction
- type objet

et un ensemble d'opérateurs et d'arguments.

3.1 Prédicats

Actuellement le système manipule cinq prédicats élémentaires (qui se réfèrent aux relations élémentaires) dont la structure générale est :



- PR sup [Arg 1, Arg 2] :

Le prédicat "sup" retourne la valeur vraie lorsque l'argument 1 est supérieur à l'argument 2. Le type d'argument autorisé est le type entier.

- PR inf [Arg 1, Arg 2] :

Le prédicat "inf" retourne la valeur vraie lorsque l'argument 1 est inférieur à l'argument 2. Le type d'argument autorisé est le type entier.

- PR comp [Arg 1, Arg 2, Arg 3] :

Le prédicat "comp" retourne la valeur vraie lorsque l'argument 2 est supérieur à l'argument 1 et inférieur à l'argument 3. Le type d'argument autorisé est le type entier.

- PR égal [Arg 1, Arg 2] :

Le prédicat "égal" retourne la valeur vraie lorsque l'argument 1 est égal à l'argument 2. Plusieurs types d'arguments sont autorisés :

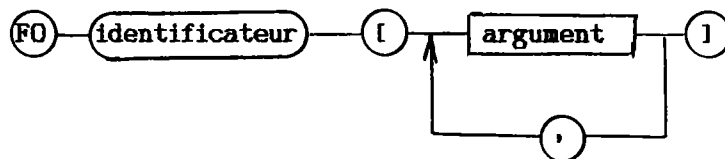
- * type entier
- * type chaîne de caractères
- * type matrice.

- PR inc [Arg 1, Arg 2] :

Le prédicat "inc" retourne la valeur vraie lorsque l'argument 1 est inclus dans l'argument 2. Le type d'argument autorisé, est le type matrice.

3.2 Fonction

Actuellement le système ne manipule qu'une seule fonction, dont la structure générale est notée :

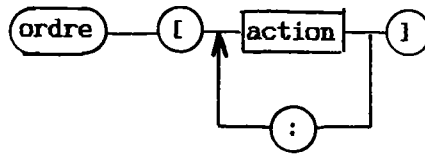


- FO quest [Arg 1] :

La fonction "quest" est du type booléen et l'argument 1 est du type entier. Le but est d'afficher la question ayant le numéro "Arg 1" à l'écran, dont la réponse est soit Oui soit Non. Si la réponse à la question est Oui, la fonction "quest" retourne la valeur vraie.

3.3 Objet

Les objets influencent la stratégie de recherche de solutions selon "l'ordre". L'élément de type objet se distingue par rapport aux autres types déjà vus, dans le fait qu'il agit directement sur l'évolution du graphe, pendant la recherche d'une gamme opératoire. Son action est comparable à celle d'un ordonnateur. Actuellement le système manipule trois "ordres" dont la structure générale est notée :



EF [---A1---] : Lors de la tentative de validation de l'objet EF par le système, celui-ci reçoit l'ordre de vérifier si l'action notée "---A1---" a déjà été établie (en amont de la transition où l'objet EF est décelé).
Si l'action en question a été établie, alors l'objet EF retourne la valeur vraie.

EP [---A1---] : Lors de la tentative de validation de l'objet EP par le système, celui-ci reçoit l'ordre de vérifier si l'action notée "---A1---" sera sélectionnée en aval de la transition où l'objet a été décelé. L'action associée à l'étape immédiatement suivante de la transition concernée, ne sera pas prise en compte. Si l'action peut-être établie, alors l'objet EP retourne la valeur vraie.

EE [---A1---] : Lors de la tentative de validation de l'objet EE par le système, celui-ci reçoit l'ordre de vérifier l'existence de l'action notée "---A1---" dans la gamme opératoire. Si l'action a été établie ou peut-être établie alors l'objet EE retourne la valeur vraie.

3.4 Définition des opérateurs

Un ensemble d'opérateurs a été défini sur les différents éléments.

- * Opérateur négation noté " / ".
Cet opérateur est utilisé pour inverser la valeur retournée par l'élément.
- * Opérateur OU noté " + ".
- * Opérateur ET noté " * ".
- * Opérateur de filtrage noté " ? ".
L'opérateur " ? " n'est utilisé que pour les éléments de type objet. Il est compatible avec les arguments de type action.

3.5 Arguments

L'utilisation des arguments dépend du type de langage dans lequel on se trouve. La notation des différents types d'arguments est différente suivant que l'on réalise un passage d'argument :

- par valeur
- par constante
- par variable

3.5.1 Types d'arguments pour un passage par valeur

Lors d'un passage d'argument par valeur on ne peut utiliser que deux types d'arguments :

Type entier :

L'entier est inscrit en clair dans la partie argument de l'élément.

ex : PR sup [10, 4]

Type chaîne de caractères :

La chaîne de caractères est inscrite entre deux apostrophes :

Le caractère blanc est pris en compte.

ex : PR égal ['Chaîne 1', 'Chaîne 1']

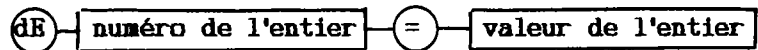
PR égal ['A', 'A']

3.5.2 Types d'arguments pour un passage par constante

Lors d'un passage d'argument par constante on peut utiliser trois types d'arguments :

Type entier :

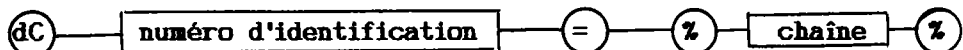
L'entier est introduit par "dE" suivi par un numéro permettant le repérage et la différenciation des entiers. L'affectation est réalisée par le signe " = ".



ex : dE3 = 14

Type chaîne de caractères :

La chaîne de caractères est introduite par "dC" suivi d'un numéro d'identification. L'affectation est réalisée par le signe " = ", et la chaîne de caractères est comprise entre deux caractères " % ".



ex : dC51 = % chaîne-M-3%

Type matrice :

Une matrice est introduite par "dM" suivi d'un numéro d'identification. L'affectation est réalisée par le signe " = ". Toutes les matrices doivent avoir le même format et être du type booléen. (Le format est prédéfini au niveau du programme).



ex : dM2 = 0 1 0 1 1
 0 0 1 0 0
 1 1 1 0 1

Remarque : L'appel des différents types d'arguments est réalisé en utilisant la définition du type de l'argument ainsi que son numéro d'identification.

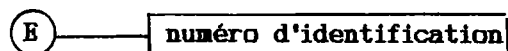
ex : PR égal [dE3, 14]
 PR inc [dM2, dM1]

3.5.3 Types d'arguments pour un passage par variable

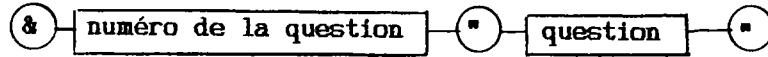
Par variable on entend, un argument dont la valeur est initialisée à chaque recherche. Chaque valeur est introduite pendant le déroulement de la recherche et uniquement en cas de besoin. L'introduction est réalisée de manière interactive par questions/réponses.

Type entier :

L'entier est introduit par la lettre "E" suivie par un numéro d'identification. Il est aussi utilisé comme numéro de la question à afficher. La réponse à cette question doit être un entier.

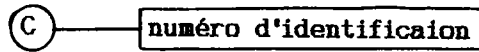


L'ensemble des questions qui se rapporte aux entiers, est stocké dans un fichier "FicQE.DAT" à l'intérieur duquel les questions sont écrites avec le format :



Type chaîne de caractères :

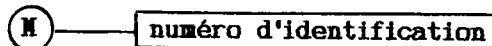
La chaîne de caractères est introduite par la lettre "C" suivie par un numéro d'identification qui est aussi utilisé comme numéro de la question à afficher. La réponse à cette question doit être une chaîne de caractères.



L'ensemble des questions qui se rapporte aux chaînes de caractères est stocké dans le fichier "FicQC.DAT. Le format est identique que celui utilisé pour les entiers.

Type matrice :

Une matrice est introduite par la lettre "M" suivie par un numéro d'identification.



Remarque : L'appel de ces différents types d'arguments est effectué en utilisant une définition du type de l'argument, suivie du nombre d'identification.

ex : PR comp [dE1, 100, E6]
PR inc [dM1, M6]

La figure III.2 récapitule les possibilités d'écriture de règles.

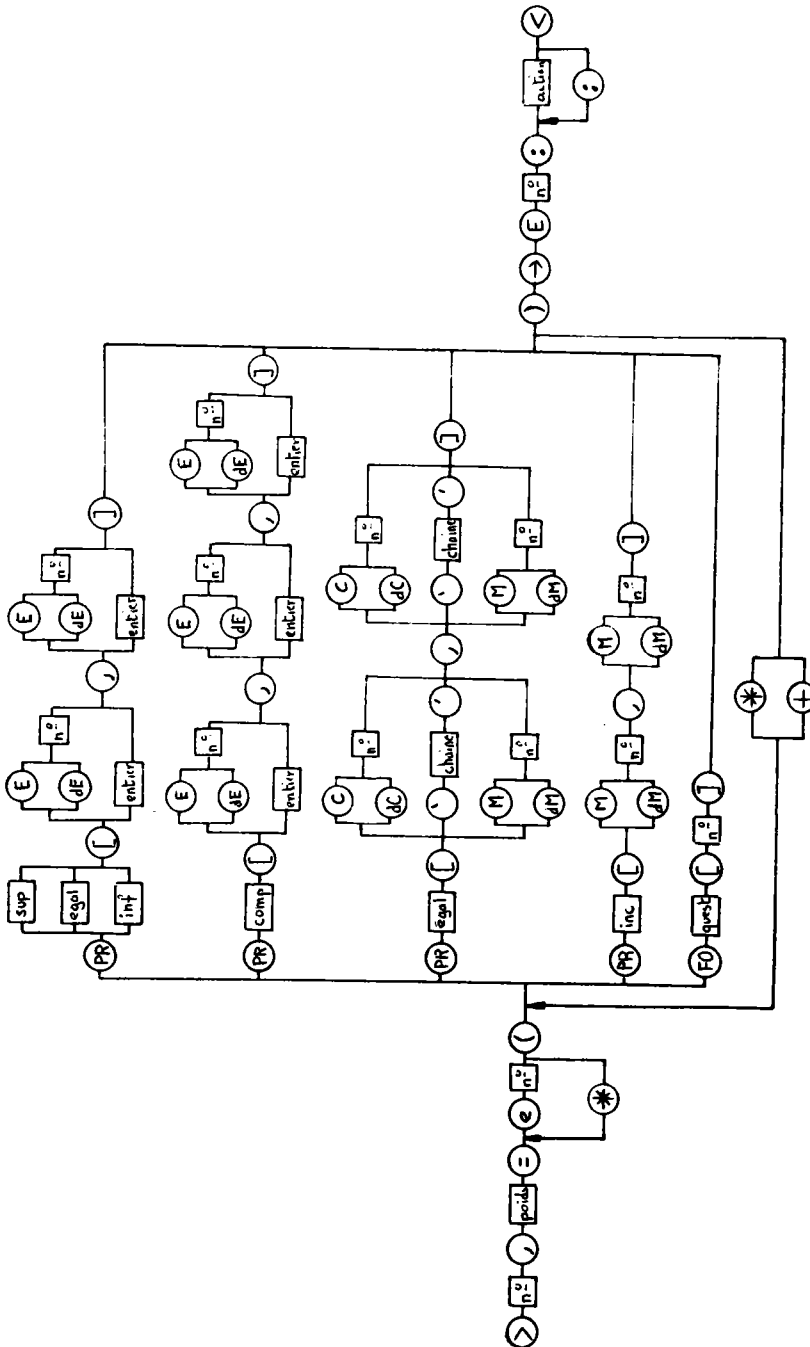
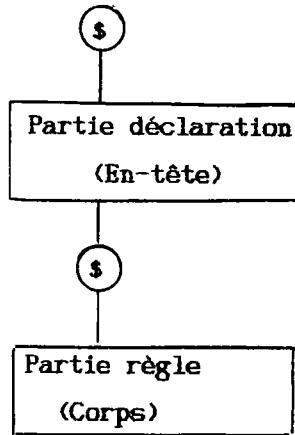


figure III.2 : Structure générale d'une règle de production.

3.6 Implantation en langage interne

A ce niveau d'implantation, la connaissance est implantée en utilisant la structure décrite au sous paragraphe 2.2. L'ensemble de ces règles s'intègre dans une base de connaissances dont la structure est la suivante :



La base de connaissances est donc partagée en deux parties :

- Partie déclaration (En-tête) qui est utilisée pour l'écriture de l'ensemble des types d'arguments utilisé pour le passage par constante.
- Partie règle (Corps) qui est utilisée pour l'écriture de la totalité des règles de production.

3.7 Implantation en langage externe

A ce niveau d'implantation on introduit de la connaissance intervenant sur les séquences d'opérations, à la suite d'une proposition (fig III.3) du système.

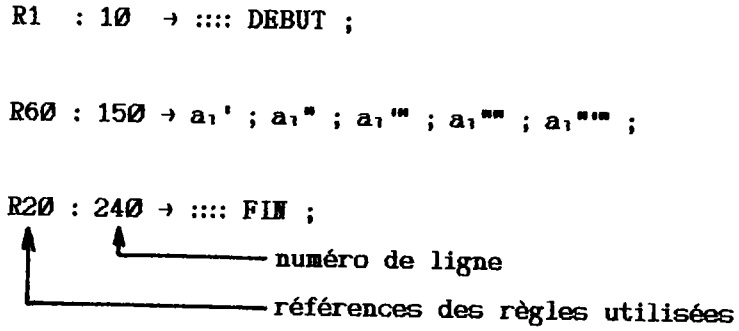


figure III.3 : Modèle de proposition de gamme.

Une proposition se présente sous la forme d'une liste d'actions, dont chaque ligne est numérotée. A chaque ligne on fait correspondre le numéro de la règle responsable du déclenchement. On remarque deux lignes particulières (DEBUT, FIN). Elles permettent d'avoir toujours une proposition, même en cas d'échec total, d'où la possibilité d'introduire une nouvelle gamme opératoire. La structure utilisée pour réaliser ces modifications est la structure (fig III.4) :

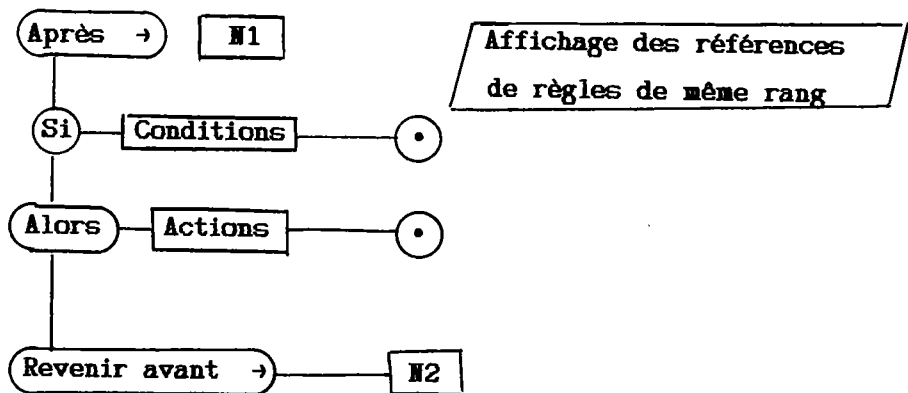


fig III.4 : Structure générale d'une construction en langage externe.

- Séquence unique, cas d'un saut conditionnel avec adjonction d'une liste ordonnée d'actions.

ex : Après → 20 R11 : R12 : R18
 Si EPI ? : a₁" : ? : ? : ? l.
 Alors
 a₃' : : a₃" : a₄" : a₅"
 a₆' : : : : a₆" .

Revenir avant → 50

- Séquence simultanée, cas d'un saut conditionnel avec adjonction de plusieurs listes d'actions parrallèles. Pour cela on utilise le caractère réservé " & " entre chaque liste.

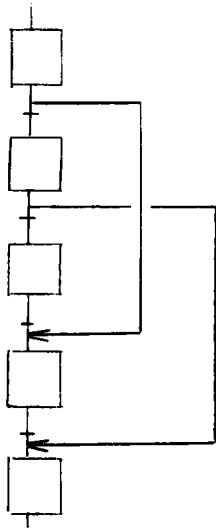
ex : Après 150 R3
 Si PR égal [E10, E2] + PR inf [dE10, 14].
 Alors
 &
 a₁-----a'₁
 a₃-----a₃
 &
 a₄-----a₂
 &
 a₆-----a'₆
 a₇-----a'₇
 a₈-----a'₈.

Revenir avant → 180

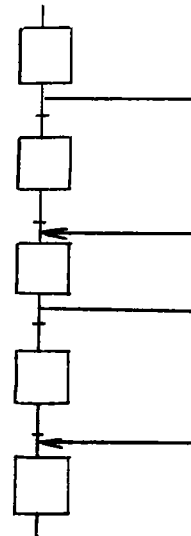
Pour éviter tout problème d'interprétation lors de l'utilisation de ce type de structure, il est indispensable de respecter les règles suivantes :

Règle 1 : Pour une proposition, dans le cas de modifications multiples, l'intersection des intervalles doit être nulle.

$$(N1i, N2i) \cap (N1j, N2j) = \emptyset$$



interdit

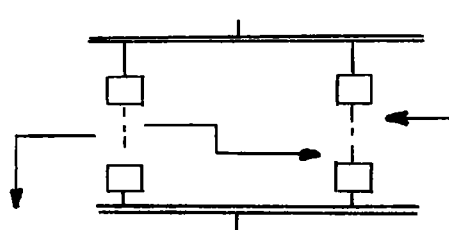


autorisé

Règle 2 : $N1 \geq 10$
 $N2 \leq$ numéro ligne fin
 $N1 < N2$

Règle 3 : La partie condition est obligatoire.

Règle 4 : Les lignes repérées par $N1$ et $N2$ doivent se référer à des règles dont les transitions correspondantes interviennent dans le même bloc. C'est à dire, si l'on décide d'un saut conditionnel dans un bloc composé d'un ensemble de séquences simultanées, le raccordement devra être fait sur la séquence concernée et à l'intérieur du bloc concerné.



interdit

4. EXPLOITATION DE LA CONNAISSANCE

L'exploitation de la base de connaissances [46, 69, 74, 75, 76] consiste à manipuler l'ensemble des règles et des éléments que l'on a définis (chapitre III-2), puis à exploiter de manière optimale le modèle correspondant (Grafcet M) pour obtenir des résultats satisfaisants.

En ayant considéré les règles de production comme des opérateurs de changement d'états, l'ensemble des solutions possibles ou l'espace potentiel de recherche est représenté par un graphe d'états de type "ou", dont la racine serait l'état "DEBUT" et l'ensemble des feuilles serait l'état "FIN".

La présence de transition de type divergence de "ET", nous a amené à considérer deux types d'états et à instaurer des niveaux d'états :

- * Etat simple i de niveau j : MS'_i

C'est un état introduit par une transition de type autre que celles de type "ET", dont la matrice colonne correspondante ne possède qu'un élément avec la valeur vraie.

- * Etat composé i de niveau j : MC'_i

C'est un état introduit par une transition de type "ET", dont la matrice colonne correspondante possède un nombre d'éléments avec la valeur vraie supérieur à 1.

Les états définis à partir de l'état initial sont de niveau 1. A chaque rencontre d'une transition de type "ET", il y a incrémentation du niveau et décrémentation lors d'une rencontre avec une transition de type convergence de "ET".

En considérant les deux types d'états, la représentation de l'espace potentiel de recherche d'un ensemble de règles de production, où il y a la possibilité de N niveaux, peut être obtenue avec N arbres d'états, de type "ou" (fig III.5).

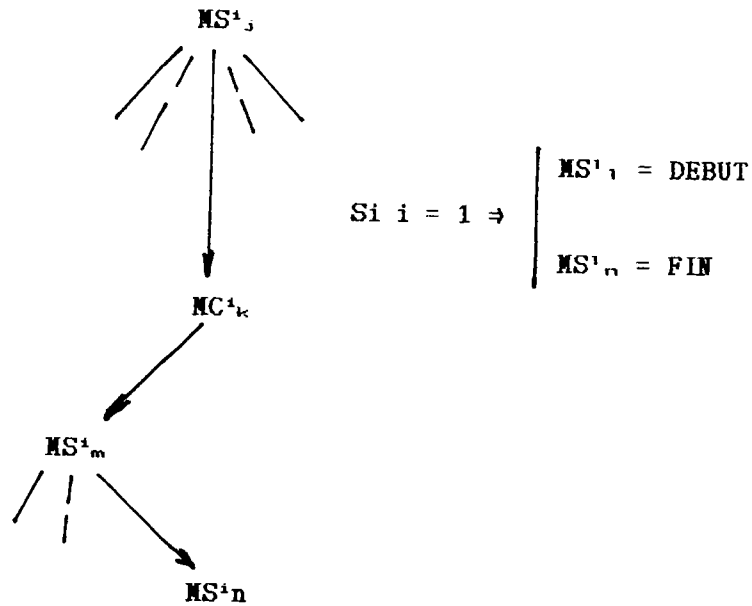


figure III.5 : Représentation de l'espace potentiel de recherche dans un niveau i .

4.1 Mécanisme d'exploitation

Le fonctionnement du mécanisme d'exploitation est basé sur un cycle de contrôle, défini comme l'ensemble des étapes nécessaires à la construction d'un nouveau noeud de l'espace de recherche (fig III.5).

La structure globale du mécanisme d'exploitation est donnée figure III.6.

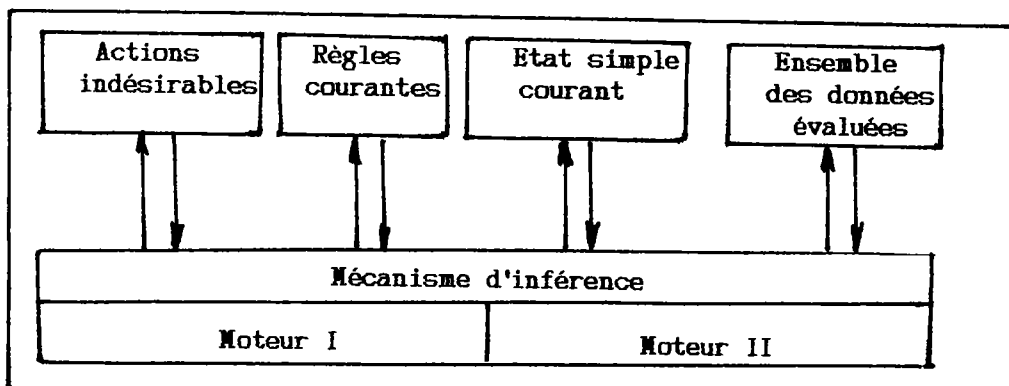


figure III.6 : Structure globale du mécanisme d'exploitation.

Actions indésirables :

C'est l'ensemble des actions que l'utilisateur ne veut pas voir apparaître dans les différentes propositions.

Règles courantes :

C'est l'ensemble des règles disponibles à chaque cycle d'inférence. Cet ensemble de règles est créé de manière dynamique.

Etat simple courant :

Représente l'état actif considéré (à l'instant t) par le mécanisme d'inférence.

Mécanisme d'inférence :

Réalise les cycles nécessaires à la construction de l'arbre de recherche et en assure la gestion. Pour cela, on utilise deux moteurs d'inférence qui ont leur propre cycle de fonctionnement (fig III.7).

Avec : I1 : Teste si la sortie du moteur I est due au besoin de valider un élément de type objet, ou à l'inférence d'une règle.

o → enclenchement du moteur II

n → création d'un nouveau noeud

I2 : Teste la fin de la recherche.

o → Fin

n → poursuite de la recherche

I3 : Teste l'arrêt du moteur II.

o → arrêt ordonné dû à l'échec ou au succès de la validation

n → poursuite de la recherche

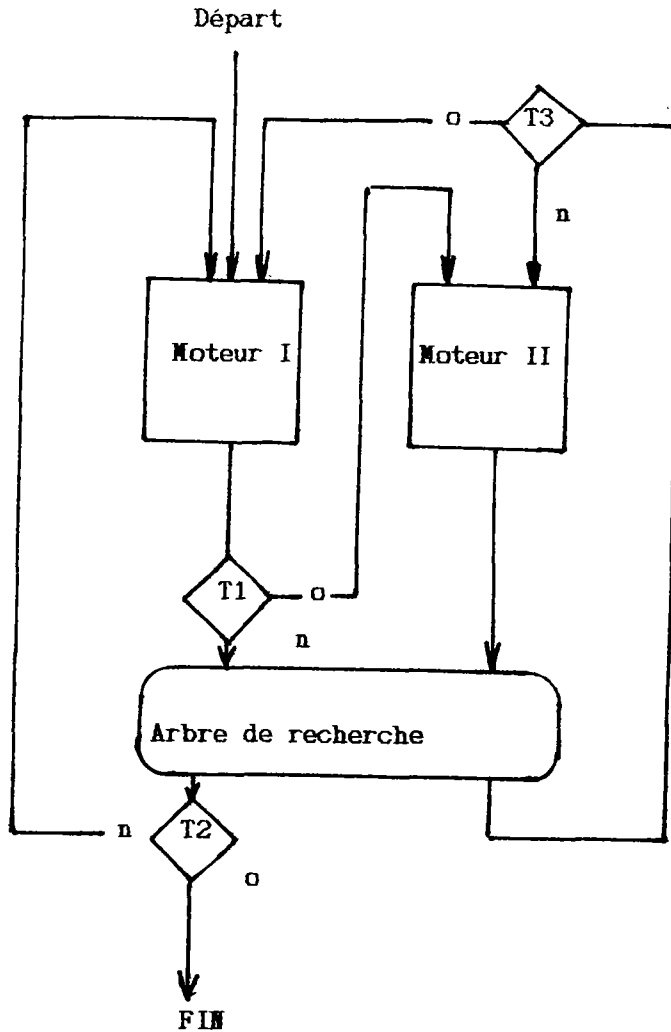


figure III.7 : Cycle général du mécanisme d'inférence.

Choix de stratégie

Il existe deux types de stratégie [65, 66] de recherche de solutions dans un graphe :

* La stratégie en largeur d'abord, qui consiste à garder l'ensemble des solutions "encore possibles" en mémoire.

Avantages : Proposition d'un ensemble de solutions sur lesquelles on pourra effectuer ultérieurement un choix.

Inconvénients : Temps de recherche important et encombrement de la mémoire, qui freinent les performances du système.

* La stratégie en profondeur d'abord, qui consiste à ne garder la trace que d'une solution, parmi l'ensemble des solutions possibles en mémoire.

Avantages : Temps de recherche rapide et encombrement mémoire moindre, qui accentuent les performances du système.

Inconvénients : Proposition d'une seule solution, qui peut ne pas satisfaire l'utilisateur.

Pour des raisons d'efficacité, notre choix s'est porté sur une stratégie en profondeur d'abord. Mais l'utilisateur a la possibilité de rejeter une proposition pour en demander une autre, d'où l'accès à l'ensemble des solutions possibles.

4.2 Caractéristiques moteur I

Le moteur I est basé sur un cycle de fonctionnement comportant trois parties principales (fig III.8).

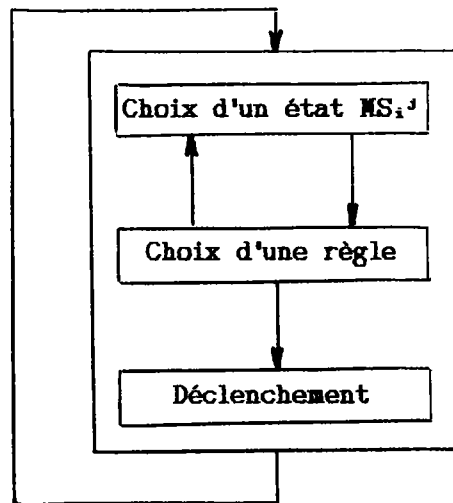


figure III.8 : Cycle global Moteur I.

Le cycle n'est interrompu que pour une :

- * vérification d'un élément de type objet,
- * recherche terminée (Echec, Succès),
- * présence d'actions indésirables.

4.2.1 Choix d'un état

L'objectif de la partie état est de présenter un état MS^i , à partir duquel pourra être déclenchée une règle de production. Pour cela on considère deux natures d'état :

- * Un état est considéré comme vivant, si après apparition de celui-ci, l'ensemble des possibilités de règles s'y appliquant n'a pas été entièrement épuisé.
- * Un état est considéré comme mort, si après apparition de celui-ci, l'ensemble des possibilités de règles s'y appliquant a été entièrement épuisé.

L'existence de ces deux natures d'état, introduit le problème de la restauration d'état antérieur qui consiste à définir :

- * Le moment de prise en compte d'une restauration d'état.
- * Les moyens à utiliser pour la restauration d'état.
- * La détection du point de redémarrage.

4.2.1.1 Détermination du cycle de prise en compte d'une restauration d'état

La remise en cause d'un état [87] peut être effectuée suivant trois cas :

- * Le premier cas consiste à remettre en cause continuellement l'état courant. A chaque cycle, le système examine la possibilité de restaurer un état antérieur, et donc d'abandonner, au-moins provisoirement l'état courant. C'est une solution peu recommandée ; d'une part pour des raisons d'efficacité, d'autre part cela suppose bien peu de confiance dans la pertinence des règles.

- * Le deuxième cas consiste à remettre périodiquement en cause l'état courant. Tous les n cycles on effectue une remise en cause de l'état courant. Cette solution ne peut être employée que pour des problèmes bien spécifiques. Elle ne considère la restauration d'état, que pour des états courants principaux et non pour tous les états.
- * Le troisième cas consiste à remettre en cause occasionnellement l'état courant. Après appréciation des possibilités d'évolution, le système commande une restauration d'état. Les possibilités d'évolution peuvent être considérées de diverses manières :
 - évolution nulle : la restauration d'un état antérieur ne sera considérée que lors d'une impasse.
 - évolution douteuse : la restauration d'un état antérieur ne sera considérée que lors d'une apparition d'un doute sur l'intérêt de poursuivre sur l'état courant.C'est une solution très couramment utilisée, car elle s'adapte à une grande variété de problèmes et intègre des qualités "d'intelligence".

Le système SAGGA ne considère le problème de la restauration d'état antérieur que lors d'une apparition de situation d'impasse. Ce choix est dû exclusivement au fait que le but est de proposer un maximum de solutions, donc a priori toutes les solutions possibles sont intéressantes.

4.2.1.2 Moyen utilisé pour la restauration d'état

La remise en cause d'un état MS^j_i , se traduit par la restauration d'un état antérieur MS^j_{i-n} et la prise en compte de celui-ci comme état courant.

Une restauration d'état est étroitement liée à la nature des modifications qu'une règle de production peut opérer sur l'état MS^j_i pour produire l'état MS^j_{i+1} . Suivant la nature de ces modifications, la restauration d'état sera obtenue soit :

- * directement par application des règles,
- * par un mécanisme intégré dans le cycle de fonctionnement.

Les deux caractéristiques à considérer pour identifier la nature des modifications sont :

- * la permutabilité des règles de production,
- * la réversibilité des règles de production.

Les propriétés des règles de production définies dans le Grafcet M leurs confèrent les caractéristiques suivantes :

- * non permutable : chaque règle est liée à un numéro d'étape, pour respecter la condition d'ordre.
- * non réversible : après validation de sa condition de franchissement, le rôle d'une règle de production est uniquement d'ajouter des actions.

Ces deux caractéristiques, nous ont amené à introduire un mécanisme de restauration d'état "Retour arrière" (Backtrack) à l'intérieur du cycle de fonctionnement.

4.2.1.3 Détermination du point de redémarrage

La détermination du point de redémarrage consiste à définir un ensemble de critères, qui sera utilisé pour estimer la pertinence d'un état, pour son utilisation comme nouvel état courant.

Dans le cas d'une restauration d'état intégré au cycle de fonctionnement on distingue deux méthodes :

- * Le backtrack systématique utilisé lorsque, pendant la recherche le sens de parcours est imposé. Le retour est imposé au dernier état activable d'un ensemble d'états activables ordonnés.
- * Le backtrack sélectif qui réalise un retour sur un état activable de l'ensemble des états activables, sans notion d'ordre.

Plusieurs critères peuvent être utilisés pour sélectionner l'état en question :

- * L'utilisation des informations obtenues lors de certains cycles précédents pour juger d'un état comme point de redémarrage.

- * L'utilisation des dépendances, lorsque le système détecte une contradiction, il l'étudie et choisit un point de redémarrage où la contradiction n'existe plus.
- * L'utilisation d'un opérateur de limitation de backtrack.
- * L'utilisation de fonctions heuristiques d'évaluation de la distance entre un état activable et un état solution.

Nous avons choisi la détermination du point de redémarrage "backtrack systématique" pour les raisons suivantes :

- * Il est impératif de respecter l'ordre, donc le sens de parcours est imposé.
- * L'on n'a pas introduit de notion de préférence entre les points de redémarrage.

4.2.2 Choix d'une règle

L'objectif de cette partie est de sélectionner une ou plusieurs règles de manière efficace. Pour cela il nous faut définir les moyens de :

- * invocation des règles de production,
- * présélection des règles de production,
- * sélection des règles de production.

4.2.2.1 Invocation des règles

Dans le type "règles de production" comportant un membre de droite et un membre de gauche, les règles peuvent être invoquées de deux façons :

- * Invocation sur le membre de gauche (MG) ; celle-ci est employée lorsque les règles de production mettent en jeu dans la partie MG, des éléments figurant dans la base de faits au moment de l'invocation des règles. Le cas échéant la règle sera retenue. Le membre de gauche est considéré comme le déclencheur de la règle. Cette invocation est du type "chaînage avant" ou "raisonnement guidé par les données".

- * Invocation sur le MD ; celle-ci est employée lorsque les règles de production mettent en jeu la partie MD, des éléments à établir. Dans cette méthode le MD (ou conclusion) est considéré comme le déclencheur de la règle. Ce type d'invocation est dit "chainage arrière" ou "raisonnement guidé par les buts et sous-buts".

Le principe de ces deux méthodes est résumé de façon schématique sur la figure III.9.

