



HAL
open science

Communication en langue naturelle avec un système d'aide à la conception d'assemblages physiques : un essai d'utilisation de réseaux sémantiques partitionnés

Mauricio Lopez

► **To cite this version:**

Mauricio Lopez. Communication en langue naturelle avec un système d'aide à la conception d'assemblages physiques : un essai d'utilisation de réseaux sémantiques partitionnés. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1979. Français. NNT: . tel-00290122

HAL Id: tel-00290122

<https://theses.hal.science/tel-00290122>

Submitted on 24 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de

**DOCTEUR INGENIEUR
«Génie Informatique»**

par

Mauricio LOPEZ

**COMMUNICATION EN LANGUE NATURELLE
AVEC UN SYSTEME D'AIDE A LA CONCEPTION
D'ASSEMBLAGES PHYSIQUES:**

**UN ESSAI D'UTILISATION DE RESEAUX
SEMANTIQUES PARTITIONNES.**

Thèse soutenue le 29 septembre 1979 devant la commission d'examen.

L. BOLLIET Président

J. FOISSEAU

A. MASSABO

J.C. LATOMBE Examineurs

J.L. LAURIERE

G. VEILLON

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de

**DOCTEUR INGENIEUR
«Génie Informatique»**

par

Mauricio LOPEZ

**COMMUNICATION EN LANGUE NATURELLE
AVEC UN SYSTEME D'AIDE A LA CONCEPTION
D'ASSEMBLAGES PHYSIQUES:**

**UN ESSAI D'UTILISATION DE RESEAUX
SEMANTIQUES PARTITIONNES.**

Thèse soutenue le 29 septembre 1979 devant la commission d'examen.

L. BOLLIET Président

J. FOISSEAU

A. MASSABO

J.C. LATOMBE Examineurs

J.L. LAURIERE

G. VEILLON

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1977-1978

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET

M. Georges LESPINARD

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Electronique - automatique
BESSON Jean	Chimie minérale
BLOCH Daniel	Physique du solide - cristallographie
BONNETAIN Lucien	Génie chimique
BONNIER Etienne	Métallurgie
* BOUDOURIS Georges	Electronique - automatique
BRISSONNEAU Pierre	Physique du solide - cristallographie
BUYLE-BODIN Maurice	Electronique - automatique
COUMES André	Electronique - automatique
DURAND Francis	Métallurgie
FELICI Noël	Electronique - automatique
FOULARD Claude	Electronique - automatique
LANCIA Roland	Electronique - automatique
LONGÉQUEUE Jean-Pierre	Physique nucléaire corpusculaire
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie - physique
PAUTHENET René	Electronique - automatique
PERRET René	Electronique - automatique
POLOUJADOFF Michel	Electronique - automatique
TRAYNARD Philippe	Chimie - physique
VEILLON Gérard	Informatique fondamentale et appliquée
* en congé pour études	

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuël	Electronique - automatique
BOUVARD Maurice	Génie mécanique
COHEN Joseph	Electronique - automatique
GUYOT Pierre	Métallurgie physique
LACOUME Jean-Louis	Electronique - automatique
JOUBERT Jean-Claude	Physique du solide - cristallographie

.../...

MM. ROBERT André	Chimie appliquée et des matériaux
ROBERT François	Analyse numérique
ZADWORNY François	Electronique - automatique

MAITRES DE CONFERENCES

MM. ANCEAU François	Informatique fondamentale et appliquée
CHARTIER Germain	Electronique - automatique
CHIAVERINA Jean	Biologie, biochimie, agronomie
IVANES Marcel	Electronique - automatique
LESIEUR Marcel	Mécanique
MORET Roger	Physique nucléaire - corpusculaire
PIAU Jean-Michel	Mécanique
PIERRARD Jean-Marie	Mécanique
SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme SAUCIER Gabrielle	Informatique fondamentale et appliquée
M. SOHM Jean-Claude	Chimie Physique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M. FRUCHART Robert	Directeur de Recherche
MM. ANSARA Ibrahim	Maître de Recherche
BRONOEL Guy	Maître de Recherche
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DRIOLE Jean	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Doré	Maître de Recherche
MATHIEU Jean-Claude	Maître de Recherche
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique) E.N.S.E.E.G.

MM. BISCONDI Michel	Ecole des Mines St. Etienne (dépt. Métallurgie)
BOOS Jean-Yves	Ecole des Mines St. Etienne (Métallurgie)
DRIVER Julian	Ecole des Mines St. Etienne (Métallurgie)

MM. KOBYLANSKI André	Ecole des Mines St. Etienne (Métallurgie)
LE COZE Jean	Ecole des Mines St. Etienne (Métallurgie)
LESBATS Pierre	Ecole des Mines St. Etienne (Métallurgie)
LEVY Jacques	Ecole des Mines St. Etienne (Métallurgie)
RIEU Jean	Ecole des Mines St. Etienne (Métallurgie)
SAINFORT	C.E.N. Grenoble (Métallurgie)
SOUQUET	U.S.M.G.
CAILLET Marcel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
COULON Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
GUILHOT Bernard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
LALAUZE René	Ecole des Mines St. Etienne (Chim. Min. Ph.)
LANCELOT Francis	Ecole des Mines St. Etienne (Chim. Min. Ph.)
SARRAZIN Pierre	Ecole des Mines St. Etienne (Chim. Min. Ph.)
SOUSTELLE Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
THEVENOT François	Ecole des Mines St. Etienne (Chim. Min. Ph.)
THOMAS Gérard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
TOUZAIN Philippe	Ecole des Mines St. Etienne (Chim. Min. Ph.)
TRAN MINH Canh	Ecole des Mines St. Etienne (Chim. Min. Ph.)

E.N.S.E.R.G.

MM. BOREL	Centre d'études nucléaires de Grenoble
KAMARINOS	Centre national recherche scientifique

E.N.S.E.G.P.

M. BORNARD	Centre national recherche scientifique
Mme CHERUY	Centre national recherche scientifique
MM. DAVID	Centre national recherche scientifique
DESCHIZEAUX	Centre national recherche scientifique

A mis viejos.

Je tiens à remercier

Monsieur Louis BOLLINET, Professeur à l'Université des Sciences Sociales de Grenoble qui m'a fait l'honneur de présider le jury de cette thèse.

Monsieur Gérard VEILLON, Professeur à l'Institut National Polytechnique de Grenoble, qui m'a accueilli dans son équipe et qui a accepté de juger ce travail.

Monsieur Jean-Claude LATOMBE, Maître-Assistant à l'Institut National Polytechnique de Grenoble, qui a dirigé la réalisation de ce travail. Je tiens à lui exprimer ma gratitude pour son accueil chaleureux et pour ses nombreux conseils et encouragements.

Monsieur Alain MASSABO, Responsable de l'équipe de Conception et Fabrication Assistée par Ordinateur à la Société BATELLE-GENEVE, qui a accepté de juger cette thèse.

Messieurs Jean-Louis LAURIERE, Maître de Conférences à l'Université de Paris VI, et Jacques FOISSEAU, Ingénieur au CERT-DERI, qui ont accepté de participer au jury.

Tous ceux qui d'une manière ou d'une autre ont contribué à la réalisation de cette thèse. Je me dois de mentionner : Augustin LUX, Julio LOPEZ, Ernest GRANDJEAN, Jacques COURTIN et Michel VERNET.

Je tiens à remercier particulièrement Augustin et Irène LUX pour l'accueil et l'amitié qu'ils m'ont témoignés, et Maryvonne DUC-LOPEZ (ma femme) pour la patience dont elle a fait preuve pendant la réalisation du manuscrit.

Je voudrais aussi remercier Madame DIAZ qui assuré la dactylographie de ce texte avec une grande compétence et avec beaucoup de rapidité, et Monsieur D. IGLESIAS et le personnel du service de reproduction qui en ont réalisé le tirage.

Le travail décrit dans cette thèse a été réalisé dans le cadre du contrat IRIA/SENSORI n° 77 052.

SOMMAIRE

INTRODUCTION -----	1
CHAPITRE I - PRESENTATION GENERALE DE L'INTERFACE -----	3
1. Les connaissances décrites à TROPIC -----	5
2. Vue d'ensemble de l'interface -----	6
3. Analyses morphologique et syntaxique -----	8
4. Réduction et filtrage des arborescences syntaxiques -----	10
5. Structure et construction du RSP -----	11
6. Conclusion - Résumé -----	15
CHAPITRE II - LA BASE DE CONNAISSANCES -----	19
1. Les réseaux sémantiques partitionnés (RSP) -----	21
2. La notion de figure -----	26
3. Description d'une classe d'objets -----	30
4. Description des relations entre objets -----	32
5. Description des propriétés d'une classe d'objets -----	36
6. Description des composants d'un assemblage -----	42
7. Les groupes d'objets -----	44
8. Les contraintes -----	48
9. Figure de conseil -----	54
10. Récapitulatif des éléments d'un RSP connus a priori des programmes de l'interface -----	56
CHAPITRE III - ANALYSES SYNTAXIQUE ET SEMANTIQUE -----	59
1. Analyses morphologique et syntaxique -----	61
2. Analyse sémantique -----	68

CHAPITRE IV - CONSTRUCTION DU RESEAU SEMANTIQUE PARTITIONNE-----	87
1. Construction du réseau correspondant à une phrase -----	89
2. Détermination des nouvelles informations apportées par une phrase -----	92
3. Insertion des nouvelles informations dans le RSP -----	104
4. Normalisation des formes réduites -----	108
 CHAPITRE V - COMMUNICATION RSP-TROPIC -----	 111
1. Le modèle de résolution de problème utilisé par TROPIC -----	113
2. Les règles de synthèse -----	116
3. Les règles d'analyse -----	126
4. Les conseils -----	135
 CHAPITRE VI - DISCUSSION, IMPLANTATION, EXPERIMENTATION -----	 137
1. Discussion -----	139
2. Implantation -----	143
3. Expérimentation -----	152
4. Trois problèmes de représentation dans le RSP -----	165
 CONCLUSION -----	 171
 APPENDICE A - LES REGLES DE TRANSFORMATION -----	 173
APPENDICE B - EXEMPLE DE DESCRIPTION D'UNE TOUR -----	179
APPENDICE C - EXEMPLE DE DESCRIPTION D'UN TRANSFORMATEUR -----	193
APPENDICE D - ALGORITHMES POUR LA CONSTRUCTION DU SOUS RESEAU CORRESPONDANT A UNE PHRASE -----	205
 BIBLIOGRAPHIE -----	 213

INTRODUCTION

Depuis cinq années, l'équipe d'Intelligence Artificielle de l'IMAG a conduit des recherches dans le domaine du traitement automatique des langues naturelles. Ces recherches ont donné lieu au développement de plusieurs outils. Parmi eux figure le système PIAF (Courtin 1977), conçu pour effectuer les analyses morphologique et syntaxique d'énoncés en langue naturelle. Dans la même voie, deux programmes pour le traitement post-syntaxique d'énoncés ont été réalisés par V.Joloboff(1978) et J.Lopez(1979) ; ils permettent notamment de construire des structures reflétant des interprétations particulières d'un énoncé. Les algorithmes de traitement de ces outils sont indépendants des données linguistiques, ce qui permet leur expansion progressive et leur application à des domaines variés.

Parallèlement, dans le cadre de la même équipe, un système de résolution de problèmes pour la conception assistée par ordinateur, TROPIC (Latombe 1977), a été réalisé. Ce système permet le traitement de problèmes de conception d'assemblages physiques et son utilisation n'est pas restreinte à une discipline technique particulière. En effet, il n'incorpore pas de connaissances physiques et technologiques a priori ; celles-ci doivent être fournies par l'utilisateur sous la forme de "règles de production" dont l'écriture est souvent difficile pour un utilisateur non spécialiste.

Notre travail établit un lien entre ces deux recherches. Il consiste en la conception et la réalisation d'un interface de communication avec le système TROPIC, incluant le système PIAF et le programme de transformation d'arborescences de J.Lopez. Cet interface permet d'exprimer les connaissances nécessaires à TROPIC en un sous-ensemble du français que nous avons essayé de rendre aussi indépendant que possible de la structure de données de TROPIC.

Nous avons utilisé le système PIAF pour réaliser les analyses morphologique et syntaxique des phrases. Le résultat de ces analyses sont des structures

arborescentes, qui sont ensuite transformées en formes réduites contenant une signification de la phrase limitée au contexte des assemblages physiques (analyse sémantique). Cette transformation est réalisée en utilisant le programme de J.Lopez. L'interprétation des formes réduites permet alors de représenter les informations qu'elles apportent dans une base de connaissances du type réseau sémantique partitionné (Hendrix, 1975). Le modèle de représentation réalisé repose sur la notion originale de figure, structure regroupant un ensemble significatif de connaissances élémentaires.

Notre travail se situe donc dans le prolongement de plusieurs recherches de l'équipe d'Intelligence Artificielle de l'IMAG.

Les aspects les plus importants de l'interface réalisé sont les suivants :

- 1/ Il intègre des programmes déjà réalisés dans l'équipe ;
- 2/ Il utilise un réseau sémantique partitionné comme structure intermédiaire entre la langue naturelle et les structures de données de TROPIC ;
- 3/ Ses programmes ont été réalisés pour être aussi indépendants que possible des connaissances spécifiques du domaine d'application.

La présentation de la thèse est organisée comme suit :

Le chapitre 1 montre le type de phrases traitées par l'interface et donne une vue globale de celui-ci.

Les chapitres 2, 3, 4 et 5 décrivent les différentes parties de l'interface.

Le chapitre 6 discute quelques aspects généraux de l'interface, montre le principe de son implantation et donne quelques exemples de son expérimentation.

Chapitre I

PRESENTATION GENERALE DE L'INTERFACE

1 - LES CONNAISSANCES DECRITES A TROPIC -----	5
2 - VUE D'ENSEMBLE DE L'INTERFACE -----	6
3 - ANALYSES MORPHOLOGIQUE ET SYNTAXIQUE -----	8
4 - REDUCTION ET FILTRAGE DES ARBORESCENCES SYNTAXIQUES -----	10
5 - STRUCTURE ET CONSTRUCTION DU RSP -----	11
6 - CONCLUSION - RESUME -----	15

1 - LES CONNAISSANCES DECRITES A TROPIC

TROPIC (Latombe 1977) est un système de résolution de problèmes dont le domaine d'application est la conception d'assemblages physiques (machines électriques par exemple). Sa fonction est de construire le modèle d'un appareil satisfaisant un cahier de charges donné.

Pour TROPIC, un assemblage est constitué d'objets appartenant chacun à une certaine classe (transformateur, circuit magnétique,...). Ces objets sont reliés par des relations de composition physique (ex: le transformateur T est composé du circuit magnétique CM et du circuit électrique CE) et par des relations d'assemblage (ex: l'enroulement électrique E entoure le noyau magnétique N). De plus, chaque objet est muni de propriétés : par exemple, à un transformateur on peut associer les propriétés puissance, fréquence, rendement, tension primaire, tension secondaire, réfrigérant, ...

L'utilisation de TROPIC n'est pas restreinte à une discipline technique particulière. Le système n'incorpore donc pas de connaissances physiques et technologiques a priori. Pour qu'il puisse construire le modèle d'un assemblage, on doit lui fournir ces connaissances. Dans le cas d'un transformateur électrique de moyenne puissance, ce sont par exemple :

- Le réfrigérant d'un transformateur est l'huile ou l'air.
- Les composants d'un transformateur sont un circuit magnétique, un circuit électrique et, si le réfrigérant du transformateur est l'huile, une cuve.
- La qualité des tôles du circuit magnétique d'un transformateur est Q1, Q2 ou Q3.
- L'échauffement (E) de l'huile dans un transformateur est fonction de la surface (S) de sa cuve et des pertes (P) dans le transformateur: $E = 445 \left(\frac{P}{S}\right)^{0,8}$.
- Si le réfrigérant d'un transformateur est l'air, alors la tension aux bornes des galettes de ce transformateur doit être inférieure à 3 kv.
- Si la puissance d'un transformateur est supérieure à 100 kva, il est recommandé de choisir l'huile pour son réfrigérant.

L'utilisateur de la version actuelle de TROPIC doit exprimer ces connaissances sous la forme de règles de production (Davis 1976) dont la syntaxe dérive de celle du langage LISP. Ces règles ont une structure qui ne correspond pas en général à celle des phrases en langue naturelle. De plus, elles doivent être classées par l'utilisateur en différentes catégories (règles de synthèse, règles d'analyse, conseils, diagnostics)¹.

Cette structuration des connaissances est orientée vers les méthodes de résolution de problèmes appliquées par TROPIC.

Dans notre travail, nous avons essayé de rendre la description des connaissances aussi indépendante que possible des structures utilisées par TROPIC. L'interface que nous avons réalisé inclut des connaissances sur les assemblages physiques et sur les besoins de TROPIC pour rétablir ces structures ultérieurement.

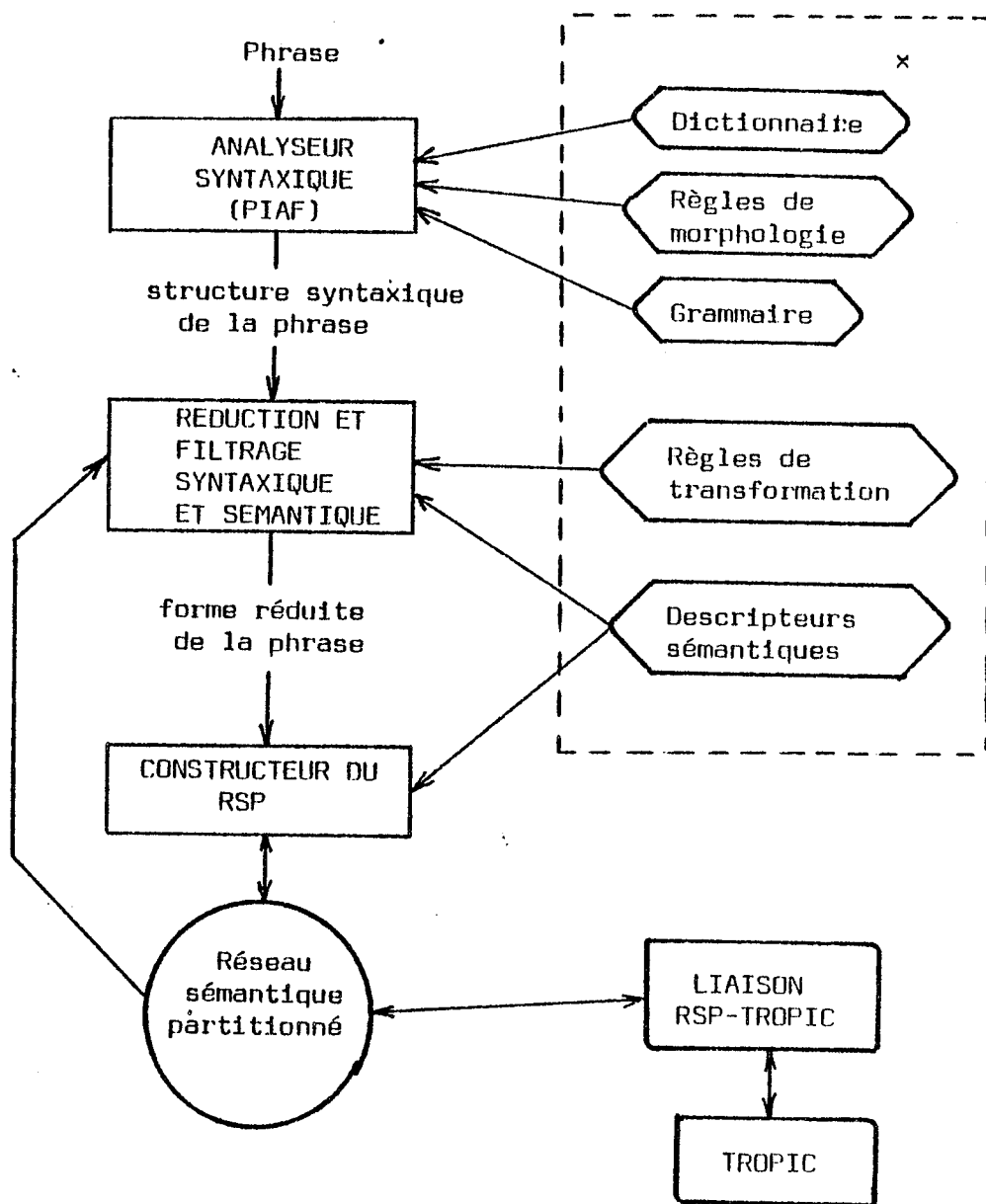
2 - VUE D'ENSEMBLE DE L'INTERFACE

La fonction d'ensemble de l'interface est de permettre la description des connaissances utiles à TROPIC par des phrases exprimées dans un sous-ensemble du français. La figure 1.1. montre la structure de l'interface. Les phrases sont traitées une à une et les connaissances correspondantes sont incorporées au fur et à mesure dans une base de connaissances constituée par un réseau sémantique partitionné (RSP).

a) L'analyse morphologique et syntaxique :

Elle est effectuée par le programme PIAF (Courtin 1977). Pour chaque phrase morphologiquement et syntaxiquement correcte, PIAF délivre une ou plusieurs arborescences de dépendances.

 (1) Les règles de production utilisées par TROPIC sont présentées au chapitre 5.



x Informations contenues dans des fichiers indépendants des programmes

Figure 1.1. - Structure de l'interface

b) La réduction et le filtrage syntaxique et sémantique :

Ils sont effectués par un programme de transformation d'arborescences réalisé par J.Lopez(1979). Les arborescences délivrées par PIAF sont soit transformées en des expressions appelées formes réduites, soit rejetées.

c) La construction du réseau sémantique partitionné :

Elle est effectuée par un programme qui incorpore les nouvelles connaissances dans un RSP (Hendrix 1975) dont la structure a été adaptée au domaine d'application de TROPIC (la conception des assemblages physiques).

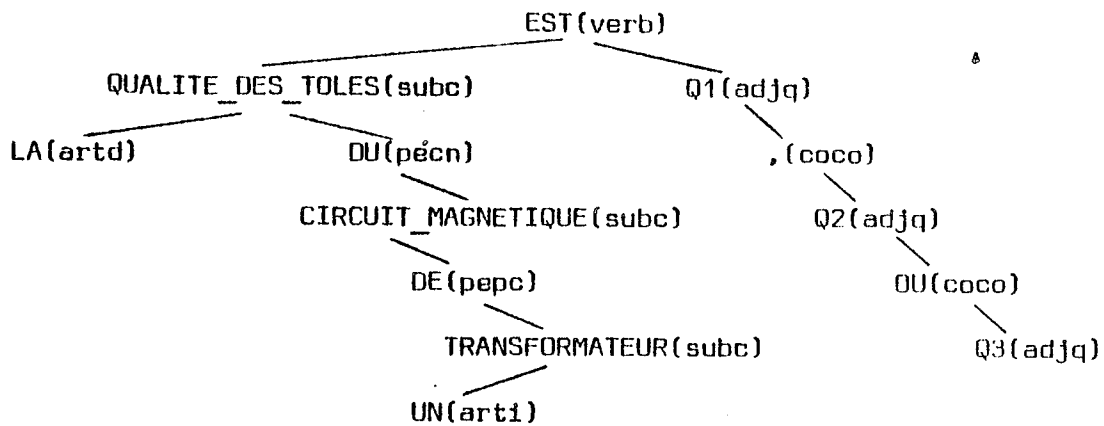
Lorsque l'ensemble de connaissances nécessaires à TROPIC a été décrit, un programme permet de traduire le contenu du RSP en règles de production directement acceptables par TROPIC.

Ils est important de noter que toutes les connaissances de l'interface relatives aux assemblages physiques, et plus spécifiquement aux assemblages électromécaniques, sont contenues dans des fichiers facilement modifiables et indépendants des programmes (cf. figure 1.1.).

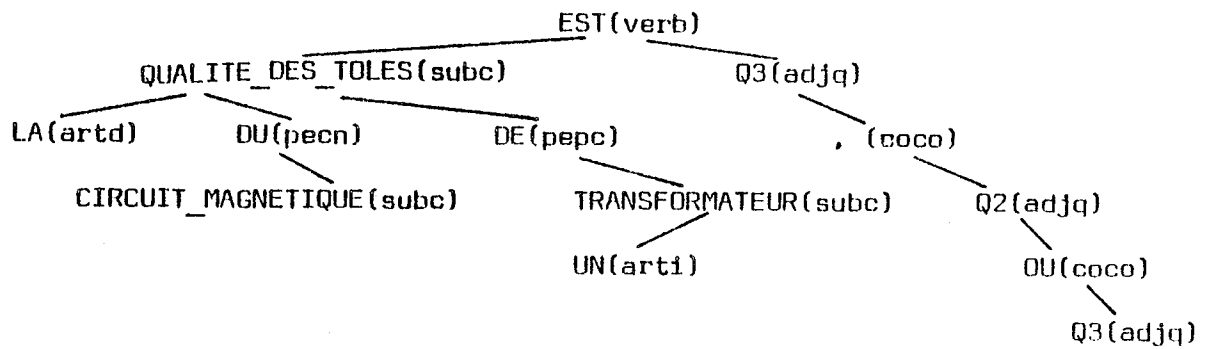
Ces différents aspects de l'interface sont étudiés en détail dans les chapitres suivants. Pour permettre une meilleure compréhension de ces chapitres nous présentons un peu plus précisément les trois phases du traitement d'une phrase dans les paragraphes ci-après.

3 - ANALYSE MORPHOLOGIQUE ET SYNTAXIQUE

La première phase du traitement d'une phrase est l'analyse morphologique et syntaxique. Elle est effectuée par le programme PIAF qui délivre une ou plusieurs arborescences de dépendances (cf. figure 1.2.). L'analyse morphologique utilise un dictionnaire que nous avons constitué avec un vocabulaire approprié à la description d'assemblages électromécaniques, et un ensemble de



(a)



(b)

Figure 1.2. Exemples d'arborescences de dépendances

règles de morphologie du français. L'analyse syntaxique utilise une grammaire de dépendances assez générale destinée à l'analyse de textes où le langage utilisé est très libre (articles de journaux par exemple). Dans la mesure où cela n'entraînait pas trop de complications, nous avons simplifié la grammaire originale de PIAF de façon à réduire le nombre de cas syntaxiques considérés.

La figure 1.2.a montre une arborescence de dépendances correspondant à la phrase "la qualité des tôles du circuit magnétique d'un transformateur est Q1, Q2 ou Q3". A chaque sommet de cette arborescence est associé un mot de la phrase accompagné de sa catégorie lexicale et de ses variables grammaticales (seule la catégorie lexicale est indiquée sur la figure). "Qualité des tôles" et "circuit magnétique" sont considérés comme des unités lexicales car elles figurent comme telles dans le dictionnaire. De même, "Q1", "Q2" et "Q3" figurent dans le dictionnaire avec la catégorie lexicale "adjq" (adjectif qualificatif).

Lorsqu'une phrase comporte des ambiguïtés syntaxiques, PIAF délivre plusieurs arborescences. Ainsi, les figures 1.2a et 1.2b correspondent à deux interprétations syntaxiques de la même phrase.

4 - REDUCTION ET FILTRAGE DES ARBORESCENCES SYNTAXIQUES

La deuxième phase du traitement d'une phrase transforme les arborescences délivrées par PIAF en une expression appelée forme réduite, directement acceptable par le constructeur du réseau sémantique et sensée contenir l'extrait des informations susceptible d'intéresser TROPIC (la "substantifique moelle" de la phrase). La transformation permet également de réduire des paraphrases telles que "un transformateur est composé d'un circuit électrique et d'un circuit magnétique" et "les composants d'un transformateur sont un circuit électrique et un circuit magnétique" à la même forme réduite.

La réduction d'une arborescence de dépendances consiste en une série de transformations. Chacune d'elles est obtenue par application de règles de transformation. La figure 1.3. montre les règles qui sont utilisées dans la figure 1.4. pour transformer l'arborescence de la figure 1.2a. La forme réduite, exprimée sous forme d'une liste, est (QUALITE_DES_TOLES(CIRCUIT_MAGNETIQUE(DE TRANSFORMATEUR) (OU Q1 Q2 Q3))).

La transformation est accompagnée de vérifications syntaxiques et sémantiques permettant d'accepter ou de rejeter l'arborescence considérée. Les vérifications sémantiques utilisent un ensemble de descripteurs sémantiques permettant notamment de distinguer les mots qui désignent des propriétés d'objets des mots qui désignent des relations entre objets. Par exemple, le descripteur (PROPRIETE QUALITE_DES_TOLES) déclare que "qualité des tôles" est le nom d'une propriété d'objet et (RELATION AUTOUR) déclare que "autour" est une relation entre objets.

Les règles de transformation et les descripteurs sémantiques utilisés par l'interface ont été définis pour le contexte des assemblages physiques.

5 - STRUCTURE ET CONSTRUCTION DU RSP

La troisième phase du traitement d'une phrase consiste à incorporer les connaissances extraites de la phrase (forme réduite) à un réseau sémantique partitionné (RSP).

Le formalisme d'un RSP est essentiellement celui d'un graphe étiqueté dont les sommets et les arcs sont répartis en espaces. Ces derniers sont utiles pour délimiter des sous-ensembles d'informations sans les déconnecter du reste du réseau.