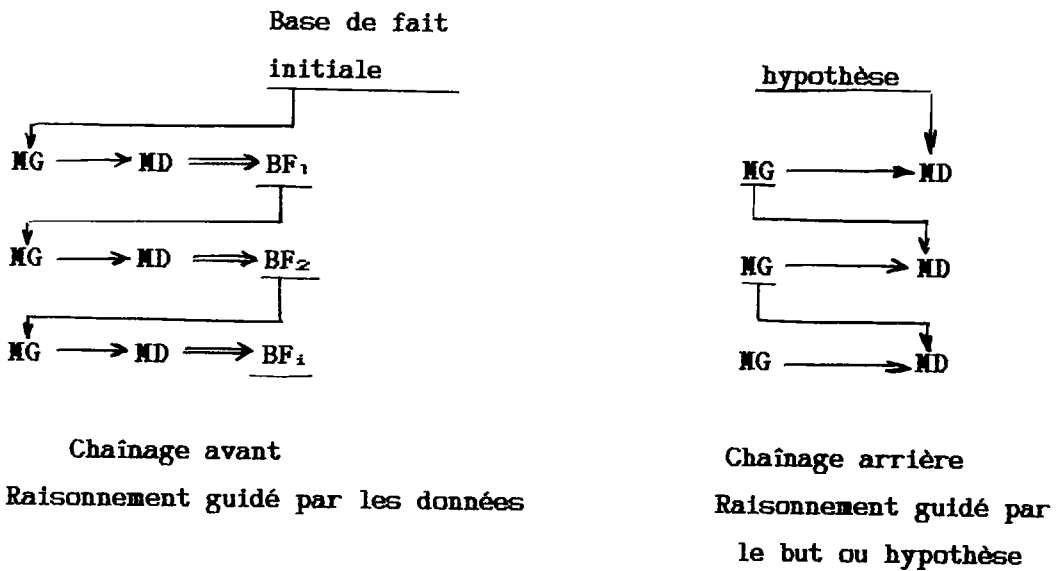


figure III.9 : Méthodes d'invocation des règles.

Le rôle du moteur I consiste à partir de la base de faits et de l'ensemble des données, de construire une liste ordonnée d'actions. A tout moment il consulte la base de faits courante ainsi que les données. Le moteur I considère les règles de production comme des règles dites "en avant", son cycle de fonctionnement est du type "chainage avant".

4.2.2.2 Présélection des règles

A partir d'un état courant, l'objectif de la présélection est de repérer toutes les règles susceptibles d'être sélectionnées. La présélection est effectuée parmi l'ensemble des règles figurant dans la base de règles courante.

La présélection consiste à retenir toutes les règles dont la partie validation est vraie.

4.2.2.3 Sélection des règles

L'objectif de la sélection est de ne considérer que les règles (parmi les règles présélectionnées) dont la partie réceptivité est vraie.

Une sélection peut être envisagée de multiples façons :

- * Sur les règles, par l'application de poids (coefficient de pondération) pour chaque règle.
- * Sur les parties réceptivité, par l'application de poids sur chaque élément, puis évaluation du poids de la partie réceptivité d'où on choisit la règle la plus adaptée à la situation.
- * Aléatoirement
- * Par méta-règles.

La solution adoptée est une sélection par évaluation locale des poids associés à chaque règle.

La sélection est faite sur la règle dont le poids est le plus fort, en cas d'égalité, la sélection est faite sur la première règle sélectionnée.

4.2.3 Partie déclenchement

La partie déclenchement est responsable de la construction de l'arbre de recherche. Le déclenchement d'une règle contribue à la création effective d'un noeud dans l'arbre de recherche. (voir annexe : Structure informatique).

4.3 Caractéristiques Moteur II

L'existence d'éléments de type objet (EE, EP) nous a amené à considérer un second moteur dont les caractéristiques sont différentes de celles du moteur I.

La validation d'un élément de type objet tel que EP [---A_i---] passe par l'établissement de l'action A_i. En effet dans ce cas A_i devient ainsi un but à atteindre d'où le type invocation des règles de production "chaînage arrière".

Le fonctionnement global du moteur II est identique au fonctionnement du moteur I (fig III.8).

4.3.1 Choix des états

La partie état du moteur II diffère de celle du moteur I par le fait que le parcours dans l'arbre d'état est réalisé dans le sens inverse, c'est à dire des feuilles vers les racines.

Donc l'arbre potentiel de recherche (fig III.10) est différent de celui développé par le moteur I (fig III.5).

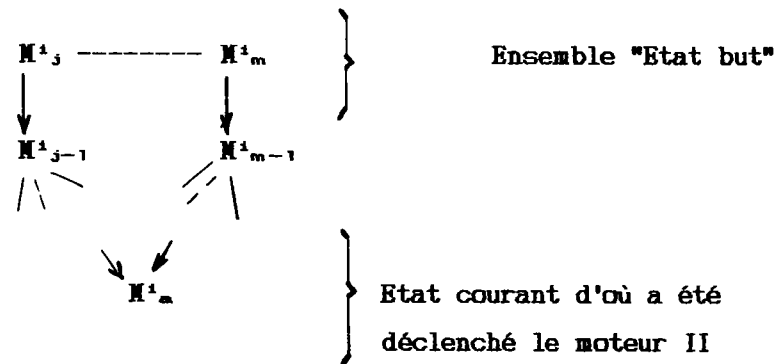


figure III.10 : Arbre (de type "ou") potentiel de recherche du moteur II.

Les notions d'état vivant et d'état mort sont aussi présentes dans le fonctionnement du moteur II.

Les remarques et les choix cités dans le cas du moteur I sont identiques pour le moteur II à savoir :

- * Détermination du cycle de prise en compte d'une restauration d'état, par situation d'impasse.
- * Mécanisme de restauration d'état, intégré dans le cycle de fonctionnement.
- * Détermination du point de redémarrage par "backtrack" systématique.

4.3.2 Choix des règles

La partie règle diffère selon la méthode d'invocation des règles de production.

L'invocation des règles du moteur II étant faite en chaînage arrière, la présélection de celles-ci sera effectuée sur la partie (5) du membre de droite de la règle de production (chapitre III-2.2).

Les remarques et les choix cités dans le cas du moteur I, en ce qui concerne la sélection et le déclenchement des règles de production restent valables pour le moteur II.

4.4 Structure de l'arbre de recherche du mécanisme d'exploitation

En résumé, le cycle de fonctionnement du mécanisme d'exploitation est la combinaison des cycles des différents moteurs (fig III.11).

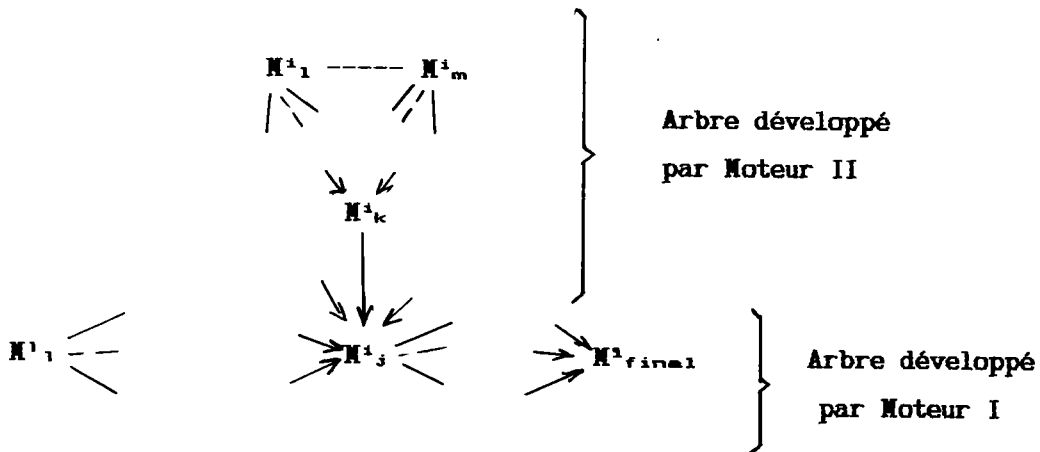


figure III.11 : Arbre potentiel de recherche du mécanisme d'exploitation.

Cette combinaison est obtenue lorsque, pendant un chaînage avant, le mécanisme est amené à valider un élément de type objet (EE, EP).

5. EXEMPLE ACADEMIQUE

Afin de montrer les possibilités actuelles du système SAGGA, nous avons écrit un exemple complet de base de règles permettant de tester :

- * Les procédures de recherche dans le cas d'un grafcet **M** complexe.
- * L'évolution des différentes stratégies, introduites par les combinaisons de plusieurs éléments de type "objet".

La figure III.12 définit la base de règles utilisée, tandis que la figure III.13 montre le Grafcet **M** correspondant.

Nous avons effectué une série de tests pour vérifier, à partir de cet exemple, le bon fonctionnement du système. Nous présentons ci-après l'un de ces tests. L'annexe "Structure informatique" résume l'ensemble des autres tests.

Nous avons résumé l'ensemble des possibilités du système SAGGA par les tableaux suivants :

avec : 1 → Résolu, 0 → Non résolu, i → interdit

Tableau 1 :

Type transition	Seq.	->ou<-	<-ou->	->et<-	<-et->
Seq.	1	1	1	1	1
->ou<-	1	1	1	1	1
<-ou->	1	1	1	0	1
->et<-	1	1	1	1	1
<-et->	1	0	1	0	1

Ce tableau récapitule, l'ensemble des associations possibles de type transition. La lecture doit s'effectuer de manière horizontale.

Tableau 4 :

I type	I	I	I	I	I	I
I transition	I	I	I	I	I	I
I type	I Seq.	I ->ou<-	I <-ou->	I ->et<-	I <-et->	I
I regle	I	I	I	I	I	I
I Seq.	I 1	I 1	I 1	I i	I i	I
I ->ou<-	I i	I i	I i	I i	I i	I
I <-ou->	I i	I i	I i	I i	I i	I
I ->et<-	I i	I 1	I i	I 1	I i	I
I <-et->	I i	I i	I Ø	I i	I 1	I

Une transition est composée d'une ou plusieurs règles. Ce tableau récapitule l'ensemble des compositions possibles, de chaque types transitions. La lecture doit d'effectuer de manière verticale.

----- Fichier REGLES -----.

```
$
$
> 1 ,1= ( ) -> E1 : : : : :DEBUT<
> 2 ,1=e1 ( ) -> E2 : : : : :FIN<
> 3 ,3= e1 ( PRegal[1,1] ) -> E3 :11:11:act1::<
> 4 ,5= e3 ( EF[?:?:act60:?:?]*EE[09:09:?:?:09] ) -> E4 :30:21:act56:4:4<
> 5 ,1= e4 ( EF[?:?:?:?:1134] ) -> E5 :33:33:act3::5<
> 6 ,1= e5 ( ) -> E2 ::::FIN<
> 7 ,1= e3 ( PRegal[E1,1] ) -> E6 :55:5:act60::<
> 8 ,1= e6 ( ) -> E5 :33:33:act3::<
> 9 ,4= e1 ( PRegal[E2,2]*EE[21:?:act303:?:?] ) -> E7 ::::~<
>10 ,1= e7 ( ) -> E8 ::::~<
>11 ,1= e8 ( ) -> E9 :2:2:act34::1134<
>12 ,1= e9 ( FQquest[8] ) -> E10 :21:21:act56::~<
>13 ,1= e10 ( ) -> E11 ::::~<
>14 ,1= e7 ( ) -> E12 ::::~<
>15 ,1= e12 ( ) -> E13 :3:3:act78::~<
>16 ,1= e13 ( ) -> E14 :31:31:act90::~<
>17 ,1= e14 ( ) -> E15 ::::~<
>18 ,1= e11*e15 ( ) -> E3 :11:11:act1::~<
>19 ,1= e13 ( PRegal[E3,3] ) -> E16 :55:44:act60::~<
>20 ,1= e16 ( ) -> E14 :31:31:act90::~<
>21 ,1= e6 ( PRegal['a',C1]+FQquest[12] ) -> E17 :66:66:act11::~<
>22 ,1= e17 ( EE[99:99:act14:?:?] ) -> E18 :77:77:act12::~<
>23 ,1= e18 ( ) -> E19 :88:88:act13::~<
>24 ,1= e19 ( ) -> E20 :99:99:act14::~<
>25 ,1= e20 ( EE[?:?:act772:?:?]*EF[77:?:act12:?:?] ) -> E21 :09:09:act15::~<
>26 ,1= e21 ( ) -> E22 :33:33:act16::~<
>27 ,1= e22 ( EF[?:?:act123:?:?] ) -> E23 :44:44:EnAvalDe123::~<
>28 ,1= e23 ( ) -> E24 :55:55:act18::~<
>29 ,1= e24 ( ) -> E2 ::::FIN<
>30 ,1= e19 ( PRinf[E4,100] ) -> E25 ::::~<
>31 ,1= e25 ( ) -> E26 ::::~<
>32 ,1= e26 ( ) -> E27 :6:6:act21::~<
>33 ,1= e27 ( ) -> E28 :6:6:act22::~<
>34 ,1= e28 ( EF[?:?:act321:?:?]*FQquest[6]+FQquest[7] ) -> E29 :6:6:act23::~<
>35 ,1= e29 ( ) -> E30 :6:6:act34::~<
>36 ,1= e30 ( ) -> E31 ::::~<
>37 ,1= e25 ( ) -> E32 ::::~<
>38 ,1= e32 ( ) -> E33 :7:7:act41::~<
>39 ,1= e33 ( ) -> E34 ::::~<
```

figure III.12 : Base de règles (Première partie).

```

>40 ,1= e25 ( ) -> E35 :::::<
>41 ,1= e35 ( ) -> E36 :8:8:act51::<
>42 ,1= e64*68 ( ) -> E37 :::::<
>43 ,1= e31*e34*e37 ( ) -> E22 :33:33:act16::<
>44 ,1= e26 ( PRsup[E5,5] ) -> E38 :::::<
>45 ,1= e38 ( ) -> E39 :::::<
>46 ,1= e39 ( ) -> E40 :7:7:act123::<
>47 ,1= e40 ( ) -> E41 :::::<
>48 ,1= e38 ( ) -> E42 :::::<
>49 ,1= e42 ( ) -> E43 :7:7:act321::<
>50 ,1= e43 ( ) -> E44 :::::<
>51 ,1= e41*e44 ( ) -> E27 :6:6:act21::<
>52 ,1= e36 ( ) -> E45 :::::<
>53 ,1= e36 ( ) -> E48 :::::<
>54 ,1= e36 ( ) -> E51 :::::<
>55 ,1= e45 ( ) -> E46 :10:10:act 101::<
>56 ,1= e46 ( ) -> E47 :::::<
>57 ,1= e48 ( ) -> E49 :11:11:act56::<
>58 ,1= e49 ( ) -> E50 :::::<
>59 ,1= e51 ( ) -> E52 :12:12:act 103::<
>60 ,1= e52 ( ) -> E53 :::::<
>61 ,1= e52 ( ) -> E56 :::::<
>62 ,1= e53 ( EF[?:?:act60:?:?] ) -> E54 :13:13:act104::<
>63 ,1= e54 ( ) -> E55 :::::<
>64 ,1= e56 ( ) -> E57 :13:13:act 105::<
>65 ,1= e57 ( ) -> E58 :::::<
>66 ,1= e55*e58 ( ) -> E59 :::::<
>67 ,1= e47*e50*e59 ( ) -> E60 :21:21:act303::<
>68 ,1= e60 ( ) -> E61 :::::<
>69 ,1= e60 ( ) -> E65 :::::<
>70 ,1= e61 ( ) -> E62 :14:14:act 401::<
>71 ,1= e62 ( ) -> E63 :14:14:act 402::<
>72 ,1= e63 ( ) -> E64 :::::<
>73 ,1= e65 ( ) -> E66 :15:15:act 403::<
>74 ,1= e66 ( F0quest[14] ) -> E67 :15:15:act 404::<
>75 ,1= e67 ( ) -> E68 :::::<
>76 ,4= e1 ( /PRegal[E2,2]*EP[31:31:?:?:?] ) -> E69 :66:66:act60::<
>77 ,1= e69 ( ) -> E70 :66:66:act67::<
>78 ,1= e70 ( EP[66:66:?:?:66] ) -> E71 :66:66:act68::<
>79 ,1= e71 ( ) -> E72 :66:66:act69:66:66<
>80 ,1= e72 ( ) -> E7 :::::<
>81 ,1= e17 ( /PRegal['a',C1] ) -> E19 :88:88:act13::<
>82 ,1= e27 ( PRegal['a',C1] ) -> E74 :44:44:act44::<
>83 ,1= e74 ( ) -> E75 :44:44:act45::<
>84 ,1= e75 ( ) -> E76 :44:44:act46::<
>85 ,1= e76 ( ) -> E30 :6:6:act34::<
>86 ,1= e22 ( /PRegal[E2,2] ) -> E77 :77:77:act771::<
>87 ,1= e77 ( ) -> E78 :77:77:act772::<
>88 ,1= e78 ( ) -> E24 :55:55:act18::<
>89 ,5= e69 (F0quest[10]*EP[?:21:act56:?:?:?]*F0quest[11]*EP[?:?:act60:?:?:?]
+ F0quest[9]*EP[09:?:act15:?:?] ) -> E79 :09:09:Act09::09<
>90 ,1= e79 ( ) -> E7 :::::<
>91 ,7= e3 ( EP[?:?:act13:?:?:?]*EF[?:?:act34:?:?:?]
+ F0quest[4]*EP[?:?:act104:?:?:?]*F0quest[5] ) -> E80 :1:68:Act68::<
>92 ,1= e80 ( ) -> E6 :55:5:act60::<

```

figure III.12 : Base de règles (suite).

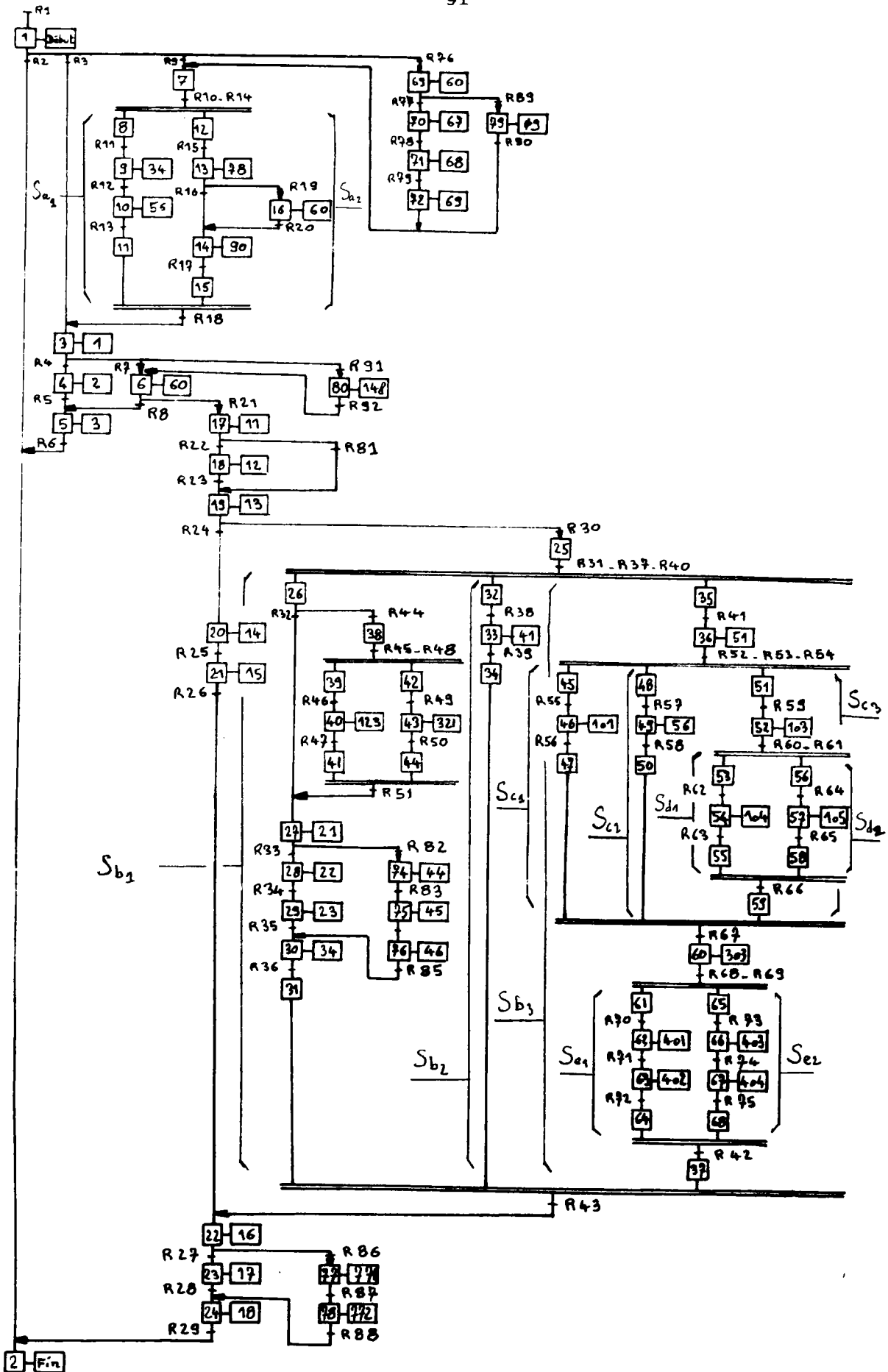


figure III.13 : Grafset correspondant à la base de règles de la figure III.12.

5.1 Test :

Ce test nous permet de valider, entre autre, les règles R89 et R91 dont la partie réceptivité est la plus complexe.

Lors des propositions, on démontre que le système SAGGA prend en compte l'ensemble des solutions des différents niveaux ainsi que de leurs combinaisons.

Introduction du nom de la B.C -> c:base2.dat

```
I-----I
I BASE DE CONNAISSANCES UTILISEE -->  c:base2.dat I
I-----I
I      *****Menu Principal***** I
I-----I
I      Recherche ->      R      I
I      Quitte   ->      Q      I
I               ----- I
I               choix ->  r      I
I-----I
I      Copyright by L.A.E.I University METZ I
```

Est-ce que la recherche est assujettie a des contraintes O/N ->n

cette question reclame comme reponse un nombre entier

" question 2 " :
Reponse ->6

cette question reclame comme reponse un nombre entier

" question 3 " :
Reponse ->3

Je vous pose la question suivante ->
"question 8"

reponse O/N ->o

Je vous pose la question suivante ->
"question 10"

reponse O/N ->o

Je vous pose la question suivante ->
"question 11"

reponse O/N ->o

cette question reclame une reponse sous forme de car. ou de ch.

" question 1 chaine de car" :
Reponse ->s

Je vous pose la question suivante ->
"question 12"

reponse O/N ->o

cette question reclame comme reponse un nombre entier

" question 4 " :
Reponse ->56

Je vous pose la question suivante ->
"question 14"

reponse O/N ->o

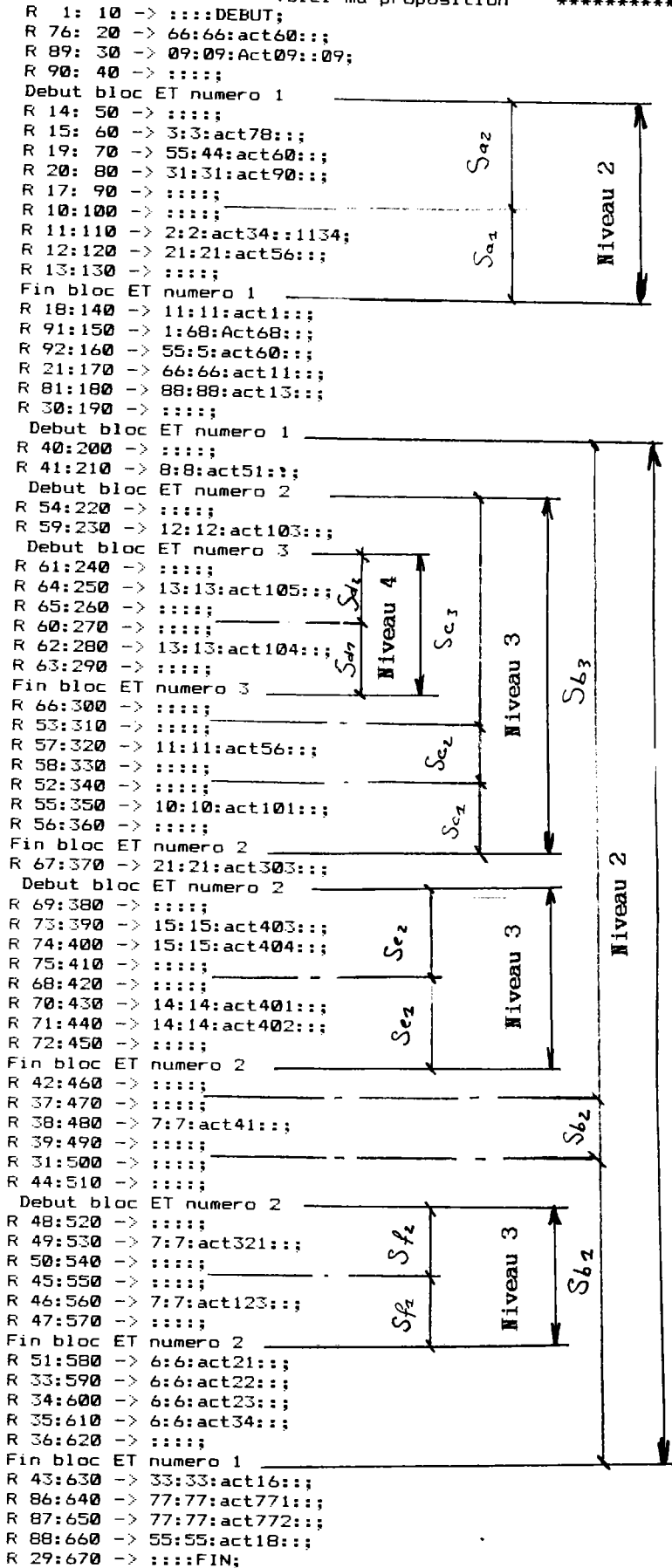
cette question reclame comme reponse un nombre entier

" question 5 " :
Reponse ->10

Je vous pose la question suivante ->
"question 6"

reponse O/N ->o

***** voici ma proposition *****



Desirez vous modifier la proposition O/N -> n

Desirez-vous une autre proposition O/N ->0

```

***** voici ma proposition *****
R 1: 10 -> :::DEBUT;
R 76: 20 -> 66:66:act60::;
R 89: 30 -> 09:09:Act09::09;
R 90: 40 -> ::::;
Debut bloc ET numero 1
R 14: 50 -> ::::;
R 15: 60 -> 3:3:act78::;
R 19: 70 -> 55:44:act60::;
R 20: 80 -> 31:31:act90::;
R 17: 90 -> ::::;
R 10:100 -> ::::;
R 11:110 -> 2:2:act34::1134;
R 12:120 -> 21:21:act56::;
R 13:130 -> ::::;
Fin bloc ET numero 1
R 18:140 -> 11:11:act1::;
R 91:150 -> 1:68:Act68::;
R 92:160 -> 55:5:act60::;
R 21:170 -> 66:66:act11::;
R 81:180 -> 88:88:act13::;
R 30:190 -> ::::;
Debut bloc ET numero 1
R 40:200 -> ::::;
R 41:210 -> 8:8:act51::;
Debut bloc ET numero 2
R 54:220 -> ::::;
R 59:230 -> 12:12:act103::;
Debut bloc ET numero 3
R 61:240 -> ::::;
R 64:250 -> 13:13:act105::;
R 65:260 -> ::::;
R 60:270 -> ::::;
R 62:280 -> 13:13:act104::;
R 63:290 -> ::::;
Fin bloc ET numero 3
R 66:300 -> ::::;
R 53:310 -> ::::;
R 57:320 -> 11:11:act56::;
R 58:330 -> ::::;
R 52:340 -> ::::;
R 55:350 -> 10:10:act101::;
R 56:360 -> ::::;
Fin bloc ET numero 2
R 67:370 -> 21:21:act303::;
Debut bloc ET numero 2
R 69:380 -> ::::;
R 73:390 -> 15:15:act403::;
R 74:400 -> 15:15:act404::;
R 75:410 -> ::::;
R 68:420 -> ::::;
R 70:430 -> 14:14:act401::;
R 71:440 -> 14:14:act402::;
R 72:450 -> ::::;
Fin bloc ET numero 2
R 42:460 -> ::::;
R 37:470 -> ::::;
R 38:480 -> 7:7:act41::;
R 39:490 -> ::::;
R 31:500 -> ::::;
R 44:510 -> ::::;
Debut bloc ET numero 2
R 48:520 -> ::::;
R 49:530 -> 7:7:act321::;
R 50:540 -> ::::;
R 45:550 -> ::::;
R 46:560 -> 7:7:act123::;
R 47:570 -> ::::;
Fin bloc ET numero 2

```



```
R 51:580 -> 6:6:act21::;
R 33:590 -> 6:6:act22::;
R 34:600 -> 6:6:act23::;
R 35:610 -> 6:6:act34::;
R 36:620 -> ::::;
Fin bloc ET numero 1
R 43:630 -> 33:33:act16::;
R 27:640 -> 44:44:EnAvalDel23::;
R 28:650 -> 55:55:act18::;
R 29:660 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> n

Desirez-vous une autre proposition O/N ->o

Je vous pose la question suivante ->
"question 7"

reponse O/N ->o

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 76: 20 -> 66:66:act60::;
R 89: 30 -> 09:09:Act09::09;
R 90: 40 -> ::::;
Debut bloc ET numero 1
R 14: 50 -> ::::;
R 15: 60 -> 3:3:act78::;
R 19: 70 -> 55:44:act60::;
R 20: 80 -> 31:31:act90::;
R 17: 90 -> ::::;
R 10:100 -> ::::;
R 11:110 -> 2:2:act34::1134;
R 12:120 -> 21:21:act56::;
R 13:130 -> ::::;
Fin bloc ET numero 1
R 18:140 -> 11:11:act1::;
R 91:150 -> 1:68:Act68::;
R 92:160 -> 55:5:act60::;
R 21:170 -> 66:66:act11::;
R 81:180 -> 88:88:act13::;
R 30:190 -> ::::;
Debut bloc ET numero 1
R 40:200 -> ::::;
R 41:210 -> 8:8:act51::;
Debut bloc ET numero 2
R 54:220 -> ::::;
R 59:230 -> 12:12:act103::;
Debut bloc ET numero 3
R 61:240 -> ::::;
R 64:250 -> 13:13:act105::;
R 65:260 -> ::::;
R 60:270 -> ::::;
R 62:280 -> 13:13:act104::;
R 63:290 -> ::::;
Fin bloc ET numero 3
R 66:300 -> ::::;
R 53:310 -> ::::;
R 57:320 -> 11:11:act56::;
R 58:330 -> ::::;
R 52:340 -> ::::;
R 55:350 -> 10:10:act101::;
R 56:360 -> ::::;
Fin bloc ET numero 2
```

```
R 67:370 -> 21:21:act303::;
  Debut bloc ET numero 2
R 69:380 -> ::::;
R 73:390 -> 15:15:act403::;
R 74:400 -> 15:15:act404::;
R 75:410 -> ::::;
R 68:420 -> ::::;
R 70:430 -> 14:14:act401::;
R 71:440 -> 14:14:act402::;
R 72:450 -> ::::;
Fin bloc ET numero 2
R 42:460 -> ::::;
R 37:470 -> ::::;
R 38:480 -> 7:7:act41::;
R 39:490 -> ::::;
R 31:500 -> ::::;
R 32:510 -> 6:6:act21::;
R 33:520 -> 6:6:act22::;
R 34:530 -> 6:6:act23::;
R 35:540 -> 6:6:act34::;
R 36:550 -> ::::;
Fin bloc ET numero 1
R 43:560 -> 33:33:act16::;
R 86:570 -> 77:77:act771::;
R 87:580 -> 77:77:act772::;
R 88:590 -> 55:55:act18::;
R 29:600 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> n

Desirez-vous une autre proposition O/N ->n

```
I-----I
I BASE DE CONNAISSANCES UTILISEE --> c:base2.dat I
I-----I
I *****Menu Principal***** I
I-----I
I      Recherche ->      R      I
I      Quitte   ->      Q      I
I               ----- I
I               choix -> q      I
I-----I
      Copyright by L.A.E.I University METZ
```

CHAPITRE IV

Application de SAGGA à la fabrication mécanique

1. INTRODUCTION

Dans ce chapitre nous présentons une application du système SAGGA sur un exemple industriel concernant la famille des pièces mécaniques prismatiques "paliers, supports, guides".

Nous montrons tout d'abord les différents menus utilisateurs affichés à l'écran lors de la génération de gammes opératoires, puis nous présentons le logiciel d'édition de gammes qui transforme les gammes opératoires en gammes détaillées. Après avoir précisé un certain nombre de références concernant le positionnement et le codage de cette famille de pièces nous montrons sur des exemples quelques possibilités du système SAGGA.

2. PRESENTATION DU SYSTEME SAGGA

Un des objectifs du système SAGGA est de minimiser l'intervention du préparateur. Après avoir rentré le code pièce, celui-ci ne fait que répondre aux questions posées par le système.

Une fois une proposition de gamme retenue, le préparateur appelle le logiciel d'édition de gamme.

La figure IV.1 représente l'organigramme des différents menus utilisés dans le système SAGGA une fois la famille déterminée.

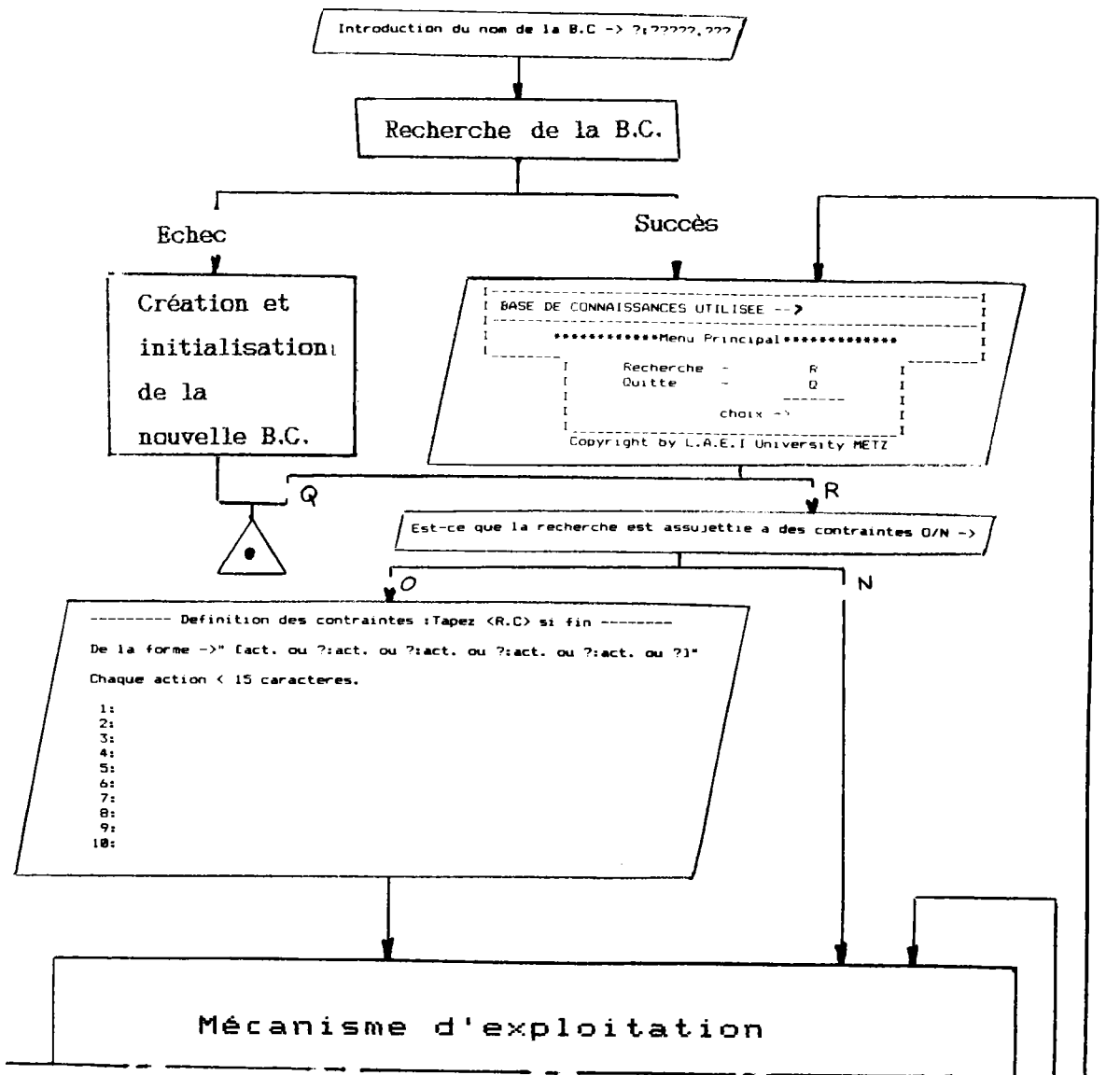


figure IV.1 : Organigramme des menus (Première partie).

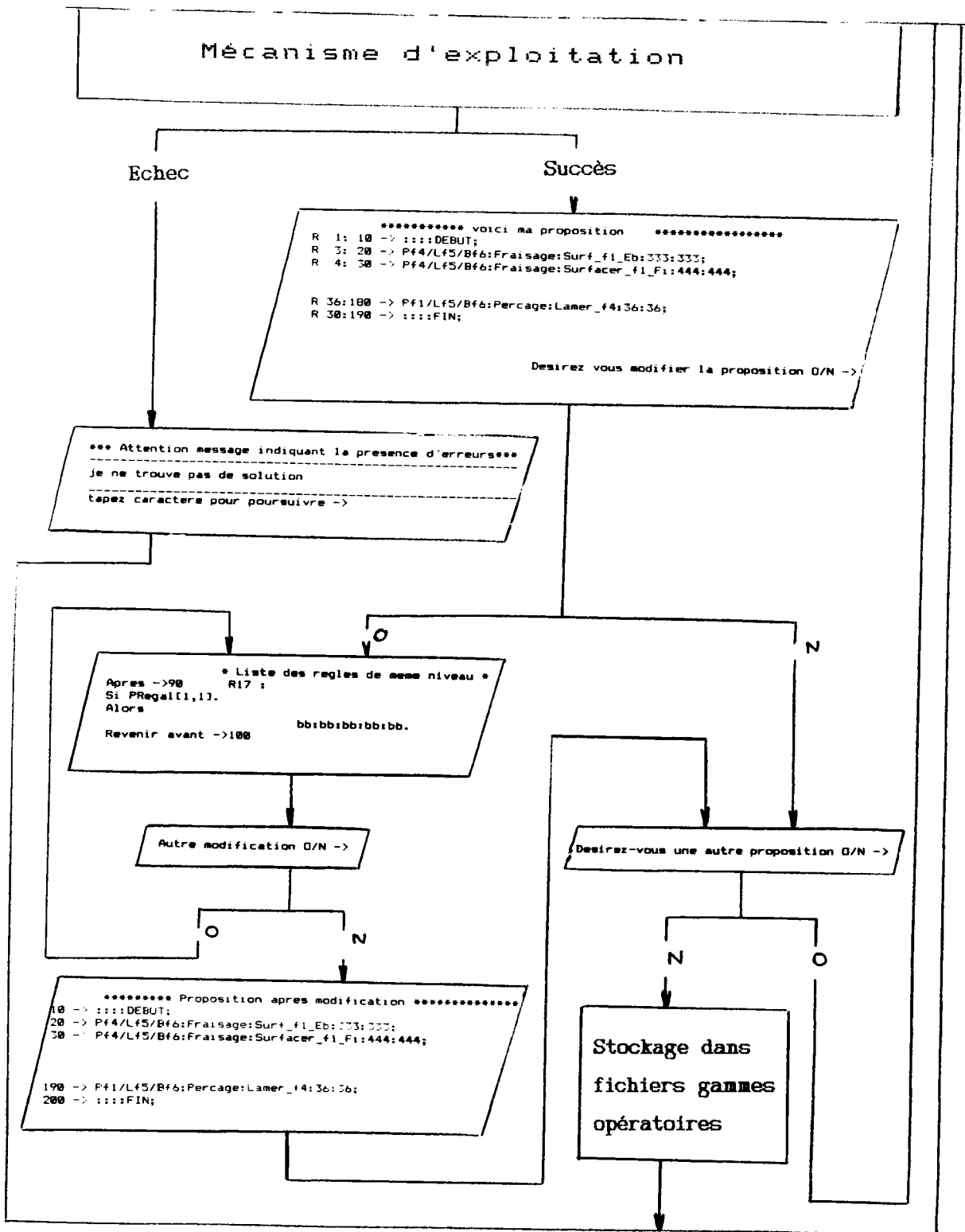


figure IV.1 : Organigramme des menus (Deuxième partie).

3. PRESENTATION DU SYSTEME D'EDITION DE GAMMES

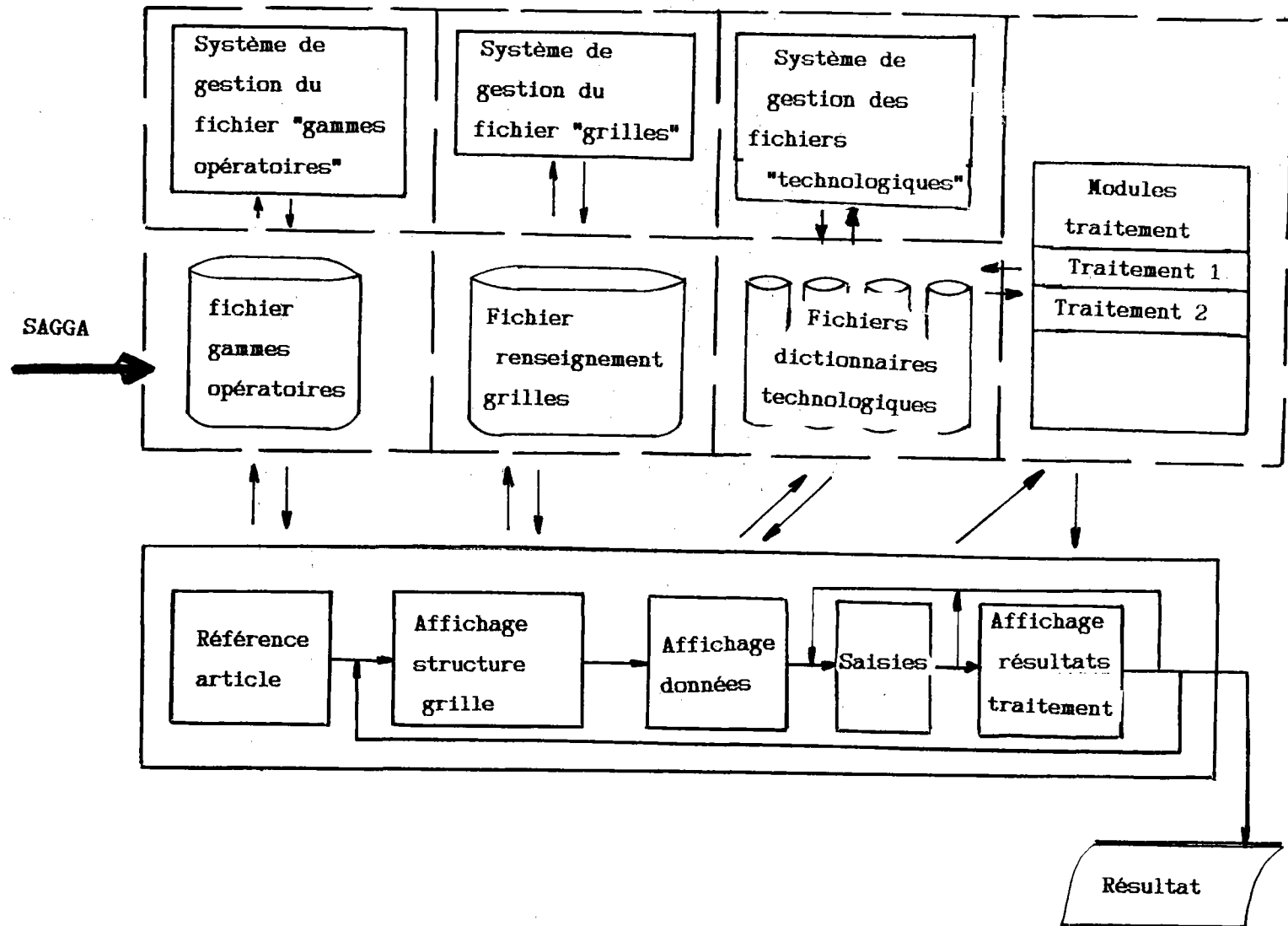
L'ordre et la nature des opérations étant définis par le système SAGGA, le système d'édition de gamme permet, de façon interactive par grilles d'affichage et de saisies, de préciser et d'éditer toutes les caractéristiques technologiques de la gamme opératoire (machine, outil, condition de coupe, calcul de temps, coût, ...).

Ce système d'édition est décrit globalement en figure IV.2. En annexe C nous présentons les caractéristiques de ce système [13, 20, 48, 64]. Il se compose de quatre programmes indépendants, et travaille sur un ensemble de fichiers séquentiels à structure indexée et accès monocritère.

Les quatres programmes en question sont appelés à partir d'un menu général :

- * GestionF : Il permet la création et la gestion de quatre fichiers technologiques
- * Grille : Grâce à un jeu de questions/réponses, il permet la création des grilles d'affichage et de saisies. Celles-ci sont stockées dans un fichier dont la gestion est assurée par le programme Grille. A noter que le programme utilise un module de traitement dans lequel est stocké un ensemble de procédures de calcul, pouvant être utilisé par les grilles.
- * Util : Après introduction du nom d'une gamme opératoire, le programme Util recherche la gamme correspondante dans le fichier gammes opératoires. Puis à chaque nom d'opération et de façon ordonnée, il affiche la grille correspondante à partir de laquelle un dialogue s'instaure, aidant ainsi à l'introduction des informations nécessaires à l'élaboration de la gamme détaillée.

figure IV.2 : Caractéristiques du système d'édition de gammes.



4. APPLICATIONS

4.1 Définition des références pièces

Le choix d'un référentiel dépend essentiellement de la famille d'articles étudiée.

Pour la famille considérée dont le volume de base est parallélépipédique, nous avons choisi un repère orthonormé à partir duquel sont définies les faces principales (voir fig IV.3).

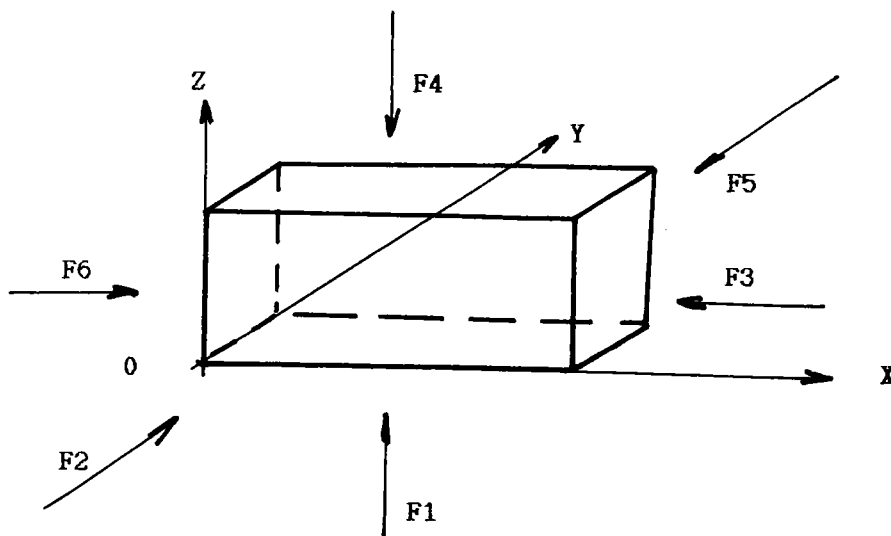


figure IV.3 : Référentiel avec indication des faces principales.

De même, afin de préciser l'orientation de la pièce par rapport au repère, nous utilisons pour cette famille, les règles d'orientation suivantes :

- * La semelle doit être parallèle au plan XY.
- * La matière doit être dans la partie positive du repère.
- * La direction de l'axe principal doit être parallèle à l'axe OX.
- * La face utilisée comme référence pour la cotation des axes de trous de fixation, doit être parallèle au plan YZ.

En résumé, chaque orientation des articles devra respecter le croquis de la figure IV.4

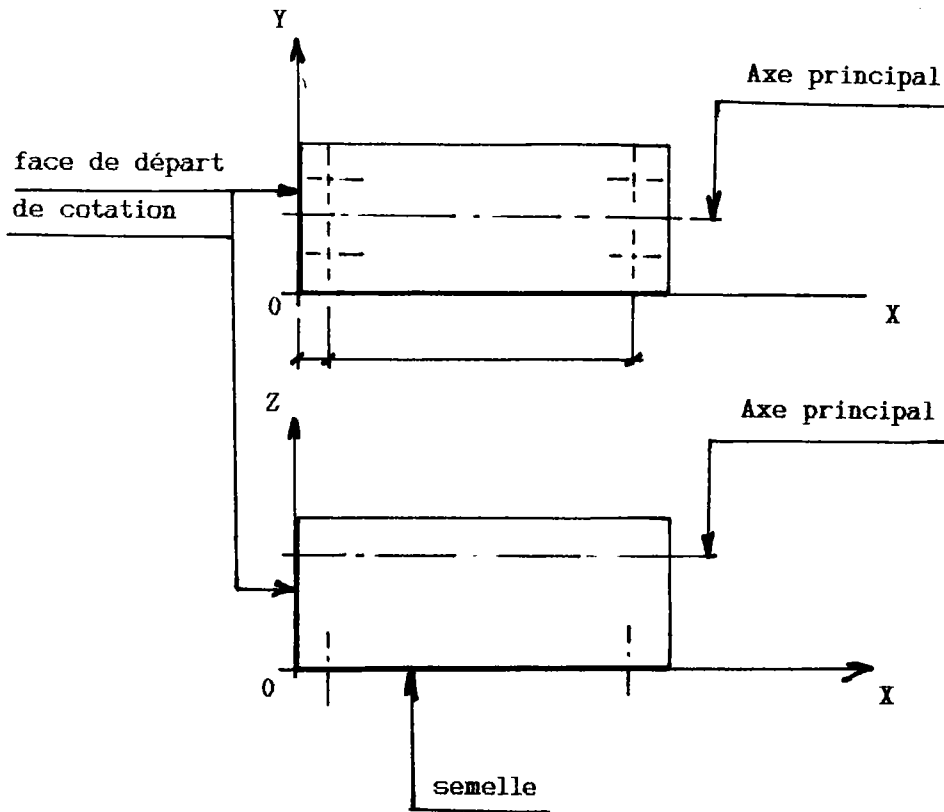


figure IV.4 : Orientation type pour la famille "paliers, supports, guide".

4.2 Définition de l'isostatisme

L'isostatisme de l'article par rapport aux faces d'appui est défini par les symboles suivants :

Appellations	Symboles
Appui plan	P
Appui linéaire	L
Butée	B
Guidage en vé	V
Centrage long intérieur	Ct_lg_int
Centrage long extérieur	Ct_lg_ext

4.3 Système de codification

Nous avons utilisé le code OPITZ à 5 digits, relatif aux pièces parallélépipédiques pour coder les articles de la famille.
(voir figures IV.5 et IV.6).

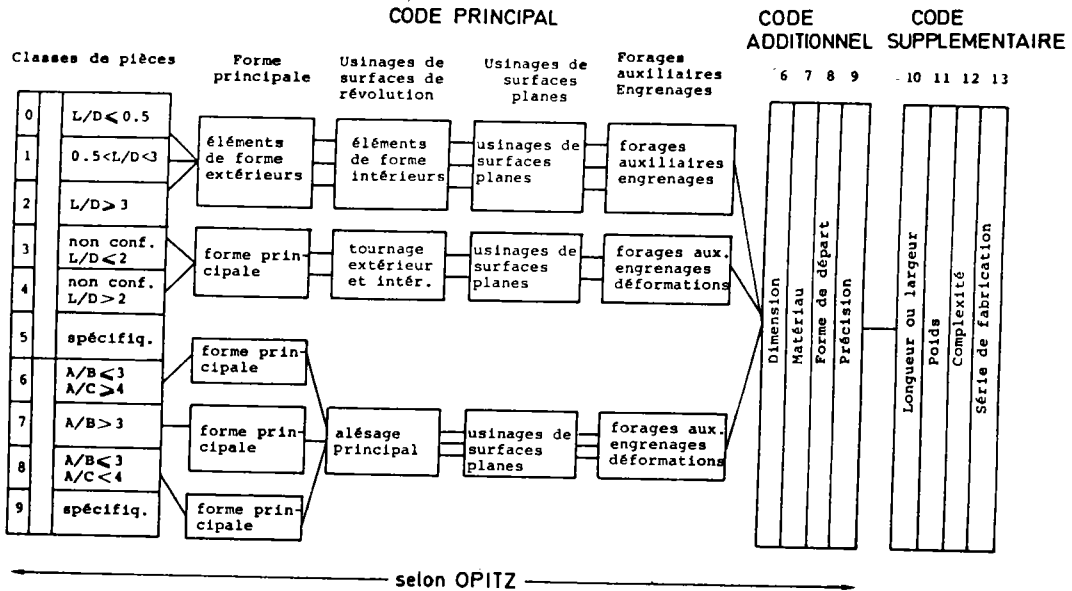


figure IV.5 : Opitz - code principal.

CODE PRINCIPAL POUR LES PIÈCES DE NON-REVOLUTION

CLASSES DE PIÈCES	FORMES PRINCIPALES - 3 DIMENSIONS	ALESAGES PRINCIPAUX (1 à 7)	USINAGES DE SURFACE	FORAGES AUXILIAIRES - DÉFORMATIONS ENGRENAGES
0	CUBE - POUTRE	0 NEANT	0 NEANT	0 NEANT
1	TRAPEZE RECTANGULAIRE - ÉCHAMBRURES DROITES DANS UNE VUE	1 UN ALESAGE PRINCIPAL LISSE	1 CHAMFREIN PORTIONNEL $L' = l/4$ p.ex. chambrées de soudure	1 UNE DIRECTION DE FORAGE
2	SUPERPOSITION DE PLUSIEURS BLOCS	2 UN ALESAGE PRINCIPAL DONT LA FORME EST VARIANTE	2 UNE SEULE FACE A USINER	2 PLUSIEURS DIRECTIONS DE FORAGE
3	PIECES CUBIQUES	3 UN ALESAGE PRINCIPAL * ELEMENTS DE FORME	3 USINAGES DE SURFACE PLANS NE NECESSITANT QU'UN SEUL SERRAGE	3 UNE DIRECTION DE FORAGE
4	3 SE COMPOSANT DE 2 OU PLUSIEURS PARTIES	4 DEUX ALESAGES PRINCIPAUX PARALLELES	4 USINAGES DE SURFACE PLANS NECESSITANT AU MOINS DEUX SERRAGES	4 PLUSIEURS DIRECTIONS DE FORAGE
5	FORME QUELCONQUE	5 PLUSIEURS ALESAGES PRINCIPAUX PARALLELES	5 ENTAILLE ET/OU RAINURE	5 DÉFORMATION SANS FORAGE
6	PIECES - CARTER	6 PLUSIEURS ALESAGES PRINCIPAUX A AXES PERPENDICULAIRES	6 4 + 5	6 DÉFORMATION AVEC FORAGE
7	FORME QUELCONQUE	7 RAINURE CIRCULAIRE SANS ALESAGE PRINCIPAL	7 SURFACE DOME	7 ENGRENAGE SANS FORAGE
8	PIECES - CARTER	8 7 + ALESAGE PRINCIPAL	8 SURFACE DE GUIDAGE	8 ENGRENAGE AVEC FORAGE
9	PLUS - PIECES	9 AUTRES CAS	9 AUTRES CAS	9 AUTRES CAS

figure IV.6 : Opitz - code pour la classe de pièces dites "non-révolution".

4.4 Exemples

L'application concerne la famille des "paliers-supports-guides".
A l'origine nous disposons de deux dessins de définition d'articles (A, B) avec leurs gammes respectives.

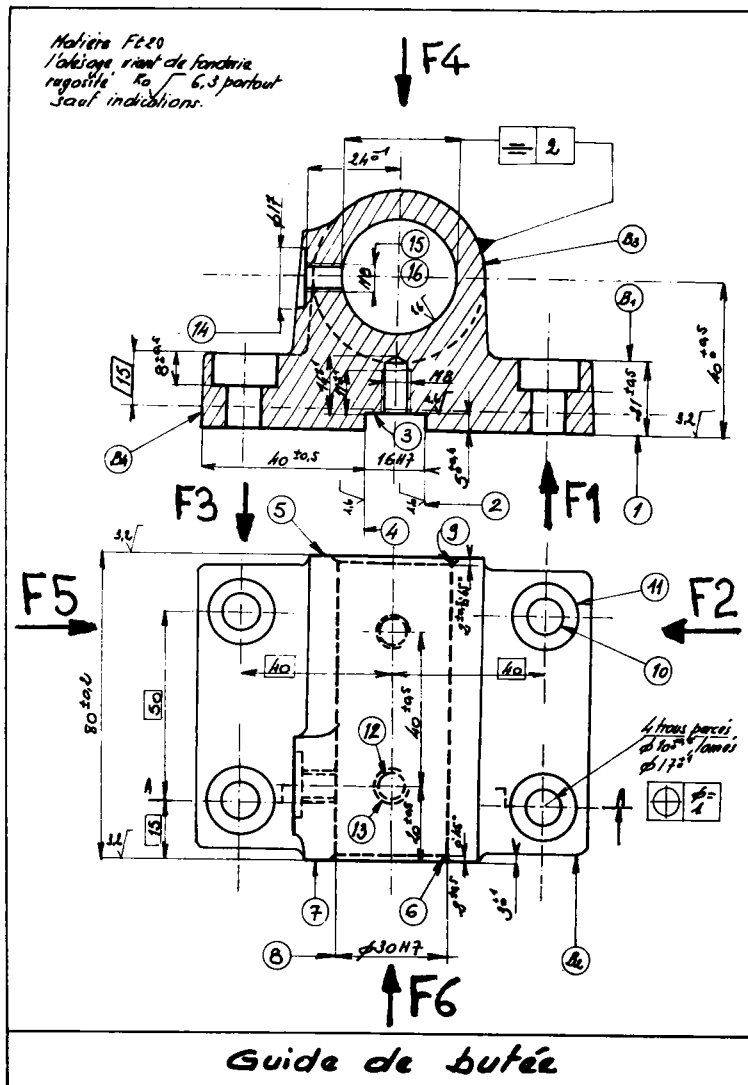
Après avoir implanté la base de connaissances correspondant à cette famille nous montrons quelques possibilités du système à l'aide de 4 exemples :

- * Recherche de gamme pour un article A.
Celle-ci est effectuée pour un lot de 8 articles. Le système SAGGA propose deux gammes satisfaisantes.
- * Recherche de gamme pour un nouvel article C.
Cette gamme concerne 100 articles. Le système SAGGA fait une proposition peu satisfaisante mais suffisamment intéressante pour être modifiée. Après ces modifications en langage externe, la base de règles est mise à jour.
- * Recherche de gamme pour un nouvel article D.
Nous devons réaliser une série de 8 articles D. Le système propose deux gammes opératoires satisfaisantes.
- * Recherche de gamme pour l'article D.
Par rapport à la précédente recherche, ici on introduit une contrainte de recherche. Du fait de la présence de cette contrainte le système SAGGA ne fait qu'une seule proposition.

4.4.1 Définition de la base de connaissances de la famille

a) Dessin de définition de l'article A (figure IV.7)

Le dessin de définition représente un guide de butée sur lequel on a précisé les faces principales ainsi que le code correspondant.



Article A

Code 8-3-1-5-2

Figure IV.7

Gamme de l'article A : série > 5

10-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e:

S.u.r.f._f.1_.E.b;3.3.3:3.3.3;

20-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e:

S.u.r.f.a.c.e.r._f.1_.F.i;4.4.4:4.4.4;

30-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e:R.a.i.n._f.1_.E.b;6.6.6:6.6.6;

40-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e:R.a.i.n._f.1_.F.i;7.7.7:7.7.7;

50-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e:R.a.i.n._f.1_.F.i;8.8.8:8.8.8;

60-:E.b.a.v.u.r.a.g.e;;

70-:C.o.n.t.r.ô.l.e.;;

80-P.f.1./L.f.5./B.f.3:F.r.a.i.s.a.g.e:

S.u.r.f._f.3.f.6_.S.i.m.u.l;1.2:1.2;

90-P.f.4./L.f.5./V.f.4:P.e.r.c.a.g.e:A.l.e.s.e.r._F.i;2.6:2.6;

100-C.o.n.t.r.ô.l.e.;;

110-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e:P.e.r.c.e.r._f.4;2.3:2.3;

120-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e:L.a.m.e.r._f.4;2.8:2.8;

130-P.f.1./L.f.2./B.f.3:P.e.r.c.a.g.e:P.e.r.c.e.r._f.5;2.9:2.9;

140-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e:L.a.m.e.r._f.5;3.1:3.1;

150-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e:T.a.r.a.u.d.e.r._f.5;3.2:3.2;

160-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e:P.e.r.c.e.r._f.1;3.3:3.3;

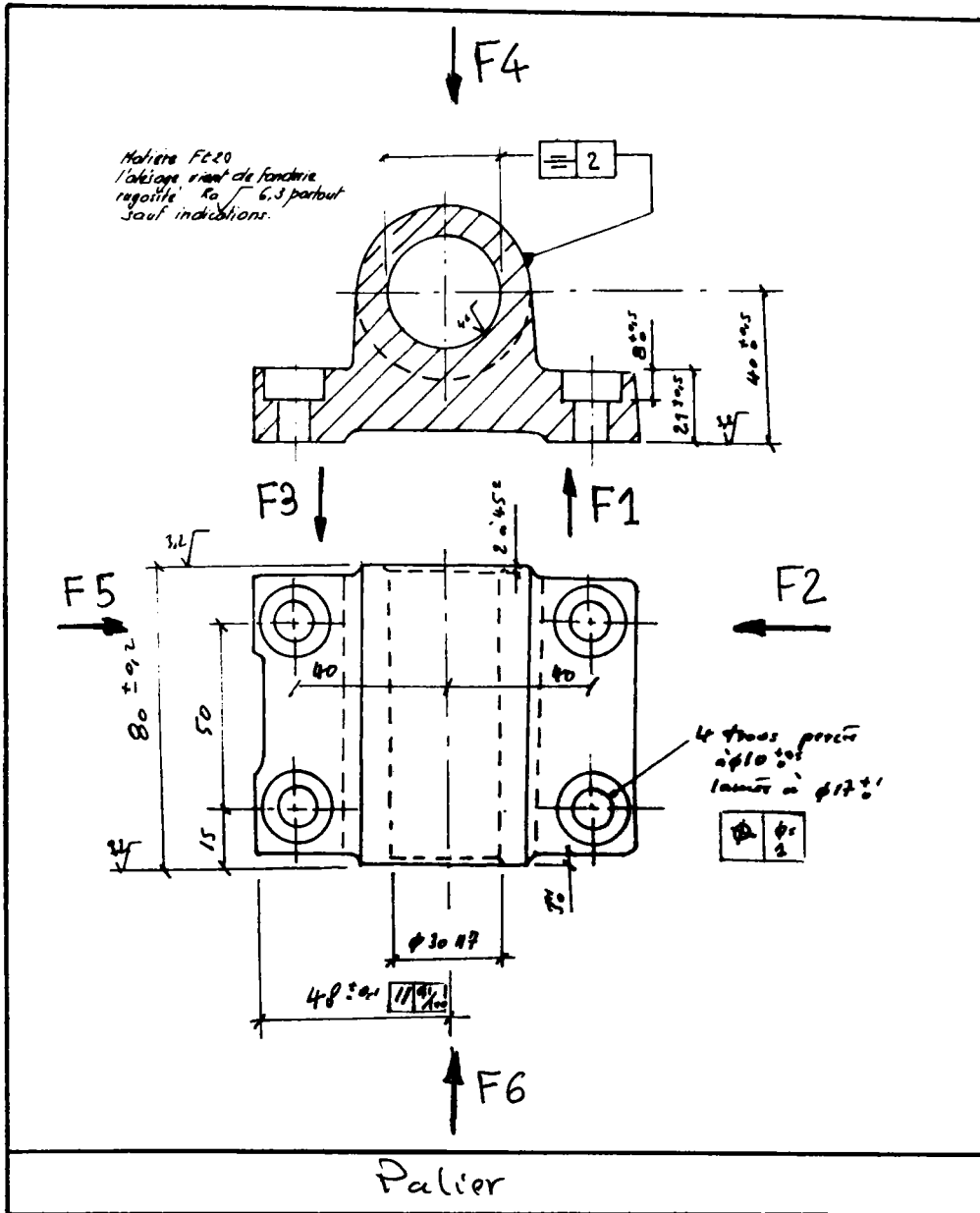
170-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e:T.a.r.a.u.d.e.r._f.1;3.4:3.4;

Gamme de l'article A : série < 10

- 10-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e;
S.u.r.f._f.1_.E.b;3.3.3:3.3.3;
- 20-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e;
S.u.r.f.a.c.e.r._f.1_.F.i;4.4.4:4.4.4;
- 30-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e;
R.a.i.n._f.1_.E.b;6.6.6:6.6.6;
- 40-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e;R.a.i.n._f.1_.F.i;7.7.7:7.7.7;
- 50-P.f.4./L.f.5./B.f.6:F.r.a.i.s.a.g.e;R.a.i.n._f.1_.F.i;8.8.8:8.8.8;
- 60-:E.b.a.v.u.r.a.g.e;;
- 70-:C.o.n.t.r.ô.l.e;;
- 80-P.f.1./V.f.4./B.f.6:T.o.u.r.n.a.g.e;D.r.e.s._f.6_.E.b;1.1:1.1;
- 90-P.f.1./V.f.4./B.f.6:T.o.u.r.n.a.g.e;D.r.e.s._f.6_.F.i;1.3:1.3;
- 100-P.f.1./V.f.4./B.f.6:T.o.u.r.n.a.g.e;A.l.e.s._f.6_.E.b;1.4:1.4;
- 110-P.f.1./V.f.4./B.f.6:T.o.u.r.n.a.g.e;A.l.e.s.e.r._1./2.F.i;1.6:1.6;
- 120-P.f.1./V.f.4./B.f.6:T.o.u.r.n.a.g.e;A.l.e.s.e.r._F.i;1.8:1.8;
- 130-P.f.1./V.f.4./B.f.6:T.o.u.r.n.a.g.e;C.h.a.n.f.r.e.i.n.e.r;1.9:1.9;
- 140-C.t...l.g...i.n.t././B.f.6:T.o.u.r.n.a.g.e;
D.r.e.s._f.3_.E.b;2.0:2.0;
- 150-C.t...l.g...i.n.t././B.f.6:T.o.u.r.n.a.g.e;
D.r.e.s._f.3_.F.i;2.1:2.1;
- 160-C.t...l.g...i.n.t././B.f.6:T.o.u.r.n.a.g.e;
C.h.a.n.f.r.e.i.n.e.r;2.2:2.2;
- 170-C.o.n.t.r.ô.l.e.;;
- 180-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e;P.e.r.c.e.r._f.4;2.3:2.3;
- 190-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e;L.a.m.e.r._f.4;2.8:2.8;
- 200-P.f.1./L.f.2./B.f.3:P.e.r.c.a.g.e;P.e.r.c.e.r._f.5;2.9:2.9;
- 210-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e;L.a.m.e.r._f.5;3.1:3.1;
- 220-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e;T.a.r.a.u.d.e.r._f.5;3.2:3.2;
- 230-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e;P.e.r.c.e.r._f.1;3.3:3.3;
- 240-P.f.1./L.f.2./B.f.6:P.e.r.c.a.g.e;T.a.r.a.u.d.e.r._f.1;3.4:3.4;

b) Dessin de définition de l'article B :
(figure IV.8)

Le dessin de définition représente un palier. Nous avons précisé les différentes faces principales ainsi que le code correspondant.



Article B

Code 8-3-1-3-1

Figure IV.8

Gamme de l'article B : série > 5

10-P.f.4./L.f.5./B.f.6:Fraisage:Surf._f.1_.Eb;3.3.3.3.3;
20-P.f.4./L.f.5./B.f.6:Fraisage:
Surfacer._f.1_.Fi;4.4.4.4.4;
30-P.f.1./L.f.2./B.f.3:Fraisage:Surf._f.5_.Fi;5.5.5.5.5;
40-Eb.avurage;;
50-Contrôle;;
60-P.f.1./L.f.5./B.f.3:Fraisage:
Surf._f.3.f.6_.Simul;1.2:1.2;
70-P.f.1./L.f.5./V.f.4:Percage:Percer._f.6;2.5:2.5;
80-P.f.1./L.f.5./V.f.4:Percage:Al.es.er._Fi;2.4:2.4;
90-Contrôle;;
100-P.f.1./L.f.5./B.f.6:Percage:Percer._f.4;2.7:2.7;
110-P.f.1./L.f.5./B.f.6:Percage:Lam.er._f.4;3.6:3.6;

Gamme de fabrication de l'article B :
série > 10

10-P.f.4./L.f.5./B.f.6:Fraisage:Surf._f.1_.Eb;3.3.3.3.3;
20-P.f.4./L.f.5./B.f.6:Fraisage:
Surfacer._f.1_.Fi;4.4.4.4.4;
30-P.f.1./L.f.2./B.f.3:Fraisage:Surf._f.5_.Fi;5.5.5.5.5;
40-:Eb.avurage;;
50-:Contrôle;;
60-P.f.1./V.f.4./B.f.6:Tournage:Dres._f.6_.Eb;1.1:1.1;
70-P.f.1./V.f.4./B.f.6:Tournage:Dres._f.6_.Fi;1.3:1.3;
80-P.f.1./V.f.4./B.f.6:Tournage:Percer;1.5:1.5;
90-P.f.1./V.f.4./B.f.6:Tournage:Al.es.er._1./2.Fi;1.6:1.6;
100-P.f.1./V.f.4./B.f.6:Tournage:Al.es.er._Fi;1.8:1.8;
110-P.f.1./V.f.4./B.f.6:Tournage:Chanfrainer;1.9:1.9;
120-Contrôle;;
130-P.f.1./L.f.5./B.f.6:Percage:Percer._f.4;2.7:2.7;
140-P.f.1./L.f.5./B.f.6:Percage:Lam.er._f.4;3.6:3.6;

4.4.2 Implantation de la base de règles

L'implantation consiste à définir le fichier règles et éventuellement les fichiers FicQ, ficQE, FicQC.

a) Définition du fichier règles (figure IV.9)

Après analyses des quatre gammes précédentes et compte tenu de leurs éléments communs et de leurs particularités nous définissons un ensemble de règles qui sera mémorisé dans un fichier nommé BASE1.DAT. Ces règles sont introduites en langage interne.

Dans la partie déclarative, nous définissons une matrice dM2 pour tester si l'article possède une rainure. Pour ce test nous ne nous intéressons qu'à la valeur du digit numéro 4. Celle-ci doit être égale à 5, si l'article possède une rainure.

Dans la partie règles, nous avons défini l'ensemble des arguments suivants :

- * E1 : représente le nombre d'articles de la série à fabriquer.
- * FO[1] : demande si dans le brut, il existe déjà un trou préformé (Axe principal).
- * M1 : représente le code de l'article.

A noter que dans la règle n°3, nous avons utilisé le prédicat PR égal [1,1], dont la valeur est toujours vraie, pour réaliser une transition de type divergence de "OU" avec la règle 2..

----- Fichier REGLES -----.