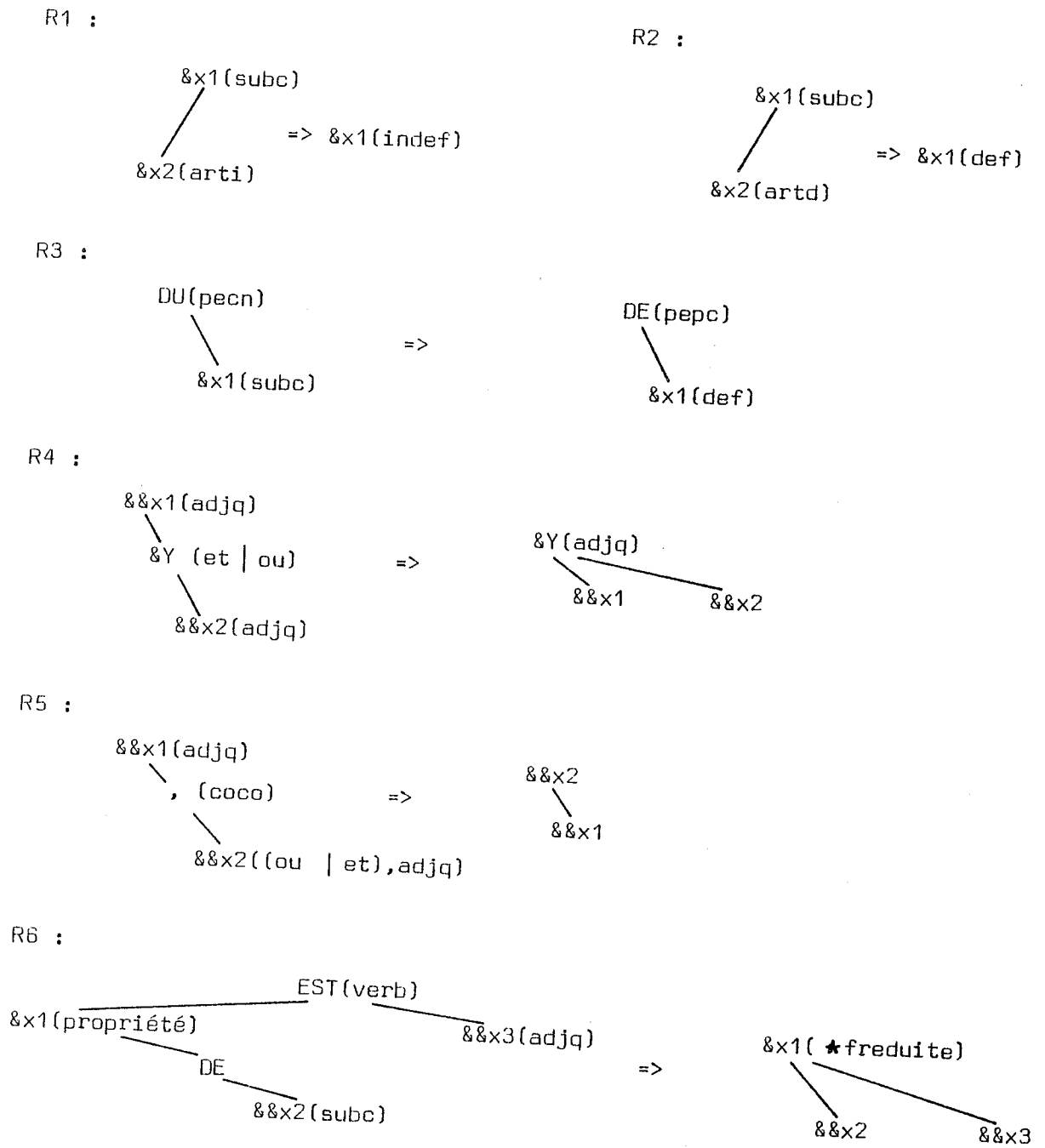


Figure 1.3. - Exemples de règles de transformation

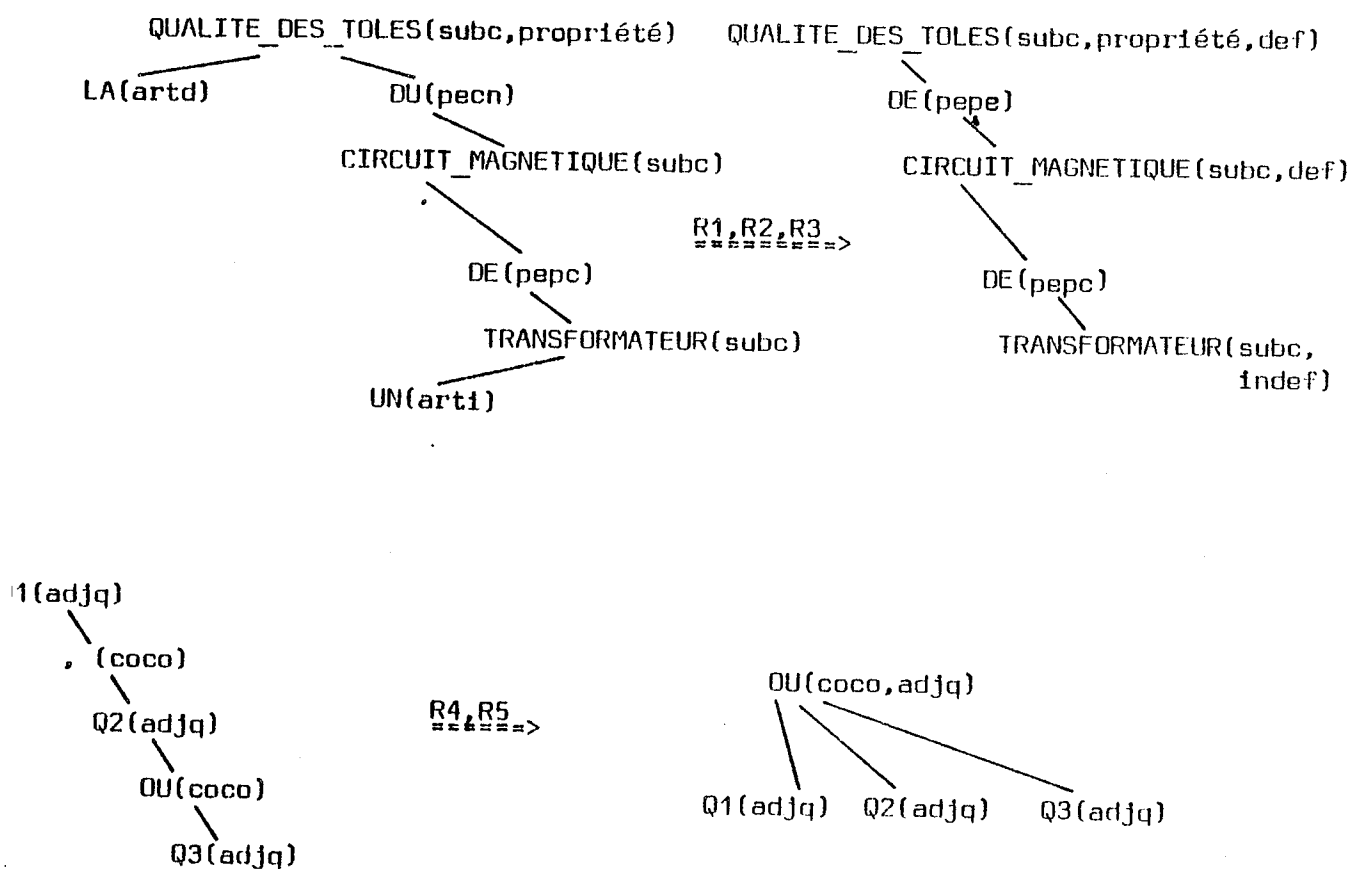


Figure 1.4. - Transformation d'une arborescence(continuée...)

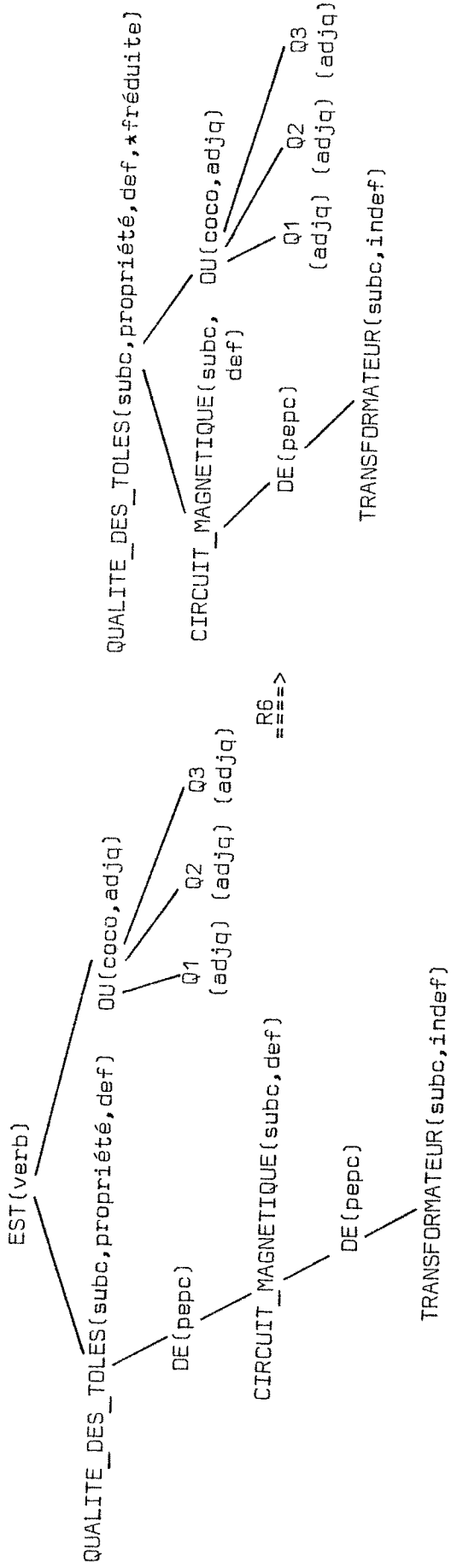


Figure 1.4. - Transformation d'une arborescence (...continuation)

Pour l'adapter à la représentation des assemblages physiques, nous avons étendu ce formalisme en distinguant plusieurs types de sommets et en définissant des ensembles caractéristiques de sommets, arcs et espaces appelés figures.

La figure 1.5. montre un exemple de RSP construit par l'interface. Les parties en trait épais et en trait pointillé constituent des figures dites de "relation entre objets" et de "valeurs possibles de propriété" respectivement. Ce RSP contient d'autres figures qui seront décrites au chapitre 2.

La construction du RSP est faite au fur et à mesure que les formes réduites sont engendrées. Le traitement de chacune de ces formes comporte trois phases :

- a) Un sous-réseau représentant la forme réduite est d'abord construit dans un espace particulier. Le programme correspondant utilise les descripteurs sémantiques attachés aux mots de la forme réduite pour choisir les figures à engendrer.
- b) Le sous-réseau est ensuite comparé au RSP courant par un procédé de "pattern matching". Il s'agit de déterminer quels éléments du sous-réseau existent déjà dans le RSP et quels éléments sont nouveaux.
- c) Les nouveaux éléments sont incorporés au RSP après un contrôle partiel et local de cohérence.

6 - CONCLUSION - RESUME

L'interface réalisé présente quatre caractéristiques principales:

- (1) Il intègre des programmes déjà réalisés dans l'équipe d'Intelligence Artificielle de l'IMAG.

- (2) Les connaissances nécessaires à TROPIC sont mémorisées dans un réseau sémantique partitionné dans le but de fournir une structure intermédiaire entre la langue naturelle et la structure de données de TROPIC.
- (3) Les programmes de l'interface ont été réalisés pour être aussi indépendants que possible des connaissances spécifiques du domaine d'application.
- (4) La structure modulaire de l'interface et l'indépendance programmes/données facilitent son évolutivité dans le cadre d'une application particulière et son adaptabilité à d'autres applications.

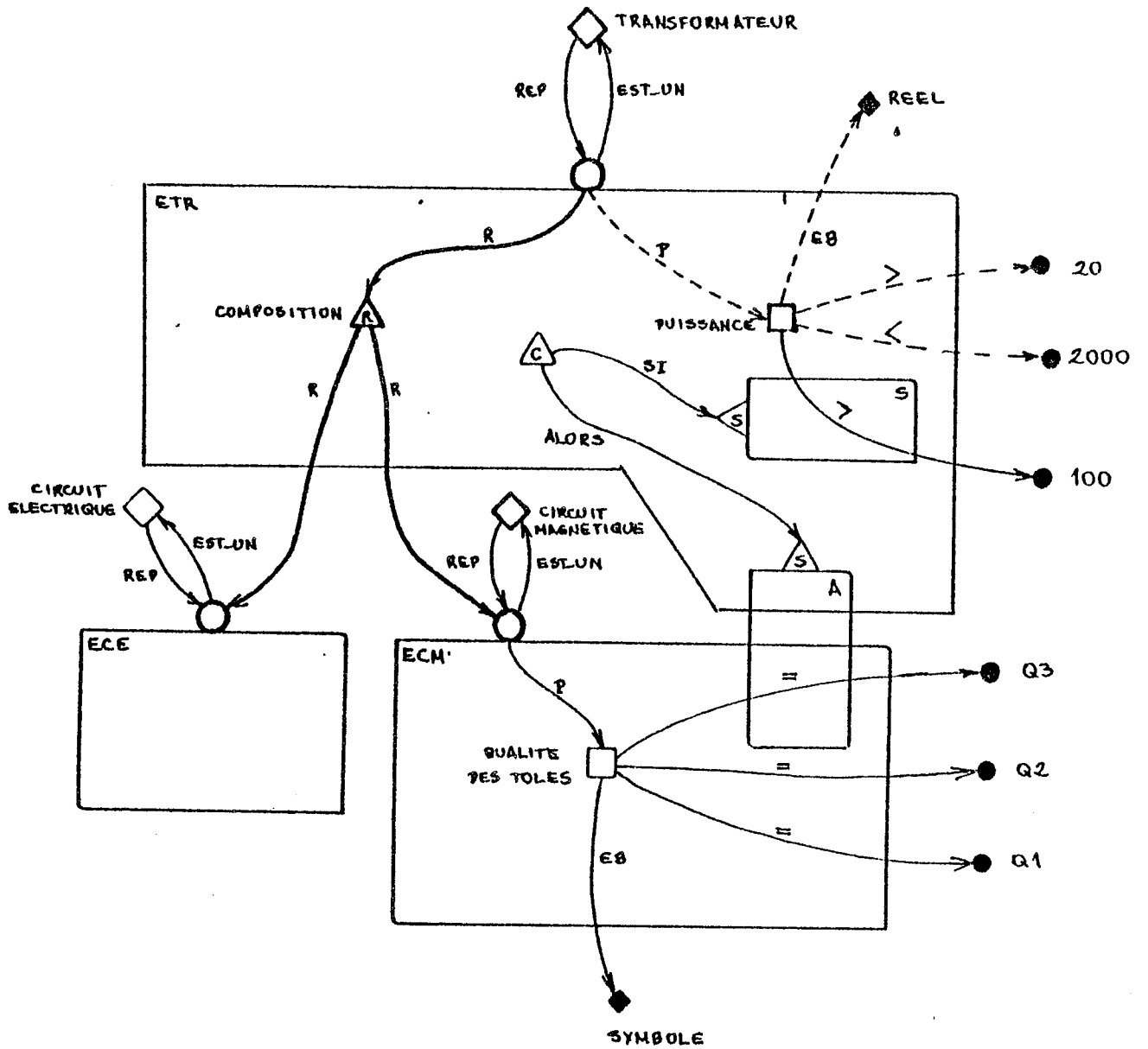


Figure 1.5.

Chapitre II

LA BASE DE CONNAISSANCES

1 - LES RESEAUX SEMANTIQUES PARTITIONNES (RSP) -----	21
1.1. Quelques antécédents -----	21
1.2. Le formalisme des RSP -----	24
2 - LA NOTION DE FIGURE -----	26
3 - DESCRIPTION D'UNE CLASSE D'OBJETS -----	30
4 - DESCRIPTION DES RELATIONS ENTRE OBJETS -----	32
4.1. La relation de composition physique -----	32
4.2. Les relations d'assemblage -----	34
4.3. Figure de "relation entre objets" -----	34
5 - DESCRIPTION DES PROPRIETES D'UNE CLASSE D'OBJETS -----	36
5.1. Figure de "valeurs possible de propriété" -----	37
5.2. Figure de "relation n-aire entre propriétés" -----	38
5.3. Figure de relation binaire entre propriétés -----	40
6 - DESCRIPTION DES COMPOSANTS D'UN ASSEMBLAGE -----	42
7 - LES GROUPEES D'OBJETS -----	44
7.1. Description des groupes d'objets -----	46
7.2. Figure de "description de groupe" -----	48
8 - LES CONTRAINTES -----	48
8.1. Représentation des contraintes dans le RSP -----	50
8.2. Figure de "situation" -----	52
8.3. Figure de "contrainte" -----	54
9 - FIGURE DE "CONSEIL" -----	54
10 - RECAPITULATIF DES ELEMENTS D'UN RSP CONNUS A PRIORI DES PROGRAMMES DE L'INTERFACE -----	56

Dans ce chapitre nous présentons la structure de la base de connaissances construite par l'interface à partir des phrases en langue naturelle.

Le formalisme utilisé est celui des Réseaux Sémantiques Partitionnés (RSP) proposé par Hendrix (1975). A l'aide de ce formalisme nous avons défini un modèle de représentation adapté au type des connaissances utilisées par TROPIC. Ce modèle s'appuie sur la notion de "figure" (sous-réseau ayant une structure particulière auquel est attachée une interprétation). Le rôle des figures est de structurer le RSP pour faciliter l'exécution de processus d'interrogation et d'interprétation.

1 - LES RESEAUX SEMANTIQUES PARTITIONNES (RSP)

1.1. Quelques antécédents

Sur un plan formel, un réseau sémantique est essentiellement un graphe étiqueté. Il est utilisé pour représenter des connaissances et leurs interdépendances. Toutefois, des modèles de représentation différents ont été élaborés sur la base de ce formalisme.

Les réseaux sémantiques ont commencé à occuper une place importante lorsque Quillian (1968, 1969) a proposé un modèle de la mémoire et du processus d'apprentissage par association. La structure utilisée par Quillian est un réseau représentant la signification des mots du langage qui correspondent à des concepts. Chaque concept est déterminé par un concept plus général et un ensemble de modificateurs (cf. figure 2.1.). Ces derniers sont exprimés en termes d'autres concepts, si bien que la signification d'un mot peut occuper toute la structure. Dans ses travaux, Quillian propose des méthodes pour identifier les relations entre concepts et pour déterminer de nouvelles relations.

Winston (1970) utilise une structure de réseau sémantique pour représenter des scènes constituées par des objets physiques comme des blocs, des pyramides, etc.

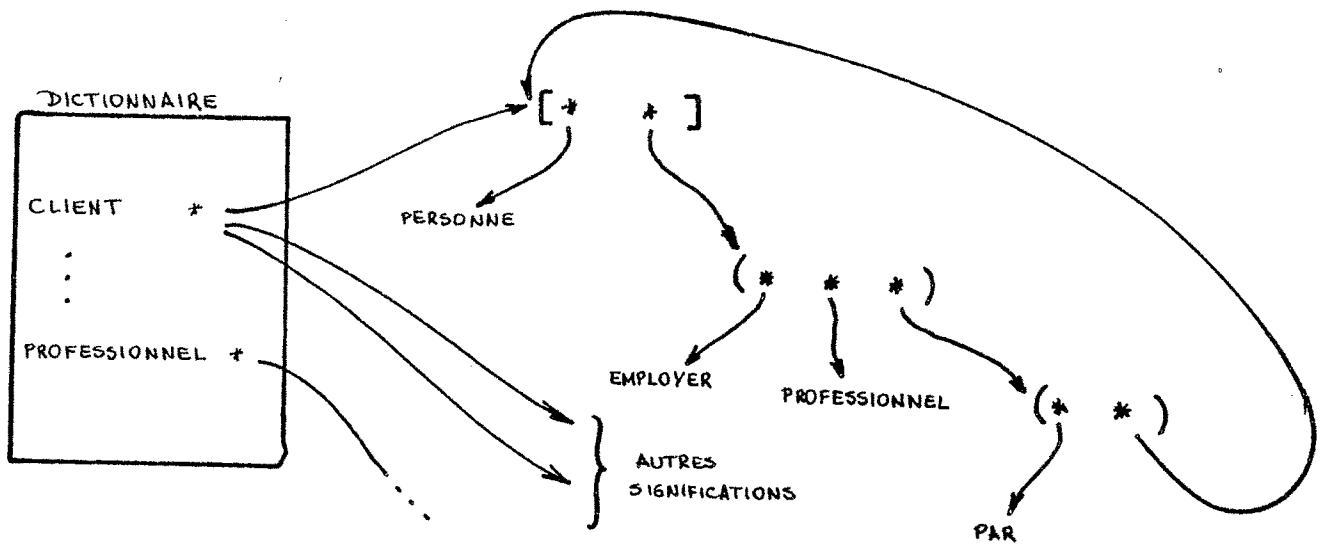
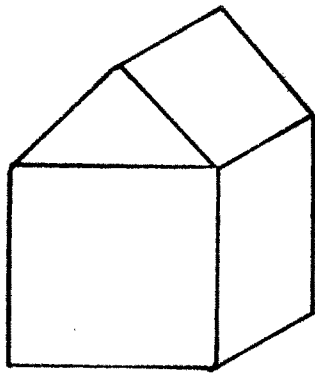


Figure 2.1.



UNE MAISON

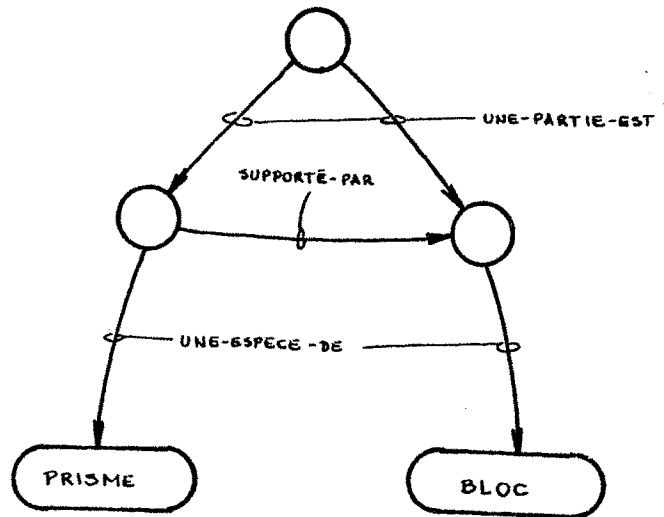


Figure 2.2.

Une notion particulièrement intéressante qu'il introduit est la notion de "groupe" définie sur des objets d'une même classe. Le but de Winston est de construire des modèles d'assemblages d'objets (cf. figure 2.2.) pour pouvoir les reconnaître lorsqu'ils apparaissent dans une scène. Chaque scène à analyser est représentée par un réseau, qui est comparé avec les modèles existants pour déterminer les similitudes et les différences.

Simmons et Bruce (1971) ont donné aux réseaux sémantiques une base plus mathématique et ont mis en évidence des relations avec d'autres systèmes formels comme les ensembles de triplets, les listes attribut-valeur, les graphes et les prédicats logiques. Ils ont aussi étudié la généralité et l'adéquation logique des réseaux sémantiques pour représenter les concepts utilisés en langue naturelle.

Les travaux de Simmons (1973) constituent une contribution significative pour donner aux réseaux sémantiques une place importante dans le traitement automatique des langues ("computational linguistics"). Ils présentent les réseaux sémantiques comme un modèle de la compréhension superficielle du langage. Du point de vue psycholinguistique, les sommets d'un réseau sont considérés comme les représentants de concepts, et les arcs comme les représentants des processus linguistiques de la pensée qui combinent les concepts pour former des descriptions en langue naturelle. Du point de vue du traitement automatique des langues, les sommets représentent des concepts lexicaux et les arcs représentent les processus utilisés pour analyser des chaînes du langage, pour construire des paraphrases, pour répondre aux questions et pour engendrer des phrases en langue naturelle. Cette théorie s'appuie sur les résultats obtenus par Quillian (1968), Rumelhart & Norman (1971), et Collins et Quillian (1971).

Woods (1975) analyse la nature et l'utilisation de différents modèles construits à l'aide de réseaux sémantiques. Il condamne le mythe de la forme canonique et montre l'importance de la définition d'une sémantique bien précise associée au

formalisme. Il soulève de nombreux problèmes concernant en particulier la représentation des relations attribut-valeur, la représentation des prédicats, et la représentation par cas, ainsi que les descriptions par "extension" et par "intension".

1.2. Le formalisme des RSP

Ce formalisme est une extension du formalisme des réseaux sémantiques, dans laquelle les sommets et les arcs sont regroupés en "espaces", eux-mêmes organisés en "vistas". Hendrix (1975) l'a introduit pour représenter des situations exigeant la délimitation de sous-ensembles d'informations, notamment la portée d'une quantification, l'implication et la description de plusieurs points de vue d'un même concept. C'est dans un esprit analogue que nous l'utiliserons.

Un RSP (la figure 2.3. en montre un exemple simple) est un graphe étiqueté dont les sommets et les arcs sont répartis en espaces. Chaque sommet ou espace est identifié par son étiquette qui lui est propre (par exemple S1, S2, S3, E1, E2, E3, U dans la figure). Chaque arc est identifié par son étiquette, son extrémité initiale et son extrémité finale (plusieurs arcs peuvent donc avoir la même étiquette : a2 par exemple). Chaque sommet et arc est contenu dans au moins un espace.

Le RSP de la figure 2.3. comporte quatre espaces E1, E2, E3 et U. E1 contient le sommet S1 et l'arc (a1,S1,S2). Dans les figures, par convention, un arc est contenu dans l'espace où se trouve son étiquette. E2 contient les sommets S2 et S3 ainsi que l'arc (a2,S2,S3). E3 contient l'arc (a2,S1,S3) et U contient l'arc (a3,S3,S2). Un arc peut donc relier deux sommets contenus dans des espaces différents et lui-même peut appartenir à un troisième espace.

A chaque espace E d'un RSP on associe une liste d'espaces de ce RSP. Cette liste, qui contient obligatoirement E, est appelée la vista de E. De l'intérieur de E, on ne "voit" que les sommets et les arcs qui sont contenus dans les espaces de la vista de E. Les vistas déterminent un ordre partiel sur

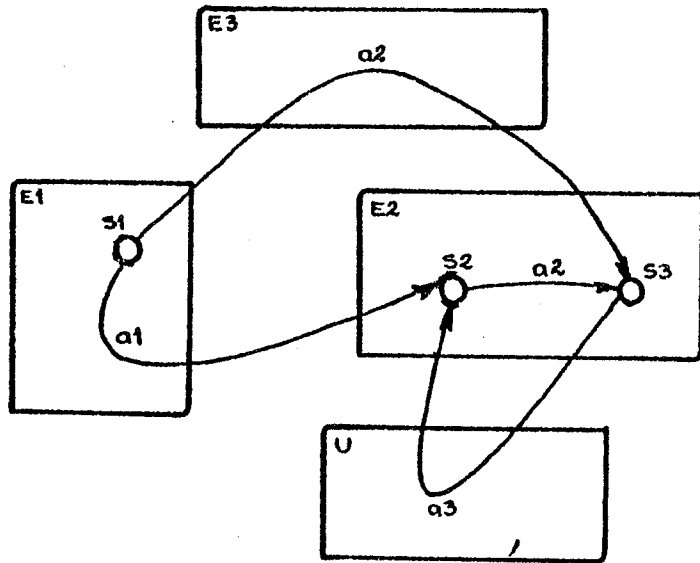


Figure 2.3.

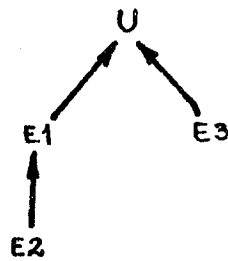


Figure 2.4.

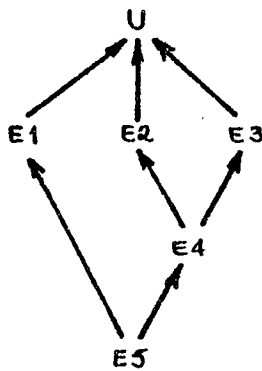


Figure 2.5.

les espaces d'un RSP. La figure 2.4. montre une organisation possible du RSP de la figure 2.3. en vistas. La vista de E2 est $(E2, E1, U)$, ce que l'on note $E2 \triangleleft (E2, E1, U)$. De même, $E1 \triangleleft (E1, U)$, $E3 \triangleleft (E3, U)$ et $U \triangleleft (U)$. La figure 2.5. montre une organisation plus compliquée d'un RSP comportant 6 espaces. On a notamment $E5 \triangleleft (E5, E4, E1, E2, E3, U)$.

Dans tout RSP, il existe un espace unique qui appartient à toutes les vistas et dont la vista ne contient que lui-même. Cet espace, que nous notons U, est appelé espace universel. Dans les figures à venir nous ne le dessinerons pas ; il sera représenté par la feuille de papier.

Il est aussi possible de donner le caractère de sommet à des espaces d'un RSP. Par exemple, la figure 2.6. montre une modification du RSP de la figure 2.3. où le sommet S4 a été associé à l'espace E3 pour former le sommet-espace S4-E3. L'arc $(a4, S1, S4-E3)$ a été ajouté dans l'espace U (non représenté), ainsi que le sommet S4.

Contrairement à Hendrix, nous permettons aux sommets et aux arcs d'être contenus simultanément dans plusieurs espaces. Le RSP de la figure 2.7. est une modification de celui de la figure 2.6. où le sommet S2 et l'arc $(a2, S2, S3)$ sont contenus simultanément dans les espaces E2 et E3 ; l'arc $(a1, S1, S2)$ est contenu dans les espaces E1 et E4.

Enfin, il est possible d'associer aux sommets, arcs et espaces d'un RSP des listes de propriétés analogues aux listes de propriétés connues en LISP.

2 - LA NOTION DE FIGURE

Les connaissances fournies à TROPIC ont des natures diverses. Des phrases comme

- "La puissance d'un transformateur est comprise entre 20 et 2000 kva",
- "Un transformateur est composé d'un circuit électrique et d'un circuit magnétique",

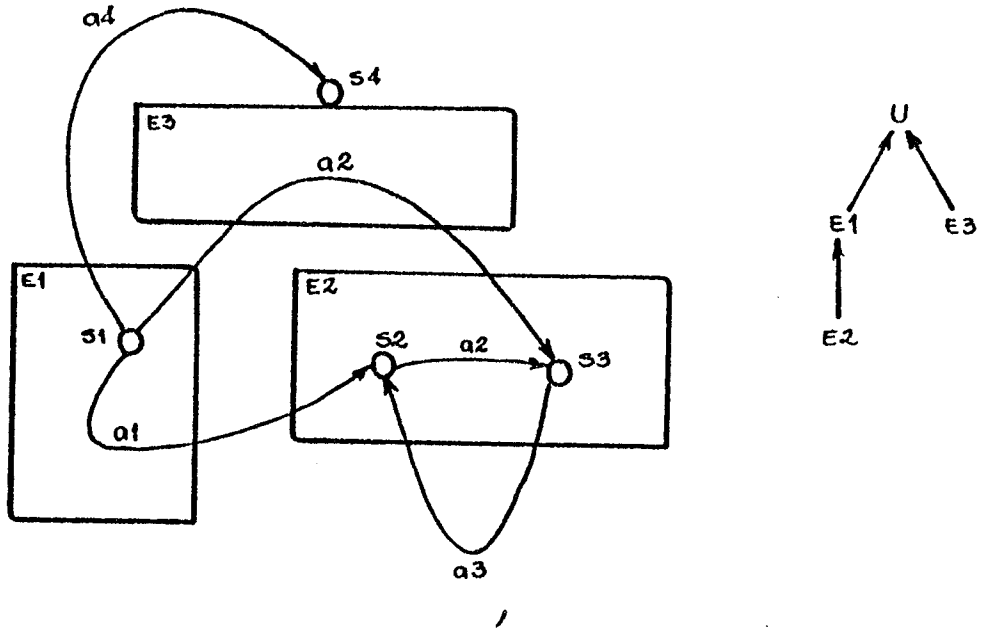


Figure 2.6.

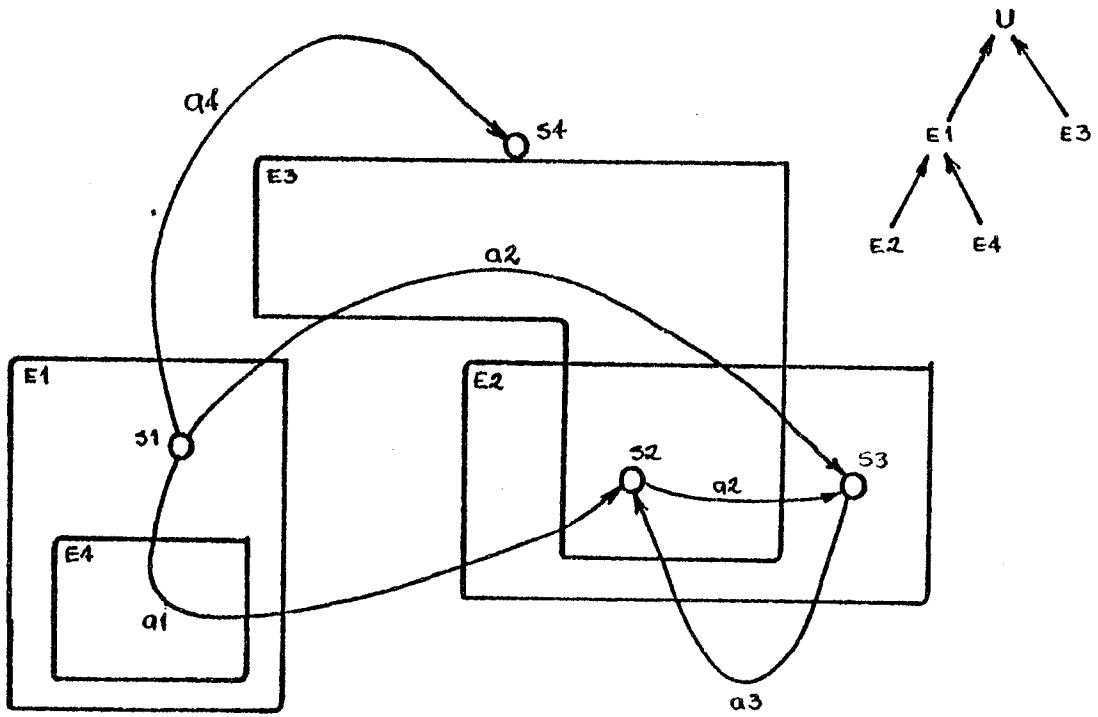


Figure 2.7.

décrivent les propriétés associées aux objets d'une certaine classe, et les relations possibles entre les objets d'une même classe ou de classes différentes¹.

Les connaissances fournies comportent aussi des lois physiques qui établissent des relations entre des propriétés d'objets, par exemple :

- "L'échauffement de l'huile (E) dans un transformateur est fonction des pertes (P) du transformateur et de la surface de radiation (S) de la cuve de ce transformateur par $E = 445 \left(\frac{S}{P}\right)^{0,8}$ ".

Par ailleurs, des contraintes imposent à TROPIC certains choix pendant la conception d'un assemblage, par exemple :

- "Si le réfrigérant d'un transformateur est l'huile, il doit être composé d'une cuve".

TROPIC accepte aussi des conseils qui guident le processus de conception :

- "Si la puissance d'un transformateur est supérieure à 100 kva, il est recommandé de choisir l'huile pour son réfrigérant".

Finalement, il est aussi possible à l'utilisateur de fournir des diagnostics pour aider TROPIC à corriger ses erreurs en cas d'échec. Nous ne traitons pas ce type de connaissance dans ce travail parce qu'il est souvent difficile de l'exprimer en langue naturelle.

Nous avons défini un modèle pour la représentation des connaissances nécessaires à TROPIC (autres que les diagnostics) en nous servant des Réseaux Sémantiques Partitionnés comme formalisme de base. La structure de ce modèle a été influencée par trois considérations principales :

(1) Il s'agit des notions de classe, de propriété et de relation au sens de TROPIC (cf. § 1 du chapitre 1).

a/ La compatibilité avec TROPIC :

Il doit exister une correspondance sémantique simple avec le modèle de représentation des connaissances utilisé par TROPIC.

b/ La nature des informations à représenter :

Il ne s'agit pas d'informations sur des objets particuliers, mais sur des classes d'objets. Ces informations incluent des contraintes à satisfaire lors du processus de conception et des conseils qui guideront ce processus.

c/ La source d'information :

La structure et les éléments du réseau doivent être aussi proches que possible de ceux des phrases en langue naturelle pour faciliter l'obtention d'informations et leur expression éventuelle en langue naturelle à partir du réseau.

Ces considérations nous ont conduits à structurer le RSP par des figures composées de sommets, arcs et espaces ayant une signification particulière. Les figures permettent le découpage des connaissances, notamment en vue de faciliter l'interprétation du RSP en termes des concepts utilisés par TROPIC.

Dans une figure, chaque sommet, arc et espace a un rôle, et en conséquence une signification déterminée, mais seulement la figure toute entière apporte une connaissance proprement dite. Par ailleurs, les figures ne sont pas disjointes elles peuvent partager des éléments jouant un rôle bien précis dans chacune d'elles. Le découpage en figures que nous avons fait est ainsi parfois arbitraire et non unique.

Dans la suite, le mot figure en caractère épais fera référence aux structures présentées ci-dessus, tandis que le mot figure en caractères minces fera référence aux illustrations du texte.

3 - DESCRIPTION D'UNE CLASSE D'OBJETS

Considérons la phrase "la puissance d'un transformateur est comprise entre 20 et 2000kva". Elle est équivalente à l'expression :
 "Vx \in TRANSFORMATEUR : la puissance de x est comprise entre 20 et 2000kva"
 où TRANSFORMATEUR est une classe d'objets. L'objet "transformateur" mentionné dans la phrase n'est donc pas un objet particulier. Nous le considérons comme le représentant générique des objets de la classe des transformateurs.

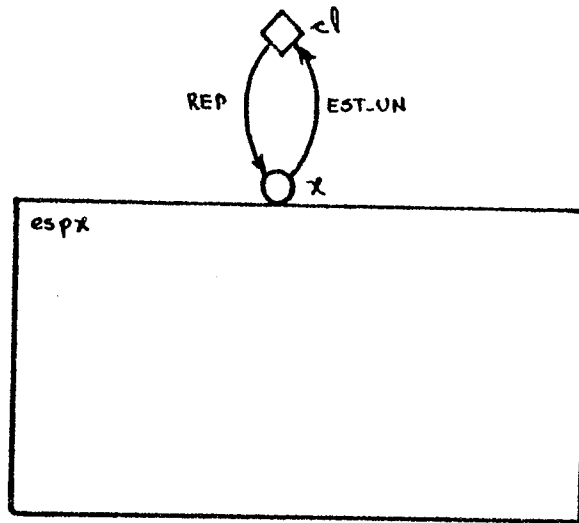
De façon analogue, dans le RSP, nous attachons toutes les connaissances sur une classe d'objets à un objet considéré comme le représentant générique de cette classe. Ces connaissances sont délimitées par une figure dite de "description de classe d'objets", montrée à la figure 2.8. :

- Le sommet \diamond représente une classe d'objets. Son étiquette cl est le nom de cette classe, qui est unique dans le RSP.(cf. Note 1 ci-dessous).
- Le sommet \circ désigne le représentant générique des objets de la classe cl. Les connaissances relatives à cette classe sont contenues dans l'espace esp_x qui est associé au sommet x pour former un sommet-espace.
- L'arc EST_UN indique que x appartient à la classe cl (cf.Note 2). Cet arc est redondant avec l'arc REP, mais son utilité sera évidente plus tard (§ 7).

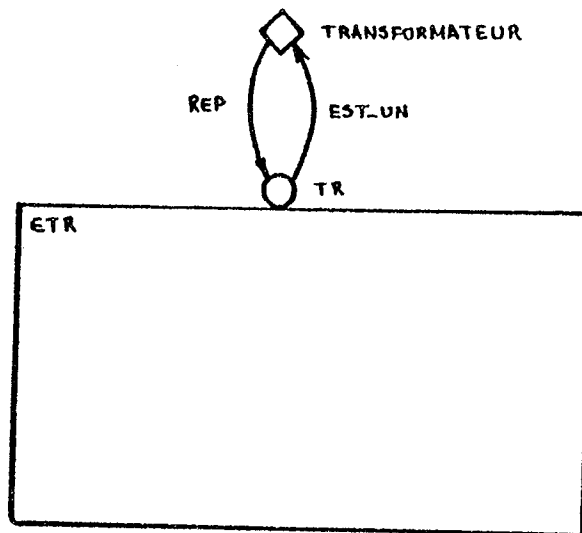
Tous les sommets et les arcs de cette figure se trouvent dans l'espace universel U.

Nous appelons esp_x le contexte de x, ou indifféremment de cl. Dans la figure 2.8b, ETR constitue le contexte de TR et de TRANSFORMATEUR.

Note 1 : Les étiquettes des sommets, arcs et espaces sont notées en lettres majuscules lorsqu'elles sont connues (REP, EST_UN dans la figure 2.8a) et en minuscules lorsqu'elles sont inconnues (cl, x dans la figure 2.8a).



(a)



(b)

Figure 2.8.

Note 2 : Lorsqu'il n'y a pas d'ambiguïté, nous faisons référence aux arcs par leurs seules étiquettes.

4 - DESCRIPTION DES RELATIONS ENTRE OBJETS

Les objets constituant un assemblage sont liés entre eux par des relations parmi lesquelles, en suivant TROPIC, nous distinguons les relations de composition physique et les relations d'assemblage.

4.1. La relation de composition physique

La relation de composition physique établit une hiérarchie entre les objets d'un assemblage. La racine de la hiérarchie est l'objet qui représente l'assemblage même. Ses successeurs immédiats sont ses composants premiers. A son tour, chacun de ces composants peut avoir des composants qui s'ajoutent à la hiérarchie. Par exemple :

- Une tour est composée d'une colonne et d'un toit.
- Une colonne est composée de blocs.
- Un toit est composé de ...

Ces phrases sont représentées par le RSP de la figure 2.9. Les sommets \triangle d'étiquette COMPOSITION représentent les relations de composition physique entre les objets. Le sommet \odot représente un groupe de blocs (cf. § 7). Chaque relation COMPOSITION est établie entre le sommet qui représente l'objet composé et les sommets qui représentent les composants. Il y a donc un ordre, indiqué par l'orientation des arcs, qui détermine quel est l'objet composé et quels sont les composants. De plus, le sommet \triangle et les arcs adjacents à ce sommet sont situés dans le contexte de l'objet composé.

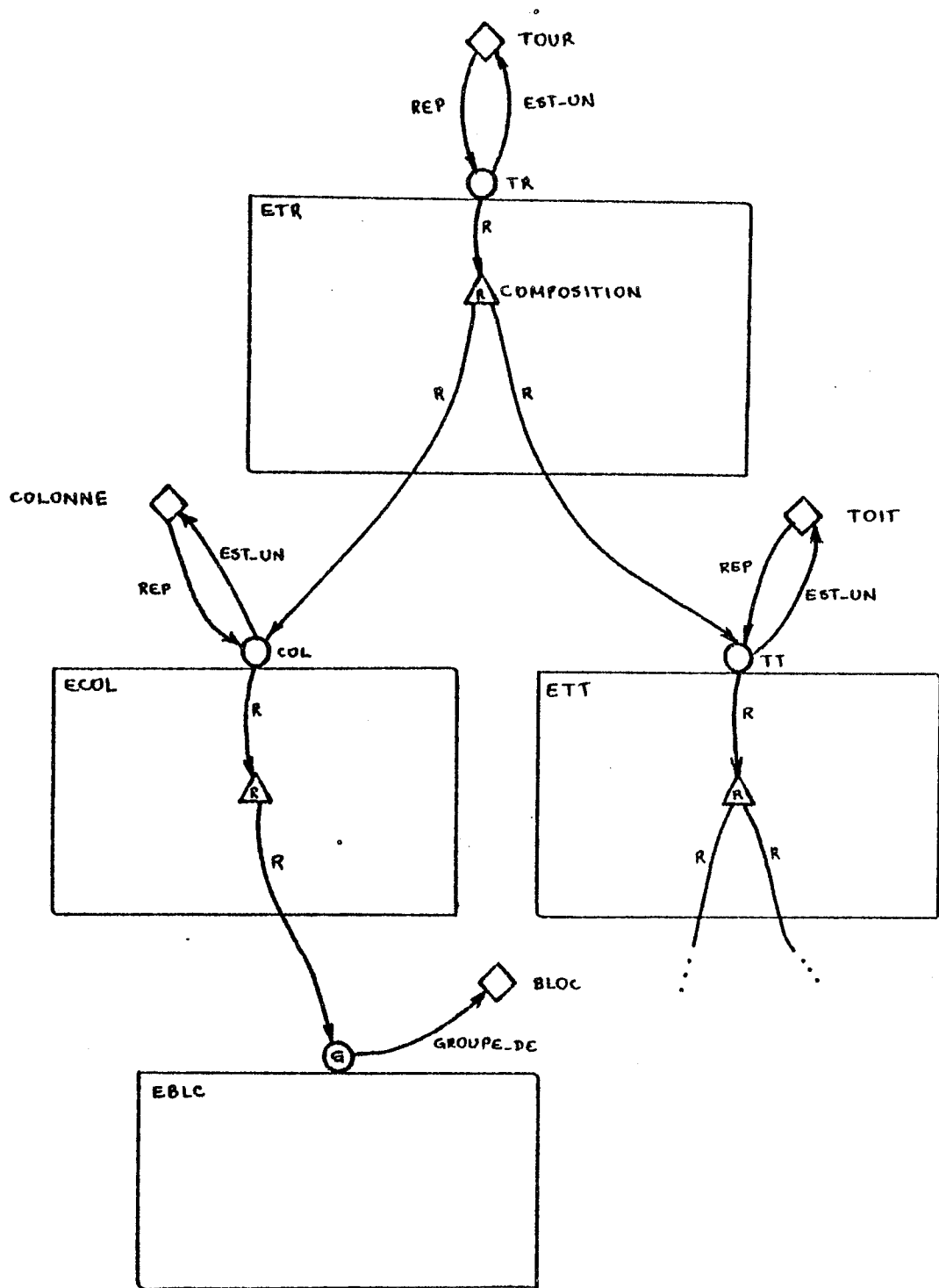


Figure 2.9.

4.2. Les relations d'assemblage

Considérons la phrase "le toit d'une tour est situé sur la colonne de la tour" qui décrit une relation d'assemblage (SUR) entre deux composants d'une tour. Si nous établissons une relation SUR entre COL et TT dans ECOL (cf. figure 2.9.), elle indiquera que tout toit est sur une colonne, alors que la phrase ne donne cette relation que lorsque la colonne et le toit sont des composants d'une même tour.

La solution que nous avons choisie pour représenter une telle phrase consiste à placer la relation SUR dans le contexte de TOUR, pour indiquer qu'elle existe seulement si les participants de la relation sont des composants d'une tour (cf. figure 2.10).

Cette solution illustre pour la première fois l'utilité du découpage du réseau en espaces. Ce découpage nous permet ici de représenter des informations concernant une même classe d'objet dans des contextes différents sans être obligé à reproduire l'objet générique dans chaque contexte. Ainsi, les informations associées à TT dans ETT décrivant un toit en général (ie. les caractéristiques intrinsèques d'un toit), tandis que les informations associées à TT dans ETR sont particulières au toit d'une tour.

4.3. Figure de "relation entre objets"

Cette figure, montrée à la figure 2.11., est suffisamment générale pour représenter la relation de composition et les relations d'assemblage :

- Les sommets x , y_1 , ..., y_n représentent des objets.
- Le sommet \triangle_R représente une relation entre objets ; son étiquette (rel) est le nom de la relation.
- Les arcs R indiquent les participants à la relation.

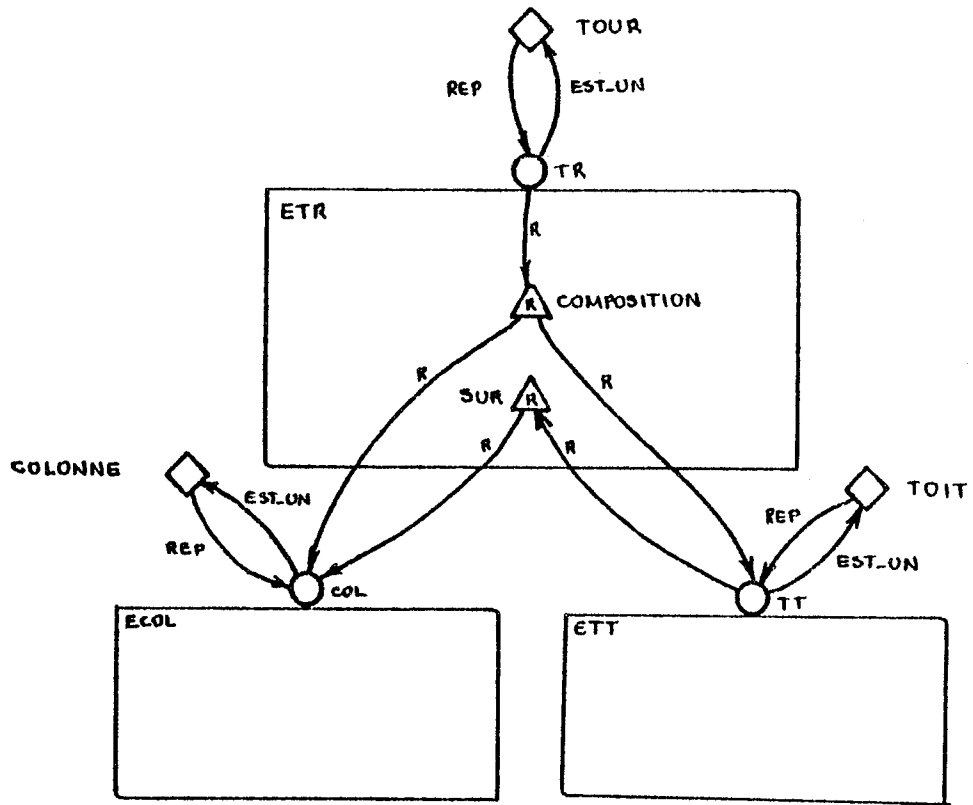


Figure 2.10.

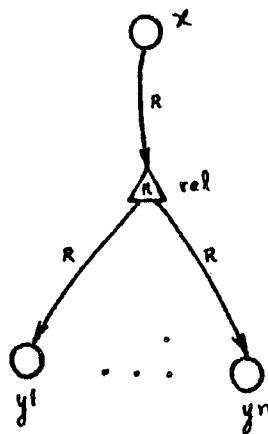


Figure 2.11.

Cette figure signifie que la relation rel existe entre x et chacun des objets y_1, \dots, y_n .

Le sommet \triangle_R est placé dans le contexte d'un ancêtre commun des objets participant à la relation. Cet ancêtre est en général spécifié dans la phrase en langue naturelle : "le toit d'une tour est placé sur la colonne de la tour". Dans les cas où il n'est pas spécifié, on détermine l'ancêtre le plus proche en suivant les relations de composition physique à partir des objets participant à la relation.

Dans le cas de la relation de composition, le sommet \triangle_R est placé dans le contexte de x , ou dans le contexte d'un ancêtre de celui-ci lorsque cet ancêtre est spécifié dans la phrase : "une colonne d'une tour est composée de deux blocs".

Remarque :

Il existe deux cas de relations entre objets qui présentent un certain intérêt théorique mais qui apparaissent assez rarement dans les cas pratiques. Le premier est celui où deux ou plusieurs relations entre objets sont mutuellement exclusives. Le deuxième cas est celui où un objet peut participer à la composition de deux assemblages différents avec des descriptions différentes selon qu'il fasse partie de l'un ou de l'autre. Ces deux cas sont discutés dans le chapitre 6.

5 - DESCRIPTION DES PROPRIÉTÉS D'UNE CLASSE D'OBJETS

Nous représentons chacune des propriétés associées à une classe d'objets par un sommet. Ce sommet est étiqueté par le nom de la propriété et est relié par un arc au représentant générique de la classe. Les valeurs possibles de la propriété sont définies par un ou plusieurs sommets reliés par des arcs au sommet désignant la propriété. L'ensemble correspondant de sommets et d'arcs constitue une figure dite "de valeurs possibles d'une propriété" qui est décrite au § 5.1.

Il est aussi possible de représenter des relations entre les valeurs de plusieurs propriétés. Deux figures dites "de relation n-aire entre propriétés" et "de relation binaire entre propriétés" ont été prévues à cet effet. Ces figures seront décrites aux § 5.2. et 5.3. respectivement.

5.1. Figure de "valeurs possibles de propriété"

Cette figure permet de représenter l'ensemble des valeurs possibles d'une propriété. Elle comporte une partie obligatoire et une partie facultative :

a) La partie obligatoire désigne par un arc d'étiquette EB l'ensemble de base dans lequel la propriété peut prendre sa valeur, par exemple : l'ensemble des réels, l'ensemble des entiers, l'ensemble des symboles. Ces ensembles sont connus par les programmes qui leur associent des règles de manipulation.

b) La partie facultative exprime des restrictions complémentaires sur les valeurs possibles de la propriété.

Par exemple, le RSP de la figure 2.12. exprime que la valeur de la propriété PUISSANCE d'un transformateur doit être un réel compris entre 20 et 2000. Le sommet \blacklozenge note l'ensemble de base des valeurs de la propriété. Les arcs $(\geq, \text{PUISSANCE}, 20)$ et $(\leq, \text{PUISSANCE}, 2000)$ codent des restrictions sur la valeur de la propriété. Les sommets \bullet notent des constantes. Ils sont placés dans l'espace U de façon à être facilement accessibles de n'importe quel point du réseau

Chaque restriction sur la valeur d'une propriété utilise une des relations =, \neq , <, >, \leq , \geq qui correspondent aux prédicats EGAL, DIFFERENT, INFÉRIEUR, SUPÉRIEUR, INFÉRIEUR OU ÉGAL, SUPÉRIEUR OU ÉGAL.

La relation = constitue un cas particulier. En effet, si on impose à une propriété d'être égale à une certaine valeur, toute autre restriction devient inutile¹. C'est pourquoi, si plusieurs relations = sont appliquées à la même propriété, nous considérons qu'elles définissent les valeurs possibles (en nombre fini) de cette propriété. Ainsi, la phrase "le réfrigérant d'un transformateur est l'air, l'huile ou le pyralène" est représentée par le réseau de la figure 2.13.

(1) Pour des raisons de simplicité nous ne traitons pas le cas général où les valeurs des propriétés sont définies par des formules combinant des conjonctions et des disjonctions de restrictions (cf. § 4.3 du chapitre 6).

Nous avons donc deux cas de figures de "valeurs possibles de propriété" :

- Le premier cas correspond à la figure 2.14a, où :
 - . Le sommet \square représente une propriété. Le nom de la propriété est donné par l'étiquette du sommet (prop).¹
 - . L'arc P indique que prop est une propriété de x.
 - . Le sommet \blacklozenge représente un ensemble connu par le système ; son étiquette eb (REEL, ENTIER, SYMBOLE, ...) détermine l'ensemble de base des valeurs de la propriété.
 - . L'arc EB indique que la propriété prop prend ses valeurs dans l'ensemble eb.
 - . Les sommets \bullet représentent des constantes. Leurs étiquettes sont les constantes mêmes.
 - . Les arcs étiquetés r_1, \dots, r_n ($\neq, <, >, \leq, \geq$) indiquent le type de restrictions appliquées (on note l'absence de =).

Cette figure indique que la propriété prop de l'objet x doit prendre une valeur dans l'ensemble eb en respectant toutes les relations r_1, \dots, r_n avec les constantes v_1, \dots, v_n .

- Le deuxième cas est celui de la figure 2.14b. Il permet d'énumérer les valeurs possibles pour la propriété prop lorsque celles-ci sont en nombre fini : les arcs étiquetés = indiquent que prop doit prendre une valeur parmi v_1, \dots, v_n . Les autres sommets et arcs ont la même interprétation que dans le cas précédent.

5.2. Figure de "relation n-aire entre propriétés"

Considérons l'énoncé "l'échauffement de l'huile dans un transformateur est égal à $445 \left(\frac{P}{S}\right)^{0,8}$, où P désigne les pertes dans le transformateur et S la surface de sa cuve".

Nous le représentons par le réseau de la figure 2.15 qui exprime que la propriété ECHAUFFEMENT DE L'HUILE de TR (le représentant générique des transformateurs) est égale au résultat de l'évaluation de l'expression lambda associée au sommet \triangle (cette expression est la valeur de la propriété EXPR associée au sommet \triangle au sens du langage LISP).² Les arcs étiquetés 1, 2, ... indiquent dans l'ordre les paramètres formels à utiliser pour évaluer l'expression, à savoir la propriété PERTES de TR et la propriété SURFACE de CU (le représentant générique des cuves). La partie en trait épais est un exemple de figure de "relation n-aire entre propriétés".

(1) Pour les identifier de façon unique, les sommets sont en réalité étiquetés par le nom de la propriété suivi du nom de la classe du représentant générique auquel il est associé (prop-cl).

(2) cf. fin du § 1.2.

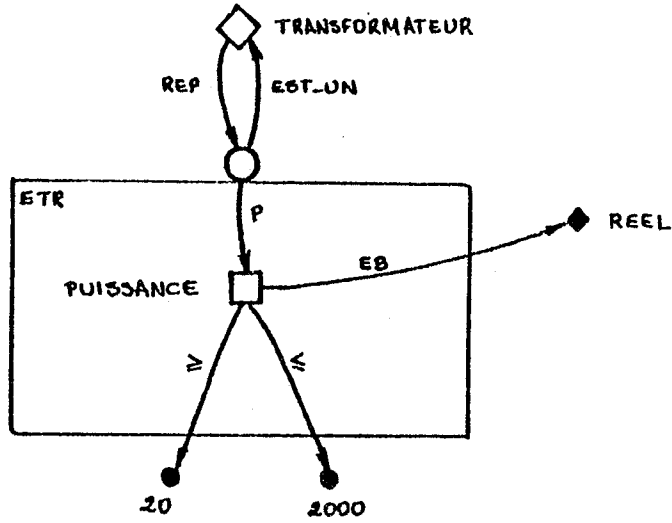


Figure 2.12.

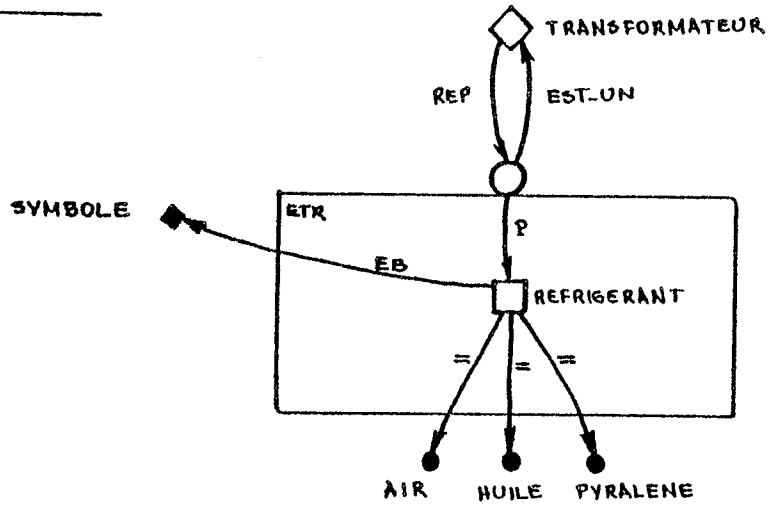
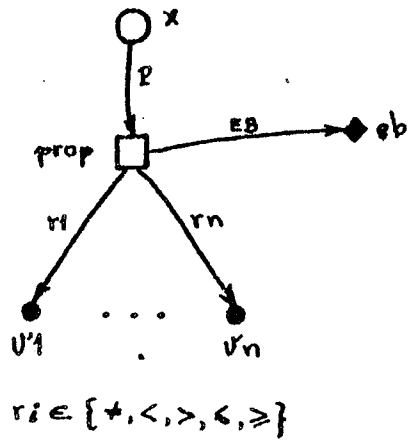
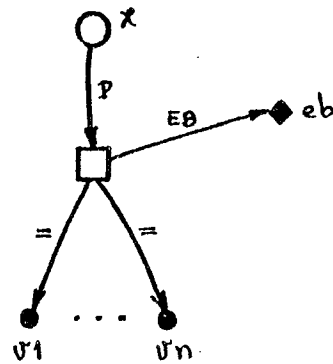


Figure 2.13.