```
$
dm2 = 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$
> 1,1= ( ) -> E1 :::::DEBUT<
> 2,1= e1 ( ) -> E2 :::::FIN<
> 3,1= e1 ( PRegal[1,1] ) -> E3 :Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333<
> 4,1= e3 ( ) -> E4 :Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444<
> 5,1= e4 ( /PRinc[M1,dM2] ) -> E6 :Pf1/Lf2/Bf3:Fraisage:Surf_f5_Fi:555:555<
> 6,1= e4 ( PRinc[M1,dM2] ) -> E5 :Pf4/Lf5/Bf6:Fraisage:Rain_f1_Eb:666:666<
> 7,1= e5 ( ) -> E7 :Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:777:777<
> 8,1= e7 ( ) -> E8 :Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:888:888<
> 9,1= e33 ( ) -> E9 ::Controle::<
> 10,1= e6 ( ) -> E33 ::Ebavurage::<
> 11,1= e9 ( PRinf[E1,10] ) -> E13 :Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:11:11<
> 12,1= e9 ( PRsup[E1,5] ) -> E10 :Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:12:12<
> 13,1= e13 ( ) -> E14 :Pf1/Vf4/Bf6:Tournage:Dres_f6_Fi:13:13<
> 14,1= e14 ( FQquest[1] ) E15: Pf1/Vf4/Bf6:Tournage:Aleser_f6_Eb:14:14<
> 15,1= e14 ( /FQquest[1] ) E16: Pf1/Vf4/Bf6:Tournage:Perceur:15:15<
> 16,1= e15 ( ) -> E17 :Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:16:16<
> 17,1= e16 ( ) -> E17 :Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:17:17<
> 18,1= e17 ( ) -> E18 :Pf1/Vf4/Bf6:Tournage:Aleser_Fi:18:18<
> 19,1= e18 ( ) -> E19 :Pf1/Vf4/Bf6:Tournage:Chanfreiner:19:19<
> 20,1= e19 ( ) -> E20 :Ct.Lg.int//Bf6:Tournage:Dres_f3_Eb:20:20<
> 21,1= e20 ( ) -> E21 :Ct.Lg.int//Bf6:Tournage:Dres_f3_Fi:21:21<
> 22,1= e21 ( ) -> E22 :Ct.Lg.int//Bf6:Tournage:Chanfreiner:22:22<
> 23,1= e32 ( EF[?:?:Rain_f1_Eb:?:?] ) -> E23 :Pf1/Lf2/Bf6:Percage:Perceur_f4:23:23<
> 24,1= e10 ( FQquest[1] ) -> E12 :Pf1/Lf5/Vf4:Percage:Aleser_Fi:24:24<
> 25,1= e10 ( /FQquest[1] ) -> E11 :Pf1/Lf5/Vf4:Percage:Perceur_f6:25:25<
> 26,1= e11 ( ) -> E12 :Pf1/Lf5/Vf4:Percage:Aleser_Fi:26:26<
> 27,1= e32 ( /EF[?:?:Rain_f1_Eb:?:?] ) -> E30 :Pf1/Lf5/Bf6:Percage:Perceur_f4:27:27<
> 28,1= e23 ( ) -> E24 :Pf1/Lf2/Bf6:Percage:Lamer_f4:28:28<

> 29,1= e24 ( ) -> E25 :Pf1/Lf2/Bf3:Percage:Perceur_f5:29:29<
> 30,1= e31 ( ) -> E2 :::::FIN<
> 31,1= e25 ( ) -> E26 :Pf1/Lf2/Bf6:Percage:Lamer_f5:31:31<
> 32,1= e26 ( ) -> E27 :Pf1/Lf2/Bf6:Percage:Tarauder_f5:32:32<
> 33,1= e27 ( ) -> E28 :Pf1/Lf2/Bf6:Percage:Perceur_f1:33:33<
> 34,1= e28 ( ) -> E29 :Pf1/Lf2/Bf6:Percage:Tarauder_f1:34:34<
> 35,1= e29 ( ) -> E2 :::::FIN<
> 36,1= e30 ( ) -> E31 :Pf1/Lf5/Bf6:Percage:Lamer_f4:36:36<
> 37,1= e22 ( ) -> E32 ::Controle::<
> 38,1= e12 ( ) -> E32 ::Controle::<
> 39,1= e8 ( ) -> E33 ::Ebavurage::<
```

figure IV.9 : Base de connaissances initiale.

b) Fichier FicQ

Nous rappelons que ce fichier doit contenir les questions se rapportant aux arguments de type fonction (FOquest). La réponse à ces questions ne peut être que O(ui) ou N(on). Pour notre exemple la question concerne l'argument FOquest [1].

FICHIER QUESTIONS dont les reponses
possibles sont O(oui) ou N(non).

& 1 "Est ce que le trou de l'axe principal est preforme en fonderie "

c) Fichier FicQE

Ce fichier FicQE contient les questions se rapportant aux arguments de type E, utilisés pour introduire des variables de type entier. Pour notre exemple la question concerne l'argument E1.

FICHIER DES QUESTIONS dont les reponses
sont du type entier.

& 1 " Quel est le nombre de pieces a realiser "

4.4.3 Exemple 1

L'exemple 1, concerne la fabrication d'une série de 7 articles A dont le brut possède un trou préformé. La figure IV.10 correspond à l'entrée des données par le préparateur tandis que la figure IV.11 montre les deux gammes proposées par SAGGA.


```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 3: 20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
R 4: 30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
R 6: 40 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Eb:666:666;
R 7: 50 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:777:777;
R 8: 60 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:888:888;
R 39: 70 -> :Ebavurage::;
R 9: 80 -> :Controle::;
R 12: 90 -> Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:12:12;
R 24:100 -> Pf1/Lf5/Vf4:Percage:Aleser_Fi:24:24;
R 38:110 -> :Controle::;
R 23:120 -> Pf1/Lf2/Bf6:Percage:Percer_f4:23:23;
R 28:130 -> Pf1/Lf2/Bf6:Percage:Lamer_f4:28:28;
R 29:140 -> Pf1/Lf2/Bf3:Percage:Percer_f5:29:29;
R 31:150 -> Pf1/Lf2/Bf6:Percage:Lamer_f5:31:31;
R 32:160 -> Pf1/Lf2/Bf6:Percage:Tarauder_f5:32:32;
R 33:170 -> Pf1/Lf2/Bf6:Percage:Percer_f1:33:33;
R 34:180 -> Pf1/Lf2/Bf6:Percage:Tarauder_f1:34:34;
R 35:190 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->O

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 3: 20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
R 4: 30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
R 6: 40 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Eb:666:666;
R 7: 50 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:777:777;
R 8: 60 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:888:888;
R 39: 70 -> :Ebavurage::;
R 9: 80 -> :Controle::;
R 11: 90 -> Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:11:11;
R 13:100 -> Pf1/Vf4/Bf6:Tournage:Dres_f6_Fi:13:13;
R 14:110 -> Pf1/Vf4/Bf6:Tournage:Ales_f6_Eb:14:14;
R 16:120 -> Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:16:16;
R 18:130 -> Pf1/Vf4/Bf6:Tournage:Aleser_Fi:18:18;
R 19:140 -> Pf1/Vf4/Bf6:Tournage:Chanfreiner:19:19;
R 20:150 -> Ct.Lg.int//Bf6:Tournage:Dres_f3_Eb:20:20;
R 21:160 -> Ct.Lg.int//Bf6:Tournage:Dres_f3_Fi:21:21;
R 22:170 -> Ct.Lg.int//Bf6:Tournage:Chanfreiner:22:22;
R 37:180 -> :Controle::;
R 23:190 -> Pf1/Lf2/Bf6:Percage:Percer_f4:23:23;
R 28:200 -> Pf1/Lf2/Bf6:Percage:Lamer_f4:28:28;
R 29:210 -> Pf1/Lf2/Bf3:Percage:Percer_f5:29:29;
R 31:220 -> Pf1/Lf2/Bf6:Percage:Lamer_f5:31:31;
R 32:230 -> Pf1/Lf2/Bf6:Percage:Tarauder_f5:32:32;
R 33:240 -> Pf1/Lf2/Bf6:Percage:Percer_f1:33:33;
R 34:250 -> Pf1/Lf2/Bf6:Percage:Tarauder_f1:34:34;
R 35:260 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->O

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 2: 20 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->N

figure IV.11 : Exemple 1. Gammes générées.

4.4.4 Example 2 :

L'exemple 2, concerne la recherche d'une gamme de fabrication pour un nouvel article.

L'objectif de cette exemple est de mettre en valeur le langage externe. Pour cela nous avons introduit l'article C composé d'un corps et d'un chapeau et dont le positionnement relatif, peut être réalisé, soit par clavette, soit par pige.

a) Dessin de définition de l'article C
(figure IV.12)


```

***** voici ma proposition *****
R 1: 10 -> :::DEBUT;
R 3: 20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
R 4: 30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
R 6: 40 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Eb:666:666;
R 7: 50 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:777:777;
R 8: 60 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:888:888;
R 39: 70 -> :Ebavurage::;
R 9: 80 -> :Controle::;
R 12: 90 -> Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:12:12;
R 25:100 -> Pf1/Lf5/Vf4:Percage:Percer_f6:25:25;
R 26:110 -> Pf1/Lf5/Vf4:Percage:Aleser_Fi:26:26;
R 38:120 -> :Controle::;
R 23:130 -> Pf1/Lf2/Bf6:Percage:Percer_f4:23:23;
R 28:140 -> Pf1/Lf2/Bf6:Percage:Lamer_f4:28:28;
R 29:150 -> Pf1/Lf2/Bf3:Percage:Percer_f5:29:29;
R 31:160 -> Pf1/Lf2/Bf6:Percage:Lamer_f5:31:31;
R 32:170 -> Pf1/Lf2/Bf6:Percage:Tarauder_f5:32:32;
R 33:180 -> Pf1/Lf2/Bf6:Percage:Percer_f1:33:33;
R 34:190 -> Pf1/Lf2/Bf6:Percage:Tarauder_f1:34:34;
R 35:200 -> :::FIN;

```

c) Modifications en langage externe

Cette gamme n'étant pas satisfaisante le préparateur demande sa modification en langage externe

```

Desirez vous modifier la proposition O/N -> 0
* Liste des regles de meme niveau *
Apres ->70 R9 :
Si PRinc[M1,dM3].
Alors
    &
    Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42
    Pf1/Lf5/Bf3:Fraisage:Rain_f4_Eb:43:43
    Pf1/Lf5/Bf3:Fraisage:Rain_f4_Fi:44:44
    &
    Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47
    Ct_Lg_ext//Bf3:Fraisage:Rain_Chap_f1:48:48
    Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:49:49
    Ct_Lg_ext//Bf3:Fraisage:Percer_f4:50:50.
Revenir avant ->80

```

```

Autre modification O/N -> 0
* Liste des regles de meme niveau *
Apres ->80 R11 : R12 :
Si EF[?:?:Rain_Chap_f1:?:?]*PRsup[E1,5].
Alors
    Pf1/Lf5/Bf3:Assemblage:CorpsChapClav:53:53.
Revenir avant ->90

```

Autre modification O/N ->

N


```

***** Proposition apres modification *****
10 -> ::::DEBUT;
20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
40 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Eb:666:666;
50 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:777:777;
60 -> Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:888:888;
70 -> :Ebavurage::;
80 -> ::::;
90 -> ::::;
100 -> Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42;
110 -> Pf1/Lf5/Bf3:Fraisage:Rain_f4_Eb:43:43;
120 -> Pf1/Lf5/Bf3:Fraisage:Rain_f4_Fi:44:44;
130 -> ::::;
140 -> ::::;
150 -> Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47;
160 -> Ct_Lg_ext//Bf3:Fraisage:Rain_Chap_f1:48:48;
170 -> Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:49:49;
180 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4:50:50;
190 -> ::::;
200 -> :Controle::;
210 -> Pf1/Lf5/Bf3:Assemblage:CorpsChapClav:53:53;
220 -> Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:12:12;
230 -> Pf1/Lf5/Vf4:Percage:Percer_f6:25:25;
240 -> Pf1/Lf5/Vf4:Percage:Aleser_Fi:26:26;
250 -> :Controle::;
260 -> Pf1/Lf2/Bf6:Percage:Percer_f4:23:23;
270 -> Pf1/Lf2/Bf6:Percage:Lamer_f4:28:28;
280 -> Pf1/Lf2/Bf3:Percage:Percer_f5:29:29;
290 -> Pf1/Lf2/Bf6:Percage:Lamer_f5:31:31;
300 -> Pf1/Lf2/Bf6:Percage:Tarauder_f5:32:32;
310 -> Pf1/Lf2/Bf6:Percage:Percer_f1:33:33;
320 -> Pf1/Lf2/Bf6:Percage:Tarauder_f1:34:34;
330 -> ::::FIN;

```

Desirez-vous une autre proposition O/N ->N

```

I-----I
I BASE DE CONNAISSANCES UTILISEE --> C:BASE1.DAT I
I-----I
I *****Menu Principal***** I
I-----I
I      Recherche -> R I
I      Quitte -> Q I
I      ----- I
I      choix -> Q I
I-----I
I Copyright by L.A.E.I University METZ I

```

Fin de la recherche

d) Etat de la base de règles après modification et avant mise à jour par le préparateur.

Nous remarquons que le système SAGGA a créé 15 règles supplémentaires, suite aux modifications précédentes. (Apprentissage).

----- Fichier REGLES -----.

```
$
dm2 = 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$
> 1,1= ( ) -> E1 ::::DEBUT<
> 2,1= e1 ( ) -> E2 ::::FIN<
> 3,1= e1 ( PRegal[1,1] ) -> E3 :Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333<
> 4,1= e3 ( ) -> E4 :Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444<
> 5,1= e4 ( /PRinc[M1,dM2] ) -> E6 :Pf1/Lf2/Bf3:Fraisage:Surf_f5_Fi:555:555<
> 6,1= e4 ( PRinc[M1,dM2] ) -> E5 :Pf4/Lf5/Bf6:Fraisage:Rain_f1_Eb:666:666<
> 7,1= e5 ( ) -> E7 :Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:777:777<
> 8,1= e7 ( ) -> E8 :Pf4/Lf5/Bf6:Fraisage:Rain_f1_Fi:888:888<
> 9,1= e33 ( ) -> E9 ::Controle:::<
>10,1= e6 ( ) -> E33 ::Ebavurage:::<
>11,1= e9 ( PRinf[E1,10] ) -> E13 :Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:11:11<
>12,1= e9 ( PRsup[E1,5] ) -> E10 :Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:12:12<
>13,1= e13 ( ) -> E14 :Pf1/Vf4/Bf6:Tournage:Dres_f6_Fi:13:13<
>14,1= e14 ( FQuest[1] ) E15:Pf1/Vf4/Bf6:Tournage:Ales_f6_Eb:14:14<
>15,1= e14 ( /FQuest[11] ) E16:Pf1/Vf4/Bf6:Tournage:Percer:15:15<
>16,1= e15 ( ) -> E17 :Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:16:16<
>17,1= e16 ( ) -> E17 :Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:17:17<
>18,1= e17 ( ) -> E18 :Pf1/Vf4/Bf6:Tournage:Aleser_Fi:18:18<
>19,1= e18 ( ) -> E19 :Pf1/Vf4/Bf6:Tournage:Chanfreiner:19:19<
>20,1= e19 ( ) -> E20 :Ct.Lg.int//Bf6:Tournage:Dres_f3_Eb:20:20<
>21,1= e20 ( ) -> E21 :Ct.Lg.int//Bf6:Tournage:Dres_f3_Fi:21:21<
>22,1= e21 ( ) -> E22 :Ct.Lg.int//Bf6:Tournage:Chanfreiner:22:22<
>23,1= e32 ( EF[?:?:Rain_f1_Eb:?:?] ) -> E23 :Pf1/Lf2/Bf6:Percage:Percer_f4:
                                         23:23<
>24,1= e10 ( FQuest[1] ) -> E12 :Pf1/Lf5/Vf4:Percage:Aleser_Fi:24:24<
>25,1= e10 ( /FQuest[1] ) -> E11 :Pf1/Lf5/Vf4:Percage:Percer_f6:25:25<
>26,1= e11 ( ) -> E12 :Pf1/Lf5/Vf4:Percage:Aleser_Fi:26:26<
>27,1= e32 ( /EF[?:?:Rain_f1_Eb:?:?] ) -> E30 :Pf1/Lf5/Bf6:Percage:Percer_f4: 27:27<
```

```

>28,1= e23 ( ) -> E24 :Pf1/Lf2/Bf6:Percage:Lamer_f4:28:28<
>29,1= e24 ( ) -> E25 :Pf1/Lf2/Bf3:Percage:Perceur_f5:29:29<
>30,1= e31 ( ) -> E2 ::::FIN<
>31,1= e25 ( ) -> E26 :Pf1/Lf2/Bf6:Percage:Lamer_f5:31:31<
>32,1= e26 ( ) -> E27 :Pf1/Lf2/Bf6:Percage:Tarauder_f5:32:32<
>33,1= e27 ( ) -> E28 :Pf1/Lf2/Bf6:Percage:Perceur_f1:33:33<
>34,1= e28 ( ) -> E29 :Pf1/Lf2/Bf6:Percage:Tarauder_f1:34:34<
>35,1= e29 ( ) -> E2 ::::FIN<
>36,1= e30 ( ) -> E31 :Pf1/Lf5/Bf6:Percage:Lamer_f4:36:36<
>37,1= e22 ( ) -> E32 ::Controle:::<
>38,1= e12 ( ) -> E32 ::Controle:::<
>39,1= e8 ( ) -> E33 ::Ebavurage:::<
>40,1= e33 ( PRinc[M1,dM3] ) -> E34 ::::;<
>41,1= e34 ( ) -> E35 ::::;<
>42,1= e35 ( ) -> E36 :Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42<
>43,1= e36 ( ) -> E37 :Pf1/Lf5/Bf3:Fraisage:Rain_f4_Eb:43:43<
>44,1= e37 ( ) -> E38 :Pf1/Lf5/Bf3:Fraisage:Rain_f4_Fi:44:44<
>45,1= e38 ( ) -> E39 ::::;<
>46,1= e34 ( ) -> E40 ::::;<
>47,1= e40 ( ) -> E41 :Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47<
>48,1= e41 ( ) -> E42 :Ct_Lg_ext//Bf3:Fraisage:Rain_Chap_f1:48:48<
>49,1= e42 ( ) -> E43 :Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:49:49<
>50,1= e43 ( ) -> E44 :Ct_Lg_ext//Bf3:Fraisage:Perceur_f4:50:50<
>51,1= e44 ( ) -> E45 ::::;<
>52,1= e39 * e45 ( ) -> E9 ::Controle:::<
>53,1= e9 ( EF[?:?:Rain_Chap_f1:?:?]*PRsup[E1,5] ) -> E46 :Pf1/Lf5/Bf3:
Assemblage:CorpsChapClav:53:53<
>54,1= e46 ( ) -> E10 :Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:54:54<

```

4.4.5 Modification en langage interne de la base de règles

L'exemple précédent a montré que la base de connaissances initiale établie à partir de deux articles seulement est insuffisante. Aussi après une analyse prospective des différents articles pouvant faire partie de cette famille nous complétons la base de connaissances.

Cette mise à jour concerne :

- * L'extension de la possibilité d'un ensemble "chapeau-corps" pour les pièces dont la série est inférieure à 10.
- * L'introduction de la possibilité de positionnement par piges.

a) Fichier règles

Le fichier règles est étendu à 64 règles.


```

>33,1= e27 ( ) -> E28 :Pf1/Lf2/Bf6:Percege:Percer_f1:33:33<
>34,1= e28 ( ) -> E29 :Pf1/Lf2/Bf6:Percege:Tarauder_f1:34:34<
>35,1= e29 ( ) -> E2 :::::FIN<
>36,1= e30 ( ) -> E31 :Pf1/Lf5/Bf6:Percege:Lamer_f4:36:36<
>37,1= e22 ( ) -> E32 ::Controle:::<
>38,1= e12 ( ) -> E32 ::Controle:::<
>39,1= e8 ( ) -> E33 ::Ebavurage:::<
>40,1= e33 ( PRinc[M1,dM3] ) -> E34 :::::<
>41,1= e34 ( ) -> E35 :::::<
>42,1= e35 ( ) -> E36 :Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42<
>43,1= e36 ( /FOquest[3] ) -> E37 :Pf1/Lf5/Bf3:Fraisage:Rain_f4_Eb:43:43<
>44,1= e37 ( ) -> E38 :Pf1/Lf5/Bf3:Fraisage:Rain_f4_Fi:44:44<
>45,1= e38 ( ) -> E39 :::::<
>46,1= e34 ( ) -> E40 :::::<
>47,1= e40 ( ) -> E41 :Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47<
>48,1= e41 ( /FOquest[3] ) -> E42 :Ct_Lg_ext//Bf3:Fraisage:Rain_Chap_f1:
48:48<
>49,1= e42 ( ) -> E43 :Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:49:49<
>50,1= e43 ( ) -> E44 :Ct_Lg_ext//Bf3:Fraisage:Percer_f4:50:50<
>51,1= e44 ( ) -> E45 :::::<
>52,1= e39 * e45 ( ) -> E9 ::Controle:::<
>53,1= e9 ( EF[?:?:Rain_Chap_f1:?:?]*PRsup[E1,5] ) -> E46 :Pf1/Lf5/Bf3:
Assemblage:CorpsChapClav:53:53<
>54,1= e46 ( ) -> E10 :Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:54:54<
>55,1= e9 ( EF[?:?:Rain_Chap_f1:?:?]*PRinf[E1,10] ) -> E47 :Pf1/Lf5/Bf3:
Assemblage:CorpsChapClav:55:55<
>56,1= e47 ( ) -> E13 :Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:56:56<
>57,1= e41 ( FOquest[3] ) -> E48 :Ct_Lg_ext//Bf3:Fraisage:Percer_f4_Pig:
57:57<
>58,1= e48 ( ) -> E43 :Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:58:58<
>59,1= e36 ( FOquest[3] ) -> E49 :Pf1/Lf5/Bf3:Fraisage:Percer_f4_Pig:59:59<
>60,1= e49 ( ) -> E39 :::::<
>61,1= e9 ( EF[?:?:Percer_f4_Pig:?:?]*PRinf[E1,10] ) -> E50 :Pf1/Lf5/Bf3:
Assemblage:CorpsChapPig:61:61<
>62,1= e50 ( ) -> E13 :Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:62:62<
>63,1= e9 ( EF[?:?:Percer_f4_Pig:?:?]*PRsup[E1,5] ) -> E51 :Pf1/Lf5/Bf3:
Assemblage:CorpsChapPig:63:63<
>64,1= e51 ( ) -> E10 :Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:64:64<

```

b) Fichier FicQ

La question n°3 permet de préciser si l'assemblage est réalisé par pige. Celle-ci correspond à l'argument FOquest [3].

FICHER QUESTIONS dont les reponses
possibles sont O(oui) ou N(non).

```

& 1 "Est ce que le trou de l'axe principal est preforme en fonderie "
& 2 "Question 2"
& 3 "Est-ce que l'assemblage est prévu avec des piges"

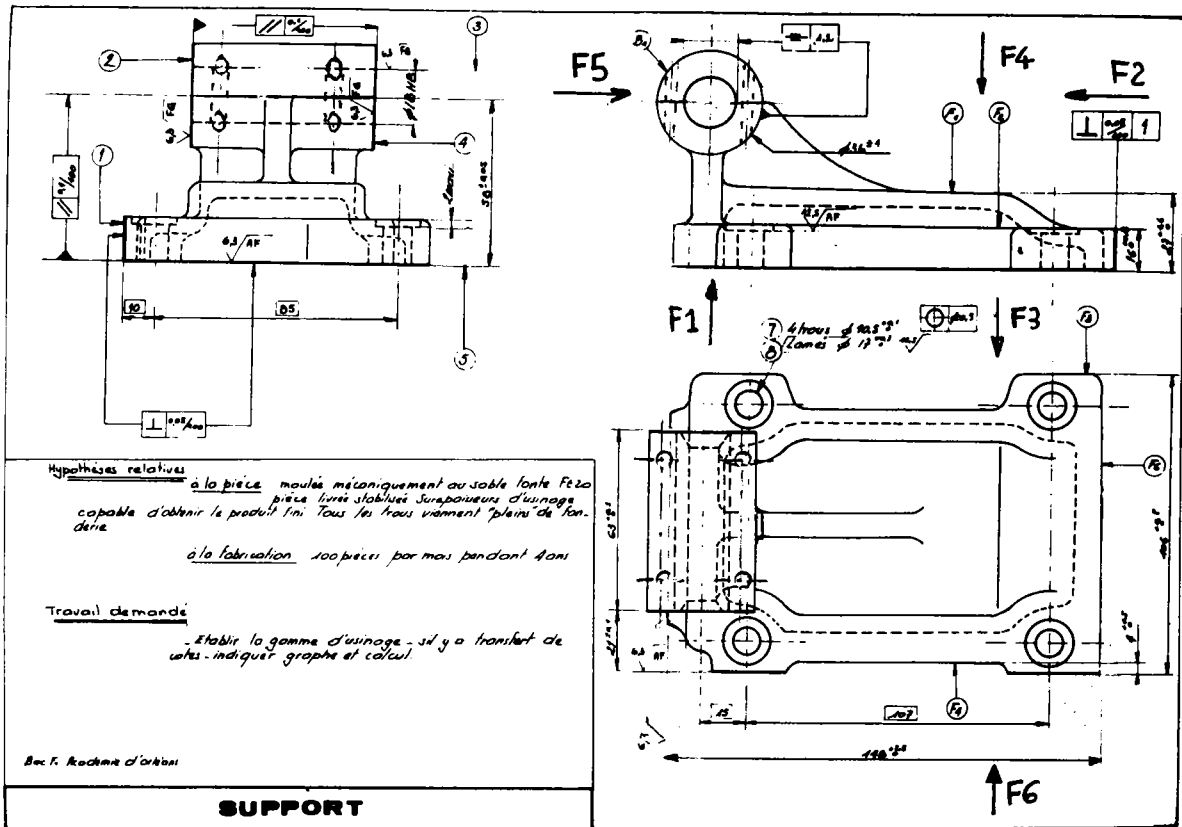
```

4.4.6 Exemple 3

Cet exemple met en évidence la possibilité de générer une gamme d'opérations pour un nouvel article D avec une proposition satisfaisante.

a) Dessin de définition de l'article C (figure IV.13)

L'article C est un nouvel article, de forme différente à celles des précédents articles, mais appartenant à la même famille selon le code.



Article C : code 8-4-5-3-1

Figure IV.13


```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 3: 20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
R 4: 30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
R 5: 40 -> Pf1/Lf2/Bf3:Fraisage:Surf_f5_Fi:555:555;
R 10: 50 -> :Ebavurage::;
R 40: 60 -> ::::;
  Debut bloc ET numero 1
R 46: 70 -> ::::;
R 47: 80 -> Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47;
R 57: 90 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4_Pig:57:57;
R 58:100 -> Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:58:58;
R 50:110 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4:50:50;
R 51:120 -> ::::;
R 41:130 -> ::::;
R 42:140 -> Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42;
R 59:150 -> Pf1/Lf5/Bf3:Fraisage:Percer_f4_Pig:59:59;
R 60:160 -> ::::;
  Fin bloc ET numero 1
R 52:170 -> :Controle::;
R 63:180 -> Pf1/Lf5/Bf3:Assemblage:CorpsChapPig:63:63;
R 64:190 -> Pf1/Lf5/Bf3:Fraisage:Surf_f3f6_Simul:64:64;
R 25:200 -> Pf1/Lf5/Vf4:Perceage:Percer_f6:25:25;
R 26:210 -> Pf1/Lf5/Vf4:Perceage:Aleser_Fi:26:26;
R 38:220 -> :Controle::;
R 27:230 -> Pf1/Lf5/Bf6:Perceage:Percer_f4:27:27;
R 36:240 -> Pf1/Lf5/Bf6:Perceage:Lamer_f4:36:36;
R 30:250 -> ::::FIN;
```

Desirez-vous une autre proposition O/N ->O

Desirez vous modifier la proposition O/N -> N

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 3: 20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
R 4: 30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
R 5: 40 -> Pf1/Lf2/Bf3:Fraisage:Surf_f5_Fi:555:555;
R 10: 50 -> :Ebavurage::;
R 40: 60 -> ::::;
  Debut bloc ET numero 1
R 46: 70 -> ::::;
R 47: 80 -> Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47;
R 57: 90 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4_Pig:57:57;
R 58:100 -> Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:58:58;
R 50:110 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4:50:50;
R 51:120 -> ::::;
R 41:130 -> ::::;
R 42:140 -> Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42;
R 59:150 -> Pf1/Lf5/Bf3:Fraisage:Percer_f4_Pig:59:59;
R 60:160 -> ::::;
  Fin bloc ET numero 1
R 52:170 -> :Controle::;
R 61:180 -> Pf1/Lf5/Bf3:Assemblage:CorpsChapPig:61:61;
R 62:190 -> Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:62:62;
R 13:200 -> Pf1/Vf4/Bf6:Tournage:Dres_f6_Fi:13:13;
R 15:210 -> Pf1/Vf4/Bf6:Tournage:Percer:15:15;
R 17:220 -> Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:17:17;
R 18:230 -> Pf1/Vf4/Bf6:Tournage:Aleser_Fi:18:18;
R 19:240 -> Pf1/Vf4/Bf6:Tournage:Chanfreiner:19:19;
R 20:250 -> Ct.Lg.int//Bf6:Tournage:Dres_f3_Eb:20:20;
R 21:260 -> Ct.Lg.int//Bf6:Tournage:Dres_f3_Fi:21:21;
R 22:270 -> Ct.Lg.int//Bf6:Tournage:Chanfreiner:22:22;
R 37:280 -> :Controle::;
R 27:290 -> Pf1/Lf5/Bf6:Perceage:Percer_f4:27:27;
R 36:300 -> Pf1/Lf5/Bf6:Perceage:Lamer_f4:36:36;
R 30:310 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->O

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 2: 20 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->N

4.4.7 Exemple 4

Cet exemple est identique à l'exemple 3, à la différence que l'on assujettit la recherche à une contrainte technologique relative à la fabrication. Elle spécifie que l'on ne peut réaliser plusieurs surfaçages simultanés.

```
I-----I
I BASE DE CONNAISSANCES UTILISEE --> C:BASE1.DAT I
I-----I
I *****Menu Principal***** I
I-----I
I Recherche -> R I
I Quitte -> Q I
I ----- I
I choix -> R I
I-----I
I Copyright by L.A.E.I University METZ I
```

Est-ce que la recherche est assujettie a des contraintes 0/N ->0

----- Definition des contraintes :Tapez <R.C> si fin -----

De la forme ->" [act. ou ?:act. ou ?:act. ou ?:act. ou ?:act. ou ?]"