(a)



(b)

Figure 2.14.

La figure 2.16 montre la forme générale de cette figure :

- les sommets x, x_1, \dots, x_n sont des représentants génériques;
- les sommets $prop, prop_1, \dots, prop_n$ sont des propriétés associées à x, x_1, \dots, x_n ;
- Le sommet noté \triangle représente la relation n-aire, qui est décrite par l'expression lambda indiquée par la propriété EXPR.
- Les arcs $1, 2, \dots, n$ indiquent chacun des paramètres de \triangle .
- L'arc r est une des relations $=, <, >, \leq, \geq$.

Cette figure indique que la valeur de $prop$ doit vérifier la relation r avec le résultat de l'évaluation de la forme F avec les paramètres indiqués.

Le sommet \triangle et les arcs de la figure sont placés dans le contexte d'un ancêtre commun dans la hiérarchie de composition des objets concernés dans la figure (celui spécifié dans la phrase en langue naturelle). Par exemple, dans la figure 2.15, ils sont placés dans le contexte de TRANSFORMATEUR (la relation de COMPOSITION entre TR et CU est supposée exister bien que non représentée sur la figure).

5.3. Figure de "relation binaire entre propriétés"

La figure précédente est utile pour représenter des connaissances telles que des lois physiques. Elle est toutefois assez lourde pour représenter une relation binaire entre deux propriétés. Pour une telle relation, nous utilisons une figure plus simple.

Par exemple, la figure 2.17a indique que la LONGUEUR de x (le représentant générique d'une classe d'objet) doit être plus grande que sa LARGEUR. La figure 2.17b indique que la HAUTEUR de x doit être égale à la HAUTEUR de y .

La figure qui correspond à ce cas est montrée à la figure 2.18, où :

- les sommets x_1 et x_2 sont des représentants génériques;
- les sommets $prop_1$ et $prop_2$ sont des propriétés associées à x_1 et x_2 .
- L'arc r est une des relations $=, <, >, \leq, \geq$.

Cette figure signifie que la valeur de $prop_1$ doit vérifier la relation r avec la valeur de $prop_2$.

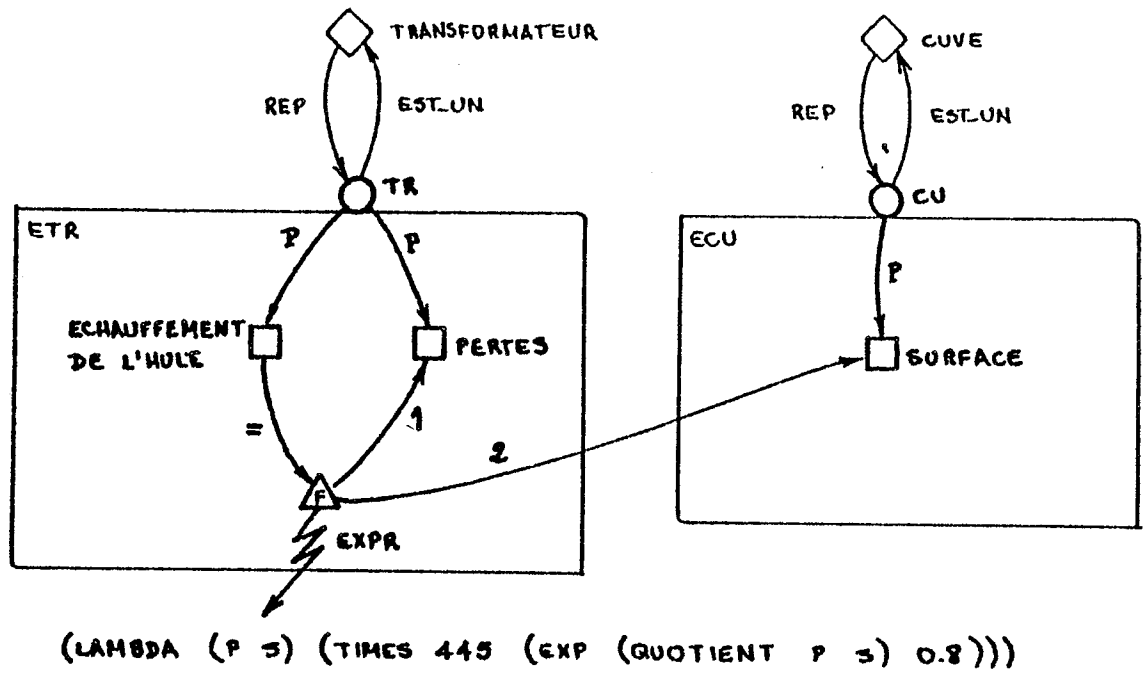


Figure 2.15.

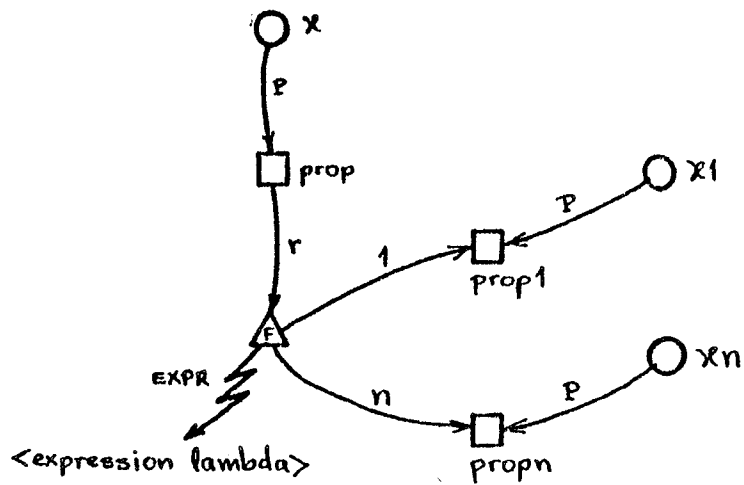


Figure 2.16.

La relation r est placée dans le contexte d'un ancêtre commun dans la hiérarchie de composition des objets concernés dans la figure. Dans le cas de la figure 2.17b, elle est placée dans le contexte de z , ce qui veut dire que x et y ont z pour ancêtre commun. Inversement, ceci indique que la relation n'est valable que si x et y sont des descendants de z .

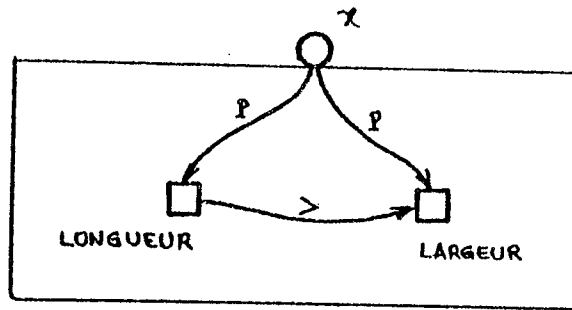
La valeur d'une propriété doit donc respecter les conditions décrites par la figure de valeurs possibles d'une propriété, par les figures de relation n -aire entre propriétés et par les figures de relation binaire entre propriétés.

6 - DESCRIPTION DES COMPOSANTS D'UN ASSEMBLAGE

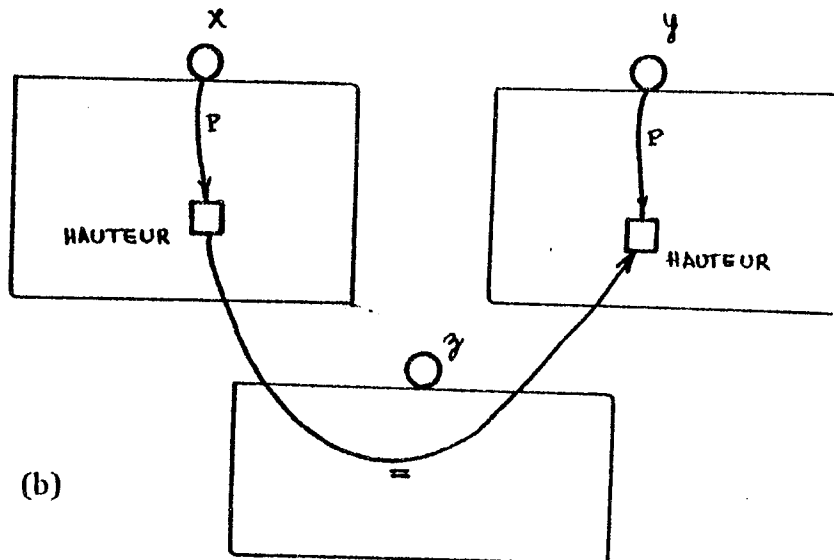
Il arrive que les objets d'une classe présentent des particularités par rapport au cas général lorsqu'ils sont des composants d'un assemblage donné. Par exemple, les phrases

- "La qualité des tôles d'un circuit magnétique est Q1, Q2 ou Q3",
 - "La qualité des tôles du circuit magnétique d'un transformateur est Q1 ou Q2",
- déterminent deux ensembles différents de valeurs possibles pour la propriété QUALITE DES TOLES d'un circuit magnétique selon que ce dernier est ou n'est pas un composant d'un transformateur.

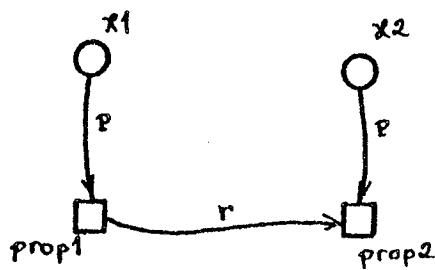
Si nous incluons les informations apportées par les deux phrases ci-dessus au réseau de la figure 2.19, le résultat est le réseau de la figure 2.20. Notons q_1 , q_2 et q_3 les arcs $(=, \text{QUALITE_DES_TOLES}, Q1)$, $(=, \text{QUALITE_DES_TOLES}, Q2)$, et $(=, \text{QUALITE_DES_TOLES}, Q3)$. Ils se trouvent dans E_{CM} pour indiquer que les informations qu'ils représentent sont applicables en général à tout circuit magnétique. Les arcs q_1 et q_2 se trouvent aussi dans E_{TR} pour indiquer que lorsqu'un circuit magnétique est un composant d'un transformateur, la qualité de



(a)



(b)

Figure 2.17.Figure 2.18.

ses tôles ne peut être que Q1 ou Q2.

Ce principe de représentation est analogue à celui qui a été retenu pour les relations n-aires et binaires entre propriétés (cf. figures 2.15 et 2.17b), et les relations d'assemblage (cf. fin du § 4.3.).

Ainsi, cette utilisation des contextes nous permet de représenter les cas particuliers qui apparaissent lorsqu'un objet peut faire partie d'un ou plusieurs assemblages, sans avoir besoin de reproduire des parties de la structure.

7 - LES GROUPES D'OBJETS

Il arrive assez fréquemment qu'un assemblage soit composé d'un ensemble répétitif d'objets disposés d'une façon régulière, par exemple, une "pile" de blocs, une "rangée" de colonnes, des objets cylindriques disposés concentriquement... Il est important de pouvoir raisonner directement sur un tel ensemble, mais aussi sur ses composants et les relations entre ceux-ci.

Considérons le cas d'une pile d'objets. Nous parlons d'une pile lorsque les objets sont situés les uns sur les autres, mais pas n'importe comment : sur chaque objet il n'y a qu'un seul objet et chaque objet est sur un seul autre. Pourtant, ceci n'est pas vrai pour tous les objets de la pile car le plus bas (ou premier) des objets n'est pas situé sur un autre objet de la pile, et le plus haut (ou dernier) des objets n'a aucun objet sur lui. Par ailleurs, nous avons un certain nombre de connaissances qui nous permettent de manipuler l'ensemble: nous savons que si l'objet A est sur l'objet B et à son tour B est sur C, alors A est au-dessus de C ; nous savons que pour enlever un objet du milieu de la pile, il faut enlever aussi ceux qui sont au-dessus de lui, etc... En somme, le mot "pile" comporte une signification assez complexe dont la connaissance nous permet de simplifier la description et la manipulation des structures concernées.

Nous appelons ce type d'ensemble d'objets groupe d'objets.

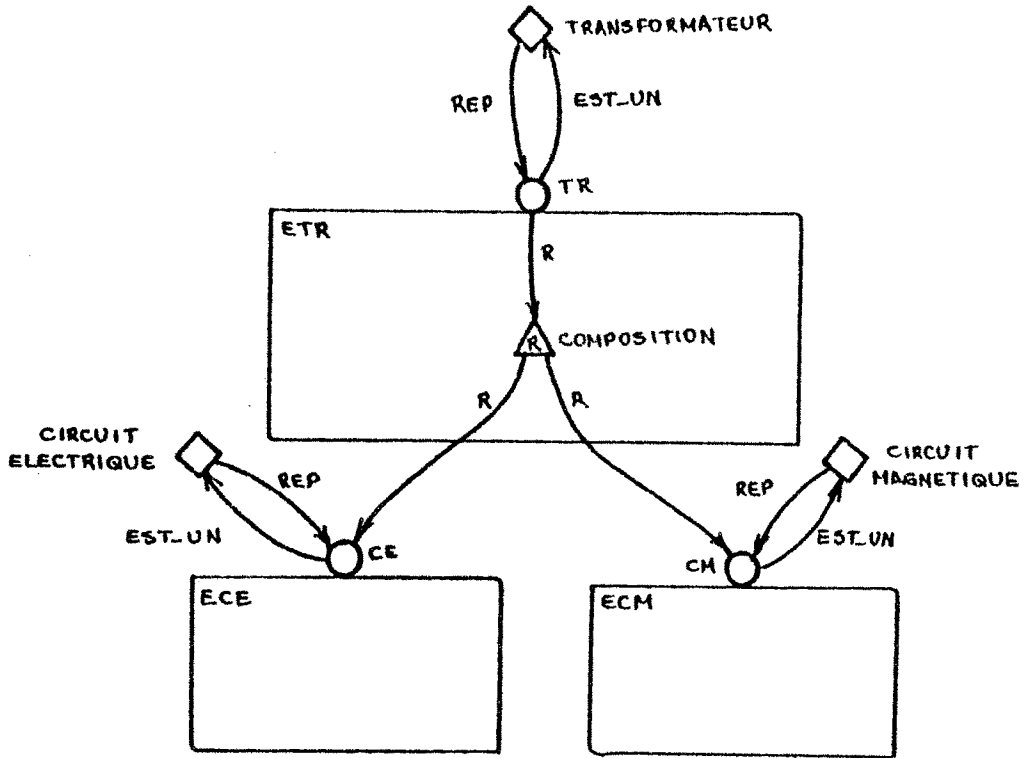


Figure 2.19.

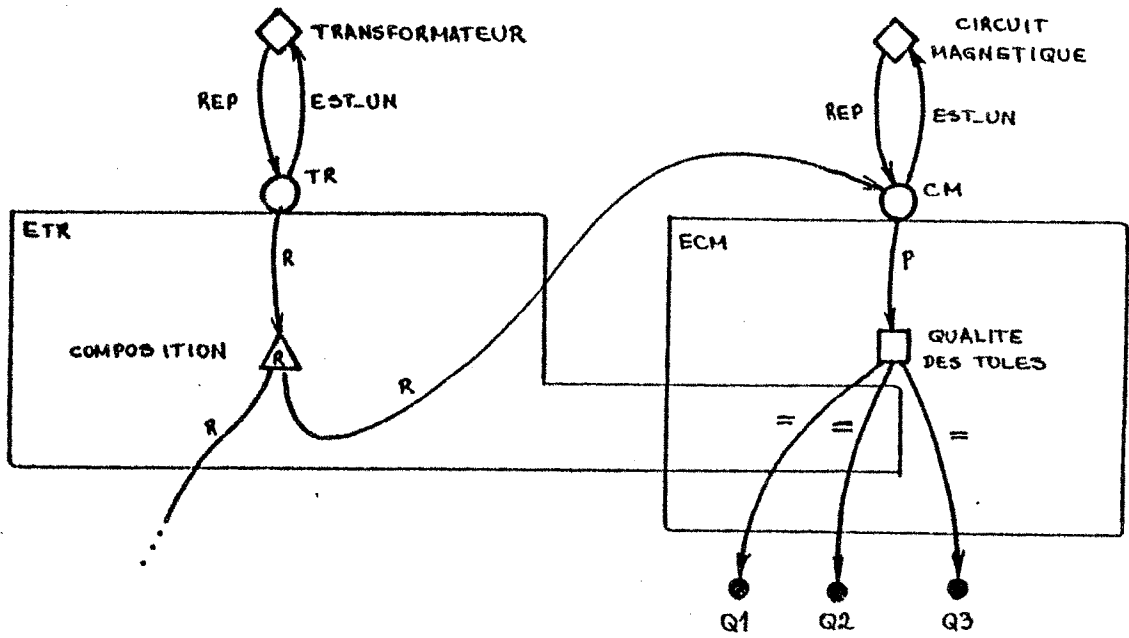


Figure 2.20.

Remarque

Nous avons introduit la notion de groupe pour des raisons d'efficacité et de concision. Notre travail sur ce sujet reste toutefois assez superficiel. Nous pensons que la définition et la représentation de la notion de groupe mériteraient d'être approfondies.

7.1. Description des groupes d'objets

La figure 2.21 représente une pile de trois blocs. Le sommet x (noté \textcircled{G}) représente en même temps le groupe en tant qu'assemblage et chacun des blocs qui forment le groupe. Nous avons accordé ce double rôle aux sommets \textcircled{G} pour tenir compte de la structure générale de l'ensemble d'objets (objets de n'importe quelle classe) d'une part, et des particularités de chacun des éléments de l'ensemble, d'autre part :

- En tant que groupe, x est décrit par ses propriétés DISP(DISPosition) et CARD (CARDinalité).
- En tant qu'objet, x est décrit par des propriétés et des relations avec d'autres objets de la même façon que les autres objets (sommets \textcircled{O}) dans le réseau.

La présence des propriétés DISP et CARD est facultative dans la description d'un groupe et leur absence indique l'indétermination du groupe. En effet, nous pouvons parler d'un groupe de trois blocs, d'un groupe de blocs disposés en pile ou tout simplement d'un groupe de blocs.

Il arrive qu'un des composants d'un groupe présente des particularités qui le distinguent des autres. Par exemple, nous pouvons avoir une pile de blocs où la section du bloc le plus bas est carrée et celle des autres blocs est ronde. Alors, nous avons besoin d'un moyen de faire référence au plus bas des blocs de la pile pour indiquer ses particularités. Nous le faisons par l'intermédiaire de la relation MEMSPEC (MEMbre SPECial) comme il est montré dans la figure 2.22. La propriété IDENT détermine le membre spécial. Le membre spécial y est placé dans l'espace Ex pour signifier que son existence n'a de sens que parce qu'il est un des membres du groupe x . De plus, $Ey \triangleleft (Ey, Ex, U)$.

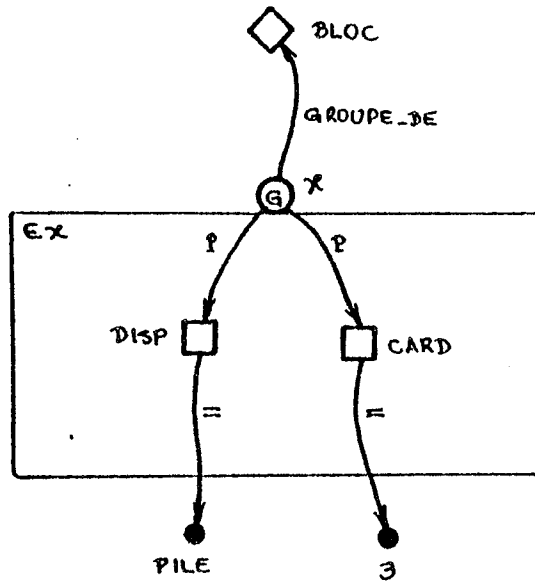


Figure 2.21.

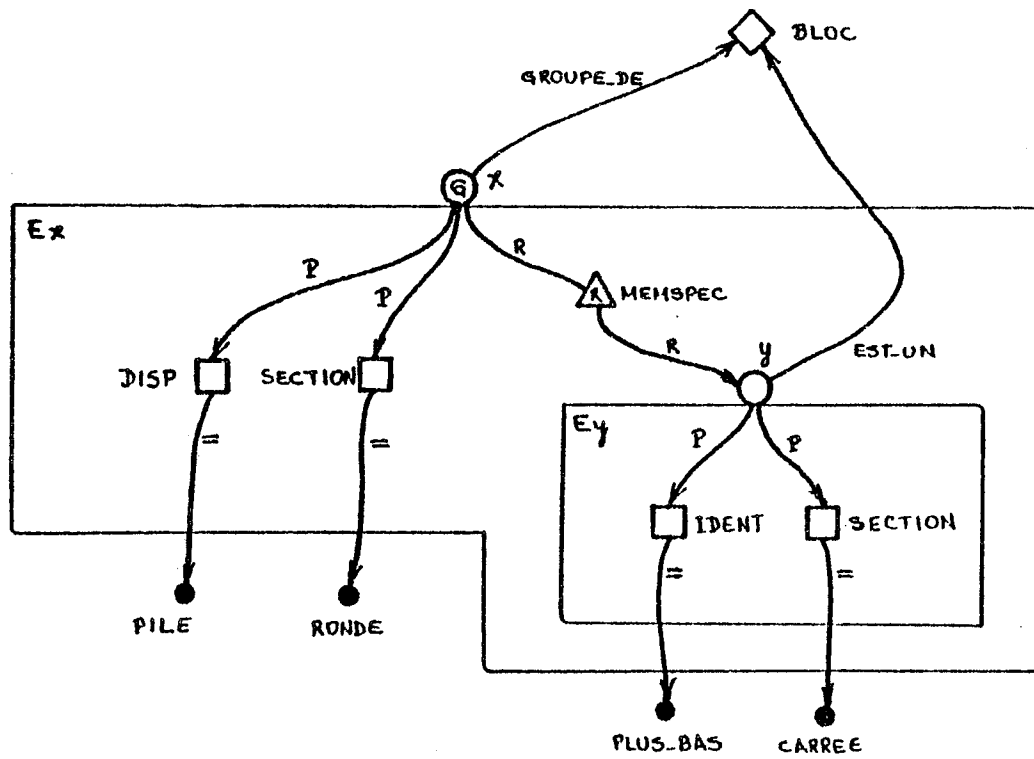


Figure 2.22.

7.2. Figure de "description de groupe"

Cette figure encadre la représentation d'un groupe. La figure 2.23 montre la structure correspondante :

- Le sommet noté \odot représente en même temps le groupe en tant que structure et chacun des objets qui forment le groupe.
- L'arc GROUPE_DE indique que chaque élément du groupe est un objet de la classe cl.
- En tant que groupe, x peut être décrit par la propriété DISP pour indiquer la disposition des objets du groupe, par la propriété CARD pour indiquer le nombre d'éléments dans le groupe, par la relation.MEMSPEC pour indiquer les membres spéciaux du groupe. L'absence totale ou partielle de cette description n'enlève pas à x son identité de groupe.
- En tant qu'objet, x peut être décrit par des propriétés et des relations avec d'autres objets, comme n'importe quel assemblage.

Remarque :

Comme les représentants génériques des classes d'objets, les groupes et les membres spéciaux de groupes sont donc représentés par des sommets \odot , auxquels sont associés des espaces. Cette même représentation permet aux sommets représentant les groupes et les membres spéciaux de groupes de participer, comme les sommets symbolisant les représentants génériques, aux figures de "relation entre objets", de "valeurs possibles de propriétés" et de "relation entre propriétés". De la même façon, le contenu de l'espace associé au sommet représentant un groupe ou un membre spécial est appelé le contexte du groupe ou du membre spécial.

8 - LES CONTRAINTES

Les contraintes sont des expressions conditionnelles de la forme
si <S1> alors <S2>

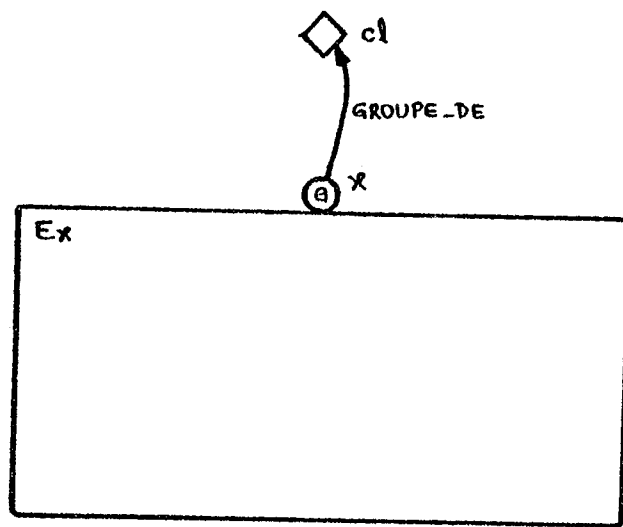


Figure 2.23.

où S1 et S2 sont des situations, par exemple :

"si les joints d'un circuit magnétique sont imbriqués à 45 degrés, alors sa qualité de tôles est Q3".

Une contrainte décrit des objets dans un cas particulier "si" définit ce cas et la partie "alors" spécifie les particularités des objets dans ce cas.

8.1. Représentation des contraintes dans le RSP

Nous représentons la contrainte ci-dessus par le réseau de la figure 2.24. Le sommet $\triangle C$ représente la contrainte. Sa partie SI est donnée par la situation contenue dans l'espace S1 et sa partie ALORS est donnée par la situation contenue dans l'espace S2. Nous pouvons constater qu'il est suffisant d'inclure dans S1 l'arc (=,JOINTS,IMBRIQUES A 45 DEGRES) pour savoir que la situation à laquelle nous faisons référence est "les joints du circuit magnétique sont imbriqués à 45 degrés". De façon analogue, il est suffisant d'inclure dans S2 l'arc (=,QUALITE DES TOLES,Q3) pour identifier la situation "la qualité des tôles du circuit magnétique est Q3".

Si une situation est précisée par l'existence de plusieurs relations entre objets et/ou plusieurs relations =, ≠, <, >, ≤, ≥ elles seront contenues dans le même espace. Par exemple, l'énoncé "Si P1 de x est égale à A, alors P2 de x est différente de P3 de x et P3 de x est supérieure à B" est représenté dans la figure 2.25. L'espace S2 contient les arcs (≠,P2,P3) et (>,P3,B) pour indiquer que la situation imposée est celle où ces deux relations se vérifient. Nous dirons que S2 représente une conjonction de situations.

Il est aussi possible que la partie SI et/ou la partie ALORS d'une contrainte soit une disjonction de situations. Par exemple, l'énoncé "si P1 de x est égale à A ou si P2 de x est différente de B, alors P3 de x est supérieure à C" est représentée dans la figure 2.26. Ici, la situation représentée par S3 est imposée soit par l'existence de la situation S1, soit par l'existence de la situation S2. Ce fait est indiqué par l'existence de deux parties SI dans la figure.

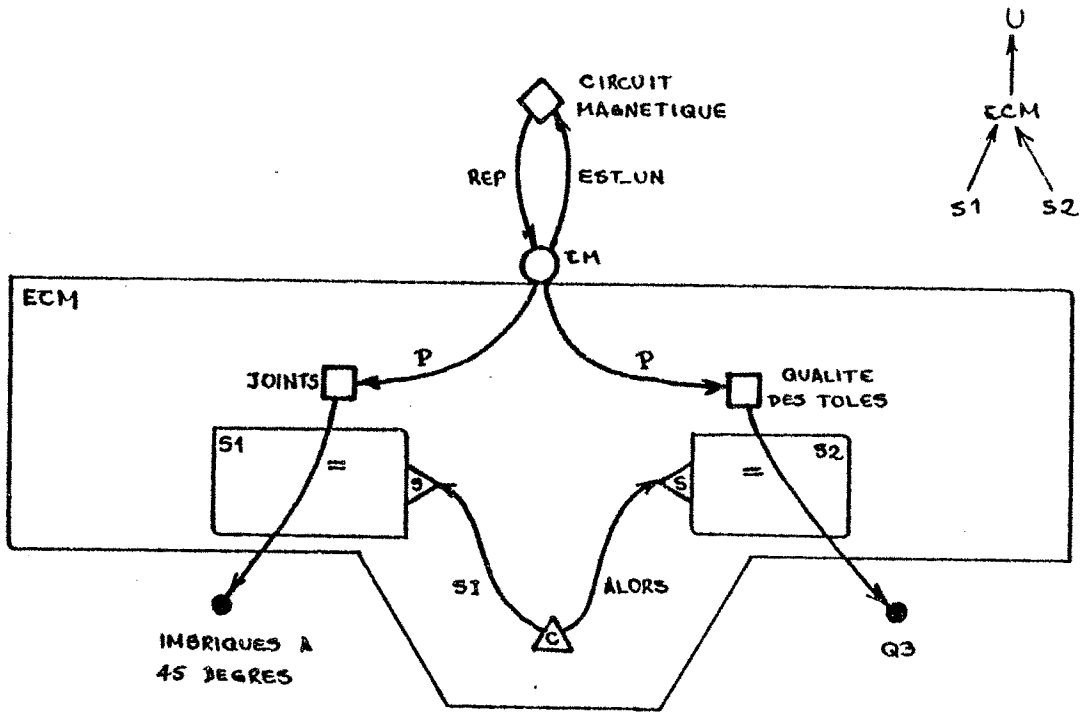


Figure 2.24.

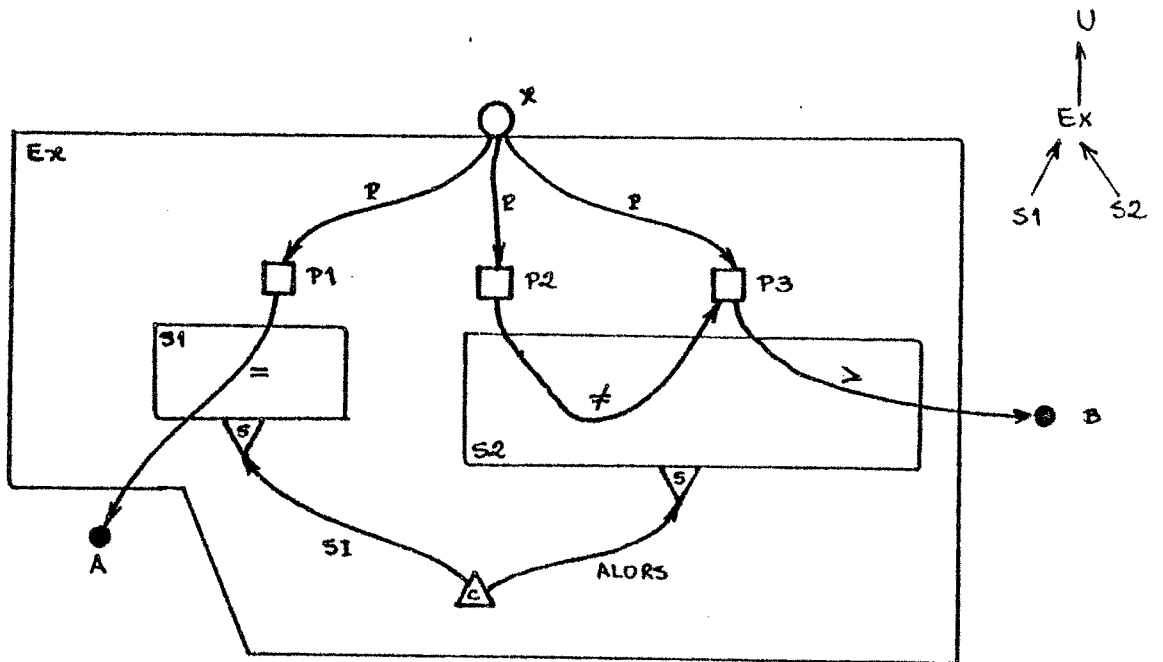


Figure 2.25.

Par ailleurs, $S1 \triangleleft (S1, Ex, U)$, $S2 \triangleleft (S2, Ex, U)$ et $S3 \triangleleft (S3, Ex, U)$.

Jusqu'ici nous avons représenté des contraintes dont les situations font référence à des informations contenues dans un même contexte. Il peut arriver que les situations soient exprimées par des informations qui se trouvent dans des contextes différents. Par exemple, l'énoncé "Si la puissance d'un transformateur est supérieure à 100, alors la qualité des tôles de son circuit magnétique doit être Q3" est représenté dans la figure 2.27. Dans ce cas, la partie SI est donnée par des informations concernant TR tandis que la partie ALORS est donnée par des informations concernant CM. Ce dernier objet représente un des composants de TR et la valeur de sa propriété QUALITE_DES_TOLES aura la valeur Q3 si la valeur de la propriété PUISSANCE de TR est supérieure à 100. Ici, $S2 \triangleleft (S2, ECM, ETR, U)$.

Dans le cas général, lorsque plusieurs assemblages sont nommés par une contrainte, celle-ci est placée dans le contexte de celui qui contient tous les autres. Si un tel assemblage n'est pas spécifié par la contrainte, on place la contrainte dans le contexte du plus proche ancêtre commun de tous les objets concernés.

8.2. Figure de "situation"

Cette figure sert à représenter une situation (cf. § 8.1.). La figure 2.28 montre le schéma correspondant, où

- le sommet noté $\triangle S$ représente la situation,
- les arcs $a1, \dots, an$ identifient les valeurs possibles de propriété et/ou les relations entre objets qui déterminent la situation,
- l'espace ES associé à $\triangle S$ contient $a1, \dots, an$.

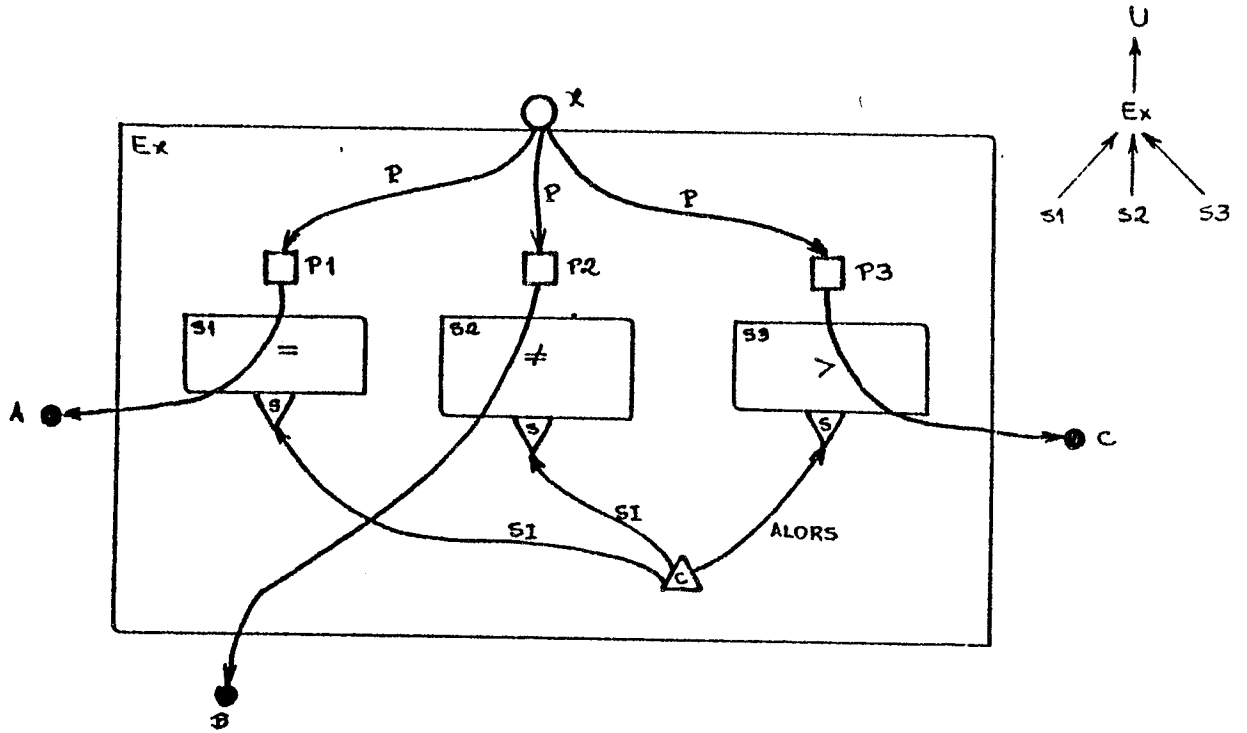


Figure 2.26

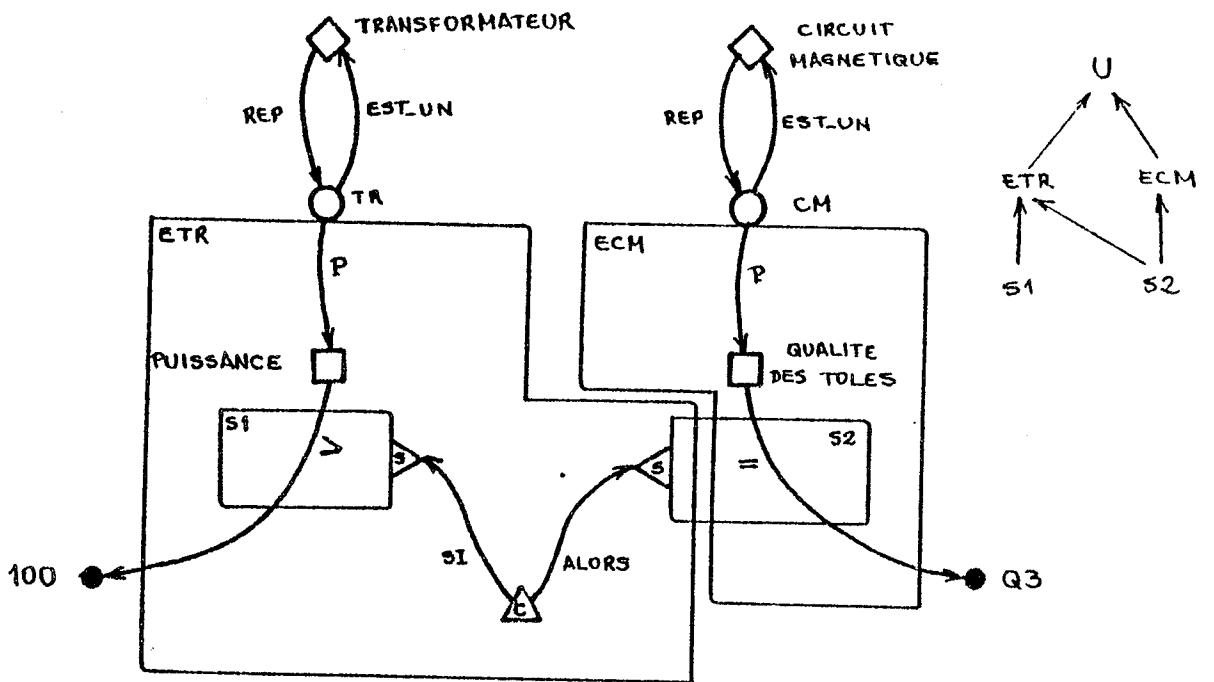


Figure 2.27.

8.3. Figure de "contrainte"

Cette figure permet la représentation des contraintes. La structure correspondante est montrée par la figure 2.29, où

- le sommet noté \triangle représente une contrainte,
- $s_1, \dots, s_m, a_1, \dots, a_n$ sont des situations reliées au sommet \triangle par des arcs SI ou ALORS.

Les arcs SI et les arcs ALORS indiquent que si au moins une des situations s_1, \dots, s_m est vérifiée, alors au moins une des situations a_1, \dots, a_n doit se vérifier aussi.

9 - FIGURE DE "CONSEIL"

Le processus de résolution du problème de conception peut être guidé dans TROPIC par des conseils sur les décisions à prendre dans des situations particulières. Ces conseils sont exprimés sous la forme d'expressions conditionnelles:

Si <A> alors (il est conseillé)

où <A> et sont des situations. Par exemple, l'énoncé

"Si la puissance d'un transformateur est supérieure à 100 kva, il est conseillé d'utiliser l'huile comme réfrigérant"

conseille de choisir l'huile comme réfrigérant d'un transformateur dans la situation où la puissance de celui-ci est supérieure à 100 kva. La figure 2.30 montre la représentation correspondante. Le sommet noté \triangle représente un conseil dont la partie SI est la situation S1 et la partie ALORS (situation conseillée) est la situation S2. S1 indique la situation "la valeur de la propriété PUISSANCE de x est supérieure à 100" et S2 indique "la valeur de la propriété REFRIGERANT de x est égale à HUILE".

Un conseil peut comporter plusieurs parties SI et/ou plusieurs parties ALORS, de la même façon que les contraintes.

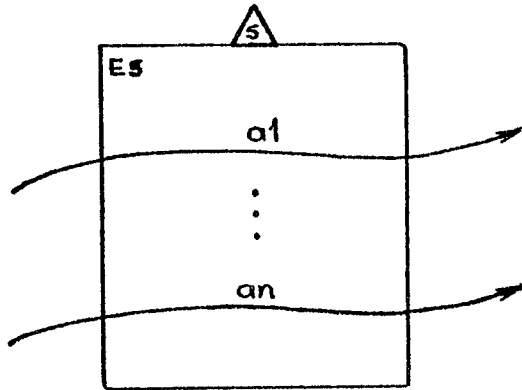


Figure 2.28.

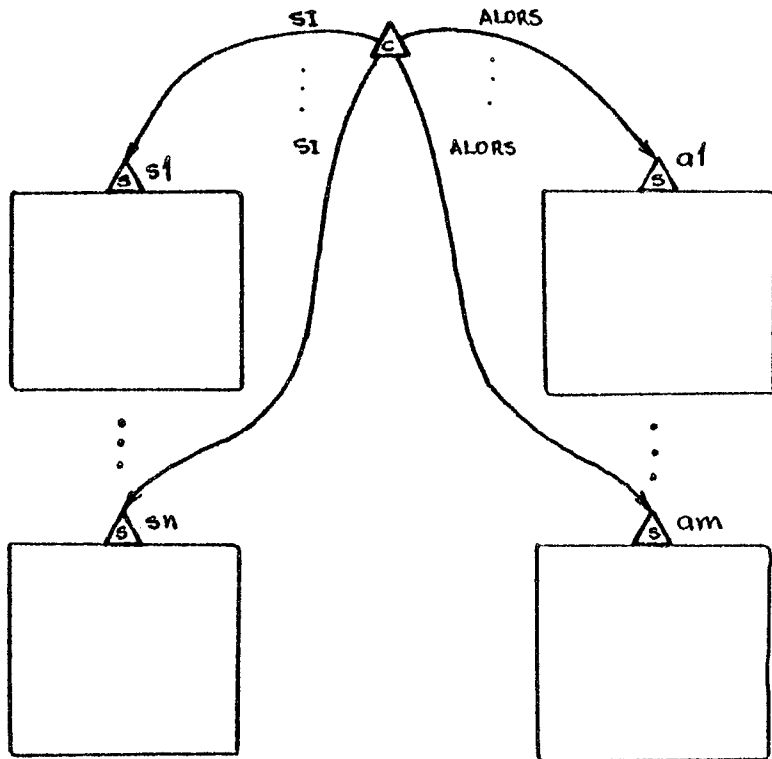


Figure 2.29.

La figure de conseil sert à structurer la représentation des conseils. La figure 2.31 montre le cas général où

- le sommet noté $\triangle A$ représente un conseil,
- $s_1, \dots, s_m, a_1, \dots, a_n$ sont des situations reliées à $\triangle A$ par des arcs SI et ALORS.

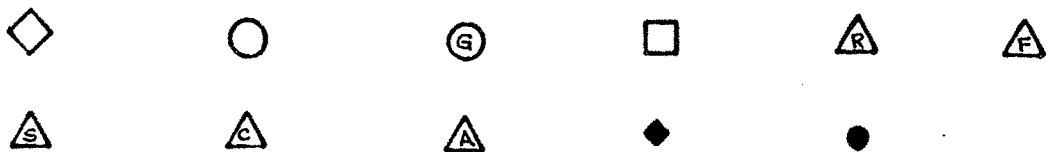
Les arcs SI et les arcs ALORS indiquent que si au moins une des situations s_1, \dots, s_m est vérifiée, alors il est conseillé qu'au moins une des situations a_1, \dots, a_n soit vérifiée aussi.

10 - RECAPITULATIF DES ELEMENTS D'UN RSP CONNUS A PRIORI DES PROGRAMMES DE L'INTERFACE

1/ Les notions fondamentales d'un RSP

- Notion de sommet étiqueté
- Notion d'arc étiqueté
- Notion d'espace
- Notion de vista
- Notion de propriété d'un sommet, d'un arc et d'un espace
- Notion de figure.

2/ Les types des sommets



3/ Les étiquettes de certains sommets

\bullet^c où c est un nombre entier ou réel

◆ REEL

◆ ENTIER

◆ SYMBOLE

△ MEMSPEC

□ DISP

□ CARD

□ IDENT

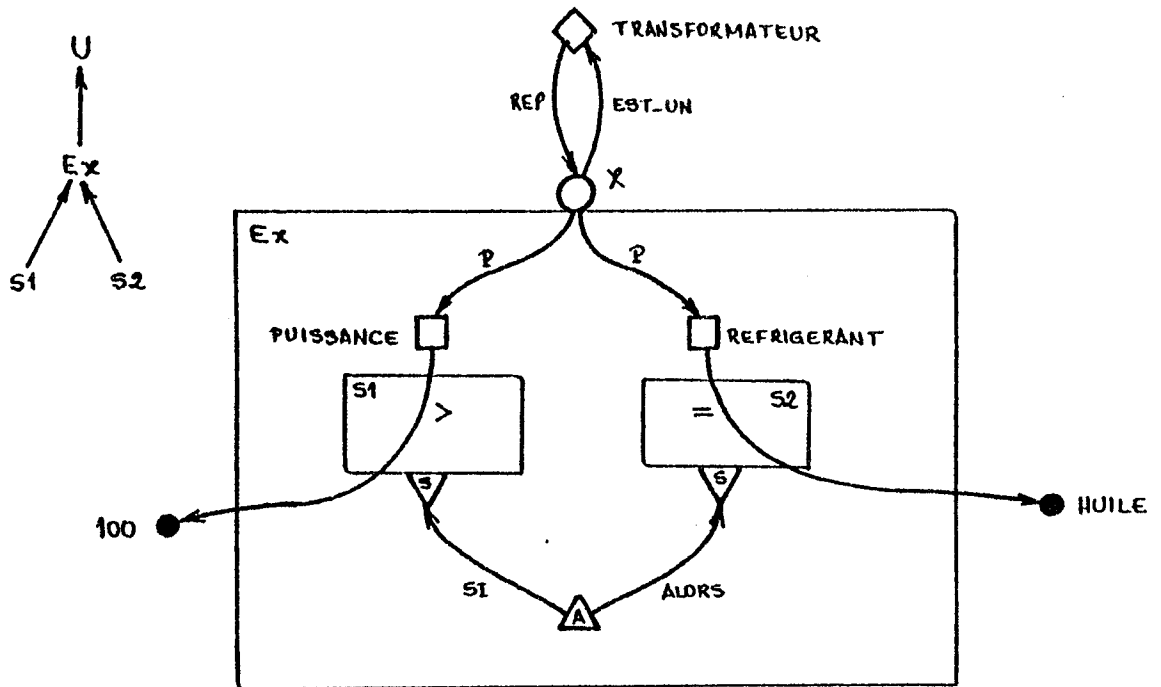


Figure 2.30.

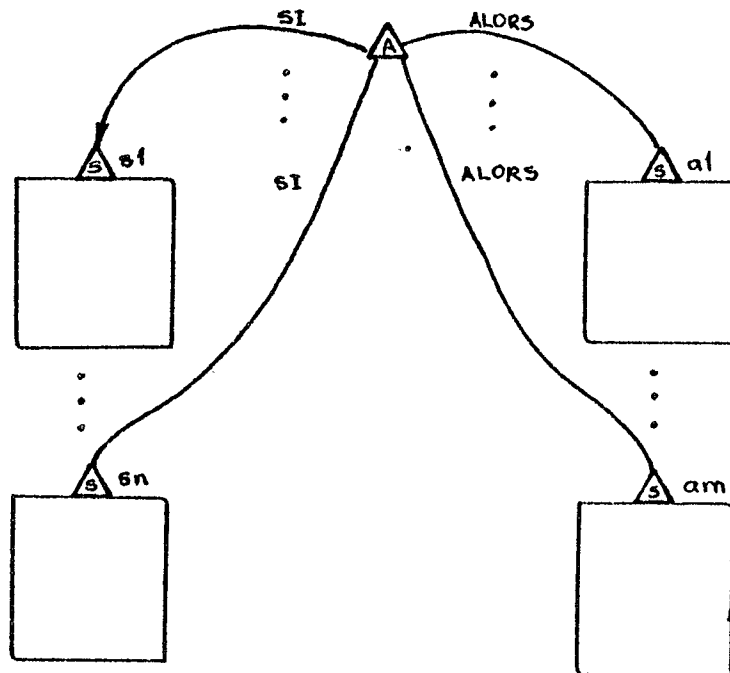
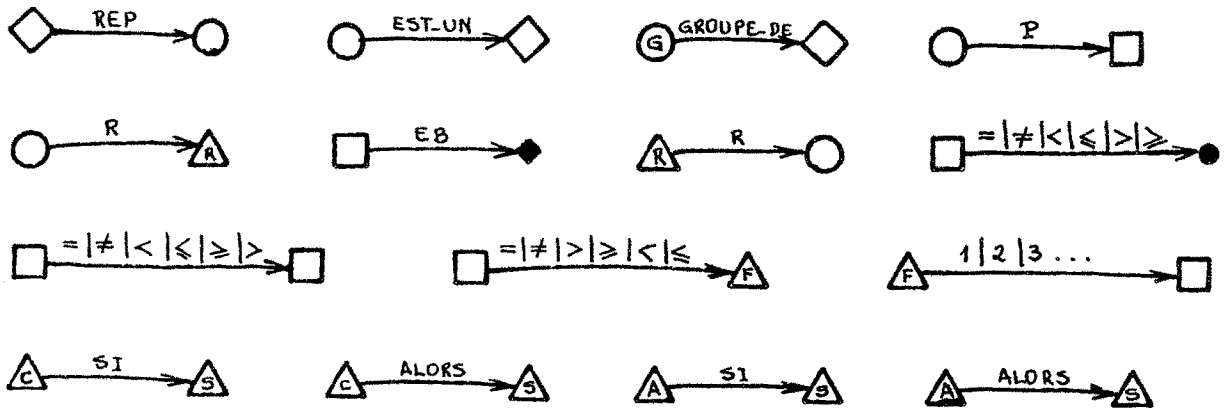


Figure 2.31.

4/ Les étiquettes de certains arcs



5/ Les types des figures

- Figure de description de classe d'objets
- Figure de relation entre objets
- Figure de valeurs possibles de propriété
- Figure de relation n-aire entre propriétés
- Figure de relation binaire entre propriétés
- Figure de groupe d'objets
- Figure de situation
- Figure de contrainte
- Figure de conseil

6/ Le langage LISP (par l'intermédiaire des propriétés des arcs, des sommets et des espaces).

CHAPITRE III

ANALYSES SYNTAXIQUE ET SEMANTIQUE

1 - ANALYSES MORPHOLOGIQUE ET SYNTAXIQUE -----	61
1.1. Analyse morphologique -----	62
1.2. Analyse syntaxique -----	63
1.3. Exemples -----	64
2 - ANALYSE SEMANTIQUE -----	68
2.1. Les formes réduites -----	68
2.1.1. Forme réduite de "propriété d'objet" -----	69
2.1.2. Forme réduite de "relation entre objets" -----	70
2.1.3. Forme réduite de "contrainte" -----	71
2.1.4. Forme réduite de "conseil" -----	72
2.2. Les descripteurs sémantiques -----	72
2.3. Construction des formes réduites -----	74
2.4. Exemples -----	79

Le traitement de chaque phrase a pour but d'incorporer les informations qu'elle apporte dans la base de connaissances. Ce traitement comporte trois phases (cf. chapitre 1) :

- a) Analyses morphologique et syntactique,
- b) Elaboration d'une forme réduite de la phrase (analyse sémantique),
- c) Incorporation dans la base des informations nouvelles.

Dans ce chapitre nous décrivons les deux premières phases.

1 - ANALYSES MORPHOLOGIQUE ET SYNTAXIQUE

Nous considérons l'analyse syntaxique comme un pas nécessaire et important vers une analyse plus profonde de l'énoncé (Iomdine et al 1977, Joloboff 1978). En effet, la syntaxe permet la distinction sujet-objet et la structuration de l'énoncé en groupes nominaux et prépositionnels, très importants dans l'analyse sémantique. Par ailleurs, en définissant des règles grammaticales appropriées, on peut aussi produire des découpages particuliers de l'énoncé et/ou détecter des incohérences au niveau syntaxique. Pour cela, l'analyse syntaxique fait fonction d'heuristique pour faciliter l'analyse sémantique ultérieure.

De son côté, l'analyse morphologique sert non seulement à déterminer les conjugaisons, mais aussi à identifier des locutions du langage, des mots composés et des formes (groupes de mots particuliers).

Nous nous sommes servis du système PIAF pour effectuer les analyses morphologique et syntaxique des phrases. Nous rappelons ci-dessous le principe du fonctionnement de ce système (cf. Courtin, 1978, pour une description détaillée) en montrant les particularités de son utilisation dans le cadre de notre travail.

1.1. Analyse morphologique

Elle est faite au moyen d'un transducteur général d'états finis utilisant un dictionnaire, un ensemble de modèles et une grammaire de validation (Grandjean 1975, Courtin 1977) :

a/ Le dictionnaire contient des bases, préfixes, suffixes et des desinences. Il inclut aussi des mots composés comme "circuit magnétique", "qualité des tôles" et des formes comme "plus grand que", "à coté de". Le caractère espace n'est pas un séparateur. Chaque entrée du dictionnaire fait référence à un modèle.

b/ Un modèle détermine, par sa participation dans la grammaire, le comportement linguistique des éléments qui lui font référence.

c/ La grammaire de validation autorise ou interdit la concaténation des différents éléments identifiés par le dictionnaire et détermine les catégories et variables syntaxiques qui leur sont associées.

Nous avons étendu le dictionnaire connu de PIAF avec un vocabulaire approprié à la description d'assemblages électromécaniques. On y trouve des mots simples et des mots composés comme "transformateur", "circuit magnétique", "réfrigérant", "coefficient de saturation", "huile", "biconcentrique", "imbriqué à 90 degrés", "kva", "hertz", etc..., ainsi que des formes comme "égal à", "plus petit que", "autour de", etc... Ils font référence à des modèles particuliers pour déterminer leur comportement syntaxique. Par exemple, la forme "égal à" fait référence à un modèle dont la catégorie syntaxique est ADJC (adjectif comparatif).

Parallèlement, nous avons défini des modèles qui déterminent le comportement sémantique des mots et des formes. Ces modèles sont utilisés dans la construction de la structure profonde (forme réduite) de la phrase (cf. § 2).

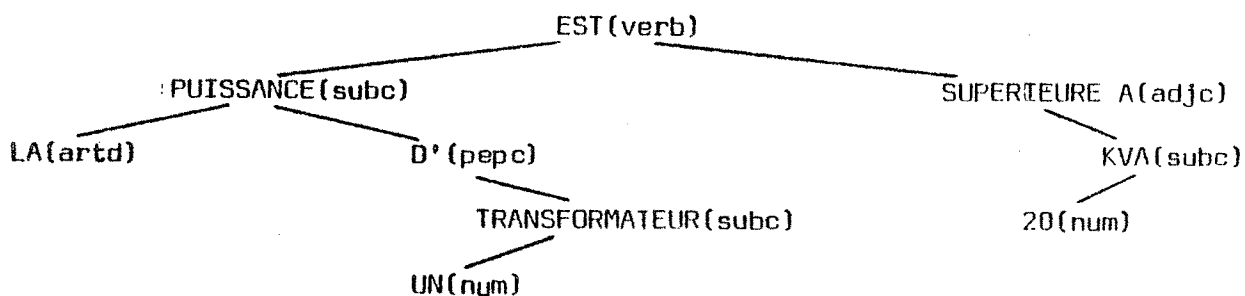
Le résultat de l'analyse morphologique est une fragmentation de l'énoncé en unités lexicales (mots ou racines de mots) accompagnées des indicateurs de classe syntaxique et des variables syntaxiques. Les ambiguïtés éventuelles sont prises en compte. Par exemple, l'analyse de "la puissance d'un transformateur est supérieure à 20 kva" produit :

LA : artd,fem,sin| .popl, fem,sin,acc
 PUISSANCE : subc,fem,sin
 D' : pepc
 UN : num
 TRANSFORMATEUR : subc,mas,sin
 EST : verb,pres,ind,tre,xat
 SUPERIEURE A : adjc
 20 : num
 KVA : subc,mas,sin

1.2. L'analyse syntaxique

Cette analyse produit une structure arborescente qui montre les dépendances syntaxiques entre les unités lexicales de l'énoncé. Ces dépendances sont définies entre les catégories syntaxiques fournies par l'analyse morphologique. Les structures sont construites à l'aide d'un ensemble de relations entre catégories. Ces relations déterminent l'emplacement des unités lexicales dans l'arborescence, laquelle est projective (Chomsky,1957).

La structure de dépendances qui correspond à la phrase "la puissance d'un transformateur est supérieure à 20 kva" est



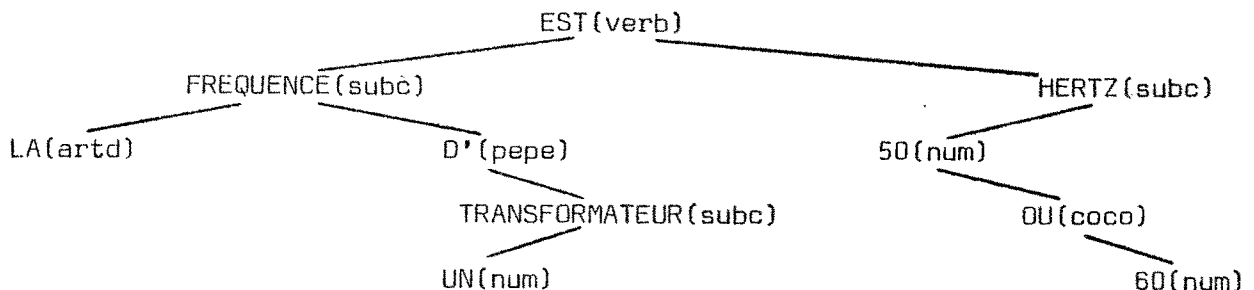
Cette arborescence a été construite grâce à des relations indiquant qu'un "subc" peut "dépendre à gauche" d'un "verb", qu'un "adj" peut "dépendre à droite" d'un "verb", etc...

Nous avons réduit l'ensemble de relations de dépendance de PIAF en éliminant les cas correspondant à des structures rarement utilisées pour exprimer les connaissances nécessaires à TROPIC. Cette simplification, qui réduit les possibilités d'ambiguïtés syntaxiques, nous a permis d'obtenir très peu d'arborescences de dépendances par phrase, et souvent une seule.