Chaque action < 15 caracteres.

- 1:[?:?:Surf_f3f6_Simul:?:?]
- 2:
- 3:
- 4:
- 5:
- 6:
- 7:
- 8:
- 9:
- 10:

***** Matrice numero = 1*****

	I	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
3	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
4	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
5	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
6	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
7	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
8	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
9	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--	--1--
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Je vous pose la question suivante ->
"Est-ce que l'assemblage est prévu avec des piges"

reponse 0/N ->0

cette question reclame comme reponse un nombre entier

" question 1 Quel est le nombre de pieces a realiser " :
Reponse ->8

Je vous pose la question suivante ->
"Est ce que le trou de l'axe principal est preforme en fonderie "

reponse O/N ->N

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 3: 20 -> Pf4/Lf5/Bf6:Fraisage:Surf_f1_Eb:333:333;
R 4: 30 -> Pf4/Lf5/Bf6:Fraisage:Surfacer_f1_Fi:444:444;
R 5: 40 -> Pf1/Lf2/Bf3:Fraisage:Surf_f5_Fi:555:555;
R 10: 50 -> :Ebavurage::;
R 40: 60 -> ::::;
  Debut bloc ET numero 1
R 46: 70 -> ::::;
R 47: 80 -> Ct_Lg_ext//Bf3:Fraisage:Surf_Chap_f1:47:47;
R 57: 90 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4_Pig:57:57;
R 58:100 -> Ct_Lg_ext//Bf3:Fraisage:Lamer_f4:58:58;
R 50:110 -> Ct_Lg_ext//Bf3:Fraisage:Percer_f4:50:50;
R 51:120 -> ::::;
R 41:130 -> ::::;
R 42:140 -> Pf1/Lf5/Bf3:Fraisage:Surf_f4_Fi:42:42;
R 59:150 -> Pf1/Lf5/Bf3:Fraisage:Percer_f4_Pig:59:59;
R 60:160 -> ::::;
  Fin bloc ET numero 1
R 52:170 -> :Controle::;
R 61:180 -> Pf1/Lf5/Bf3:Assemblage:CorpsChapPig:61:61;
R 62:190 -> Pf1/Vf4/Bf6:Tournage:Dres_f6_Eb:62:62;
R 13:200 -> Pf1/Vf4/Bf6:Tournage:Dres_f6_Fi:13:13;
R 15:210 -> Pf1/Vf4/Bf6:Tournage:Percer:15:15;
R 17:220 -> Pf1/Vf4/Bf6:Tournage:Aleser_1/2Fi:17:17;
R 18:230 -> Pf1/Vf4/Bf6:Tournage:Aleser_Fi:18:18;
R 19:240 -> Pf1/Vf4/Bf6:Tournage:Chanfreiner:19:19;
R 20:250 -> Ct.Lg.int//Bf6:Tournage:Dres_f3_Eb:20:20;
R 21:260 -> Ct.Lg.int//Bf6:Tournage:Dres_f3_Fi:21:21;
R 22:270 -> Ct.Lg.int//Bf6:Tournage:Chanfreiner:22:22;
R 37:280 -> :Controle::;
R 27:290 -> Pf1/Lf5/Bf6:Perceage:Percer_f4:27:27;
R 36:300 -> Pf1/Lf5/Bf6:Perceage:Lamer_f4:36:36;
R 30:310 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->O

```
***** voici ma proposition *****
R 1: 10 -> ::::DEBUT;
R 2: 20 -> ::::FIN;
```

Desirez vous modifier la proposition O/N -> N

Desirez-vous une autre proposition O/N ->N

```
I-----I
I BASE DE CONNAISSANCES UTILISEE --> C:BASE1.DAT I
I-----I
I *****Menu Principal***** I
I-----I
I Recherche -> R I
I Quitte -> Q I
I ----- I
I choix -> Q I
I-----I
Copyright by L.A.E.I University METZ
```

CHAPITRE V

Conclusion

La génération automatique des gammes de fabrication, interface dans le système de production entre le système de conception assistée par ordinateur et la fabrication automatisée, est un problème qui n'a pas trouvé à ce jour de solutions satisfaisantes. Pour réaliser cette tâche délicate du bureau des méthodes, les techniques actuellement utilisées sont :

- * La génération de gammes par variantes basée sur le concept de technologie de groupe. Ces techniques nécessitent la définition de familles de produits et de gammes standards. Elles ne sont pas capables de générer véritablement de nouvelles gammes. Ces méthodes s'appliquent principalement à des productions diversifiées stables dans le temps et présentent des variantes mineures. Elles sont très lourdes à implanter et manquent de flexibilité pour les adapter à d'autres produits et d'autres entreprises.

- * La définition d'un modèle informatique des procédés de fabrication correspondant à la pièce la plus complexe de la famille. Dans ces techniques la gamme correspondant à une nouvelle pièce est un sous-ensemble du modèle de la famille. Ces familles regroupent des produits qui ont même fonction et même forme. Les logiciels sont spécifiques à chaque famille et une modification du modèle entraîne la réécriture d'une partie du logiciel. Cette méthode s'applique surtout aux familles stables de moyennes séries.
- * L'utilisation de l'intelligence artificielle basée généralement sur l'emploi des systèmes experts générateur de plan. Cette approche très séduisante se heurte cependant au problème général du développement des systèmes experts qui sont non adaptatifs, hautement spécifiques et non flexibles.

Dans ce rapport nous présentons un nouveau type de systèmes de génération automatique de gammes de fabrication, qui comprend trois niveaux :

1. Reconnaissance de familles de produits par des méthodes de classification automatique basées sur la notion de mesure de ressemblance.
2. Définition d'une base de connaissances primitive par famille, représentée par un Grafcet modifié.
3. Utilisation de ces bases de connaissances par famille par un système expert particulier, pour concevoir les gammes de fabrication de nouveaux produits et renforcer ces bases de connaissances par apprentissage avec maître, à l'aide d'un langage externe.

Notre étude a porté principalement sur les niveaux 2 et 3, et a donné lieu à l'écriture d'un logiciel de génération de gammes : SAGGA. Au niveau 2, l'ensemble des gammes de fabrication appartenant à chaque famille est représenté par un graphe orienté où les sommets sont des opérations et les branches les conditions géométriques et techniques de réalisation de ces opérations. Ce graphe définit le modèle de fabrication des produits d'une famille. D'un point de vue mathématique, ce modèle est formalisé comme un réseau de Pétri particulier monovalué où les opérations sont les étapes et les conditions de fabrication les transitions. Les propriétés de ce réseau appelé Grafcet M sont déduites de celles du GRAFCET qui est utilisé pour la commande de systèmes séquentiels. Le graphe représentant le modèle de fabrication est défini par une suite de règles de production codées, qui constituent la représentation interne de la base de connaissances associée à chaque famille.

Au niveau 3, cette base de connaissances est exploitée par un mécanisme d'inférence permettant une recherche en chaînage arrière avec backtracking.

Le système expert génère alors la ou les gammes de fabrication correspondant au nouveau produit, à partir de la base de faits composée du code du produit et des renseignements complémentaires donnés par le préparateur. La gamme est alors complétée à l'aide d'un éditeur de texte spécialisé interactif, qui tient compte de contraintes technologiques de production.

Si la gamme proposée n'est pas satisfaisante, le préparateur modifie celle-ci à l'aide de règles en langage externe qui permettent la mise à jour de la base de connaissances. Des exemples sur des pièces industrielles prismatiques montrent les particularités du système SAGGA qui présentent les avantages suivants :

- * Utilisation d'une organisation à technologie de groupe, qui assure une rationalisation de la production.
- * Définition d'une base de connaissances en partie modulaire, ou partie structurée.
- * Utilisation des règles de fabrication mécanique non génériques, faciles à définir, ce qui rend plus aisée la mise à jour progressive de la base de connaissances.
- * Génération de toutes les gammes possibles.
- * Implantation du système sur un micro-ordinateur.

Par ailleurs la partie génération de gammes peut-être utilisée seule hors contexte système de codification de technologie de groupe.

Dans ce cas il suffit de définir une base de connaissances portant sur les caractéristiques de la pièce brute et finie qui définira directement les procédés technologiques de fabrication. Mais dans ce cas la définition de cette base de connaissances sera beaucoup plus délicate.

De même cette partie utilisant les techniques des systèmes experts pourrait aisément s'adapter selon nous à la génération de gammes d'assemblage, à la génération de programme très structuré, tels que les post-processeurs pour machines à commande numérique ainsi qu'à la simulation de processus discret. Dans cette deuxième application il faudrait revoir le mécanisme d'inférence en supprimant le chaînage arrière.

Enfin l'éditeur de bordereau de fabrication seul, par ses possibilités de saisies interactives, recherche des données technologiques, calcul des temps, de conditions de coupe, ... est en lui-même un logiciel d'aide à l'édition des gammes, qui supprime les tâches répétitives sans intérêt.

ANNEXE A

1. GLOSSAIRE

Article : a

C'est un élément physique et indécomposable, du point de vue de la fabrication, qui entre dans la composition d'une nomenclature.

Produit : Pi

Un produit est un ensemble d'articles qui compose entièrement une nomenclature. Il peut être vendu en l'état par le centre de production (prêt à l'utilisation par le client).

Famille d'article : FAi

Ensemble d'articles proches au sens des valeurs prises par les paramètres qui interviennent en fabrication (formes géométriques, méthodes de fabrication...).

Famille de produits : FPi

Ensemble de produits proches au sens des valeurs prises par les paramètres qui interviennent en gestion de production (temps, coût, ...).

Type de poste de travail : TP

Sous ensemble d'un système physique de production dont tous les éléments :

- * peuvent exécuter les mêmes opérations standards,
- * ont des caractéristiques de fabrication équivalentes ou voisines.

Opération élémentaire : OE

Tâche élémentaire qui a pour but de rapprocher l'article de son état final, et de ce fait engendre une plus value sur l'article.

- ex :
- * Action d'un outil sur une surface.
 - * Suite de mouvements concourant à la réalisation d'un travail précis et indécomposable sur un article.

Opération standard : OS

Ensemble d'opérations élémentaires de même type.

- ex :
- * Chariottage de a.
 - * Chariottage de b.

Attention ne pas conclure qu'une opération standard est l'ensemble des opérations élémentaires réalisées avec un même outil, sur une même pièce. Car il faut envisager le cas d'un outil multifonction.

Gamme de fabrication : G

Toute gamme est définie comme étant une suite ordonnée d'opérations. La nature des opérations indique le type de la gamme de fabrication.

Exemples : Gamme d'usinage, gamme d'assemblage, gamme de contrôle.

On peut aussi remonter les gammes mixtes, c'est à dire qu'elles incluent des types différents d'opérations.

Nomenclature

Liste des articles qui composent un produit. Cette liste est structurée en niveaux dans laquelle un article apparaît autant de fois qu'il est utilisé.

Plan directeur de production

Plan glissant à moyen terme qui exprime les prévisions de production établies en quantités par produit sur une période déterminée.

Programme de fabrication

Programme glissant à court terme, dont l'objectif est de traduire le plan directeur de production. Il exprime sur une période désirée, les quantités d'articles à fabriquer sur des types de poste de travail définis.

Ordonnancement d'atelier

Affectation dans le temps des postes de travail disponibles, aux opérations standards, de façon à réaliser le programme de fabrication.

Long terme

Période de temps nécessaire pour étudier et mettre en place un système de production nouveau, ou pour remanier en profondeur un système de production existant.

Le long terme intéresse la politique de l'entreprise.

Entrées

Sorties

- | | |
|--------------------------|---------------------|
| - Etude de marché | - Moyens en homme |
| - Capacité financière | - Moyens en machine |
| - Données technologiques | - Part du marché |
| - Données techniques | |
| - Ressources humaines | |

Moyen terme

Période de temps nécessaire à l'ajustement des postes disponibles aux besoins réels ou évalués par famille de produits.

L'objectif est d'entraîner une simplification ainsi qu'une amélioration des produits existants en vue d'obtenir une meilleure rentabilité.

Le moyen terme intéresse la stratégie de l'entreprise.

Entrées

Sorties

- | | |
|------------------------------|---------------------|
| - Programme de vente | - Plan directeur |
| - Contraintes financières | - Approvisionnement |
| - Contraintes humaines | |
| - Contraintes technologiques | |
| - Contraintes techniques | |

Court terme

Période de temps sur laquelle on distribue les tâches par rapport au temps, en tenant compte des capacités du système de production.

Le court terme intéresse la tactique de l'entreprise.

Entrées

Sortie

- | | |
|---------------------------------------|------------------|
| - Commande ferme (interne ou externe) | - Ordonnancement |
| - Moyens disponibles | général |
| - Capacités d'approvisionnement | |

Très court terme

Période dans le temps sur laquelle on affecte dans le détail les moyens aux tâches, à partir du programme de fabrication.

Le très court terme intéresse la marche de l'entreprise.

Entrées

Sorties

- | | |
|-----------------------------------|------------------|
| - Programme de fabrication | - Ordonnancement |
| - Approvisionnement à court terme | d'atelier |
| - Ordonnancement général | |
| - Postes disponibles | |
| - Aléas de la production | |

ANNEXE B

1 . * Un réseau est sauf pour un marquage initial noté Mo (état initial du système), si au cours de son évolution aucune étape ne peut posséder plus d'un jeton. Les étapes du réseau correspondent ainsi à des conditions logiques. Ceci permet de vérifier qu'une condition logique vraie (étape activée) a été remise à faux (étape désactivée) après utilisation.

* Un réseau est vivant, pour un marquage initial Mo, si au cours de son évolution, toute transition est franchissable. Ceci permet de s'assurer que le système ne puisse se bloquer.

* Un conflit indique la présence d'une décision à prendre. C'est à dire lorsque plusieurs transitions sont sensibilisées avec une étape précédente commune, qu'une seule transition sera tirée et que le choix de l'une d'elles sera arbitraire. Ceci provient de la notion d'indévisibilité du jeton, ce qui interdit toute confusion entre le "ou exclusif" et le "et logique".

BIBLIOGRAPHIE

1. Practical Methods and Techniques, new ventures in CAD/CAM
Matyas Horvath : Budapest Technical University
Budapest - Hungary
Andras Markus : Computer and Automation Institut
Budapest - Hungary
1983
2. Development of Manufacturing Logic and its use in generative process planning.
Edward J. Adlard - Metcut Research Associates Inc.
3. Contribution à l'étude de la reconnaissance de familles de pièces en fabrication mécanique.
Minot - Thèse de docteur Ingénieur - Université Metz - 1983
4. Manufacturing planning evolution with artificial intelligence with application toward machining operations.
William J. Zdeblick : Metcut Research Associates Inc. Cincinnati, OH. USA
Brian E. : Owen Graduate School of Management - Vanderbilt University - Nashville, TN. USA
5. An interactive Process Planning System for Prismatic Parts (ICAPP) -
H. Eskicioglu, B.J. Davies - Department of Mechanical Engineering UNITS
Annals of the CIRP Vol 32/1/1983

6. Computer - Aided Process Planning -
Dell K. Allen, Paul R. Smith - Computer Aided Manufacturing Laboratory -
Brigham Young University - Provo, Utah 84602 - October 15, 1980
7. SAVEGAM, logiciel d'élaboration et de valorisation des gammes de
fabrication - M. Brun : Weco.SA - M. Moriceau : Save
8. EXCAP - The use of expert systems in process-planning -
B.J. Davies, I.L. Darbyshire - Annals of the CIRP Vol 33/1/1984
9. Process Plan Representation and Manipulation in generic expert planning
systems - J.P. Tsaucy, Y. Lagoude - LIFIA (Imag) B.P 68 38402 Saint
Martin d'Hères, France
10. Représentation et exploitation de connaissances expertes en génération
de plans - Y. Descotte - Thèse 3è cycle 1981 - INPG - Grenoble, France
11. Application d'un Système Expert à la Génération de Gammes de
Fabrication de Produits en Caoutchouc -
Norberto R. Iudica - Battelle Institut e.v. - Am Römerhof 35, D-6000
Frankfurt 90 - Federal Republic of Germany
12. GT and Capp Cut Work-in-Process Time 80% -
D. Holtz - Editor MANAGEMENT
13. Les SIAD : outil, concepts et mode d'action -
J.C. Courbon - Université de Genève - AFCET INTERFACES - Juillet 1983
14. Méthode de reconnaissances de groupements technologiques et de familles
de pièces.
Y.Lemoine, B.Mutel - Journée ADEPA 1982
15. AUTOGAM - J. Crapart - CETIM information n°88 Déc 84

16. Implantation Assistée par Ordinateur de la Technologie de Groupe -
J. Minot, B. Mutel - Congrès AFCET Automatique - Besançon Nov 83
17. Recherche des dépendances entre typologie de production, de pilotage et
niveau de planification - C. Pourcel - Automatique Innovation 1985
18. Analyse de quelques caractéristiques pouvant intervenir dans une
typologie des systèmes de gestion de production -
F. Roubellat - L.A.A.S. - CNRS France - Septembre 1985
19. Le Système de Production Automatisée Flexible : L'assistance
informatique pour leur conception -
C. Dupont-Gatelmand, J.P. Vignnot - Renault Machines Outils -
Centre Parly 2 - B.P. 70 - 78152 Le Chesnay, France
20. Structuration des bases de données techniques autour de nomenclatures
et gammes mères en vue de l'élaboration automatique de gammes -
J.P. Campagne, J. Peyron, M. Temani, G.R.A.S.P., INSA -
Villeurbanne Cédex 69621, France
21. Base de données pour la gestion d'un atelier flexible -
P. Baptiste, J. Faurel - GRECO - CNRS - INSA, Villeurbanne Cédex, France
22. OPITZ - Klassifizierungssystem -
Industrie Anzeiger, 87, 1965 Technische Hochschule - Aachen
23. Typologie de la production et des systèmes associés -
G. Achard, F. Bonneau, J.M. Proth - F.W.E.G.E. 5 Juin
24. Gestion de la production : Terminologie A.F.W.O.R.
25. Proposition de termes et de définitions concernant "CAM" -
CIRP - STC 0 - Working Group on Manufacturing Systems - 1977

26. Guide du Bureau des Méthodes - G. Bérangée - Edition Des Forges
27. Précis méthodes d'usinage - Nathan - A.F.N.O.R.
28. Première approche pour un langage de gestion -
J.M. Proth - INRIA Nancy 54500 Vandoeuvre - F.N.E.G.E. 29 Novembre 1985
29. An introduction to the DCLASS System Computer Aided Manufacturing
Laboratory BRIGHAM Young University Provo, Utah 1982
30. Analyse et simulation en langage APL de système de commande décrits
par des réseaux de Pétri -
J. Renacher - Thèse de docteur de spécialité EEA - Toulouse 1977
31. Application of Petri net based models in the modelling and analysis of
Flexible manufacturing systems -
Manjunath Kamath, N. Viswanadham - IEEE 1986
32. Les réseaux de Pétri et leurs applications -
G. Vidal Naquet - AFCET/INTERFACES n°43 1986
33. Synthèse programmée des automatismes logiques décrits par réseaux de
Pétri : Une méthode de mise en oeuvre sur microcalculateur -
M. Silva, R. David - R.A.I.R.O. Automatique/Systems Analysis and Control
Vol 13 n°4 1979
34. Conception des systèmes séquentiels totalement autotestables à partir
des réseaux de Pétri -
J. Marin, C. André, F. Boeri - R.A.I.R.O. Automatique Vol 10 n°11 1976
35. Réseaux de Pétri : Théorie et Analyse - G.W. Brams - Edition : Masson
36. Réseaux de Pétri : Modélisation et applications -
G.W. Brams - Edition : Masson

37. Réseaux de Pétri Synchronisés -
M. Moalla, J. Pulou, J. Sifakis - E.N.S. d'informatique et de
mathématiques appliquées - R.R. n°80 - Septembre 1977
38. Réseaux de Pétri et automates programmables -
J.M. Toulotte - Automatisation - Juillet, Août 1978
39. Le Grafcet - M. Blanchard - Edition Cepadues
40. Grafcet et logique industrielle programmée -
S. Theilliez, J.M. Toulotte - Editions Eyrolles
41. Interprétations algébriques et algorithmiques et temporisations -
Par le groupe Systèmes logiques de l'AFCEC - Le nouvel Automatisation -
Septembre 1983
42. Rapport de la commission de normalisation de la représentation du
cahier des charges d'un automatisation logique - Par le groupe Systèmes
logiques de l'AFCEC Automatisation - Mars, Avril 1978
43. Automatismes logiques : Grafcet ou réseau de Pétri -
M. Blanchard - Le nouvel Automatisation - Mai 1979
44. Etude comparative de deux outils de représentation : Grafcet et Réseau
de Pétri - R. Valette - Le nouvel Automatisation - Décembre 1978
45. Automatisation des équipements de production - J.P. Frachet -
Support de cours - ISMCM - 1985
46. Principles of artificial Intelligence -
Nils J., Nilsson - Tioga Publishing Co
47. Méthode générale d'analyse d'une application informatique -
X. Castellani - Edition Masson

48. d Base II - SGBD relationnel - Alshton - Tate
49. An introduction to automated Process planning Systems -
Tien-Chien Chang, R.A. Wysk - Editors Prentice Hall
50. The winter Annual Meeting of the American Society of Mechanical
Engineers - Miami Beach, Florida November 17-22 - 1985 - Editor ASME
51. Computer Aided Design and Manufacturing -
V. Rembold, R. Dillmann - Editor Springer Verlag
52. Process-Planning Program report of CAM-I 1979 - CAM-I
53. A system of computer-aided process-planning for machine parts -
K.H. Tempelhof - Advanced Manufacturing Technology. IFIP 1980
54. Semi-generative process-planning for part manufacturing -
M. Horwath - Detroit - USA - PROLAMAT 79
55. The TAUPROG System Family. Application experiences and new results of
development - T. Toth, D. Wadasz - Paris - PROLAMAT 85
56. Automated coding and process selection - Ch. Emerson - ACAPS 9th North
A. Manufacturing Res. Conf. Mai 81
57. TOSICAP - A system of computer-aided process-planning system for
rotational parts. S. Zhang, W.D. Gao - Annals of the CIRP Vol 33/1/1984
58. Integrated generation of drawing, process plans and M.C. Tapes
W. Eversheim, H. Fuchs - AUTAP Detroit - PROLAMAT 79
59. XPLANE, a generative computer-aided process-planning system for part
manufacturing - A.M. Erve, H.J.J. Kals - Annals of the CIRP Vol 35/1/86

60. OPEX - An expert system for CAPP - M. Ganes and all -
Institut "Sozef Stefan" - Lubiana. Y.
61. The scape and limitations of expert system technology - D. Partridge -
6th International Workshop on Expert Systems - Avignon 86
- 62 Contribution à la classification automatique de données de production -
A. Nadif - Thèse de doctorat - Université de Metz - 1987
63. Contribution of data analysis method to group technology implantation -
A. Bruyand, B. Mutel - Congrès IFIP - PROLAMAT 85 - Paris
64. Etude d'un système interactif d'aide à la décision -
M. Spadoni - DEA Production Automatisée - Oct 83
- 65 Control Structure in expert system -
J.P. Laurent - T.S.I. Vol 3 n°3 1984
66. Les moteurs de systèmes experts : Typologie, comparaisons et guide du
concepteur - J.P. Laurent - Rapport - Université de Savoie
73011 Chambéry, France
67. Knowledge representation - J.L. Laurière - T.S.I. Vol 1 n°2, 1982
68. Experts systems - J.L. Laurière - T.S.I.
69. Applications de l'intelligence artificielle : Les systèmes experts -
A. Bonnet - R.A.I.R.O. Informatique/Computer Science Vol 15 n°4, 1981
70. Etude des contradictions entre les données dans les systèmes de
diagnostic - J.G. Ganascia - Rapport L.R.I.
Université Paris Sud 91405 Orsay, France
71. Représentation des connaissances dans les systèmes experts -
S. Pinson - R.A.I.R.O. Informatique/Computer Vol 15 n°4, 1981

72. Decision support using causation knowledge base -
K. Nakamura, S. Iwai, T. Sawaragi - I.E.E.E. n°6, 1982
73. Design of a question-answering system for data base interfaces using
similarity knowledge bases - K. Nakamura - Ph D dissertation,
Kyoto University, 1983
74. L'intelligence artificielle dans la génération de plans - F. Giannesini
Le nouvel automatisme - Décembre 1980
75. Intelligence artificielle et construction de modèles dynamiques -
F. Rechenmann - I.A. et Productique - Novembre 1984
76. La logique mathématique et l'intelligence artificielle - J. Fargues -
AFCET/INTERFACES n°44, Juin 1986
77. Syntax and semantics of programming languages versus CAD/CAM
Requirements : The roles of artificial intelligence languages -
J.P. Crestin - Rapport E.N.S.E.E.I.H. 31071 - Toulouse, France
78. Knowledge - Based Facility Design - Edward L. Fisher, Shimon Y. Nof -
Institute of Industrial Engineers 1984 - Annual International Industrial
Engineering Conference Proceedings
79. Manufacturing Planning evolution with artificial intelligence with
application toward machining operations -
William J. Zdeblick - Manufacturing Tech. Div. Metcut Research
Associates Inc. Cincinnati, Ohio - USA, Brian E. Barkocy - Owen Graduate
School of Management Vanderbilt University Nashville, Tennessee - USA
80. Systèmes industriels d'intelligence artificielle - Lucas Pun -
Editions édiTests 1984
81. AUTOCAP : Interactive Computer Assisted System for Turning Parts -
Ph D Thesis - VMIST, October 1979

82. LISP - C. Queinnec - Edition Eyrolles
83. A new approach to automatic generation of process-planning -
M. Spadoni, B. Mutel - 19th CIRP Conf. in Manufacturing Systems -
Philadelphia Juin 87
84. Système flexible de préparation du travail et Technologie de Groupe -
M. Spadoni, B. Mutel - Usinica - Paris - Mai 85
85. An Automated Process Planning and Selection Program APPAS -
Ph D Thesis, Purdue University, May 1977
86. Etude d'une fonction de classement et du codage de pièces mécaniques,
dans le contexte de la T.G.A.O.
M. Costantini - Thèse de 3è cycle - Université de Metz - 1987
87. Learning Heuristics in Second Generation Expert Systems -
Walter Van de Velde - 1986 -
Vrije Universiteit Brussel Pleinlaan 2, B-1050 Brussels, Belgium

ANNEXE C

ANNEXE D

ANNEXES

C. Structures informatiques.....	1
1. Introduction	1
2. Introduction des types d'arbres.....	2
2.1 Ensemble noeuds "contrôle" et "en avant".....	2
2.2 Ensemble noeuds "en arrière".....	3
3. Contrôle de la connaissance.....	3
3.1 Définition de l'algorithme.....	4
4. Procédure de validation des réceptivités.....	10
5. Partie recherche.....	10
5.1 Algorithme moteur I.....	11
5.2 Algorithme moteur II.....	16
D. Système d'édition de gammes détaillées.....	33
1. Introduction	33
2. Fichiers technologiques.....	34
3. Programme Gestion F.....	35
4. Programme Grille	39
5. Programme Util	46
6. Exemple d'utilisation.....	49

ANNEXE C

SAGGA :

Structures informatiques

1. INTRODUCTION

Dans la partie "structures informatiques" nous présentons les caractéristiques informatiques de la méthode utilisée dans le système SAGGA. Cette méthode est essentiellement basée sur la construction et la gestion d'arbres.

Dans le paragraphe 2 nous introduisons les différents types d'arbres et nous définissons l'ensemble des noeuds utilisés pour chacun d'eux. Puis dans le paragraphe 3 nous proposons une méthode pour la vérification de la "bonne écriture" de la base de règles, c'est une vérification structurelle et non syntaxique. Enfin dans le paragraphe 4 nous exposons la méthode utilisée pour la validation de la partie réceptivité et dans le paragraphe 5 nous établissons les caractéristiques de fonctionnement de la partie recherche proprement dite.

2. INTRODUCTION DES TYPES D'ARBRES

Dans son fonctionnement le système SAGGA fait appel à un ensemble de noeuds, qui peut différer suivant le type d'arbre à développer. Nous distinguons trois types d'arbres :

- * Un arbre "contrôle", développé pour le contrôle de la structure de la base de règles.
- * Un arbre "en avant", développé lors d'une recherche par chaînage avant (moteur I).
- * Un arbre "en arrière", développé lors d'une recherche par chaînage arrière (moteur II).

Pour ces différents types d'arbre nous utilisons deux ensembles différents de noeuds :

- * Un premier ensemble utilisé pour les arbres "contrôle" et "en avant".
- * Un deuxième ensemble concerne les arbres "en arrière".

2.1 Ensemble noeuds "contrôle" et "en avant"

Le premier ensemble de noeuds est composé de huit types de noeuds :

- Noeud "II" : caractérise le noeud, donc la racine de l'arbre.
- Noeud "SE,n" : indique que la règle n appartient à une transition de type "séquence".
- Noeud "DO,n" : indique que la règle n appartient à une transition de type "divergence OU".
- Noeud "CO,n" : indique que la règle n appartient à une transition de type "convergence OU".
- Noeud "EE" : indique l'introduction d'un état composé MC^j .
- Noeud "BR,n" : indique que la règle n appartient à une transition de type "divergence ET".

- Noeud "CE,n" : indique que la règle n appartient à une transition de type "convergence ET".
- Noeud "AA" : indique l'insertion d'un arbre de recherche de type "en arrière".

2.2 Ensemble noeuds "en arrière"

Le deuxième ensemble de noeuds est composé de neuf types de noeuds :

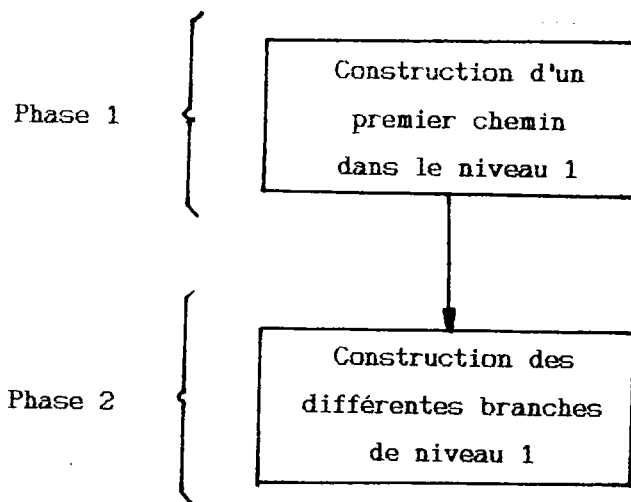
- Noeud "aa,n" : indique un état but, le numéro n qui s'y rapporte précise la règle associée.
- Noeud "CO" : introduit un ensemble de règles intervenant dans une transition de type "convergence OU".
- Noeud "BO,n" : indique que la règle n intervient dans une transition de type "convergence OU".
- Noeud "DO,n" : indique que la règle n intervient dans une transition de type "divergence OU".
- Noeud "CE,n" : indique que la règle n intervient dans une transition de type "convergence ET".
- Noeud "BR" : introduit une séquence simultanée.
- Noeud "DE,n" : indique que la règle n intervient dans une transition de type "divergence ET".
- Noeud "SA" : indique l'insertion d'un arbre par un ensemble de sous-buts.
- Noeud "sa,n" : indique un sous état but, introduit par la règle numéro n.

3. CONTROLE DE LA CONNAISSANCE

Il est impératif de vérifier la cohérence de la base de règles avant toute exploitation de celle-ci.

Suivant le principe que "Des règles bien écrites génèrent un Grafcet **M** bien formé", nous avons basé le contrôle de la base de règles sur la vérification du Grafcet **M** qu'elles génèrent. Cette **vérification** est obtenue par construction du Grafcet **M**, correspondant à la base de règles, en mémoire centrale de l'ordinateur.

La méthode utilise deux phases principales :



La figure Ac.1 représente l'algorithme proposé pour la méthode de contrôle, et la figure Ac.2 représente un exemple d'arbre de contrôle généré par l'algorithme.

3.1 Définition de l'algorithme

L'algorithme de contrôle de la base de règles est composé de sept procédures :

* **"Sélection des Règles"** : Après lecture de la règle associée au dernier noeud créé, la procédure "Sélection des Règles" présélectionne un ensemble de règles. Cette opération est effectuée sur l'ensemble des règles sauf les règles faisant partie d'une transition déjà construite et les règles déjà utilisées.

* "Type transition" : Au vu de l'ensemble des règles sélectionnées, la procédure "Type transition" recherche le type de la transition correspondante. Le résultat doit correspondre au tableau des compositions présenté au paragraphe III.5. Dans le cas où il n'y a pas correspondance, le système signale une erreur.

Dans le cas d'une transition de type :

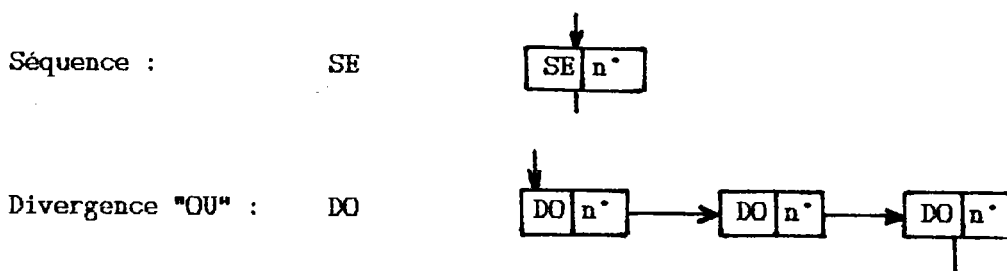
- "SE" : Le système vérifie si l'élément 5
(voir sous-paragraphe III.2.2) est unique.
- "DO" : Le système vérifie si les éléments 5 sont différents
et uniques.
- "CO" : Le système vérifie si les éléments 5 sont identiques
et uniques.
- "DE" : Le système vérifie si l'élément 5 est unique.
- "CE" : Le système vérifie si l'élément 5 est unique.