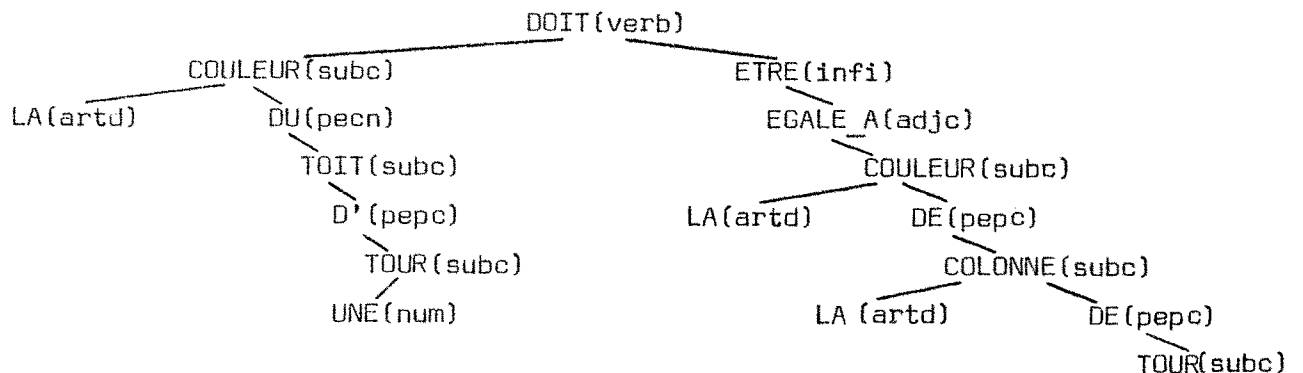
1.3. Exemples

Nous montrons ci-dessous les structures de dépendances de quelques phrases caractéristiques :

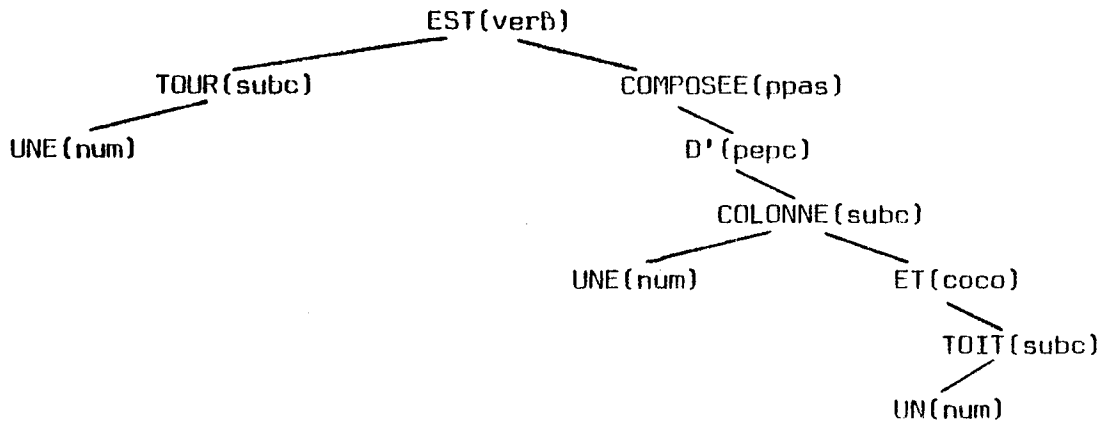
(1) La fréquence d'un transformateur est 50 ou 60 hertz :



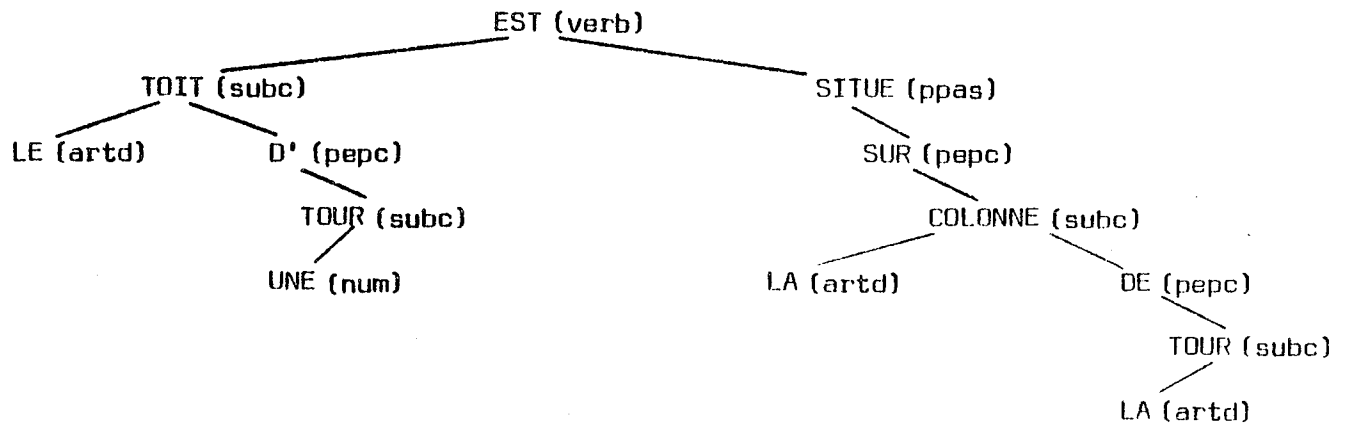
(2) La couleur du toit d'une tour doit être égale à la couleur de la colonne de la tour :



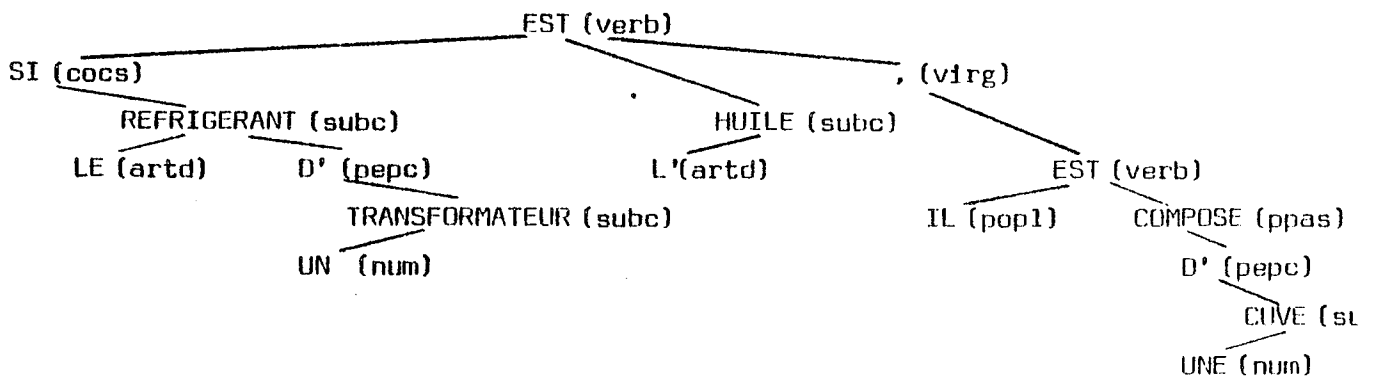
(3) Une tour est composée d'une colonne et d'un toit :

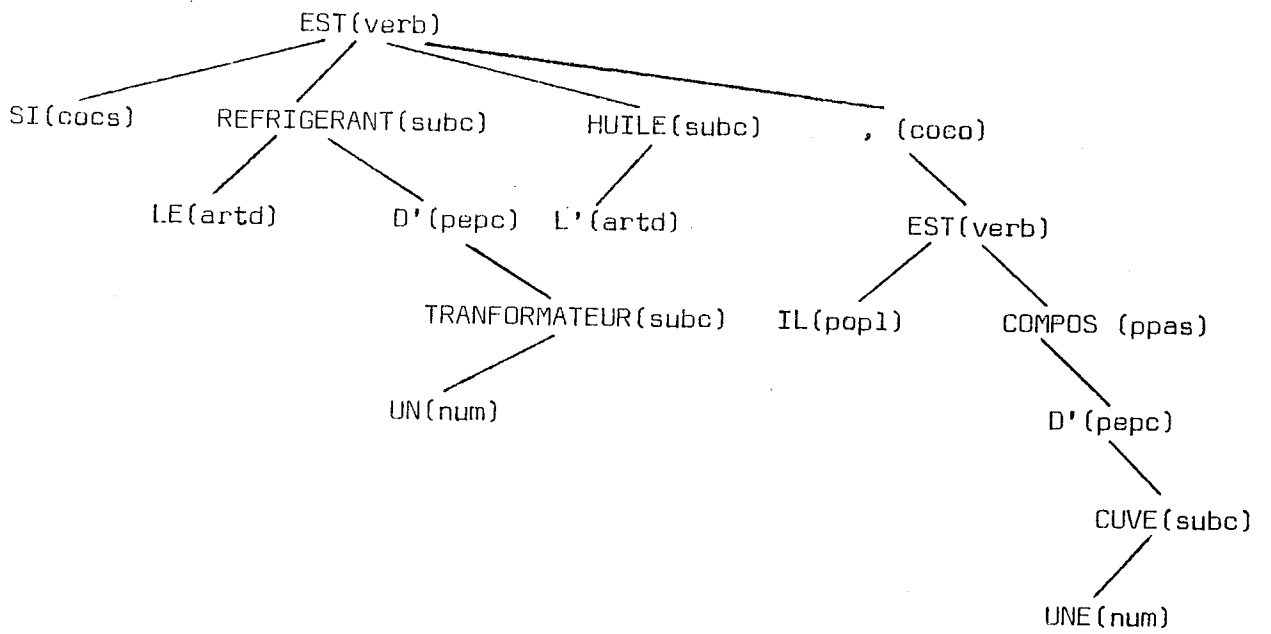
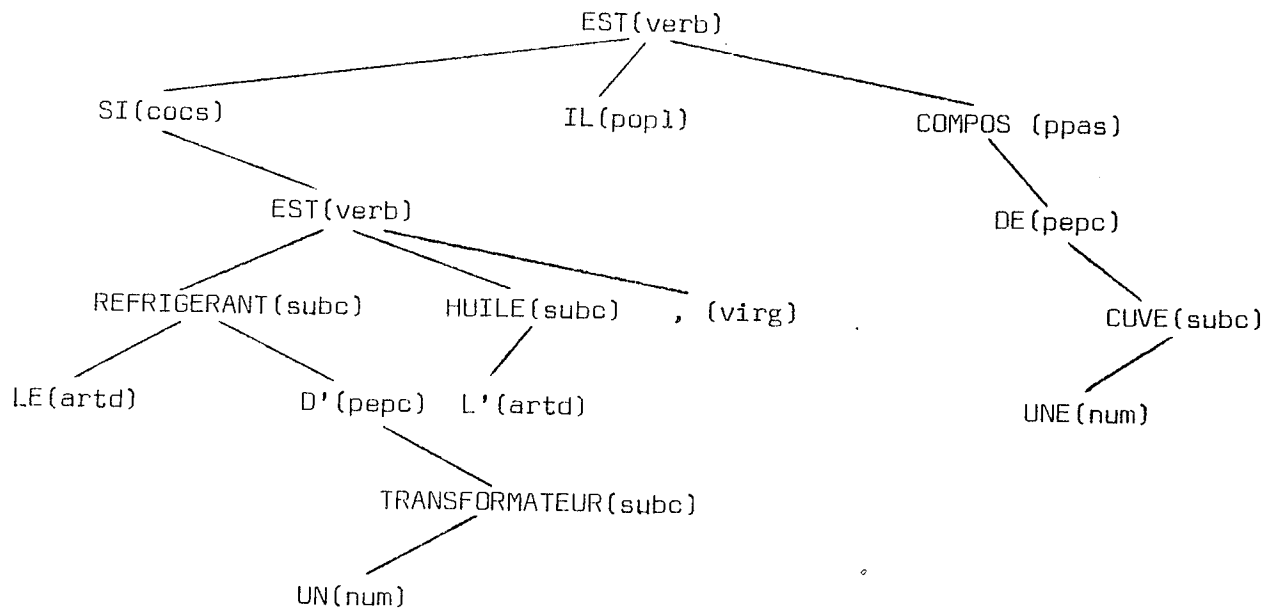


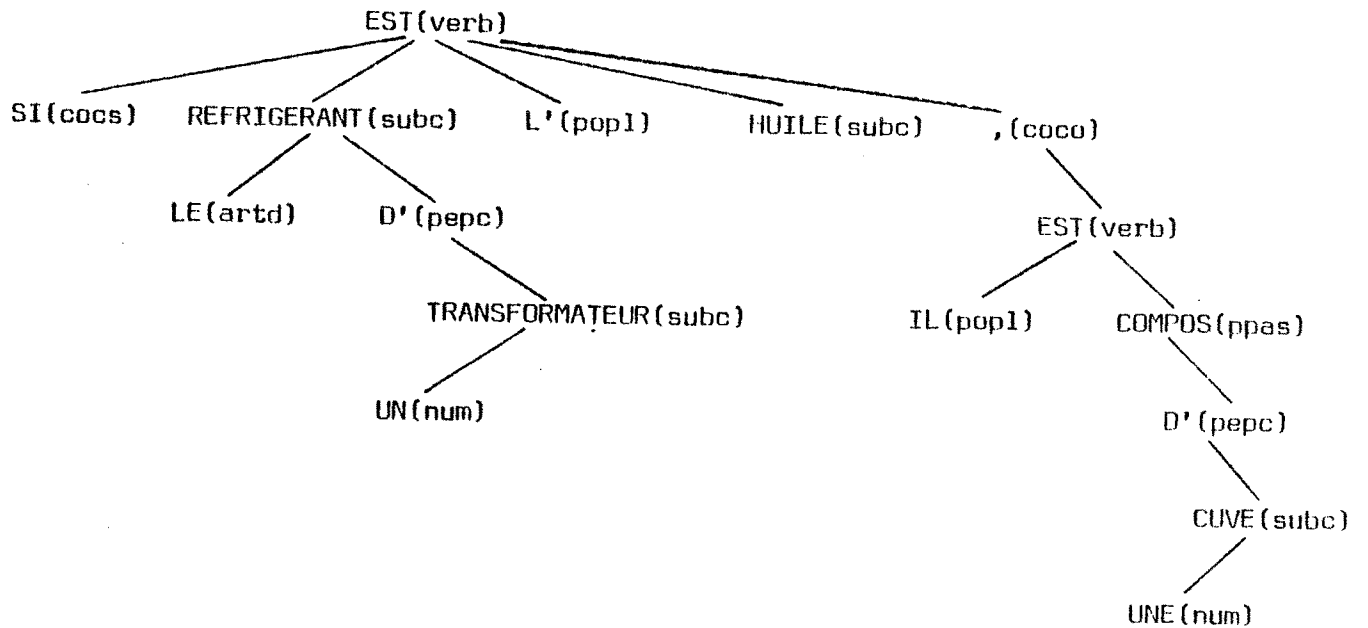
(4) Le toit d'une tour est situé sur la colonne de la tour



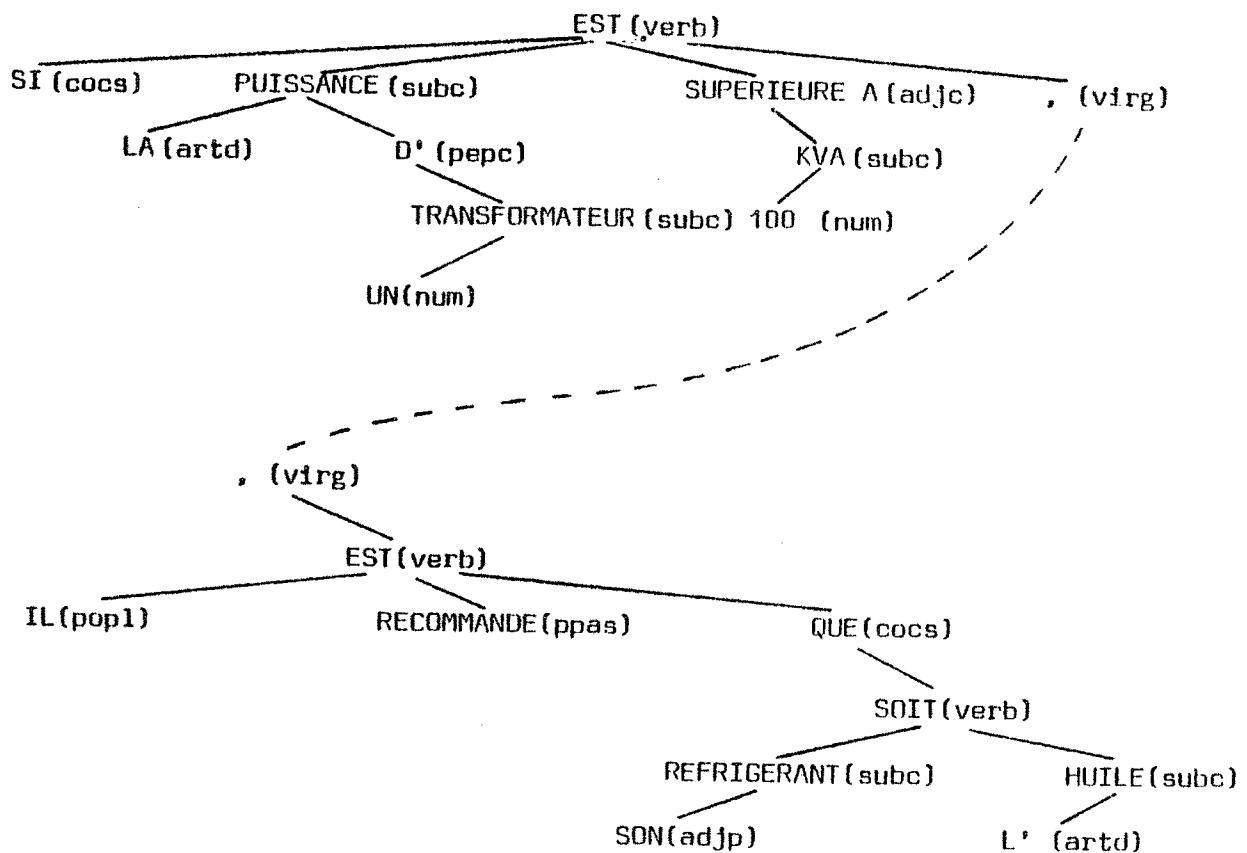
(5) Si le réfrigérant d'un transformateur est l'huile, il est composé d'une cuve :







(6) Si la puissance d'un transformateur est supérieure à 100 kva,
il est recommandé que son réfrigérant soit l'huile :



2 - ANALYSE SEMANTIQUE

De chaque phrase on extrait une forme réduite sensée contenir la signification de la phrase dans le contexte des assemblages physiques. Cette forme est déterminée à partir des structures de dépendances de la phrase fournies par PIAF par un processus qui comporte trois types d'opérations exécutées itérativement :

- . Identification des éléments pouvant jouer un rôle dans une forme réduite.
- . Validation des relations entre les différents éléments.
- . Détermination du rôle de chaque élément.

Pendant ce processus, le système utilise un ensemble de descripteurs sémantiques, associés aux mots de la phrase. Ces descripteurs indiquent le rôle des mots et permettent aussi de réaliser certaines vérifications sémantiques. D'autres vérifications sont faites à l'aide des connaissances déjà existantes dans le réseau.

Nous présentons successivement les formes réduites (§ 2.1.), les descripteurs sémantiques utilisés pour leur construction (§ 2.2.), le mécanisme qui transforme les arborescences de dépendances en formes réduites (§ 2.3.) et quelques exemples illustratifs (§ 2.4.).

2.1. Les formes réduites

Les formes réduites ont une structure arborescente qui ressemble à la représentation par cas (Fillmore, 1968), très souvent utilisée pour représenter l'interprétation conceptuelle de phrases ("structure profonde"). A titre d'illustration, les formes réduites qui correspondent aux phrases du § 1.3. sont :

- 1 - (FREQUENCE TRANSFORMATEUR(OU (= 50) (= 60)))
- 2 - (COULEUR (TOIT (DE TOUR)) (= (COULEUR (COLONNE (DE TOUR)))))
- 3 - (COMPOSITION TOUR (ET COLONNE TOIT))
- 4 - (SUR (TOIT (DE TOUR)) (COLONNE (DE TOUR)))

- 5 - (COND (REFRIGERANT TRANSFORMATEUR (= HUILE))
 (COMPOSITION TRANSFORMATEUR CUVE))
- 6 - (CONS (PUISSANCE TRANSFORMATEUR (> 100))
 (REFRIGERANT TRANSFORMATEUR (= HUILE)))

Nous distinguons à ce niveau quatre familles de connaissances qui concernent respectivement les propriétés d'objets, les relations entre objets, les contraintes et les conseils. Pour les représenter nous avons défini quatre structures de formes réduites. Nous décrivons ci-dessous ces quatre structures à l'aide d'une notation BNF, sans vouloir fournir une grammaire pour leur traitement.

<forme réduite> ::= <fréduite de propriété>|<fréduite de relation> |
 <fréduite de contrainte>|<fréduite de conseil>

2.1.1. Forme réduite de "propriété d'objet" (exemples 1 et 2)

<fréduite de propriété> ::= (<propriété><objet1><description de propriété>)
 <propriété> ::= chaîne alphanumérique ayant l'indicateur sémantique PROPRIETE
 <objet1> ::= <classe>|(<classe><modificateur1>)
 <classe> ::= chaîne alphanumérique ayant l'indicateur sémantique CLASSE
 <modificateur1> ::= (DE <objet1>) |
 (IDENT (= <identificateur>)) (DE <objet1>)
 <identificateur> ::= chaîne alphanumérique ayant l'indicateur sémantique MEMSPEC
 <description de propriété> ::= <relation primitive>|
 (OU <relation primitive><relation primitive>⁺)|
 (ET <relation primitive><relation primitive>⁺)
 <relation primitive> ::= (<nom de rep><valeur de rep>)
 <nom de rep> ::= = | ≠ | > | ≥ | ≤ | <
 <valeur de rep> ::= <constante>|<référence à propriété>|
 <expression fonctionnelle>

* indique que l'élément concerné peut apparaître une ou plusieurs fois.

<constante> ::= toute valeur numérique ou chaîne alphanumérique
 <référence à propriété> ::= (<propriété><objet 1>)
 <expression fonctionnelle> ::= (FONCTION <nom de fonction><paramètre>⁺)
 <nom de fonction> ::= chaîne alphanumérique
 <paramètre> ::= <référence à propriété>

Commentaires :

- <propriété> indique le nom de la propriété d'objet : PUISSANCE, HAUTEUR, COULEUR ...
- <objet 1> identifie l'objet dont on décrit la propriété. Un objet peut être identifié par le nom de sa classe (TRANSFORMATEUR, TOUR,...) ou par ce nom accompagné d'un modificateur.
- Le modificateur DE indique que l'objet est un composant d'un autre objet. "La colonne de la tour" correspond à (COLONNE (DE TOUR)).
- Le modificateur IDENT sert à repérer un objet particulier d'un groupe. "Le plus haut bloc de la colonne" correspond à (BLOC (IDENT (= PLUS HAUT))(DE COLONNE)).
- <description de propriété> sert à indiquer les valeurs possibles ou les restrictions imposées à la valeur d'une propriété. Il y a plusieurs cas :
 - a/ la valeur de la propriété est précisée par des relations avec des constantes : (QUALITE DES TOLES CIRCUIT_MAGNETIQUE (= Q3)), (REFRIGERANT TRANSFORMATEUR (OU (= AIR) (= HUILE))), (PUISSANCE TRANSFORMATEUR (ET (> 20)(< 2000))).
 - b/ la valeur de la propriété est précisée par une relation avec la valeur d'une deuxième propriété : (COULEUR (TOIT (DE TOUR)) (= (COULEUR (COLONNE (DE TOUR)))))).
 - c/ la valeur de la propriété est précisée par une relation avec le résultat d'une fonction : (ECHAUFFEMENT DE L'HUILE TRANSFORMATEUR (= (FONCTION EH (PERTES TRANSFORMATEUR)(SURFACE (CUVE (DE TRANSFORMATEUR)))))), où la fonction EH est définie ailleurs (cf. § 5.2. du chapitre 2).

2.1.2. Forme réduite de "relation entre objets" (exemples 3 et 4)

<fréduite de relation> ::= (<relation><objet primaire><objet secondaire>)
 <relation> ::= chaîne alphanumérique ayant l'indicateur sémantique RELATION.
 <objet primaire> ::= <objet 1>
 <objet secondaire> ::= <objet>|(ET <objet><objet>⁺)

```

<objet> ::= <objet1>| (<classe><modificateur2>)
<modificateur2> ::= (CARD (<nom de rep><entier>)
                    (DISP (= (<disposition>))
                    (CARD (<nom de rep><entier>))(DISP (= <disposition>)))
<disposition> ::= chaîne alphanumérique ayant l'indicateur sémantique DGROUPE.
<entier> ::= nombre entier supérieur à 0

```

Commentaires :

- <relation> indique le nom de la relation qui existe entre l'objet primaire et l'objet secondaire. Par exemple (COMPOSITION TRANSFORMATEUR CUVE) ,(AUTOUR ENROULEMENT NOYAU_MAGNETIQUE). Si la relation existe entre l'objet primaire et plusieurs objets, ceci est indiqué par un objet secondaire composé par une énumération (conjonction) de ces objets, par exemple : (COMPOSITION TOUR(ET COLONNE TOIT)).

- Le modificateur CARD indique qu'un objet secondaire est un groupe d'objets dont on précise la cardinalité. "Une colonne est composée de trois blocs" correspond à (COMPOSITION COLONNE (BLOC (CARD (= 3)))).

- Le modificateur DISP indique la disposition des objets d'un groupe. "Une pile de trois blocs" correspond à (BLOC (CARD (= 3))(DISP (= PILE))).

2.1.3. Forme réduite de "contrainte" (exemple 5)

```

<fréduite de contrainte> ::= (COND <si> <alors>)
<si> ::= <situation>
<alors> ::= <situation>
<situation> ::= <fréduite1>|
              (OU <fréduite1><fréduite1>+)|
              (ET <fréduite1><fréduite1>+)
<fréduite1> ::= <fréduite de propriété>|<fréduite de relation>

```

Commentaires :

- la partie <si> indique les conditions d'application de la contrainte. La partie <alors> indique les conséquences de son application.

Exemples :

- (COND (REFRIGERANT TRANSFORMATEUR (= HUILE))
 (COMPOSITION TRANSFORMATEUR CUVE))
- (COND (ET (HAUTEUR TOUR (> 100)) (POIDS (TOIT (DE TOUR)) (> 8))
 (SECTION (COLONNE (DE TOUR)) (= RONDE)))

2.1.4. Forme réduite de "conseil" (exemple 6)

<fréduite de conseil> ::= (cons <si><alors>)

Commentaire :

La partie <si> indique la situation dans laquelle le conseil est applicable.
La partie <alors> indique les faits conseillés.

Exemples :

- (CONS (PUISSANCE TRANSFORMATEUR (> 100))
 (REFRIGERANT TRANSFORMATEUR (= HUILE)))
- (CONS (HAUTEUR TOUR (> 150))
 (ET (HAUTEUR (TOIT (DE TOUR)) (> 20))
 (HAUTEUR (BLOC (DE (COLONNE (DE TOUR)))) (> 30))))

2.2. Les descripteurs sémantiques

Les données syntaxiques fournies par l'arborescence de dépendances d'une phrase sont insuffisantes pour déterminer la signification de celle-ci. Il est nécessaire de disposer de connaissances sur le contexte de l'application pour pouvoir établir la relation de chaque élément ou groupe d'éléments de la structure syntaxique avec ce contexte. Dans notre interface, ces connaissances sont données par des descripteurs sémantiques qui indiquent les rôles conceptuels que peuvent jouer les mots apparaissant dans la structure syntaxique d'une phrase. En général, ces rôles correspondent aux rôles accordés aux éléments des formes réduites.

Chaque descripteur associe un ou plusieurs indicateurs sémantiques à un mot. L'ensemble des descripteurs constitue un dictionnaire où chaque entrée a la forme /<mot>/<indicateurs sémantiques>/.

Certains indicateurs servent à déterminer le rôle que peut jouer un mot dans une forme réduite et à valider certaines relations entre les éléments de cette dernière (cas des conjonctions) :

- /transformateur/classe/... indique que "transformateur" est le nom d'une classe d'objets
- /circuit magnétique/classe/
- /puissance/propriété/... indique que "puissance" désigne une propriété d'objet
- /forme/propriété/
- /section/propriété,forme/... indique que "section" désigne une propriété décrivant la forme d'un objet
- /sur/relation/... indique que "sur" désigne une relation entre objets
- /composition/relation/
- /autour de/relation/
- /pile/dgroupe/... indique que "pile" est une disposition possible pour les éléments d'un groupe
- /rangée/dgroupe/
- /plus bas/memspec/... indique que "plus bas" identifie un membre spécial d'un groupe
- /plus haut/memspec/
- /dernier/memspec/

Certains indicateurs servent à valider des relations entre les éléments de la forme réduite et, éventuellement, à compléter l'énoncé. Par exemple, savoir que "rouge" et "vert" sont des "couleurs" permet de valider l'énoncé "la couleur d'un toit est rouge ou verte", et de compléter l'énoncé "un toit est rouge ou vert" pour obtenir l'énoncé précédent (cf. §3 du chapitre 6). Voici quelques exemples :

- /carré/forme,section/... indique que "carré" est une valeur possible des propriétés "forme" et "section"
- /pyramidal/forme/
- /rouge/couleur/

Il existe aussi des indicateurs servant à identifier des mots qui peuvent jouer un rôle syntaxique et/ou sémantique pendant l'interprétation de l'énoncé mais qui n'apparaissent pas dans la forme réduite résultante :

- /hertz/unités/... indique que "hertz" est le nom d'une unité de mesure
- /mètre/unités/

D'autres indicateurs sont engendrés dynamiquement pendant la construction de la forme réduite pour transmettre des informations à des étapes différentes de la transformation de l'arborescence. Ceci est le cas des indicateurs "comp" et "nocomp" (cf. §2.3 et 2.4).

Les indicateurs sémantiques sont donc utilisés pendant l'interprétation de la phrase pour déterminer le rôle de ses éléments, permettant ainsi la construction de la forme réduite correspondante et la réalisation des vérifications sémantiques. De cette façon, les informations contenues dans l'arborescence de dépendances d'une phrase se trouvent augmentées par les indicateurs sémantiques associés à certains mots. Les arborescences des exemples 1, 4, 5 et 6 du § 1.3. nous apparaissent alors comme celles des figures 3.1, 3.2, 3.3 et 3.4 respectivement.

2.3. Construction des formes réduites

La forme réduite d'une phrase est obtenue à partir de l'arborescence de dépendances par une série de transformations. Ces dernières sont effectuées par un programme réalisé par J.Lopez(1979). Nous présentons ci-dessous le principe de fonctionnement de ce programme dans le cadre de notre application. Les exemples de la section 2.4. serviront d'illustration.

Chaque transformation résulte de l'application d'une règle de transformation. Elle est accompagnée de vérifications syntaxiques et sémantiques destinées à accepter ou rejeter l'arborescence considérée. Une règle est composée de :

- Une production $PG \Rightarrow PD$, où PG et PD sont des arborescences paramétrées (voir exemple 1 du § 2.4.). Chaque sommet est soit une variable (&x1,&x2...), soit une constante (DU,DE,EST,...), accompagnée par une liste d'indicateurs (subc,artd,propriété,...) qui correspondent aux indicateurs syntaxiques et sémantiques des mots.

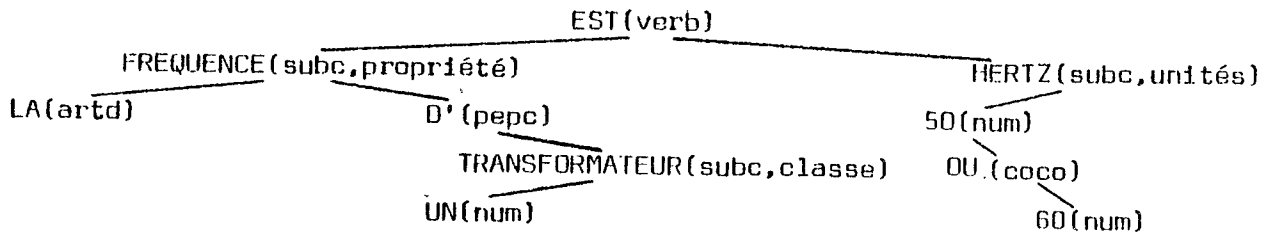


Figure 3.1.

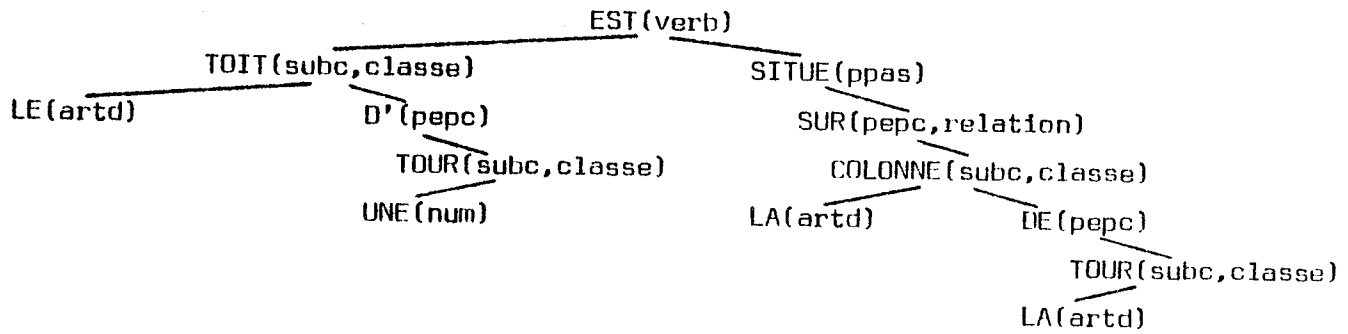


Figure 3.2.

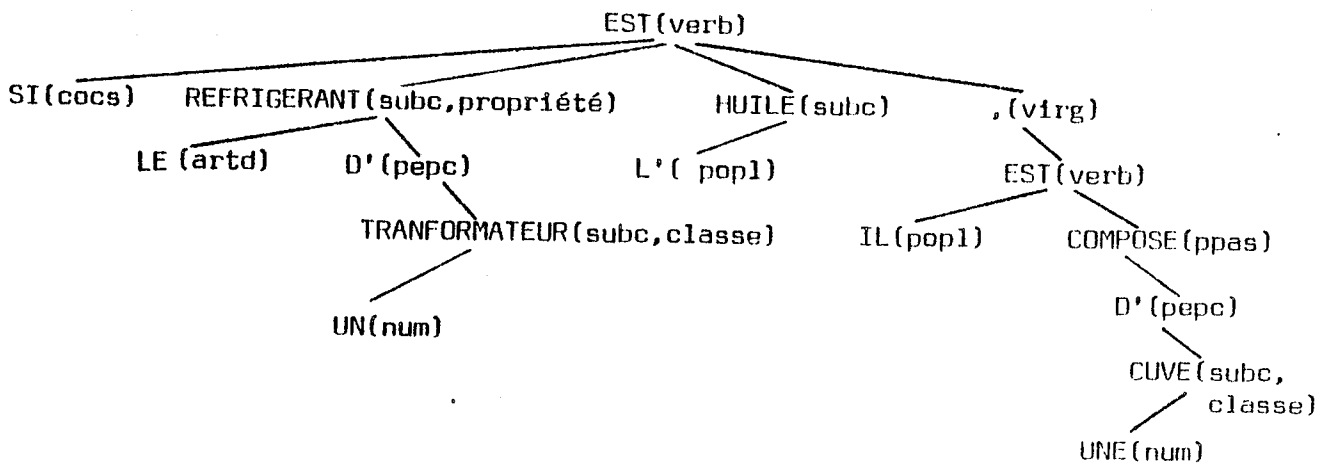


Figure 3.3.

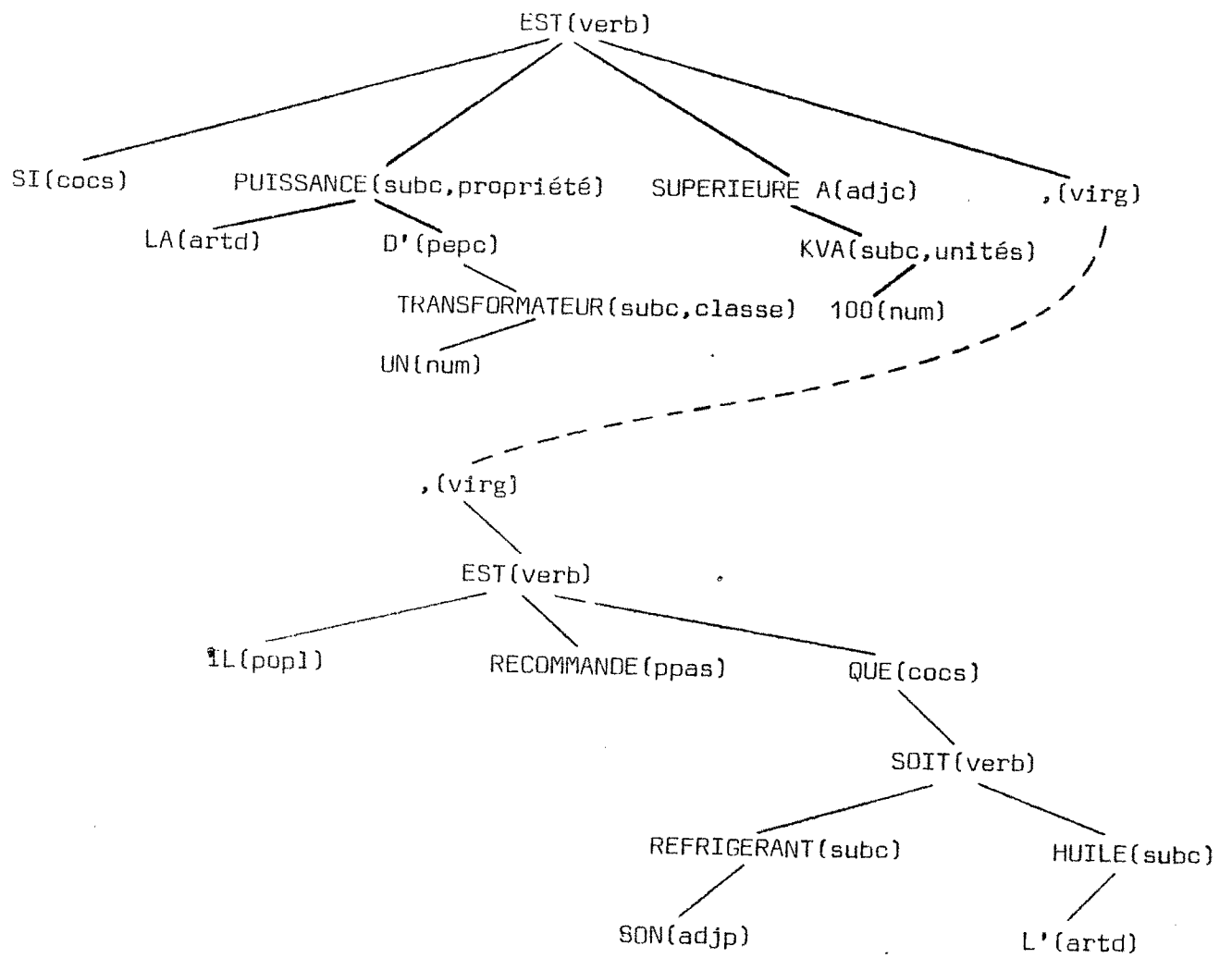


Figure 3.4.

- Un ensemble de conditions ECONO sur les relations entre les sommets de l'arborescence. Ces conditions expriment des vérifications syntaxiques et sémantiques à effectuer.
- Un ensemble d'effets secondaires (ESEC) causés par l'application de la règle.

Une règle de transformation est applicable à une sous-arborescence A de l'arborescence de dépendances considérée si l'arborescence de gauche PG de la règle "coïncide" avec A et si les éléments de A satisfont les conditions imposées par ECONO. La coïncidence entre les éléments des arborescences est définie ainsi :

- Deux constantes identiques coïncident si les indicateurs associés à la constante de PG apparaissent dans la liste des indicateurs de la constante de A (le symbole "{" indique des alternatives).
- Une variable de PG coïncide avec une constante de A si les indicateurs associés à la variable apparaissent dans la liste des indicateurs de la constante, ou si la constante apparaît parmi les indicateurs de la variable.

L'application de la règle a pour effet de remplacer A par l'arborescence de droite PD de la règle où :

- les variables sont remplacées par les constantes correspondantes de A ;
- les indicateurs associés aux sommets de PD sont ajoutés à ceux qu'avaient déjà leurs correspondants de A.

En plus, les actions indiquées dans ESEC sont exécutées.

Ainsi, l'application d'une règle correspond à l'identification, vérification et interprétation (partielles) d'une partie de l'énoncé, et à la construction (partielle aussi) de la partie correspondante de la forme réduite.

La transformation d'une arborescence est arrêtée lorsqu'il n'y a plus de règle applicable. L'arborescence obtenue doit vérifier deux conditions pour être acceptée comme forme réduite :

- 1/ Chaque sommet a dû participer à au moins une transformation (ceci pour assurer une interprétation complète).
- 2/ La racine de l'arborescence doit avoir l'indicateur "*fréduite".

Le processus s'arrête aussi lorsqu'une règle d'arrêt (règles dont la partie droite est nil) est appliquée (voir exemple 2). L'arborescence considérée est alors rejetée par le programme. Ceci permet l'élimination des structures correspondant à des phrases sémantiquement incorrectes, ou à des interprétations syntaxiques indésirables dues aux ambiguïtés dans les relations de dépendances.

Les règles de transformation sont groupées en plusieurs ensembles spécialisés (voir appendice A) : règles d'arrêt, règles pour le traitement des conjonctions, règles générales, règles pour le traitement des propriétés d'objets, des relations entre objets, des contraintes, des conseils, des groupes, ... Le programme qui les applique les considère dans un certain ordre : d'abord les règles d'arrêt, puis celles qui traitent les conjonctions et enfin les autres règles.

Si à une phrase correspondent plusieurs arborescences de dépendances, elles sont traitées une à une. L'interface conserve la première forme réduite correcte obtenue. Si aucune arborescence ne conduit à une forme réduite, la phrase est rejetée.

2.4. Exemples

Les exemples ci-dessous illustrent les différents aspects de la construction d'une forme réduite.

Exemple 1 :

Soit la phrase "la fréquence d'un transformateur est 50 ou 60 hertz" dont l'arborescence de dépendances (avec les indicateurs sémantiques) est montrée par la figure 3.1. (§ 2.2.).

Aucune règle d'arrêt n'est applicable.

La règle C03 (cf. appendice A) :

$\&\&X1(\text{subc|adjc|adjq|num})$

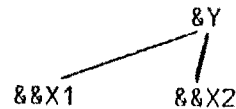
$\&Y$ (et ou)

=>

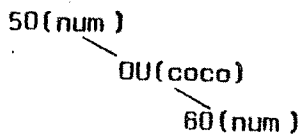
$\&\&X2(\text{subc|adjc|adjq|num})$

ECOND : (CL($\&\&X1$) = CL($\&\&X2$))

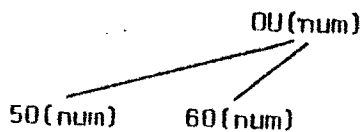
ESEC : (CL($\&Y$) = CL($\&\&X1$))



est appliquée à la sous-arborescence.



pour donner



La règle exige (ECOND) que les classes lexicales (CL) de 50 et 60 soient identiques, ce qui est le cas puisque leur classe lexicale est "num". Ceci permet de regrouper les sommets 50 et 60 sous la conjonction OU. Par ailleurs, les effets secondaires (ESEC) changent la classe lexicale de OU pour lui donner celle des éléments qu'il regroupe, ce qui permettra ensuite sa participation à d'autres transformations.

Les règles R1 et R2 :

$$\begin{array}{l} \text{\&X1(subc)} \\ \diagdown \\ \text{\&X2(artd)} \end{array} \Rightarrow \text{\&X1(defn)},$$

$$\begin{array}{l} \text{\&X1(subc)} \\ \diagdown \\ \text{\&X2(un|une)} \end{array} \Rightarrow \text{\&X1(indef)}$$

sont appliquées à

$$\begin{array}{l} \text{FREQUENCE(subc,propriété)} \\ \diagdown \\ \text{LA(artd)} \end{array}$$

et à

$$\begin{array}{l} \text{TRANSFORMATEUR(subc,classe)} \\ \diagdown \\ \text{UN(num)} \end{array}$$

pour produire

FREQUENCE(subc,propriété,defn) et TRANSFORMATEUR(subc,classe,indef)

Ainsi, les articles définis et indéfinis disparaissent laissant comme trace de leur existence les indicateurs "defn" et "indef"¹.

La règle R5 :

$$\begin{array}{l} \text{\&X1(unités)} \\ \diagdown \\ \text{\&\&X2(num)} \end{array} \Rightarrow \text{\&\&X2}$$

s'applique à

$$\begin{array}{l} \text{HERTZ(subc,unités)} \\ \diagdown \\ \text{OU(num)} \end{array}$$

pour donner

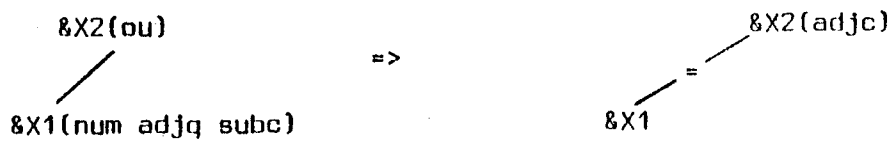
OU(num),

ce qui élimine les unités car TROPIC suppose que les valeurs numériques

(1) Les indicateurs "defn" et "indef" sont utilisés pendant la "normalisation" de la forme réduite (cf. §4 du chapitre 4).

sont données dans un système d'unités prédéfini. Ici, l'indicateur sémantique "unités" permet l'identification de HERTZ comme unité de mesure.

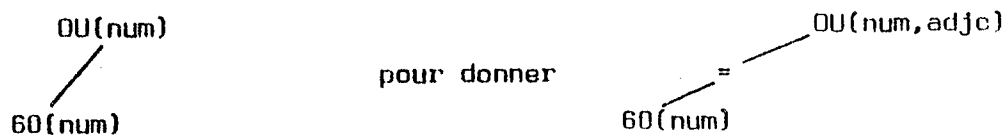
La règle R8 :



est appliquée deux fois de suite, d'abord à

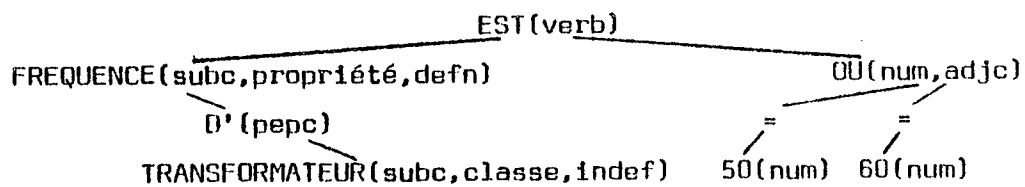


puis à

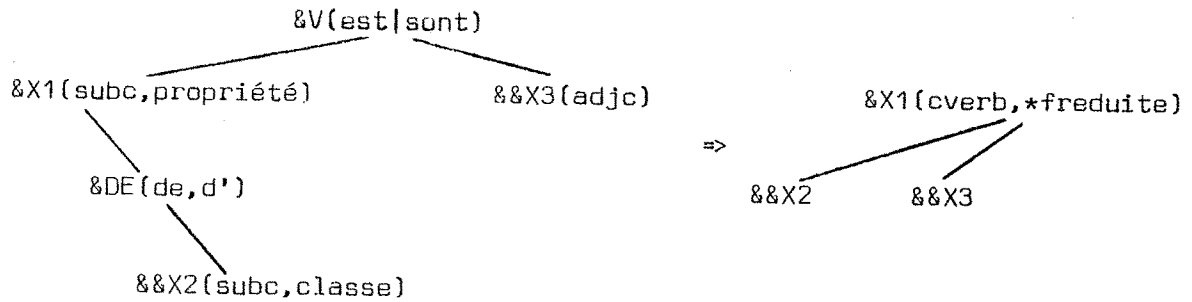


Ceci correspond à supposer que l'intention de la phrase est d'exprimer une relation d'égalité (on suppose que "est 50 ou 60" veut dire "est égal à 50 ou à 60").

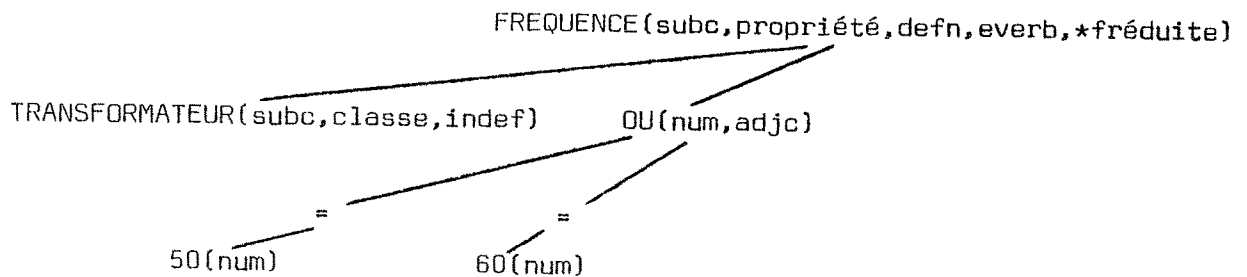
A ce moment, les transformations nous ont conduits à l'arborescence



La règle R22 :



est alors appliquée pour produire



La règle R22 permet d'identifier une des structures arborescentes qui correspondent à une description de propriété d'objet. Les indicateurs sémantiques "propriété" et "classe" contribuent à spécifier la structure et à valider le rôle des éléments dans la forme réduite. La valeur de la propriété est donnée par une sous-arborescence marquée "adjc" (adjectif comparatif) car elle est toujours exprimée par une relation (=, ≠, <, ...) avec une constante.

Aucune autre règle n'est alors applicable. Par ailleurs, la racine de l'arborescence est munie de l'indicateur "*fréduite" et tous les sommets ont participé à au moins une transformation. La structure est donc acceptée comme forme réduite et linéarisée pour donner

(FREQUENCE TRANSFORMATEUR (OU (= 50) (= 60)))

Commentaire :

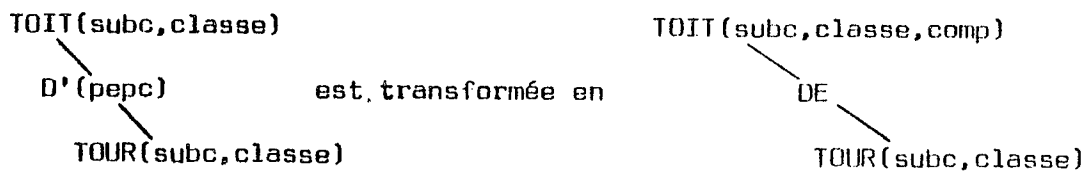
Les règles de transformation utilisent donc les diverses caractéristiques des arborescences de dépendances : relations de dépendances, indicateurs syntaxiques et indicateurs sémantiques. Ainsi, déterminer l'applicabilité d'une règle de transformation correspond à identifier dans la structure syntaxique les candidats à jouer un rôle dans la forme réduite et à vérifier qu'ils satisfont les conditions exigées pour être acceptés comme tels. L'application d'une règle correspond à choisir les éléments porteurs d'information et à les ranger (souvent modifiés par la présence d'autres éléments) dans une structure de forme réduite. Etant donné que chaque transformation est partielle et indépendante, l'ensemble de transformations doit être conçu pour permettre leur enchaînement jusqu'à l'obtention d'une forme réduite ou le refus de la structure considérée.

Suivant ce procédé, il est possible de réduire des paraphrases à la même forme réduite. Par exemple, la phrase "Un transformateur a une fréquence de 50 ou 60 hertz" est réduite à la même forme que la phrase traitée ci-dessus à l'aide des règles de l'appendice A.

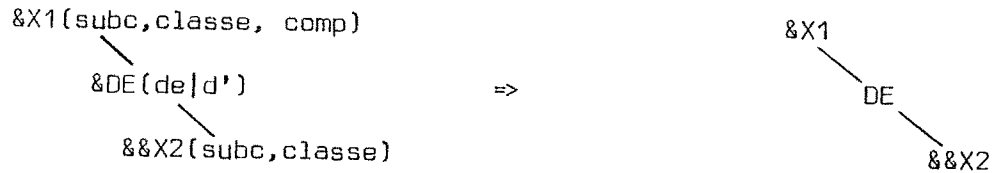
Exemple 2 :

A la phrase "le toit d'une tour est situé sur la colonne de la tour" correspond l'arborescence de dépendances de la figure 3.2. (cf. § 2.2.).

La sous-arborescence

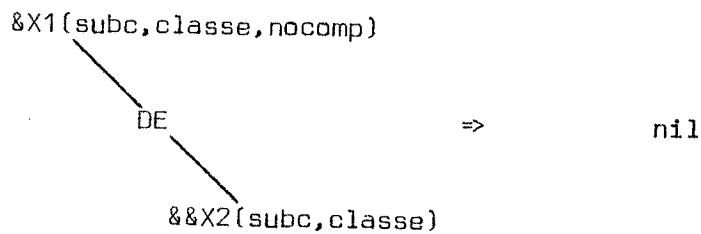


par application de la règle R45 :

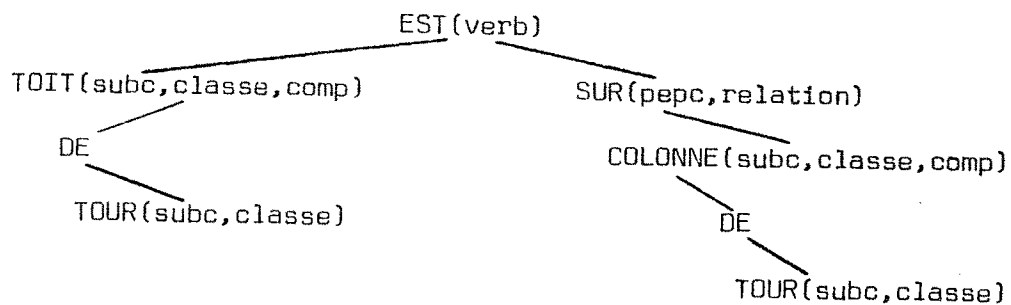


ESEC : (SI (COMPOSANT &X1 &&X2) ALORS (IS(&X1) = comp) SINON
(IS (&X1) = nocomp))

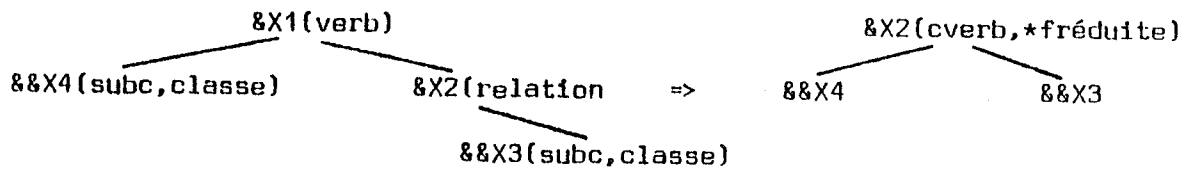
Cette règle a pour but de vérifier que &X1 est un des composants de &&X2. Le prédicat COMPOSANT(TOIT, TOUR) est alors évalué dans le RSP courant. S'il existe une relation de composition (cf. § 2 du chapitre 2) entre TOUR et TOIT dans le RSP, l'application de la règle aura l'effet secondaire de donner à TOIT l'indicateur sémantique (IS) "comp" (ce qui est le cas ici). Dans le cas contraire, TOIT recevra l'indicateur "nocomp", ce qui permettra l'application de la règle d'arrêt



Dans le premier cas, la structure deviendra



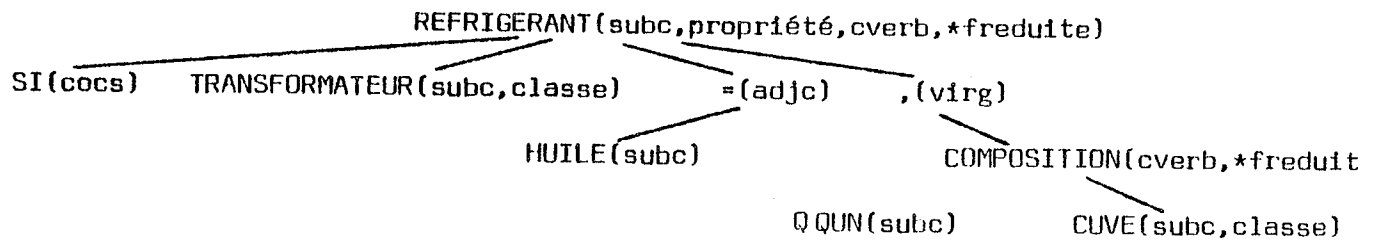
pour permettre l'application de la règle R61 :



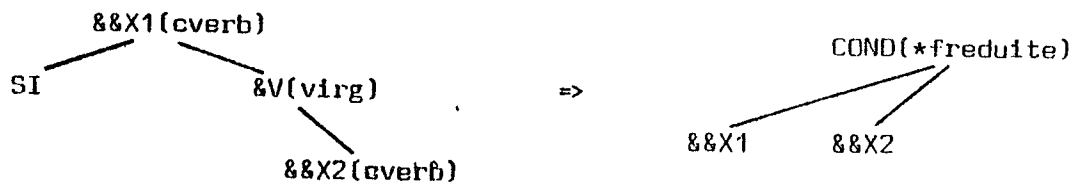
et produire la forme réduite de relation entre objets
(SUR (TOIT (DE TOUR)) (COLONNE (DE TOUR)))

Exemple 3 :

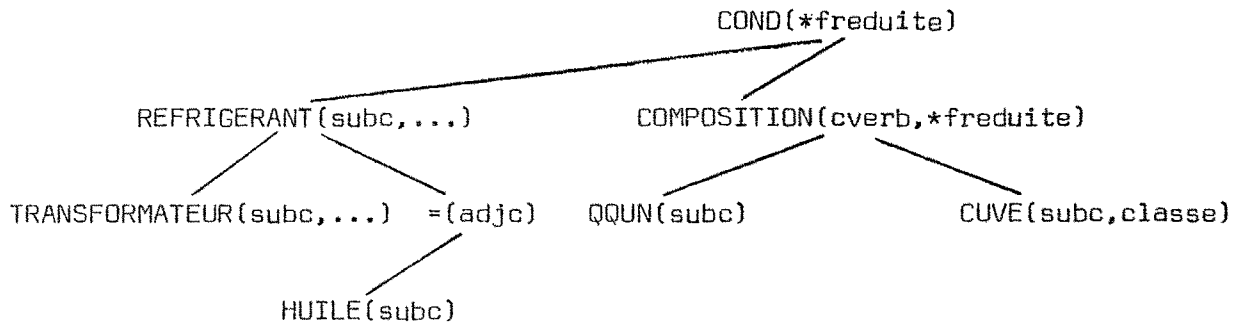
Pour la contrainte "si le réfrigérant d'un transformateur est l'huile, il est composé d'une cuve", PIAF produit quatre arborescences de dépendances. Trois d'entre elles sont rejetées grâce aux règles d'arrêt REJ1 et REJ3. Pour la quatrième, montrée par la figure 3.3., les parties "si" et "alors" sont réduites par des transformations analogues à celles montrées ci-dessus pour obtenir



La règle R81 :



est alors appliquée pour produire



ce qui correspond à la forme réduite

```

(COND (REFRIGERANT TRANSFORMATEUR (= HUILE))
 (COMPOSITION QQUN CUVE))
  
```

où QQUN remplace le pronom personnel "il" apparaissant dans la phrase (...il est composé...). Ceci constitue une référence anaphorique locale résolue en remplaçant QQUN par l' <objet primaire> de la partie <si> de la forme (cf. §4 du chapitre 4). Ici, il s'agit de TRANSFORMATEUR et on obtient finalement

```

(COND (REFRIGERANT TRANSFORMATEUR (= HUILE))
 (COMPOSITION TRANSFORMATEUR CUVE))
  
```

Chapitre IV

CONSTRUCTION DU RESEAU SEMANTIQUE PARTITIONNE

1 - CONSTRUCTION DU RESEAU CORRESPONDANT A UNE PHRASE -----	89
2 - DETERMINATION DES NOUVELLES INFORMATIONS APORTEES PAR UNE PHRASE-	92
2.1. Comparaison des figures de valeurs possibles de propriétés ---	95
2.2. Comparaison des figures de relation n-aire entre propriétés --	96
2.3. Comparaison des figures de relation binaire entre propriétés--	98
2.4. Comparaison des figures de relation entre objets -----	100
2.5. Comparaison des contraintes et des conseils -----	100
2.6. Exemple 1 -----	102
2.7. Exemple 2 -----	102
3 - INSERTION DES NOUVELLES INFORMATIONS DANS LE RSP -----	104
3.1. Cas des figures de description des propriétés -----	105
3.2. Cas des figures de relation entre objets -----	106
3.3. Cas des contraintes et des conseils -----	106
4 - NORMALISATION DES FORMES REDUITES -----	108

Dès que la forme réduite d'une phrase est élaborée, les informations nouvelles qu'elle transmet sont ajoutées au RSP courant. Le traitement correspondant comporte trois phases :

- a) Génération d'un réseau à partir de la forme réduite.
- b) Comparaison de ce réseau avec le RSP courant pour identifier les nouvelles informations.
- c) Insertion des nouvelles informations dans le RSP avec une vérification partielle de la cohérence.