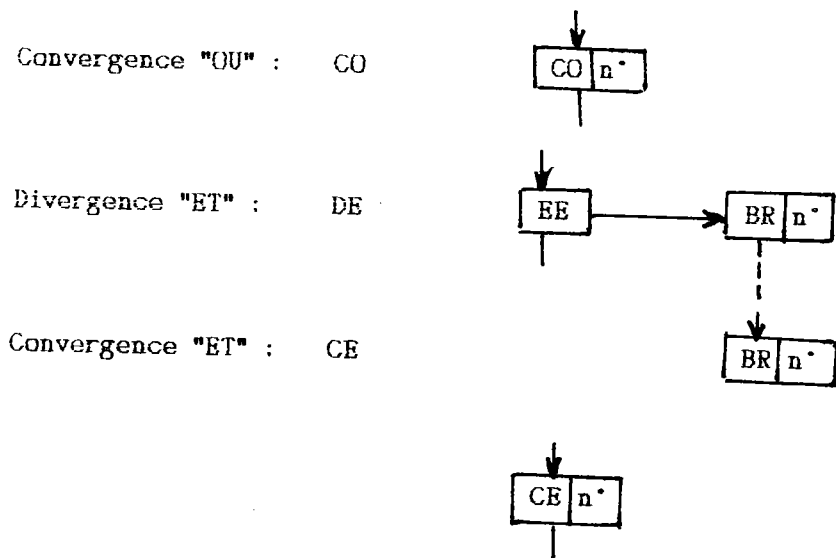
Attention, dans le cas où une transition de type "CE" appartient à un ensemble de transitions de type "CO", celui-ci est prioritaire par rapport au type "CE".

* "Création transition" : Le type de transition ayant été défini, la procédure "création transition" est chargée de la création dynamique du noeud correspondant et l'adapte sur l'arbre en cours de construction.

Constructions des noeuds en fonction du type des transitions :

Types noeuds Correspondances sur l'arbre de recherche





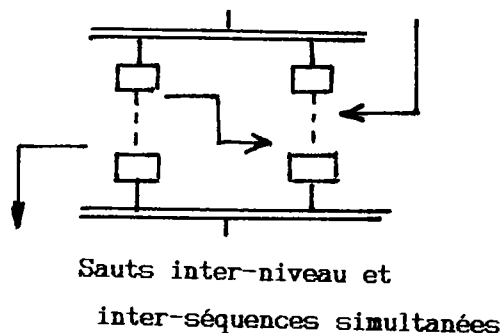
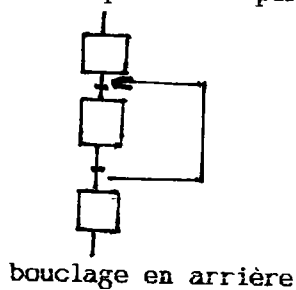
* "Déclenchement" : Après création d'un nouveau noeud, la procédure "Déclenchement" se positionne sur le dernier noeud créé et déclenche la règle qui s'y rapporte.

* "Règles Déjà Utilisées" : Lorsque le nombre de règles sélectionnées est nul, et que le dernier noeud créé est du type "CO,n", la procédure "Règles Déjà Utilisées" recherche des règles déjà utilisées. Cette recherche s'effectue sur l'ensemble des règles appartenant :

- au même niveau que la dernière transition.
- à la partie **avale** du noeud de type "DO,n" d'où est parti le sous-arbre en question.

Puis le système SAGGA teste si la dernière règle déclenchée appartient à cet ensemble de règles, dans le cas d'un résultat négatif, il y a détection d'une erreur.

Cette procédure permet de tester les erreurs suivantes :



* "Bactrack (1)" : Lorsque la construction de l'arbre s'est arrêtée sur un noeud de type "CO,n" et que le système n'a décelé aucune erreur, la procédure "Bactrack (1)", remonte l'arbre jusqu'au premier noeud de type "DO,n", d'un niveau égal ou supérieur. A partir de ce noeud le système commence la construction d'un nouveau sous-arbre. Dans le cas où la remontée s'effectue jusqu'au noeud initial "II", le contrôle est terminé avec la mention "succès".

* "Bactrack (2)" : Après détection d'une transition de type "convergence ET", la procédure "Bactrack (2)" est chargée de remonter l'arbre jusqu'à l'un des noeuds suivants : "EE", "DO", "BR". Si le noeud rencontré est :

"EEi" : Le système teste si la règle de type "convergence de ET" de même niveau (CEi) peut être validée. Dans le cas où le test est positif, il y a création du noeud "CE,n". Dans le cas où le test est négatif il y a détection d'une erreur. Ce test nous permet de respecter le tableau des combinaisons (paragraphe III.5).

"DO,n" : Il y a continuation dans la construction de l'arbre.

"BR,n" : Le système teste si la règle associée au noeud "CEi,n" correspondante à l'état composé, peut être validée. Dans le cas où le test est négatif, la règle correspondante est déclenchée, et il y a construction de la séquence correspondante. Dans le cas où le test est positif, il y a détection d'une erreur. Ces erreurs peuvent être soit :

- Une mauvaise écriture de la règle de type "CE".
- Une intersection de transition (voir paragraphe III.5).

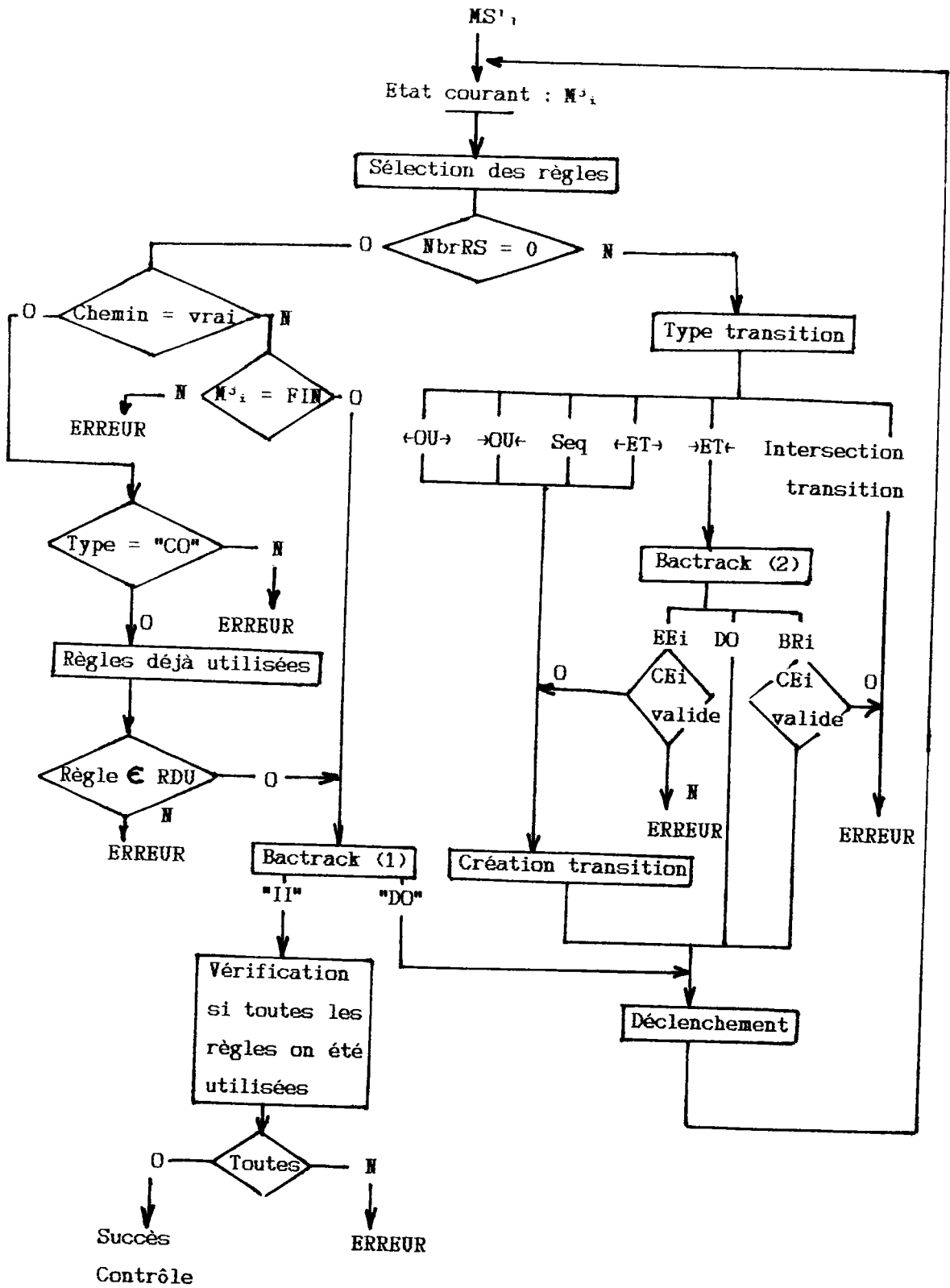


figure Ac.1 : Algorithme proposé par la méthode de contrôle.

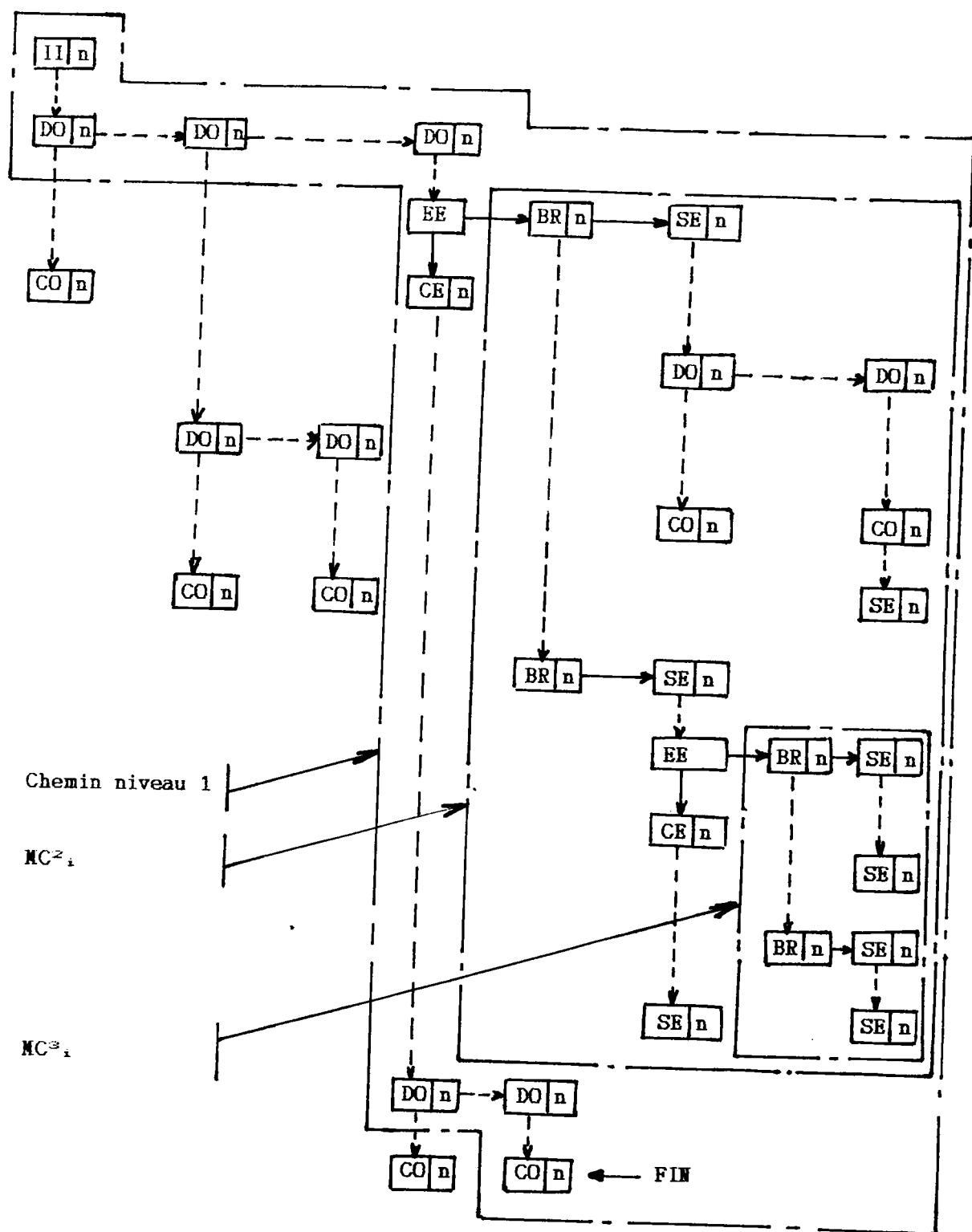


figure Ac.2 : Exemple de représentation graphique de l'arbre de contrôle, en mémoire centrale.

4. PROCEDURE DE VALIDATION DES RECEPTIVITES

Lors d'une recherche, la procédure de validation des parties réceptivité est importante, car elle intéresse directement les performances du système.

Une réceptivité est formée d'une disjonction de monômes, elle ne peut être validée que si l'un de ceux-ci est vrai. La procédure de validation d'un monôme opère par une analyse globale du celui-ci qui lui permet d'établir une priorité parmi les éléments qui le composent. Le critère de priorité est fonction du type de chaque élément qui est établi, par ordre décroissant de la manière suivante :

- * prédicat - fonction
- * objet EF
- * objet EE
- * objet EP

Cette priorité est nécessaire, car il est inutile d'entamer une procédure de validation d'un élément de type EP, si dans le même monôme, il existe un élément de type fonction qui ne pourra être validé.

Tenant compte du fait qu'un élément de type objet influe sur le développement de l'arbre, lorsqu'une règle a été validée avec la présence d'un élément de type objet, le système teste la présence d'autres monômes, et ainsi vérifie la possibilité d'une validation différente de la règle.

5. PARTIE RECHERCHE

La partie recherche représente le noyau du système SAGGA. Celle-ci se compose essentiellement du moteur I et du moteur II (figure III.7), dont le fonctionnement prend en compte l'ensemble des situations décrites dans les tableaux énoncés dans le paragraphe IV.5.

5.1 Algorithme moteur I

L'algorithme général est représenté par la figure Ac.3 et en figure Ac.4 nous avons illustré un exemple d'arbre de recherche développé par cet algorithme. Il se compose de huit procédures principales :

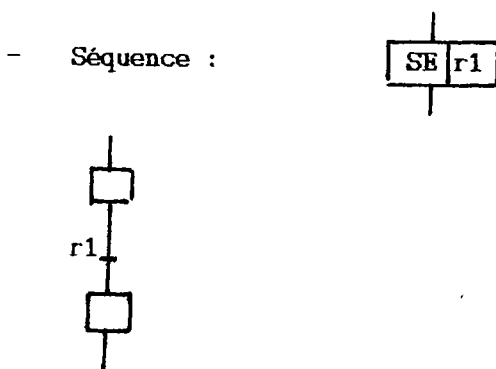
* "Sélection des règles" : A partir d'un état courant MS^j_i , la procédure "Sélection des règles" présélectionne les règles. Cette opération est effectuée sur l'ensemble des règles, sauf celles utilisées dans des transitions déjà validées et celles qui interviennent dans une transition déjà construite.

* "Type de Transition" : Au vu de l'ensemble des règles présélectionnées, la procédure "Type de Transition" recherche le type de transition correspondant à cet ensemble.

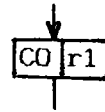
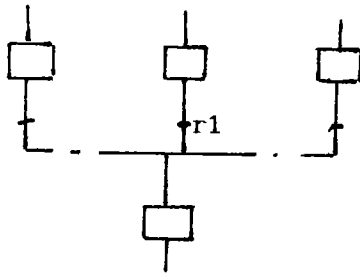
On retrouve les quatres types de bases :

- convergence de "OU"
- divergence de "OU"
- convergence de "ET"
- divergence de "ET"

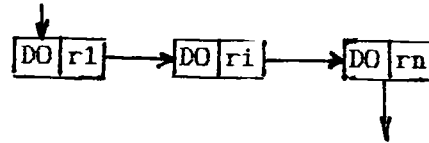
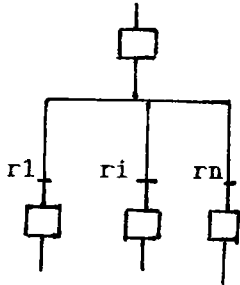
* "Création Transition" : Après définition du type de la transition, la procédure "Création Transition" est utilisée pour la création effective des noeuds correspondants ainsi que l'adaptation sur l'arbre en cours de développement. A chaque transition correspond des noeuds particuliers :



- Convergence de "OU" :

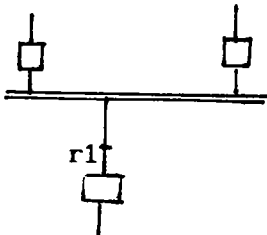


- Divergence de "OU" :

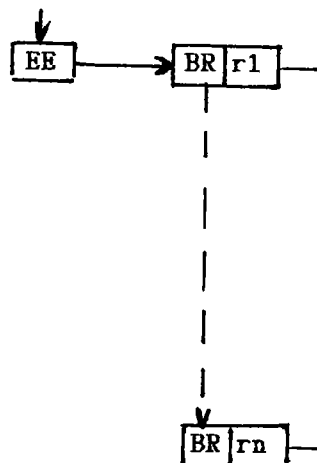
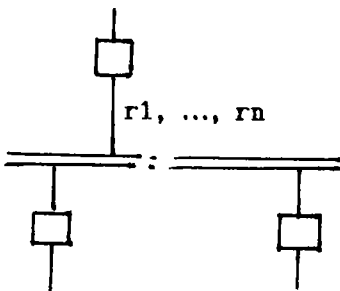


à noter que l'ordre de création des noeuds respecte les différents poids de chaque règle.

- Convergence de "ET" :



- Divergence de "ET" :



* "Validation Règle" : Après positionnement sur le dernier noeud créé, la procédure "Validation Règle" est utilisée pour valider la partie réceptivité de la règle associée au noeud. La validation est effectuée en respectant les directives citées dans le paragraphe Ac.4.

* "Déclenchement" : Après validation de la règle, la procédure "Déclenchement" applique la règle et ainsi modifie l'état courant MS_i.

* "Backtrack (AV,1)" : Dans le cas où il y a échec dans une tentative de validation de la partie réceptivité d'une règle, le système fait appel à la procédure "Backtrack (AV, 1)" qui est chargé de remonter l'arbre de recherche. L'arrêt de la procédure "Backtrack (AV,1)" est effectué dans trois cas, à la rencontre d'un noeud de type :

- "DO" non encore exploité, c'est un état activable.
- "AA" qui indique l'insertion d'un arbre de type "arrière", dans ce cas il y a appel du moteur II.
- "II" signifie que la totalité des possibilités a été exploitée et débouche ainsi sur un échec.

* "Backtrack (AV,2)" : Lorsqu'un ensemble de séquences simultanées est en cours de validation, et que le moteur I vient de finir de valider l'une d'entre elles, il fait appel à la procédure "Backtrack (AV,2) pour remonter cette dernière séquence et en recommencer une autre. Pour cela il remonte jusqu'au premier noeud de type "BR" rencontré et non encore exploité.

* "Backtrak (AV,3)" : Après la rencontre et la validation d'une transition de type "CEi", la procédure "Backtrack (AV,3) remonte l'arbre jusqu'au noeud de type EEi (de même niveau). Puis il y a création du noeud CE, à partir du noeud "EE".

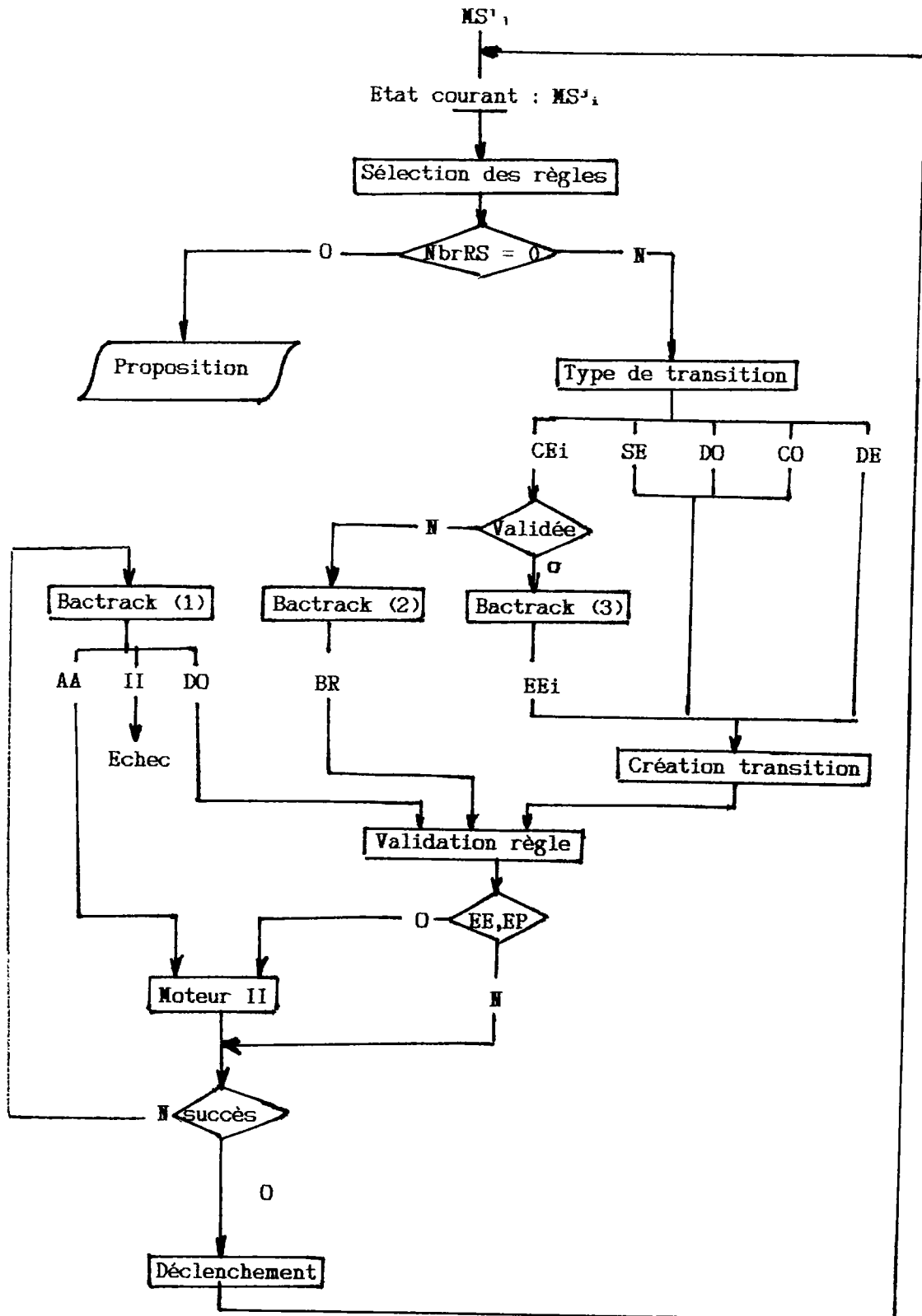


figure Ac.3 : Algorithme moteur I.

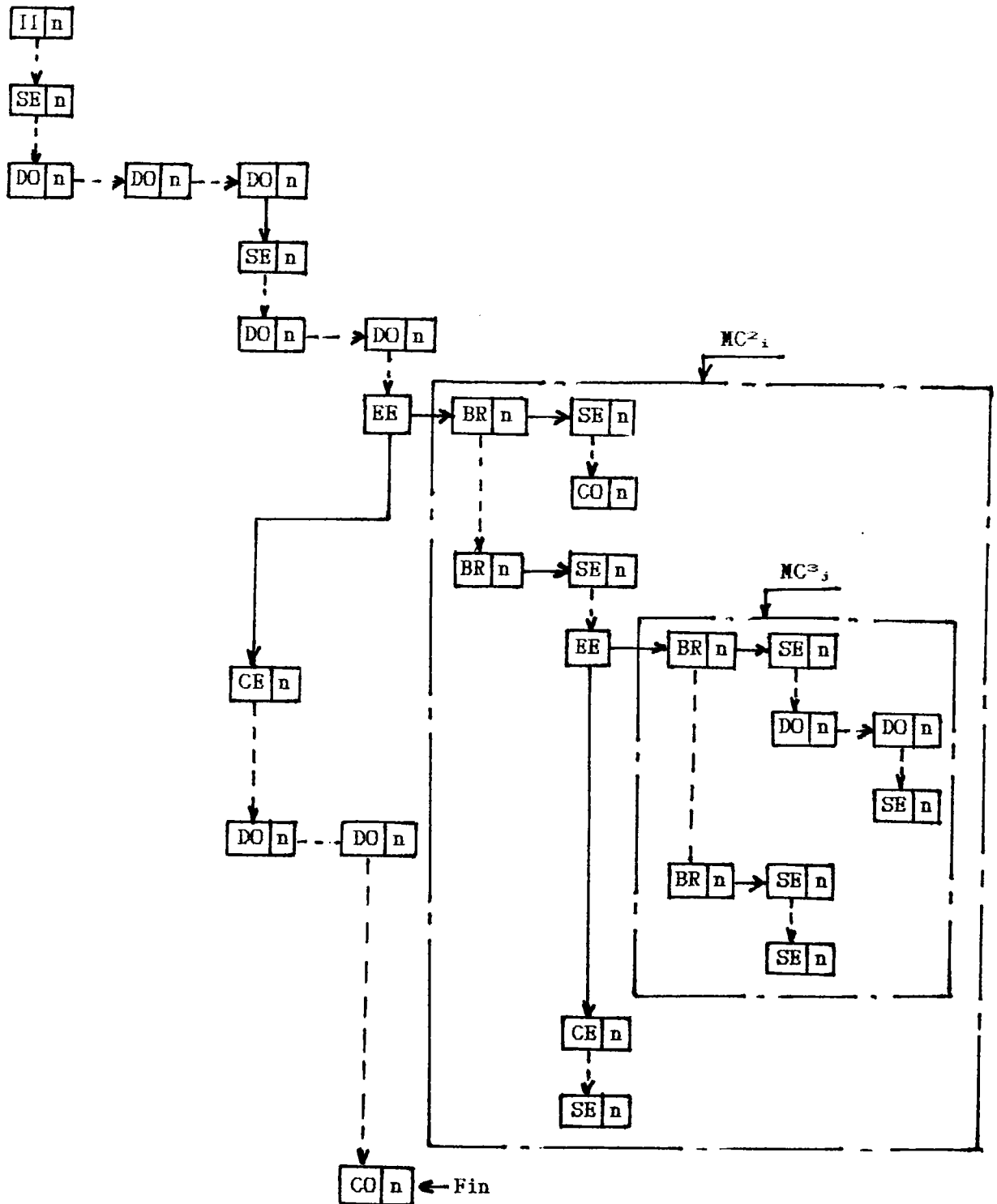


figure Ac.4 : Exemple de représentation d'arbre de recherche
développé par le Moteur I.

5.2 Algorithme moteur II

Le moteur II peut être sollicité pour deux raisons (figure Ac.5) :

- Etablissement d'un fait.
- Lors d'un retour arrière (procédure Backtrack (AV,1)).

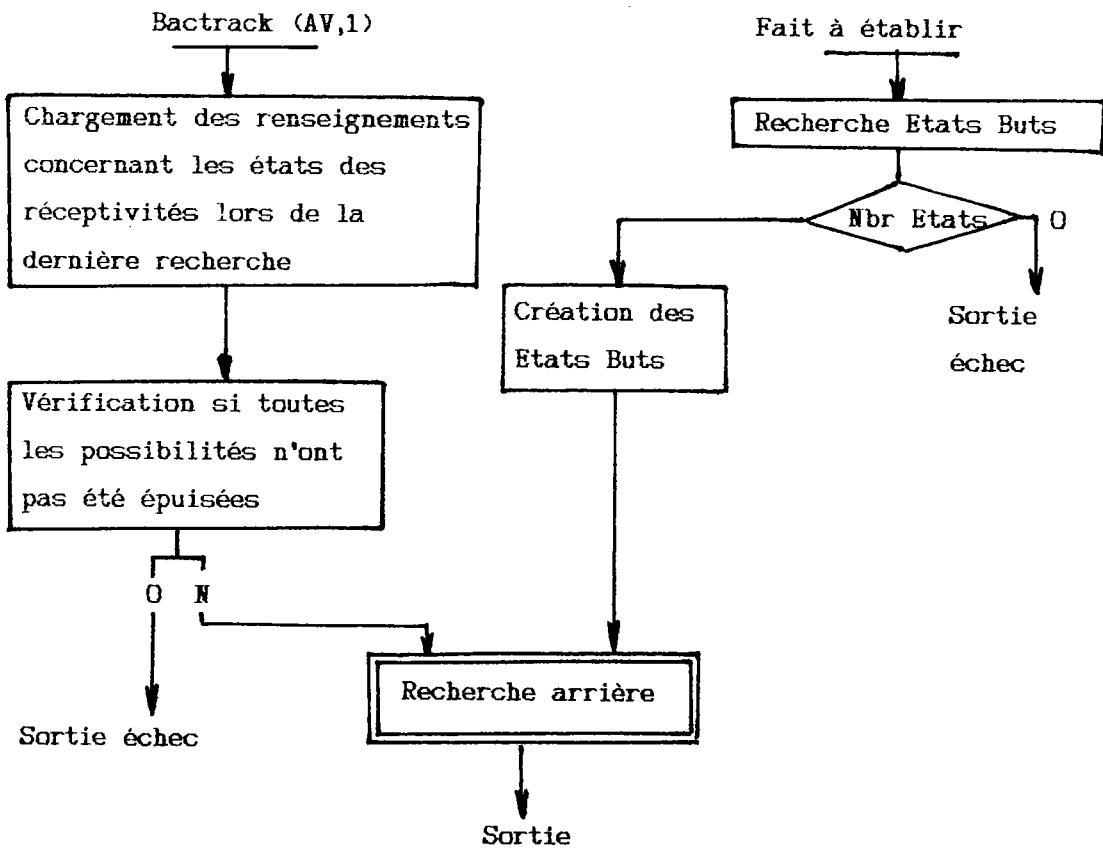


figure Ac.5 : Algorithme général du moteur II.

Lorsque le moteur II est chargé d'établir la valeur de vérité d'un élément type objet, il recherche l'ensemble des états buts possible. Cette recherche s'effectue sur les états buts dont les règles qui s'y rapportent ne figurent pas dans les transitions validées et n'ont pas encore été utilisées. Dans le cas où il n'existe pas d'état but, le moteur II sort en échec. Dans le cas où il existe au-moins un état but, il y a création d'un noeud de type "AA" pour signaler l'insertion d'un arbre type "arrière", et autant de noeuds de type "aa" que d'états buts sélectionnés.

A noter que si l'état but se trouve à la sortie d'une convergence de "OU" il ne sera sélectionnée qu'une règle parmi l'ensemble des règles constituant la transition. (figure Ac.6)

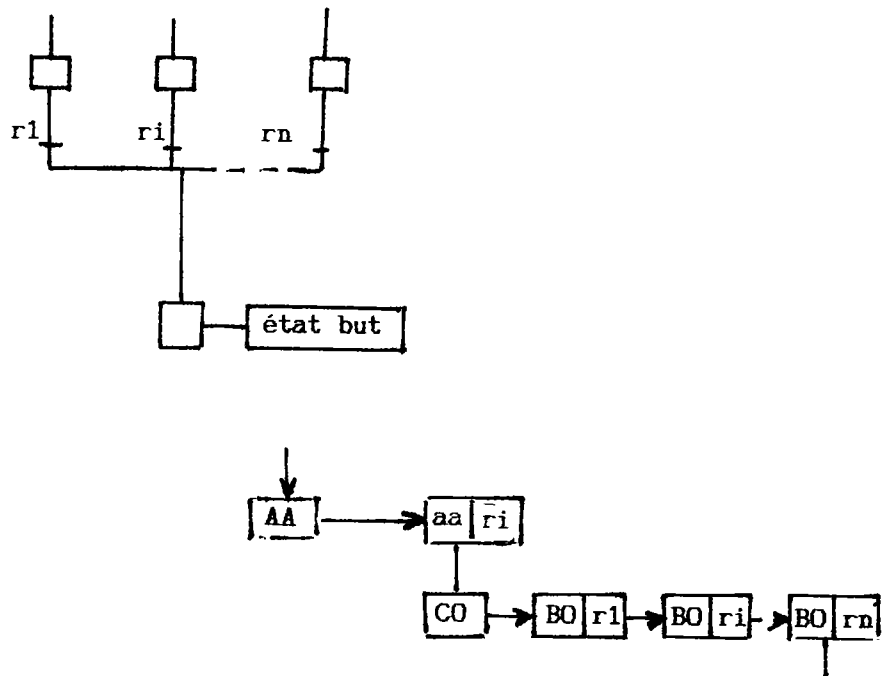


figure Ac.6 : Illustration du cas, où l'état but se trouve à la sortie d'une convergence de "OU".

Pour l'établissement de l'ordre de création des différents noeuds, le moteur II se réfère au poids des règles associées à ces états buts.

Dans le cas où le moteur II est sollicité par la procédure "Backtrack (AV,1)", il prend connaissance de l'état des réceptivités de la dernière recherche effectuée dans l'arbre en question. Puis il vérifie si toutes les possibilités ne sont pas épuisées, si c'est la cas alors il y a sorti en échec, sinon la recherche est entamée.

La recherche proprement dite est réalisée par procédure "Recherche arrière". La figure Ac.18 représente un exemple d'arbre développé par le moteur II, et la figure Ac.17 représente l'algorithme de la procédure "Recherche arrière" qui se compose de huit procédures principales :

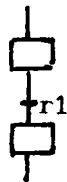
* **"Sélection Règles"** : Après lecture de la règle associée au dernier noeud créé, la procédure "Sélection Règles" présélectionne un ensemble de règles. Cette opération est effectuée sur l'ensemble des règles sauf les règles faisant partie d'une transition déjà construite et les règles déjà utilisées.

* **"Type Transition"** : Après présélection d'un ensemble de règles, la procédure "Type Transition" est chargée de définir le type de transition correspondant à cet ensemble de règles. On retrouve les cinq transitions de base. Dans le cas où la transition est du type "CO", celle-ci est immédiatement créée, sinon la moteur II teste si la fin de la recherche dite "en arrière" est atteinte.

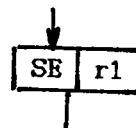
* **"Création Transition"** : Après présélection des règles et définition de type de transition, la procédure "Création Transition" réalise la construction des noeuds et l'adapte à l'arbre en cours de développement.

A chaque transition correspond des **noeuds différents** :

- Séquence :

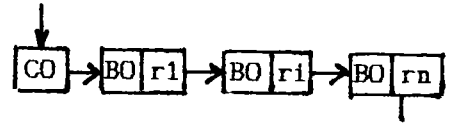
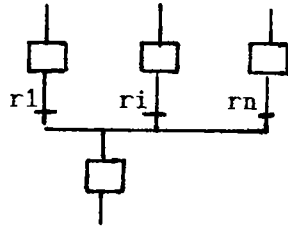


Représentation Grafcet M

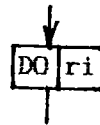
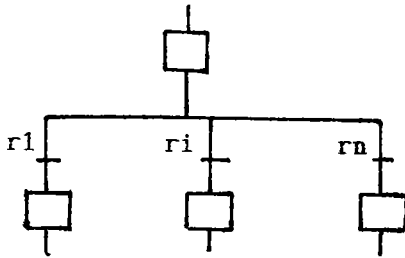


Correspondance dans
l'arbre de recherche

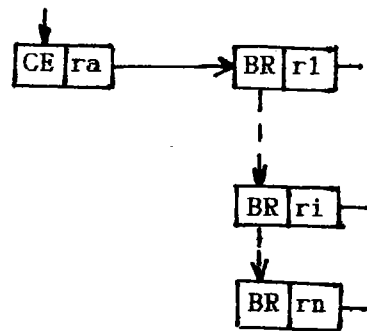
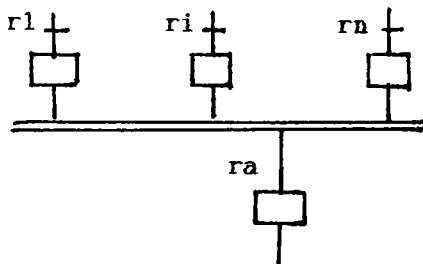
- Convergence de "OU" :



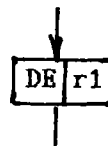
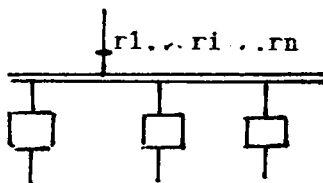
- Divergence de "OU" :



- Convergence de "ET" :



- Divergence de "ET" :



* "Insertion Début" : Lorsqu'un état but ou sous-but se situe dans un état composé (figure Ac.8) il est impératif de réaliser une insertion

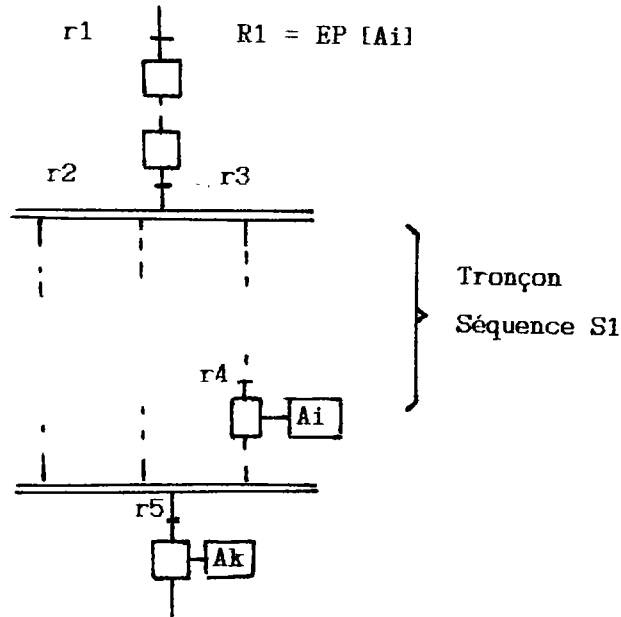


figure Ac.7 : Exemple où l'insertion est obligatoire.

Une insertion est définie comme un changement d'état but, dans le cas de la figure Ac.7, on passe de l'état but Ai à l'état but Ak . Ce qui se traduit par la validation de l'ensemble de l'état composé. Cette opération est réalisée en deux phases :

- Phase 1 : Détection d'une insertion.

Une insertion est décelée lors de la validation d'un noeud type "DE" et de la non existence d'un noeud type "CE" figure Ac.8 :

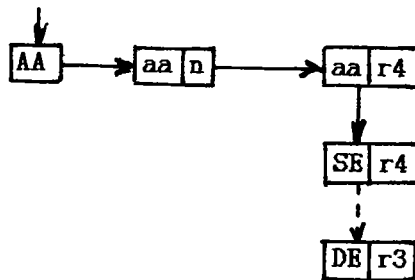


figure Ac.8 : Détection d'une insertion dans l'arbre de recherche.

- Phase 2 : Réalisation du début d'insertion.

Un début d'insertion est réalisé par la modification de l'état but et le stockage temporaire du tronçon de la séquence notée S1 figure Ac.9.

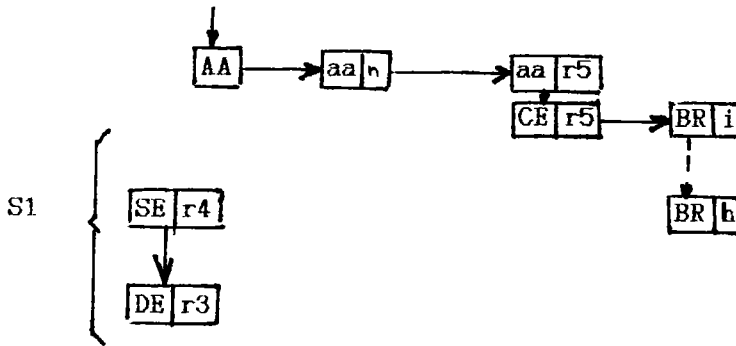


figure Ac.9 : Phase d'insertion.

* "Insertion Fin" : La procédure de fin d'insertion est réalisée en deux phases :

- Phase 1 : La détection d'une fin d'insertion est obtenue lorsque la règle du premier état but Ai est appliquée par le système.
- * Phase 2 : La réalisation de l'insertion qui consiste à rattacher le tronçon de la séquence stockée temporairement figure Ac.10.

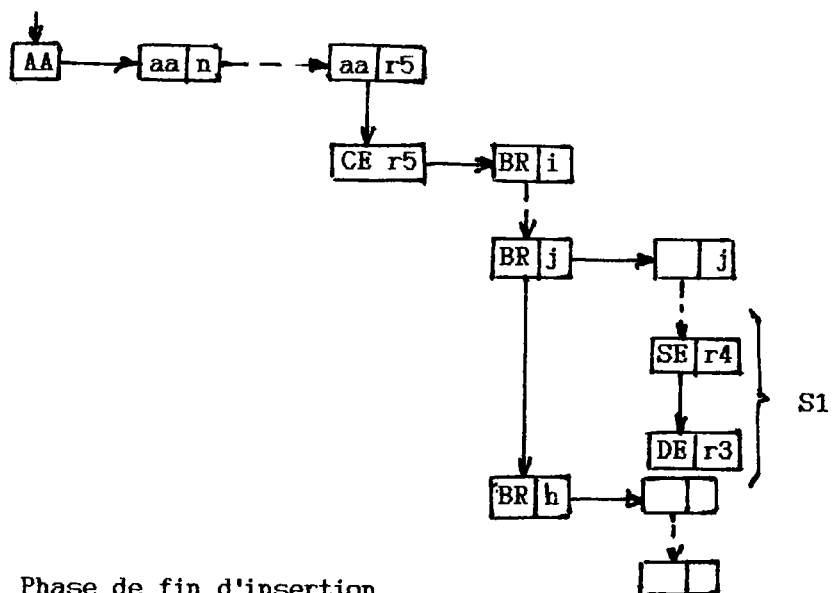


figure Ac.10 : Phase de fin d'insertion.

* "Recherche Etat Sous-But" : Lors d'une recherche arrière le moteur II peut être amené à valider des éléments de type objets. La validation ne sera déclenchée que lorsque l'arbre de recherche "arrière" sera terminé, donc pendant la recherche les éléments objets seront empilés dans la table dite de renseignements (états des réceptivités). La procédure "Recherche Etat Sous-But" n'est déclenchée que lorsque l'objet est du type EP ou EE et qui n'a pu être validé par la procédure "Objet à Valider". Le déclenchement de la procédure "Recherche Etat Sous-But" se traduit par la recherche des règles dont les états correspondants coïncident avec l'état précisé dans l'objet. Cette recherche s'effectue sur l'ensemble des règles ne faisant pas partie des transitions déjà construites et des règles déjà utilisées. Dans le cas où il y a échec, c'est à dire aucun état sous-but possible, le moteur II se positionne sur le noeud dont la règle contient l'objet responsable de l'échec et essaie de valider la règle avec un autre monôme. Dans le cas où il y a succès, alors le Moteur II crée les noeuds nécessaires.

* "Création Sous-But" : Un ensemble de sous-buts ayant été sélectionné, la procédure "Création Sous-Buts" est chargée de créer les noeuds correspondant et de les adapter sur l'arbre en cours de développement. L'ordre de création des différents noeuds est assujetti au poids des règles figure Ac.11 et figure Ac.12.

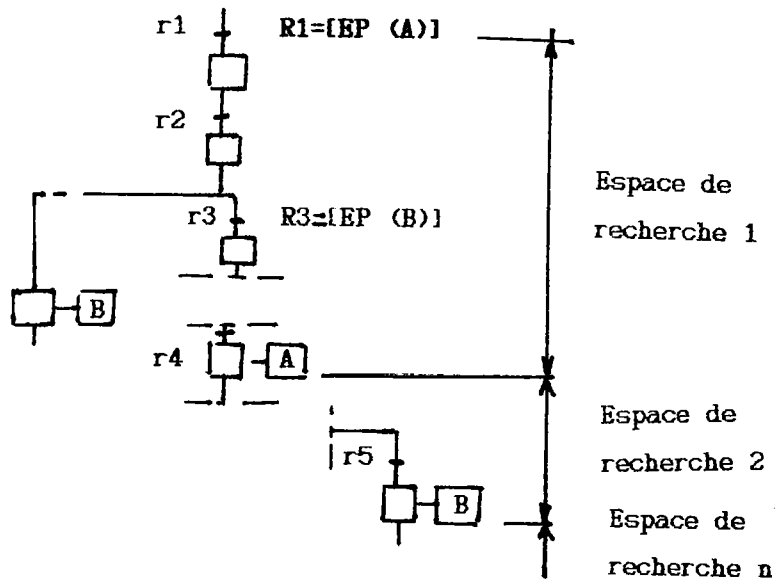


figure Ac.11 : Représentation d'une situation de sous-but en Grafset
M.

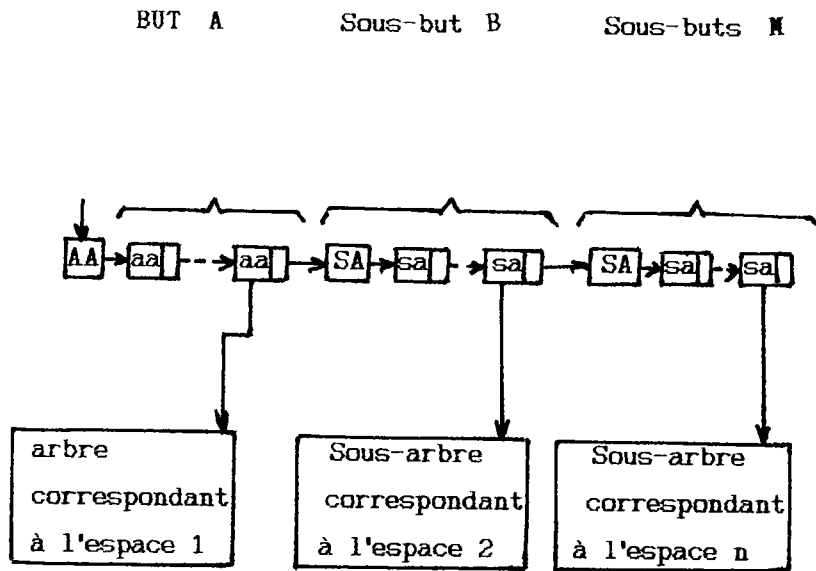
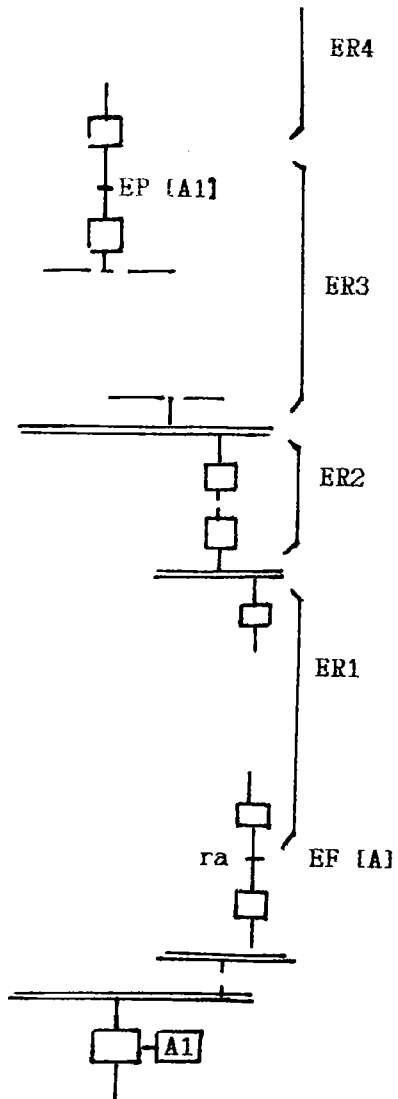


figure Ac.12 : Représentation de la situation figure Ac.11 dans l'arbre de recherche.

* "Objet à Valider" : Lorsque le développement de l'arbre de recherche "arrière" est terminé et en fonction des éléments de type objets empilés (Renseignement état réceptivités), la procédure "Objet à Valider" est chargée d'établir la valeur de vérité de chacun d'eux. La lecture de chaque élément de type objet de la pile, est réalisée de manière séquentielle par le haut de la pile. La vérification est spécifique à chaque type d'objet.

"EF" : La validation est effectuée sur l'ensemble des règles déjà utilisées et situées en amont de la règle à qui appartient l'objet EF figure Ac.13, figure Ac.14.



L'ensemble des règles sur lequel
va être effectué le test est noté
ER tel que :

$$ER = ER1 + ER2 + ER3 + ER4$$

figure Ac.13 : Situation représentant le cas d'un élément de type EF
à valider lors d'une recherche arrière.

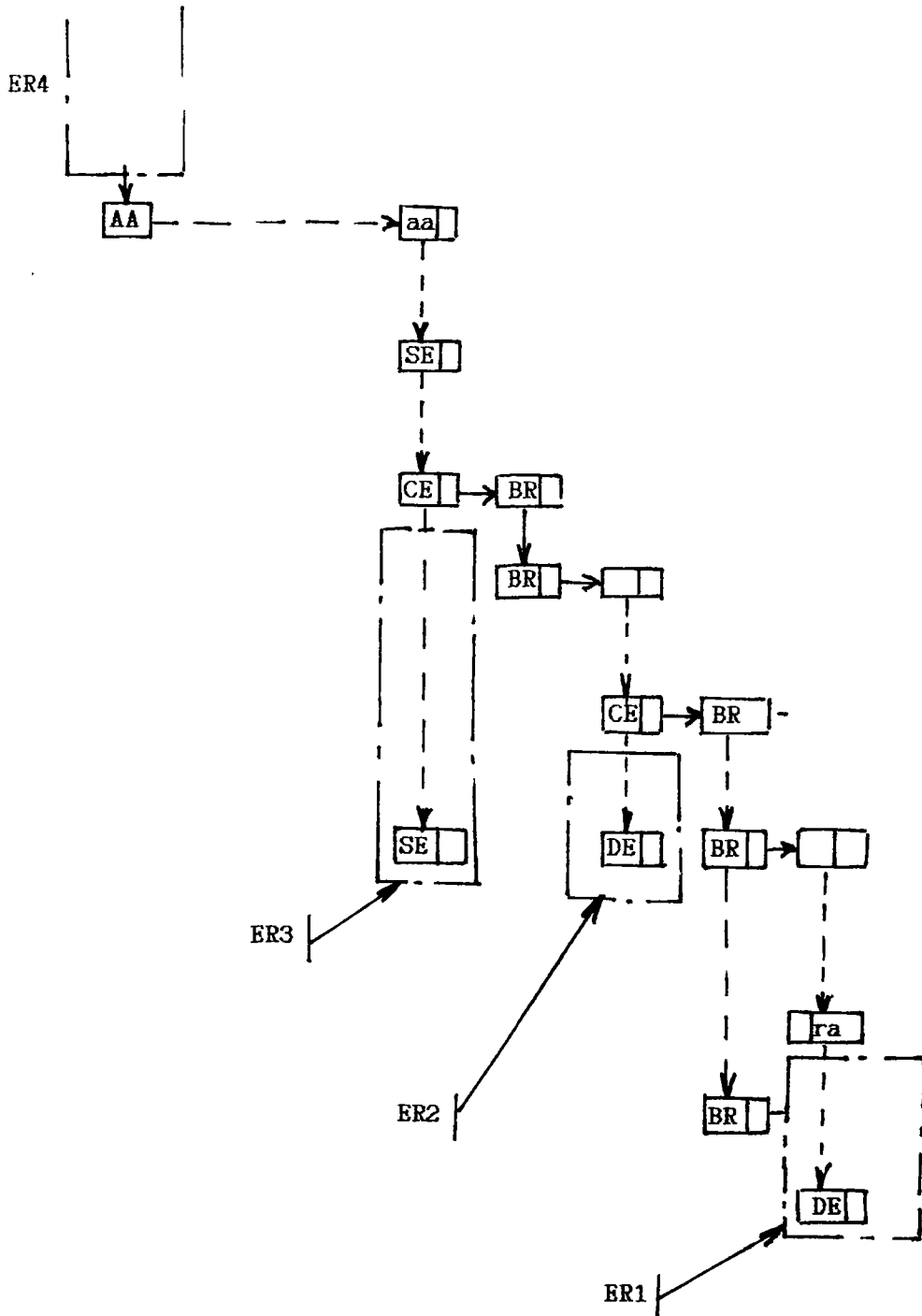


figure Ac.14 : Représentation dans l'arbre de recherche, l'ensemble des règles à considérer lors d'une validation d'un objet "EF".

"EP" : La recherche de la valeur de vérité est effectuée sur l'ensemble des règles situé en aval de la règle à qui appartient l'objet EP figure Ac.15.

$$ER = ER1 + ER2$$

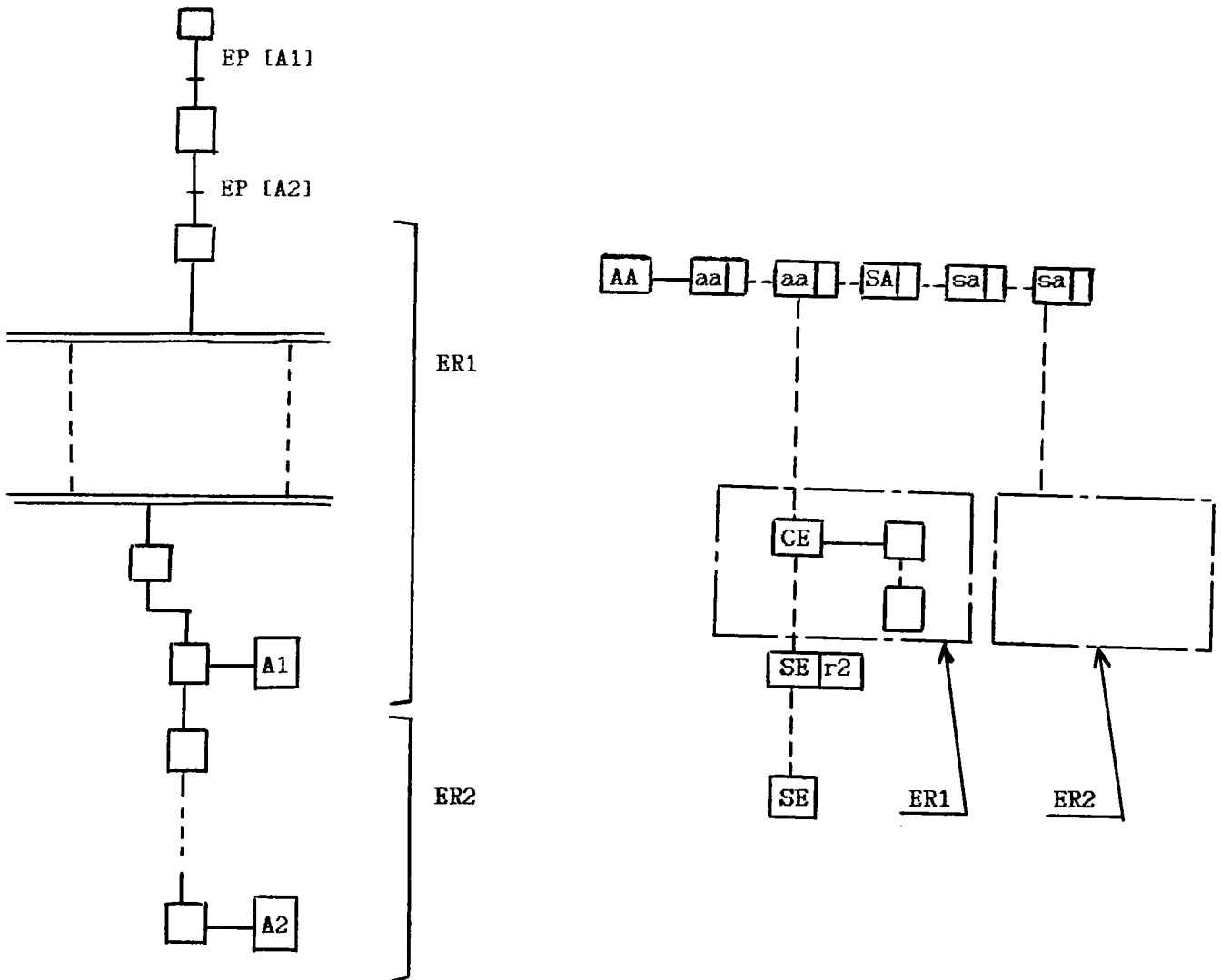
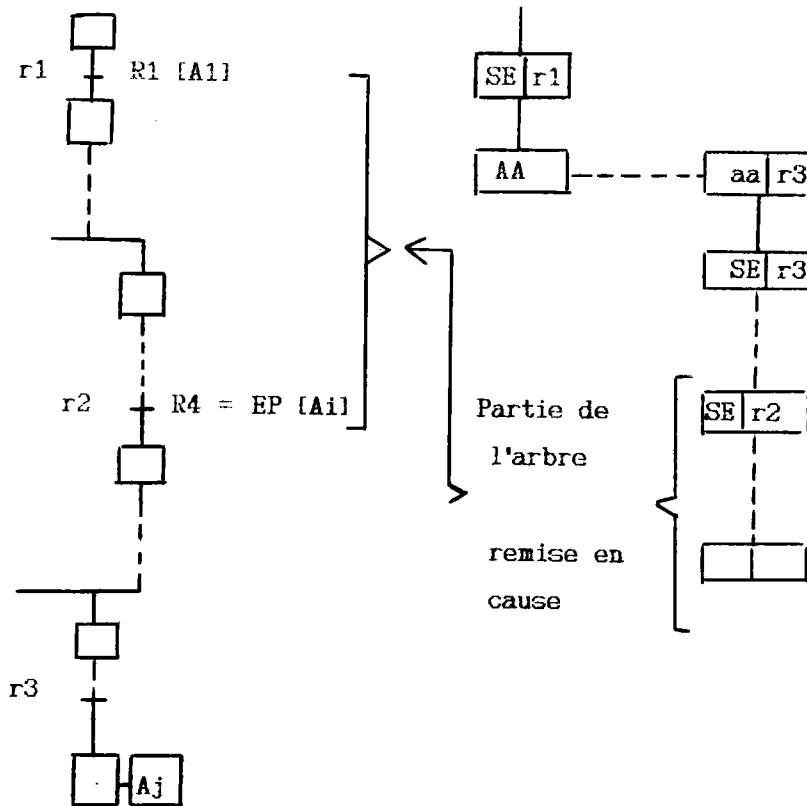


figure Ac.15 : Ensemble des règles correspondantes aux transitions franchies.

"EE" : Pour la validation de ce type d'objet la procédure le considère d'abord comme un type EE, si il y a échec le considère ensuite comme un type EP.

* "Destruction d'arbre" : Lors d'un échec après tentative de validation d'un élément de type objet. Le Moteur II par l'intermédiaire de la procédure "Destruction arbre" remet en cause une partie de l'arbre développé. C'est la partie qui se situe entre le noeud associé à la règle responsable de l'échec et la feuille de l'arbre (figure Ac.16).



Représentation Grafcet M

Représentation arbre de
recherche

r1 = règle responsable de l'appel du moteur II

r2 = règle responsable de l'échec

figure Ac.16 : Illustration d'une remise en cause de l'arbre de recherche, après un échec dû à un élément de type objet.

* "Validation Règle" : A partir du dernier noeud créé, la procédure "Valide Règle" est chargée de rechercher la valeur de vérité de la partie réceptivité de la règle associée à ce noeud. Dans le cas où dans le monôme en cours d'étude, il y a la présence d'un élément de type objet, celui-ci est empilé dans la table représentative de l'état des réceptivité. L'ensemble de ces objets sera étudié lorsque l'arbre de recherche "arrière" sera terminé.

* "Autre monôme" : En cas d'échec lors de la validation d'un élément de type objet, la procédure "Autre monôme" recherche s'il existe un autre monôme non encore exploité qui permettrait de valider la réceptivité. En cas d'échec, la règle est rejetée puis la procédure "Destruction arbre" est activée.

* "Backtrack (AR,2)" : Quand il y a nécessité de remonter une séquence faisant partie d'un ensemble de séquences simultanées, le Moteur II fait appel à la procédure "Backtrack (AR,2)". L'arrêt est déclenché lors de la rencontre d'un noeud de type :

- "BR", c'est le cas où l'on se trouve en cours de validation de séquences simultanées.
- "CE", indique la fin de validation d'un ensemble de séquences simultanées.

* "Backtrack (AR,3)" : Quand il y a échec lors d'une tentative de validation d'une règle, la procédure "Backtrack (AR,3)" est chargée de remonter l'arbre de recherche et ne s'arrête qu'à la rencontre d'un noeud de type :

- "BO" : L'arrêt est demandé, car on se trouve au niveau d'une transition de type "CO" ce qui se traduit par une nouvelle possibilité de chemin (état vivant).
- "sa" : Possibilité de développer un arbre "arrière" à partir d'un nouvel état but.

- "SA" : Echech d'un sous-arbre, le pointeur se positionne directement sur la règle qui a été à l'origine du développement de ce sous-arbre. Puis la procédure "Backtrack (AR,3)" continue en séquence.
- "AA" : Signifie l'épuisement de toutes les possibilités de l'arbre "arrière", il y a sortie échec.

De plus, l'arrêt est déclenché lorsque la procédure "Backtrack (AR,3)" décèle la présence d'un élément de type objet dans un monôme utilisé pour la validation d'une règle, et la présence d'autres monômes non encore exploités, dans cette même réceptivité.

* "Backtrack (AR,4)" : Dans le cas d'un succès de la recherche arrière, la procédure "Backtrack (AR,4)" est utilisée pour quitter dans de bonnes conditions l'arbre de développement arrière.

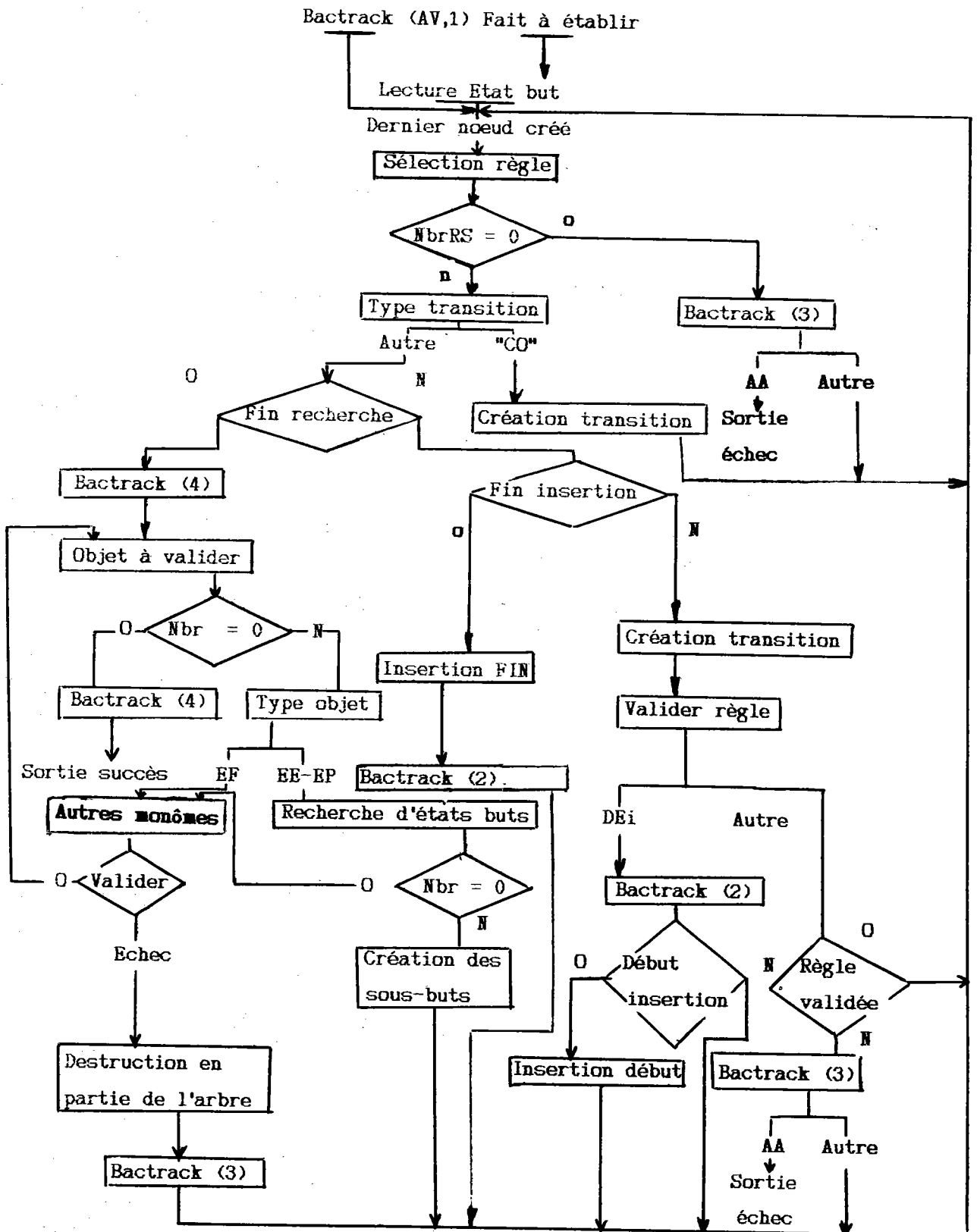


figure Ac.17 : Algorithme général de la procédure de "Recherche arrière".

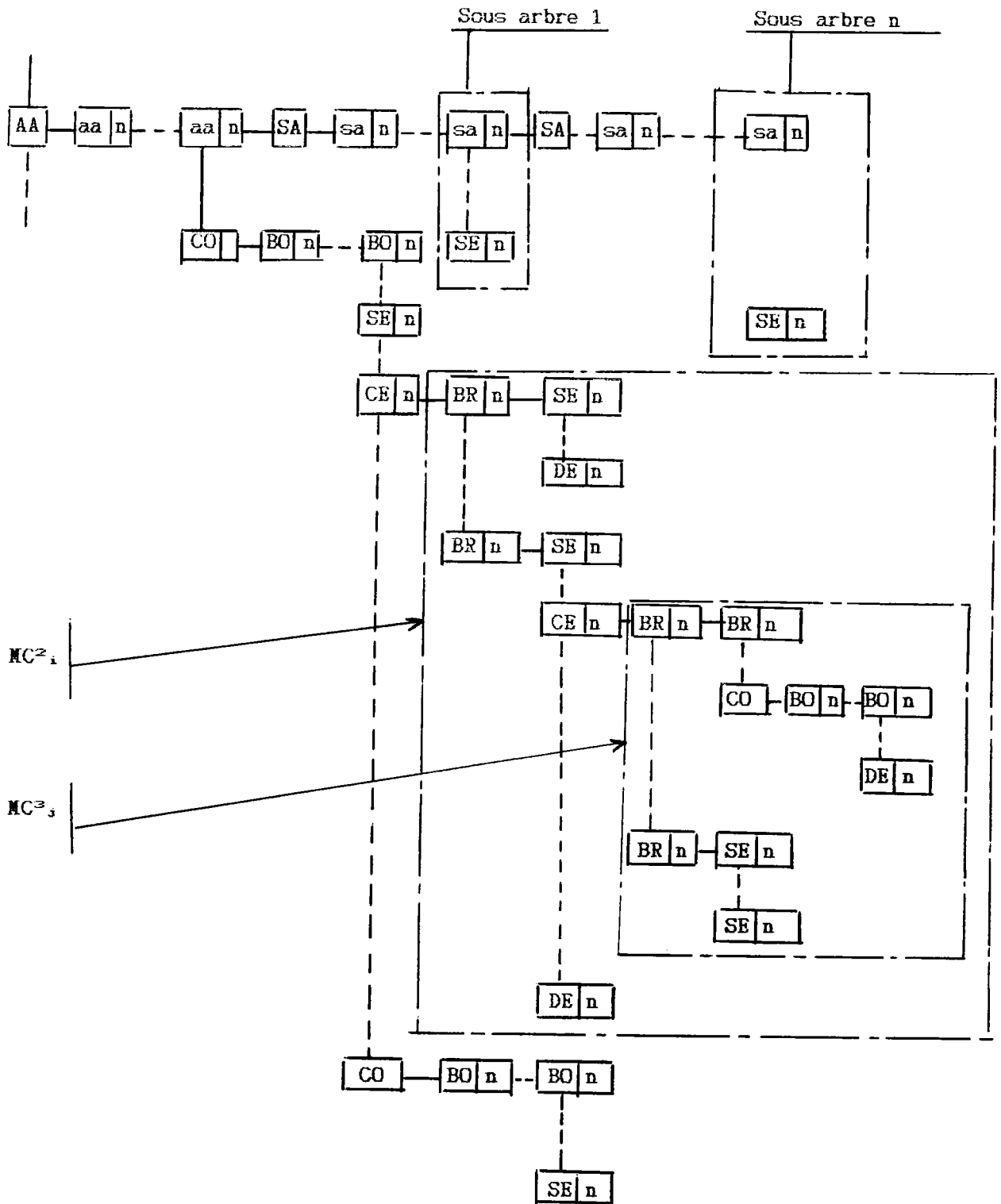


figure Ac.18 : Exemple de structure d'arbre développé par le Moteur II.

ANNEXE D

Systeme d'edition de gammes detaillées

1. INTRODUCTION

L'annexe D concerne la spécification de la structure des fichiers technologiques et des différents programmes (voir paragraphe IV.3) d'un point de vue de l'utilisateur, qui compose le système d'édition de gammes détaillées. Puis nous terminons par un exemple concernant la fabrication mécanique.

2. FICHIERS TECHNOLOGIQUES

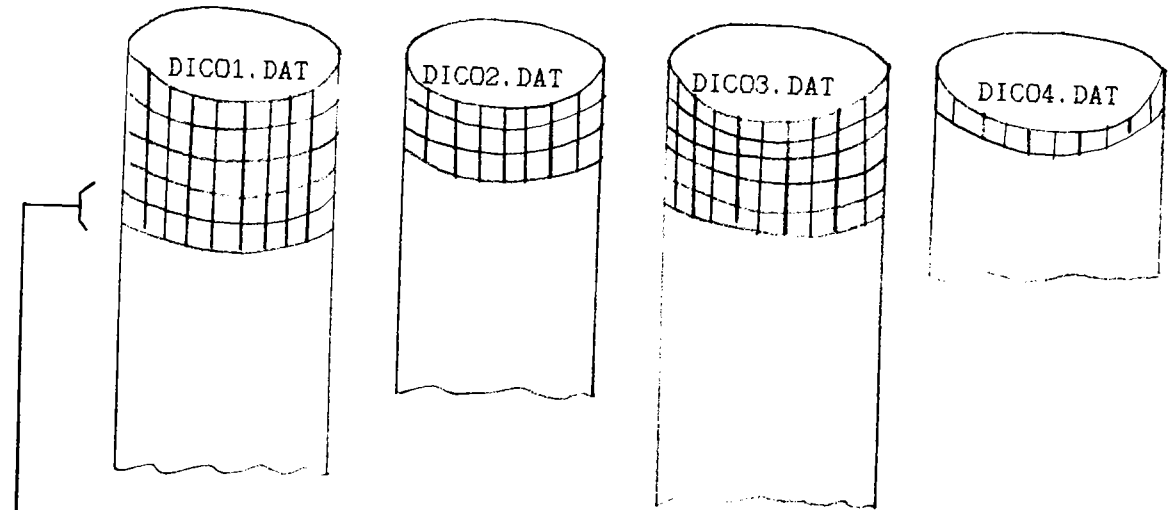
L'ensemble des données est utilisé par 3 programmes :

- GESTION F - GRILLE - UTIL -

GESTION F : utilisé pour la lecture, la modification et la destruction des données.

GRILLE
UTIL exploitent l'ensemble des données (En lecture)
contenues dans les fichiers dictionnaires.

- Fichiers : (fichiers indexés)



- Enregistrement :

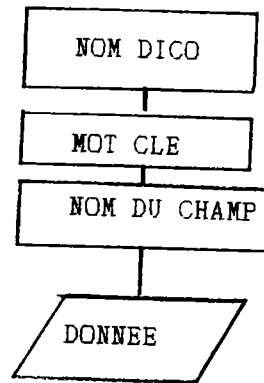
CLE	NOM(1)	NOM(2)	NOM(3)	NOM(4)	NOM(5)	NOM(6)	NOM(7)
-----	--------	--------	--------	--------	--------	--------	--------

- Champ :

D.O.N.N.E.E.

(25 caractères maximum)

Pour accéder a une donnée il faut :

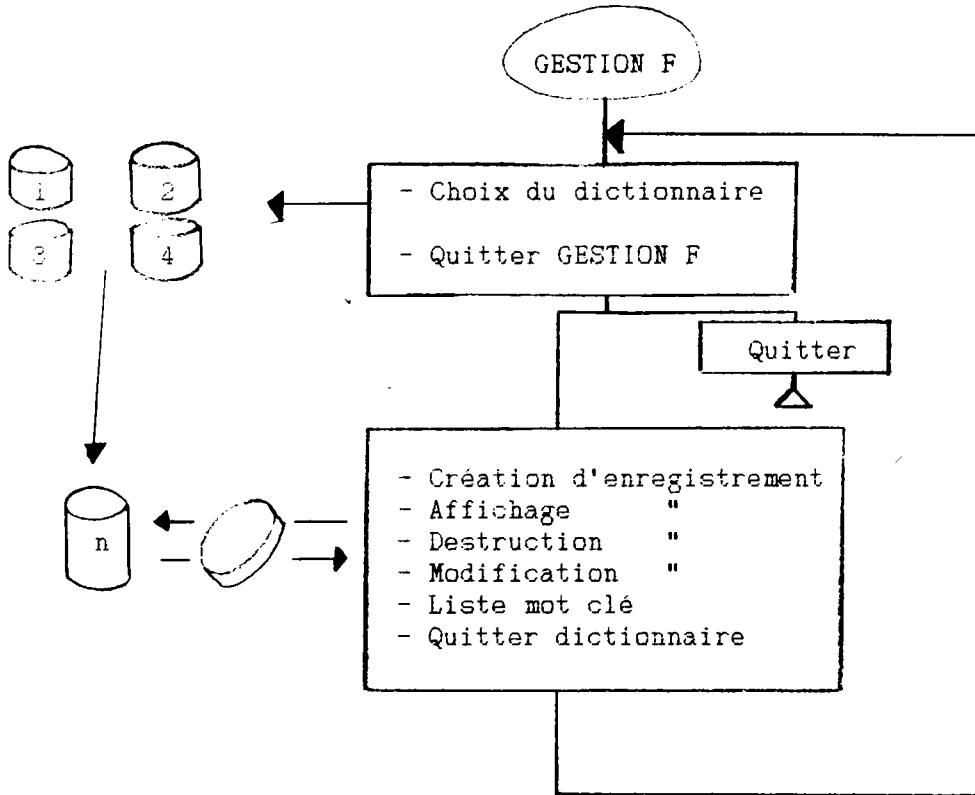


- Préciser le nom du fichier dictionnaire
(DICO1.DAT, DICO2.DAT, DICO3.DAT, DICO4.DAT)
- Préciser l'enregistrement grâce à des clés d'accès
- Préciser le nom du champ dans lequel celle-ci est stockée.
(NOM1, NOM2, NOM3, NOM4, NOM5, NOM6, NOM7)

3. PROGRAMME GESTION F

GESTION F permet la création et la gestion complète de 4 fichiers dictionnaires ayant une même structure (ce nombre pourra être étendu à 16).

Organisation générale de GESTION F



Utilisation de GESTION F

- Chargement de GESTION F
- Affichage d'un menu général

CHOIX DU DICTIONNAIRE	1, 2, 3, 4 ;
QUITTER GESTION F	Q ;

REPONSE :	☒
-----------	-------------------------------------

Réponse 1	ouverture de DICO1.DAT
Réponse 2	" de DICO2.DAT
Réponse 3	" de DICO3.DAT
Réponse 4	" de DICO4. DAT
Réponse Q	sortie du programme GESTION F

- Entrée de données.

CLE	:	DONNEE 1	:
NOM1	:	DONNEE 2	:
NOM2	:	DONNEE 3	:
NOM4	:		:
NOM5	:		:
NOM6	:	DONNEE 4	:
NOM7	:		:

- Affichage d'un nouveau menu

FAITES VOTRE CHOIX :	E :	ENREGISTRER
	N :	NE PAS ENREGISTRER
	R :	RECOMMENCER

Celui-ci nous laisse 3 possibilités :

E : Enregistrement de l'enregistrement et retour au menu précédent
N : Retour au menu précédent
R : Réécriture ou correction des données

- Retour au menu précédent
- Retour au menu principal

REPONSE A :

- Affichage d'un menu

AFFICHAGE :	NOM DE L'ENREGISTREMENT :	
	<R.C> POUR QUITTER "AFFICHAGE" :	

2 possibilités :

<R.C> = Retour au menu principal
<NOM (CLE)> = Affichage de l'enregistrement correspondant

- Entrée d'un mot clé
- Affichage de l'enregistrement correspondant accompagné d'un menu.

AFFICHAGE :	TAPEZ <ESPACE>	
-------------	----------------	---

Celui-ci nous permet de revenir au menu précédent

- Retour au menu précédent
- Retour au menu principal

REPONSE M :

- Affichage d'un menu

MODIFICATION : NOM DE L'ENREGISTREMENT :  <R.C> POUR QUITTER " MODIFICATION"
--

2 possibilités :

<MOT CLÉ> = Affichage de l'enregistrement correspondant
<R.C> = Retour au menu principal

- Entrée d'un mot clé
- Affichage de l'enregistrement correspondant accompagné d'un menu

MODIFIER : NOUVELLES VALEURS <R.C> POUR QUITTER
--

2 possibilités :

<NOUVELLES VALEURS> = Modification
<R.C> = Retour au menu "MODIFICATION"
Pour non modification et passage au champ suivant faire
<R.C>

- Modification
- Affichage d'un nouveau menu

MODIFIER : E = ENREGISTREMENT N = NE PAS ENREGISTRER

VOTRE CHOIX : 

Celui-ci nous laisse 2 possibilités

E : enregistrement des nouvelles valeurs
N : annuler les modifications

- Retour au menu modification
- Retour au menu principal

REPONSE D :

- Affichage d'un menu

DESTRUCTION : NOM DE L'ENREGISTREMENT <R.C> POUR QUITTER "DESTRUCTION"

2 possibilités :

<MOT CLE> = Destruction de l'enregistrement correspondant
<R.C> = Retour au menu principal

- Retour au menu principal

REPONSE T :

- Affichage de la liste des mots clés dans l'ordre alphabétique
- Retour au menu principal (TAPEZ <ESPACE>)

REPONSE Q :

- Affichage du menu général

REPONSE Q :

- Sortie de GESTION F

4. PROGRAMME GRILLE

GRILLE permet la création et la gestion du fichier ART.DAT. Celui-ci contient tous les paramètres nécessaires à l'affichage d'une grille.

1) Composition d'une grille

- a) Structure : On dispose de 10 champs d'affichage de structure de 25 caractères chacun. Il est à noter qu'une fois ces champs créés, on ne peut plus les modifier, il faut détruire la grille.

exemple de structure :

	PRIX/KG	POIDS	TOTAL

- b) Données : Ces données sont celles contenues dans les fichiers dictionnaires DICO. On dispose de 10 possibilités d'affichage. Il est important de remarquer que ce sont des paramètres d'accès à un champ de données d'un fichier dictionnaire que l'on fournit, donc ces données pourront être modifiées à volonté par le programme GESTION F, à la suite d'une mise à jour par exemple.

exemple :

	PRIX/KG	POIDS	TOTAL
FONTE	10		
ACIER	7		

Les données affichées sont FONTE - 10 - ACIER - 7.

Dans ce cas s'il y a un changement de prix, il suffit de modifier le champ dans lequel se trouve les prix. Cette modification se fera avec le programme GESTION F.

- c) Saisies : On dispose de 10 champs de saisie. Une saisie peut être de 3 types :

- Chaîne de caractères
- Entier
- Réel

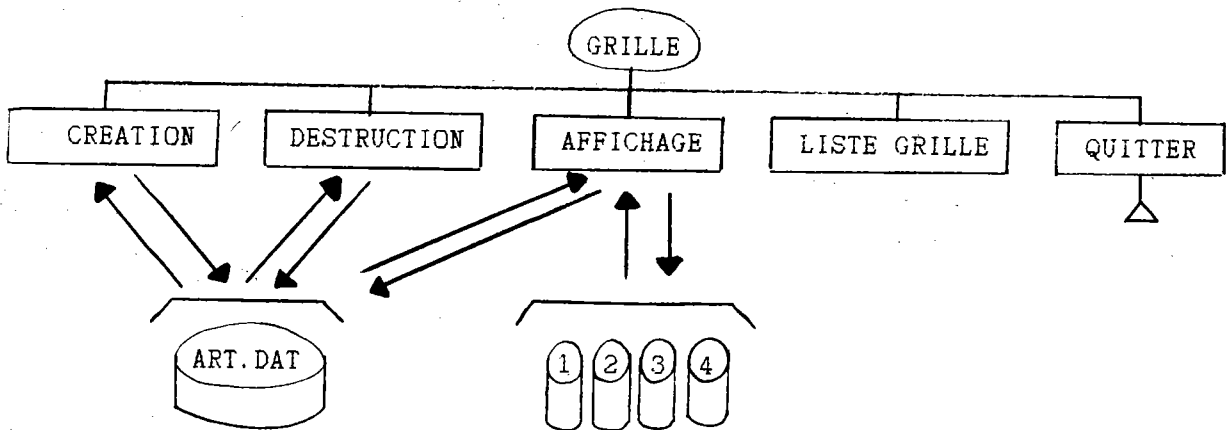
On peut réaliser un traitement sur chacune d'elle, il suffit de préciser le n° de celui-ci. (Voir module TRAIT).

exemple :

	PRIX/KG	POIDS	TOTAL
FONTE	10	30	300
ACIER	7	10	70

Les saisies sont : 30 - 10 .
300 et 70 sont les résultats des traitements associés à chaque saisie.
Les traitements consistent à faire 10 x 30 puis à afficher le résultat.

2) Organigramme général de GRILLE



3) Utilisation de GRILLE

- Chargement de GRILLE
- Affichage d'un menu principal

CREATION GRILLE	: C
AFFICHAGE GRILLE	: A
DESTRUCTION GRILLE	: D
LISTE GRILLES	: T
QUITTER "GRILLE"	: Q

VOTRE CHOIX : ☐

REPONSE C :

- Affichage d'un menu

CREATION : NOM DE LA GRILLE : ☐
<R.C> POUR QUITTER "CREATION"

2 possibilités :

<NOM GRILLE> = Affichage d'un menu
<R.C> = Retour au menu principal

- Entrer nom de la grille
- Affichage d'un menu

NOMBRE DE CHAMPS "STRUCTURE" (<10) : ☐

On choisit le nombre de champs de structure que l'on veut créer.

Réponse : $0 < \text{nombre} \leq 10$ Affichage du menu concernant les
paramètres de ces champs.
Si Nombre = 0 Affichage du menu concernant les
champs de "Données"

- Entrée = 1
- Affichage menu

CHAMP NUMERO 1 : NL = _____ NC = _____

CHAMP A AFFICHER : _____ ;

ATTRIBUT : I>nverse, N>ormal, S>ouligné ?

CONFIRMATION DES DONNEES O/N ?

Par un jeu de questions-réponses, on rentre les paramètres d'affichage :

- NL = numéro de ligne $7 \leq \text{NL} \leq 19$
- NC = numéro de colonne $0 \leq \text{NC} \leq 77$
- CHAMP A AFFICHER < 25 caractères
- ATTRIBUT : concerne l'affichage
Inverse = vidéo inverse
Normal = vidéo normale
Souligné = le champ sera souligné

- Confirmation : On conserve la possibilité de recommencer l'introduction des données. Ce menu sera réaffiché autant de fois qu'il y a de champs à créer.
- Plus de champ "STRUCTURE" à créer et confirmation.
- Affichage d'un menu.

NOMBRE DE CHAMPS "DONNEE" (<10) ? 

On choisit le nombre de champs d'affichage de données.

0 < nombre ≤ 10 affichage du menu concernant les paramètres d'entrée.

Si nombre = 0 affichage du menu concernant les saisies.

- Entrée = 1
- Affichage menu

CHAMP NUMERO 1 :
NL = _____ NC = _____
NOM DU DICO = _____
CLE DE L'ENREGISTREMENT = _____
NOM DU CHAMP = _____
ATTRIBUT : I>nverse, N>ormal, S>ouligné ?
CONFIRMATION DES DONNEES O/N ?

Par un jeu de questions-réponses on rentre les paramètres suivants :

NL = numéro de ligne coordonnées d'affichage

NC = numéro de colonne de la donnée

NOM DU DICO = nom du dictionnaire
(DICO1.DAT ; DICO2.DAT ; DICO3.DAT ; DICO4.DAT)
dans lesquels se trouve la donnée à afficher.

CLE DE L'ENREGISTREMENT = nom de la clé de l'enregistrement où se trouve la donnée à afficher.

NOM DU CHAMP = nom du champ dans lequel se trouve la donnée à afficher

ATTRIBUT : choix de l'attribut d'affichage
I = inverse vidéo
N = vidéo normal
S = caractère souligné.

CONFIRMATION DES DONNEES O/N ?

O = OUI

N = NON Recommencer.

Ce menu sera réaffiché autant de fois qu'il y a de champ à créer.

- Plus de champ à créer et confirmation.
- Affichage d'un menu.

NOMBRE DE CHAMPS DE "SAISIE" (<10) ?

On choisit le nombre de saisies désirées.

0 < nombre < 10

Affichage du menu concernant les paramètres d'entrée.

Si nombre = 0

Affichage du prochain menu.

- Entrée = 1
- Affichage menu

CHAMP NUMERO 1 :

- NL = _____ NC = _____

- TYPE DE SAISIE A EFFECTUER :

C>haîne E>ntier R>éel ? ____

si réponse E.R :

Borne inf. = _____

Borne sup. = _____

- NUMERO DE TRAITEMENT A EFFECTUER = _____

- RENTRER MENU CONCERNANT LA SAISIE :

CONFIRMATION DES DONNEES O/N ? ____

Introduction des paramètres :

- NL = numéro de ligne
- NC = numéro de colonne
- TYPE DE SAISIE A EFFECTUER : Si la saisie est du Type E ou R alors il faut préciser :
 - C = chaîne de caractère
 - E = nombre entier
 - R = nombre réel
 - BORNE INF = borne inférieure
 - BORNE SUP = borne supérieure
- NUMERO DE TRAITEMENT A EFFECTUER
 - <R.C> = s'il n'y a pas de traitement
 - <numéro> = la saisie sera appelée par le traitement correspondant
- MENU : chaque saisie possède son propre menu (<1 ligne>).

CONFIRMATION DES DONNEES O/N ?

- O = confirmation et passage au menu suivant
- N = recommencer

Ce menu sera réaffiché autant de fois qu'il y a de champ de saisie à créer.

Plus de champ à créer et confirmation ⇒ passage au menu suivant.

- Affichage menu

ENREGISTREMENT DE LA GRILLE O/N : ____

- "O" = grille à enregistrer
- "N" = grille à ne pas enregistrer

- Affichage d'un menu

AUTRE CREATION DE GRILLE : O/N ?

- "O" = retour au menu création
- "N" = retour au menu principal

REPONSE A :

- Affichage d'un menu

AFFICHAGE : NOM DE LA GRILLE :
TAPEZ <R.C> POUR QUITTER "AFFICHAGE"

<R.C> retour au menu principal
<nom grille> affichage de la grille concernée

REPONSE D :

- Affichage d'un menu

DESTRUCTION : NOM DE LA GRILLE :
 <R.C> POUR QUITTER

<R.C> retour au menu principal
<nom grille> destruction de la grille concernée

REPONSE T :

- Affichage de la liste des noms de grilles dans l'ordre alphabétique.
- Tapez <ESPACE> retour au menu principal

REPONSE Q :

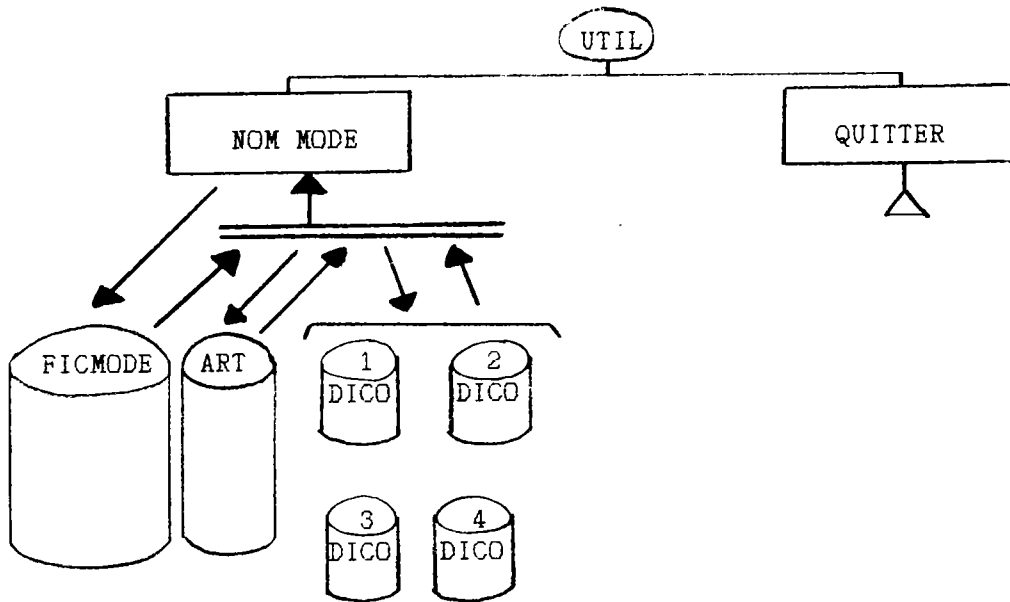
- SORTIE DE "GRILLE"

5. PROGRAMME UTIL

UTIL est le programme utilisateur, c'est à dire celui qui utilise toutes les fonctions qui ont été créés par les programmes GESTION F - GRILLE - MODE.

L'introduction du nom d'un mode entraîne le processus d'affichages et de saisies.

1) Organigramme général de UTIL



La structure du menu est définie comme une grille.

2) Utilisation de UTIL

- Chargement de UTIL
- Affichage d'un menu

RENTREZ NOM MODE :
<R.C> POUR QUITTER

Le programme UTIL appelle la grille MENU UTIL

2 possibilités :

- | | |
|--------------|--|
| - <nom mode> | L'affichage séquentiel des grilles
contenues dans ce mode |
| - <R.C> | Sortie de UTIL |

MODULE TRAIT

Fonctionnement :

C'est la partie la plus importante car c'est elle qui est à la base de la décision. Lors d'une fin de saisie (saisie sur lequel doit être effectué un traitement numéro n) le programme envoie le numéro de traitement au module qui sélectionne ce dernier.

Caractéristiques d'un traitement :

Un traitement doit être écrit en PASCAL, ce qui lui confère toutes les possibilités de celui-ci. Donc le module TRAIT doit être compilé avec le programme GRILLE et UTIL.

Il est à noter qu'il a été mis à la disposition de l'utilisateur un ensemble de procédures facilitant l'affichage sur l'écran et permettant :

- l'affichage de message en bas de l'écran
- un accès en lecture à la base de données.

Procédures:

- NOR : sélectionne la vidéo normale
- INV : sélectionne la vidéo inverse
- SOULIGNE : écriture soulignée des prochains caractères
- GO TO XY (X,Y) : positionne le curseur X = ligne Y = colonne
- RECH DONNEE (DICTIONNAIRE, ARTICLE, CHAMP, DONNEE) :
procédure permettant l'accès à la base de données

ENTREES	{	DICTIONNAIRE	= nom du dictionnaire où se trouve l'article concerné.
		ARTICLE	= nom (mot clé) de l'article où se trouve le champ.
		CHAMP	= nom du champ où se trouve la donnée.

SORTIE	{	DONNEE	= contient la donnée .
--------	---	--------	------------------------

- CHAI REEL (CHAINE, REEL, BIEN) ;
procédure de conversion de chaîne de caractères en nombre réel.

ENTREE : CHAINE : chaîne de caractères à convertir.

SORTIES : REEL : nombre réel correspondant
BIEN : valeur booléenne
vraie bien déroulée
faux erreur sur l'écriture de la chaîne

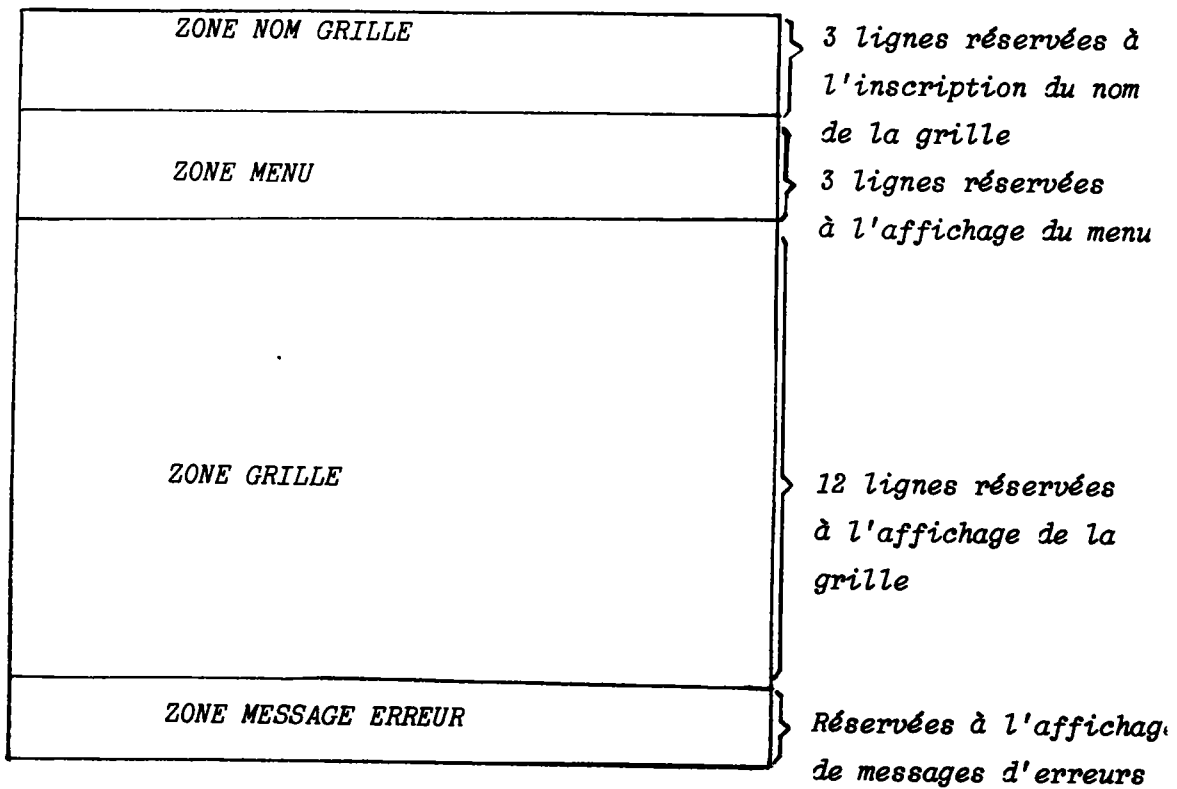
- CHAI ENTIER (CHAINE, ENTIER, BIEN) ;
procédure de conversion de chaîne de caractères
en nombre entier.

ENTREE : CHAINE : chaîne de caractères à convertir.

SORTIES : ENTIER : nombre entier correspondant
BIEN : valeur booléenne
vraie conversion correcte
faux erreur

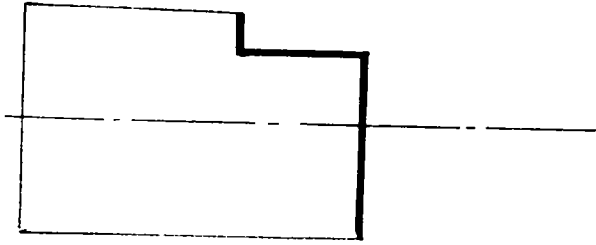
6. EXEMPLE D'UTILISATION

Pour une bonne utilisation de l'écran, il est important de délimiter des zones d'affichages spécifiques. Pour cela, on a partagé l'écran en quatre zones



Le but de cet exemple est de donner un aperçu des possibilités et de l'utilité d'un tel système pour l'élaboration de gammes d'usinages.

Réalisation d'une opération de dressage et de surfacage sur une pièce cylindrique

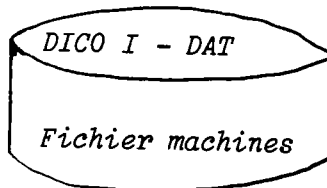


1/ Phase d'implantation

- Création de la B.D. Utilisation du programme

GESTION F

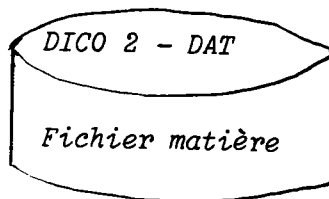
. Fichier machines



Contient les renseignements concernant chaque type de machine.

Le nom de la machine est utilisé comme clé d'accès

. Fichier Matière



Répertoire des matières et des caractéristiques correspondant

Le nom de la matière étant utilisé comme clé d'accès

. GRILLE PRISES FR : Identique à PRISES TR
mais concerne le fraisage

. GRILLE MACHINES : Cette grille concerne le
choix de la machine suivant deux paramètres :
- le volume hors tout de la pièce à usiner
- le nombre de pièces

. GRILLE SURFACAGE : Cette grille concerne
l'opération de surfacage en fraisage. On a
l'affichage de tous les types d'outils
permettant cette opération.

Après l'introduction de plusieurs paramètres,
celle-ci nous affiche la vitesse de rotation,
l'avance adéquate avec le temps et le coût
d'usinage.

. GRILLE DRESSAGE : Identique à Surfacage mais
concerne une opération de dressage sur un tour

. GRILLE COMPTABLE : Grille destinée au service
comptable. Après avoir introduit les coordonnées
du préparateur et de la pièce, celle-ci affiche
le temps total d'usinage avec son coût respectif.

- Création des modes

On va créer un mode nommé "CAME" qui a pour
objectif l'affichage des grilles dans un ordre
imposé.

MACHINES

PRISES PR

PRISES FR

DRESSAGE

SURFACAGE

COMPTABLE

2/ Phase utilisation

UTIL

On sélectionne le mode "CAME" et l'on obtient dans l'ordre
l'affichage des grilles suivantes :

*** GRILLE MACHINES ***

-RENSEIGNEMENTS MACHINES-

ENCOMBREMENT PIECE - 25

NOMBRE DE PIECES - 1

PHASE DE TOURNAGE O/N : 0 TOUR APPROPRIE

TOUR T1

PHASE DE FRAISAGE O/N : 0 FRAISEUSE APPROPRIEE

FRAIS-F1

PHASE DE PERCAGE O/N : N PERCEUSE APPROPRIEE

PERANT

TAPEZ <ESPACE> POUR GRILLE SUIVANTE

*** GRILLE PRISES TR ***

-MONTAGE D'USINAGE-

CHOIX: MANDRIN MORS DURS : 1
MANDRIN MORS DOUX : 2
PLATEAU : 3

REPOSE - 1

TEMPS DE MONTAGE : 300

COUT DE L'OPERATION : 1.2

TAPEZ <ESPACE> POUR GRILLE SUIVANTE

Légende

Ecriture manuscrite : Saisie



: Affichage d'une donnée de la B.D.



: Affichage du résultat d'un traitement

*** GRILLE PRISES FR ***

-MONTAGE D'USINAGE-

CHOIX:

ETAU	: 1
PLATEAU	: 2
BRIDES	: 3

REPONSE - 1

TEMPS DE MONTAGE : 600
COUT DE L'OPERATION : 2.4

TAPEZ <SPACE> POUR GRILLE SUIVANTE

*** GRILLE DRESSAGE ***

-OPERATION DRESSAGE-

OUTILS DISPONIBLES:

CHAR. A DROITE (ARS) : 1 CHAR. (P10) : 3
CHAR.DRES. (ARS) : 2 COUTEAU (P10) : 4

MATIERE A USINER - XC30
EPAISSEUR AUSINER - 2
FINITION DESIREE - 1
TYPE D'OUTIL : 2
DIAMETRE PIECE - 40

TOUR/MN AVANCE
500 0.1

TEMPS D'USINAGE : 30
COUT DE L'OPERATION : 0.15

TAPEZ <SPACE> POUR GRILLE SUIVANTE

Légende:

Ecriture manuscrite : Saisie



: Affichage d'une donnée de la B.D.



: Affichage du résultat d'un traitement

*** GRILLE SURFACAGE ***

*-OPERATION SURFACAGE-

OUTILS DISPONIBLES:

FRAISE 2 TAILLES : 1
FRAISE TOURTEAU : 2

FRAISE 2 TAILLES (P10) : 3
FRAISE TOURTEAU (P10) : 4

MATIERE A USINER : XC38
LONG = 30 LARG = 20

TYPE FRAISE : 3

NOMBRE DE DENTS = 8

EPAISSEUR DE PASSE = 3

TOUR/MM

200

AVANCE

0.18

TEMPS D'USINAGE : 120

COUT DE L'OPERATION : 0.7

TAPEZ <ESPACE> POUR GRILLE SUIVANTE

*** GRILLE COMPTABLE ***

NOM PREPARATEUR : DUPONT

NOM PIECE : TETON

TEMPS TOTAL D'USINAGE : 1050

COUT TOTAL : 4.45

DATE: 21-10-83

TAPEZ <ESPACE> POUR GRILLE SUIVANTE

- TAPEZ <ESPACE> POUR QUITTER-

Légende:

Ecriture manuscrite : Saisie



: Affichage d'une donnée de la B.D.



: Affichage du résultat d'un traitement