Ces trois phases sont décrites aux paragraphes 1, 2 et 3 respectivement. Un autre aspect du traitement appelé normalisation du RSP, passé sous silence jusque là, sera décrit au § 4.

Initialement le RSP est vide, i.e. il ne contient que l'espace universel U.

1 - CONSTRUCTION DU RESEAU CORRESPONDANT A UNE PHRASE

La forme réduite d'une phrase est interprétée pour construire un sous-réseau du RSP. Cette interprétation repose sur une correspondance assez directe entre les structures des formes réduites (cf. § 2.1. du chapitre 3) et les figures utilisées pour représenter les connaissances dans le RSP (cf. chapitre 2).

Le sous-réseau est construit dans un espace particulier, noté CP, que nous appelons le contexte de la phrase.

Un programme écrit à partir de la définition syntaxique des formes réduites (donnée sous forme BNF au § 2.1. du chapitre 3) identifie et interprète les différents éléments de chaque forme réduite pour construire le sous-réseau correspondant. La description de ce programme, qu'il serait fastidieux d'inclure ici, est donnée dans l'appendice D. Nous l'illustrons ci-après par un exemple simple.

Soit la forme réduite :

(FREQUENCE TRANSFORMATEUR (OU (= 50) (= 60))).

Il lui correspond le sous-réseau de la figure 4.1., où la partie en trait épais est une **figure** de propriété d'objet. Ce réseau a été construit en interprétant la forme réduite selon le modèle :

<fréduite de propriété> ::= (<propriété><objet><description de propriété>)

L'interprétation suivant ce modèle a comporté plusieurs pas :

a) L'identification <objet> \equiv <classe> \equiv TRANSFORMATEUR conduit à localiser dans le RSP le représentant générique de la classe TRANSFORMATEUR (sommet \bigcirc TR dans la figure). Si la classe TRANSFORMATEUR n'existe pas dans le RSP, elle est créée ainsi que son représentant générique et l'espace associé. Nous utilisons le sommet \bigcirc qui joue le rôle de représentant générique de la classe TRANSFORMATEUR pour construire le sous-réseau de la phrase.

b) L'identification <propriété> \equiv FREQUENCE conduit à créer dans CP un sommet \square étiqueté FREQUENCE et à relier TR par un arc P dans CP. Ainsi, nous disposons de la propriété FREQUENCE du transformateur TR.

c) L'identification <description de propriété> \equiv (OU (= 50)(= 60)) conduit à établir dans CP deux relations = entre le sommet \square et les constantes 50 et 60 (Rappel : les sommets représentant des constantes se trouvent toujours dans l'espace U).

Remarque :

Pendant la construction du sous-réseau, si un sommet ou un arc existe déjà dans le RSP, il n'est pas reproduit mais simplement inclus aussi dans l'espace CP

Dans le cas des contraintes et des conseils, les parties <si> et <alors> des formes réduites sont représentées dans des espaces différents contenus dans le contexte de la phrase. Par exemple, la forme :

(COND (JOINTS CIRCUIT MAGNETIQUE (= IMBRIQUES A 45 DEGRES))
(QUALITE DES TOLES CIRCUIT MAGNETIQUE (= Q3)))

est représentée par le réseau de la figure 4.2.

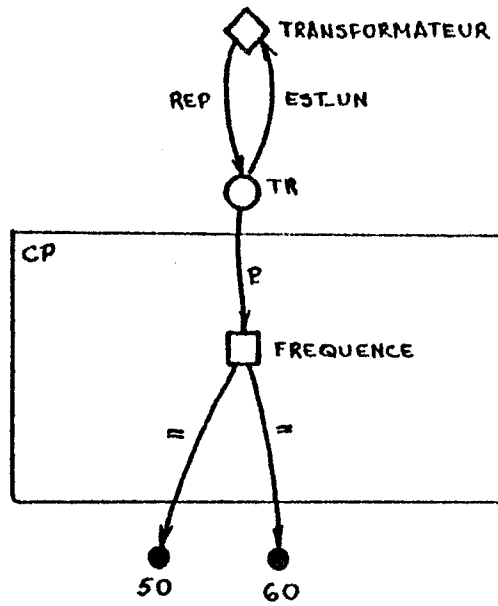


Figure 4.1.

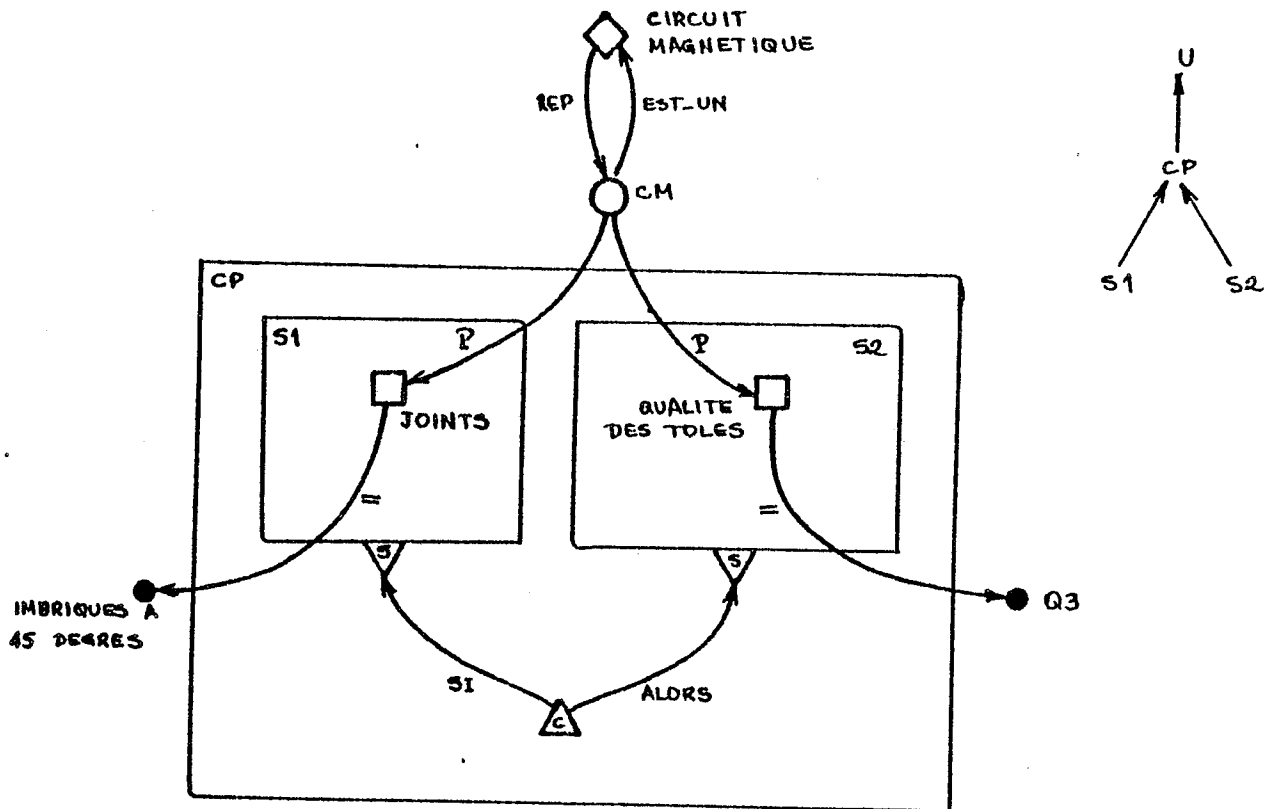


Figure 4.2.

D'une manière générale, les programmes qui interprètent les formes réduites (cf. appendice D) engendrent un ensemble de **figures** partageant éventuellement des éléments. Le sous-réseau créé a donc la structure générale schématisée par la figure 4.3.

En même temps que la forme réduite est interprétée, on prépare la phase suivante du traitement en déterminant le sous-ensemble du RSP susceptible d'être modifié par les informations apportées. Nous le délimitons par le contexte de l'objet le plus englobant mentionné dans la forme ou, à défaut, par le contexte du plus proche ancêtre commun des objets mentionnés. Dans la suite, nous appelons ce contexte le contexte de modification et nous le notons CM.

Dans les exemples précédents les contextes de modification sont donc celui de TRANSFORMATEUR et celui de CIRCUIT MAGNETIQUE respectivement car ils sont les seuls objets mentionnés dans chaque cas. Pour une forme réduite telle que :

(SUR (TOIT (DE TOUR)))(COLONNE (DE TOUR)),

représentée par le sous-réseau de la figure 4.4., le contexte de modification est déterminé en utilisant la hiérarchie de composition. Celle-ci montre que TOUR est l'assemblage le plus englobant mentionné. Le contexte de modification est donc le contexte de TOUR.

2 - DETERMINATION DES NOUVELLES INFORMATIONS APPORTEES PAR UNE PHRASE

Le sous-réseau construit à partir d'une forme réduite est comparé au reste du RSP courant afin de déterminer quelles sont les informations nouvelles apportées par la phrase traitée.

Cette comparaison est effectuée entre le sous-réseau RP contenu dans le contexte de la phrase⁽¹⁾ CP et le sous-réseau RM contenu dans le contexte de modification⁽¹⁾ CM.

 (1) Les sommets de U adjacents au sous-réseau participent aussi à la comparaison.

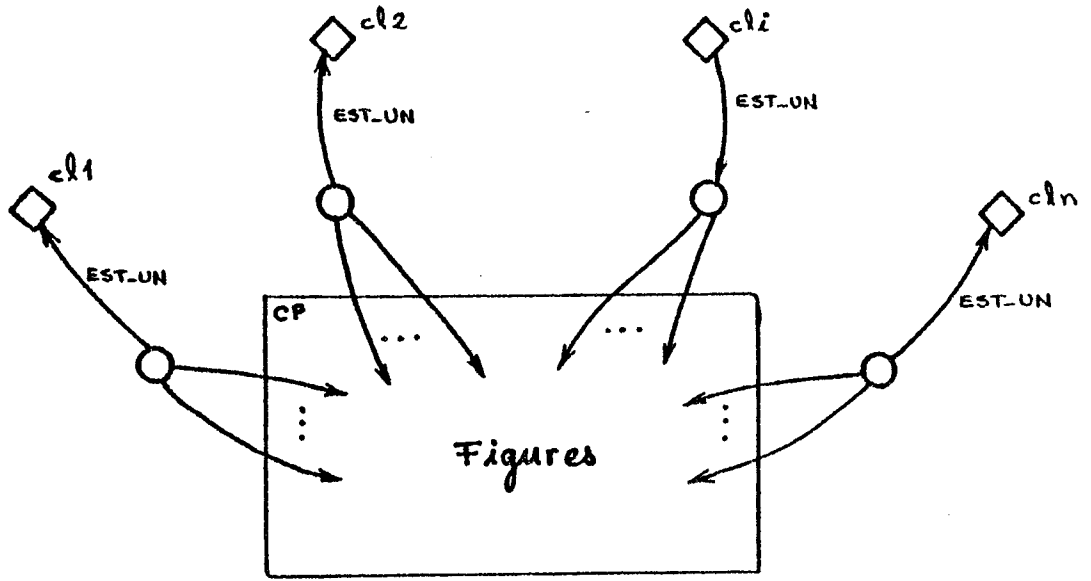


Figure 4.3.

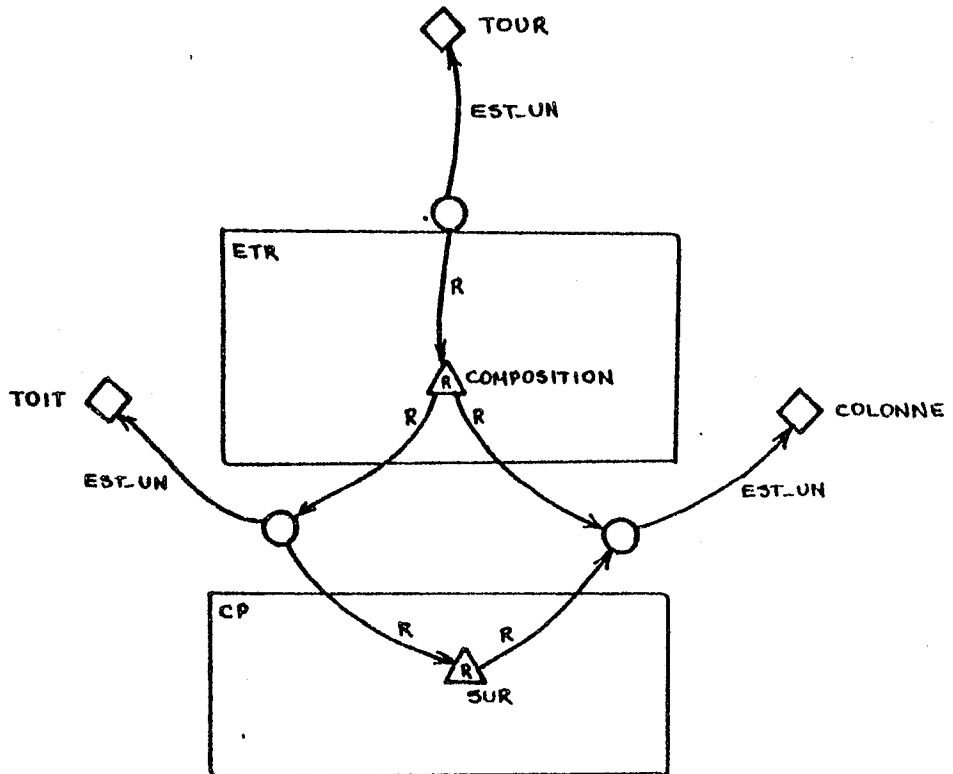


Figure 4.4.

Le but est d'établir une correspondance ("pattern-matching") entre les éléments de RP et ceux de RM. Plus précisément, cette correspondance

- (a) identifie des éléments de RP qui existent déjà dans RM,
- (b) identifie des éléments de RP qui n'existent pas dans RM,
- (c) identifie des éléments de RM susceptibles d'être affectés (problème de cohérence) par la présence des nouveaux éléments contenus dans RP.

L'identification de ces trois ensembles repose sur la coïncidence entre les sommets et les arcs des deux sous-réseaux RP et RM.

La coïncidence est définie comme suit :

- Deux sommets \bigcirc coïncident s'ils sont liés par des arcs EST_UN au même sommet \diamond
- Deux sommets \textcircled{G} coïncident s'ils sont liés par des arcs GROUPE_DE au même sommet \diamond
- Deux sommets \triangle_R coïncident s'ils portent la même étiquette.
- Deux sommets \square coïncident s'ils portent la même étiquette et si leurs arcs P incidents ont le même sommet \bigcirc pour extrémité initiale.
- Deux sommets \triangle_F coïncident si leurs propriétés EXPR sont identiques et si leurs arcs 1,2,...,n pointent sur des sommets \square qui coïncident.
- Deux sommets \bullet coïncident s'ils portent la même étiquette.
- Deux arcs coïncident si leurs étiquettes sont identiques, leurs extrémités initiales coïncident et leurs extrémités finales coïncident.
- Aucune autre coïncidence n'est possible.

Remarque :

Dans de nombreux cas, déterminer la coïncidence de deux sommets ou de deux arcs consiste à simplement constater qu'il s'agit du même sommet ou du même arc.

Sur cette base, la correspondance entre les éléments de deux sous-réseaux est déterminée en comparant successivement chaque figure F1 de RP avec une figure F2 de RM (définie plus loin). Le résultat de la comparaison comporte en général (exception faite des contraintes et des conseils) trois listes d'arcs ⁽¹⁾

(1) Les listes ne contiennent que des arcs car la coïncidence entre deux arcs implique la coïncidence de leurs extrémités initiales et finales.

- (a) Une liste d'arcs qui coïncident (LCOUPLE),
- (b) une liste d'arcs de F1 qui n'ont pas de coïncident dans F2 (LNONC1),
- (c) une liste d'arcs de F2 qui n'ont pas de coïncident dans F1 (LNONC2).

Dans le cas d'une contrainte ou d'un conseil dont l'ensemble des parties si et des parties alors comprend un total de n figures, le résultat est une liste contenant n fois trois listes LCOUPLE, LNONC1, LNONC2.

Nous avons défini la comparaison de figures séparément pour les figures de valeurs possibles de propriété, de relation n-aire entre propriétés, de relation binaire entre propriétés, de relation entre objets, de contrainte et de conseil. Ci-dessous nous décrivons la comparaison dans chaque cas. Nous l'illustrons ensuite sur deux exemples.

2.1. Comparaison des figures de valeurs possibles de propriété

Soit F1 une figure de valeurs possibles de propriété contenue dans CP. Sa structure générale est celle montrée à la figure 4.5a. Le premier pas de sa comparaison avec RM consiste à chercher dans CM une figure de valeurs possibles de propriété F2 (cf. figure 4.5b) telle que son arc p' coïncide avec l'arc p de F1. S'il n'existe pas une telle figure, le résultat de la comparaison est LCOUPLE=NIL, LNONC1=(p,r1,?,rn), LNONC2=NIL. Si une telle figure existe, le résultat de la comparaison est donné par :

Début

LCOUPLE := (p') ;

LNONC1 := (r1,r2,...,rn) ;

LNONC2 := (r1',r2',...,rm') ;

pour chaque REL1 dans LNONC1 faire

pour chaque REL2 dans LNONC2 faire

si REL1 coïncide avec REL2 alors

. on élimine REL1 de LNONC1 ;

. on élimine REL2 de LNONC2 ;

. REL2 est inclus dans LCOUPLE ;

fsi ;

fin.

2.2. Comparaison des figures de relation n-aire entre propriétés

Soit F1 une figure de relation n-aire entre propriétés contenue dans CP. Sa structure générale est celle montrée à la figure 4.6a. Le premier pas de la comparaison avec RM consiste à chercher dans CM une figure F2 de relation n-aire entre propriétés (cf. figure 4.6b) telle que son arc p' coïncide avec l'arc p de F1. S'il n'existe pas une telle figure, le résultat de la comparaison est LCOUPLE=NIL, LNONC1=(p,r), LNONC2=NIL. Si une telle figure existe, le résultat de la comparaison est donné par :

Début

LCOUPLE := (p') ;

si r coïncide avec r' alors

. on inclut r' dans LCOUPLE ;

. LNONC1 := NIL ;

. LNONC2 := NIL ;

sinon

LNONC1 := (r) ;

si f coïncide avec f' alors LNONC2 := (r')

sinon LNONC2 := NIL ;

fsi ;

fsi ;

fin.

Si plusieurs figures peuvent être comparées à F1, les comparaisons sont effectuées pour chacune d'elles selon l'algorithme ci-dessus jusqu'à l'obtention

- soit d'un résultat qui comporte la liste LNONC1 vide : c'est le cas où la relation existait déjà dans CM,
- soit d'un résultat avec la liste LNONC2 non-vide : c'est le cas où la fonction est la même, mais sa relation avec la propriété est différente.

Si aucun de ces deux résultats n'est obtenu, le résultat final considéré est LCOUPLE=(p'), LNONC1=(r), LNONC2=NIL.

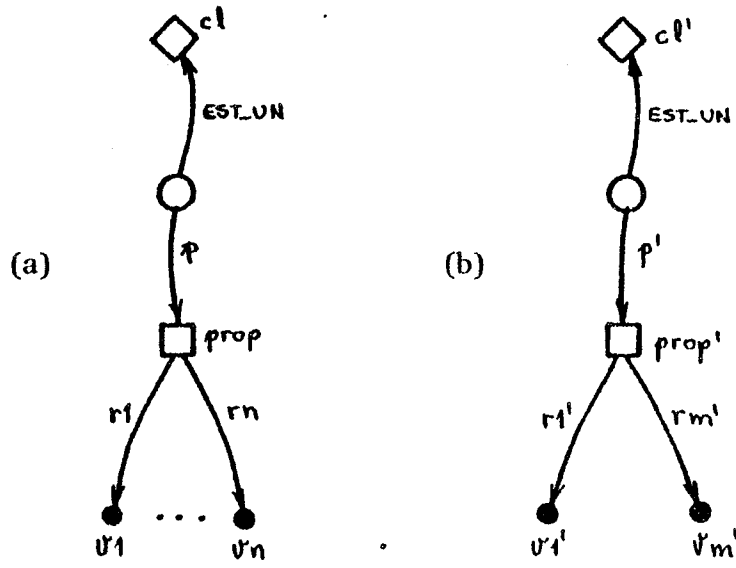


Figure 4.5.

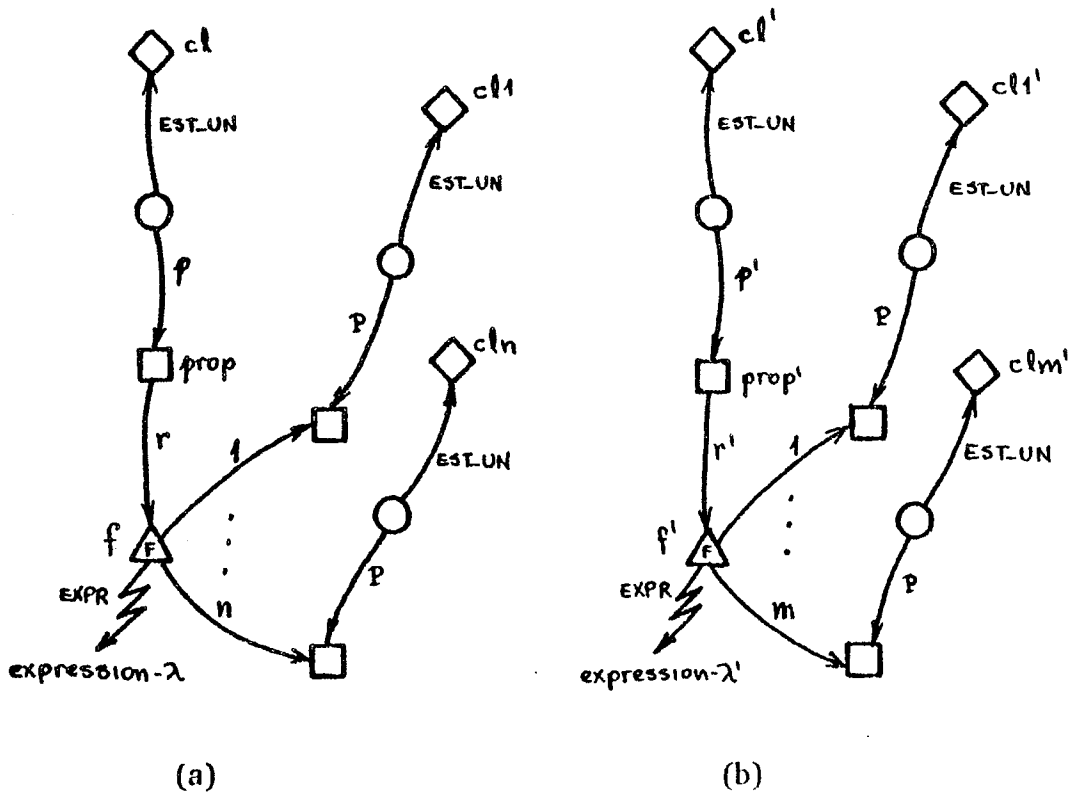


Figure 4.6.

2.3. Comparaison des figures de relation binaire entre propriétés

Soit F1 une figure de relation binaire entre propriétés contenue dans CP. Sa structure générale est celle montrée à la figure 4.7a. Le premier pas de sa comparaison avec RM consiste à chercher dans CM une figure F2 de relation binaire entre propriétés (cf. figure 4.7b) telle que son arc p' coïncide avec l'arc p de F1. S'il n'existe pas une telle figure, le résultat de la comparaison est LCOUPLE=NIL, LNONC1=(p,r), LNONC2=NIL. Si une telle figure existe, le résultat de la comparaison est donné par :

Début

LCOUPLE:=(p');

si r coïncide avec r' alors

. on inclut r' dans LCOUPLE;

. LNONC1:=NIL;

. LNONC2:=NIL;

sinon

LNONC1:=(r);

si prop2 coïncide avec prop2' alors LNONC2:=(r')

sinon LNONC2:=NIL;

fsi;

fin

Si plusieurs figures peuvent être comparées à F1, les comparaisons sont effectuées pour chacune d'elles selon l'algorithme ci-dessus jusqu'à l'obtention :

- soit d'un résultat qui comporte la liste LNONC1 vide : c'est le cas où la relation existait déjà dans CM,
- soit d'un résultat avec la liste LNONC2 non-vide : c'est le cas où il existe déjà une relation différente avec la même propriété.

Si aucun de ces deux résultats n'est obtenu, le résultat final considéré est LCOUPLE=(p'), LNONC1=(r), LNONC2=NIL.

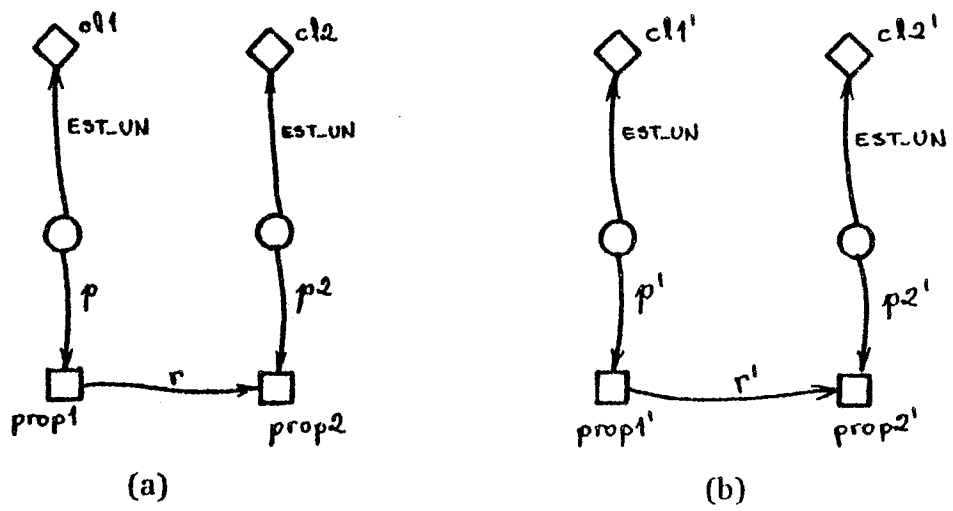


Figure 4.7.

2.4. Comparaison des figures de relation entre objets

Soit F1 une figure de relation entre objets contenue dans CP. Sa structure générale est celle montrée à la figure 4.8a. Le premier pas de sa comparaison avec RM consiste à chercher dans CM une figure F2 de relation entre objets (cf. figure 4.8b) telle que son arc r' coïncide avec l'arc r de F1. S'il n'existe pas une telle figure, le résultat de la comparaison est (LCOUPLE=NIL, LNONC1=(r,r1,...,rn), LNONC2=NIL. Si une telle figure existe, le résultat de la comparaison est donné par :

Début

```
LCOUPLE:=(p');
LNONC1:=(r1,...,rn);
LNONC2:=(r1',...,rm') ;
  pour chaque rel1 dans LNONC1 faire
    pour chaque rel2 dans LNONC2 faire
      si rel1 coïncide avec rel2 alors
        . on inclut rel2 dans LCOUPLE ;
        . on élimine rel1 de LNONC1 ;
        . on élimine rel2 de LNONC2 ;
      fsi ;
```

fin.

2.5. Comparaison des contraintes et des conseils

Dans le cas d'une contrainte ou d'un conseil ayant n figures dans l'ensemble de ses parties <si> et m figures dans l'ensemble de ses parties <alors> , sa comparaison avec RM produit une liste contenant (m+n) fois les listes LCOUPLE, LNONC1 et LNONC2. L'algorithme suivi est le suivant :

Début

```
N:=0 ;
M:=0 ;
L:=NIL ;
```

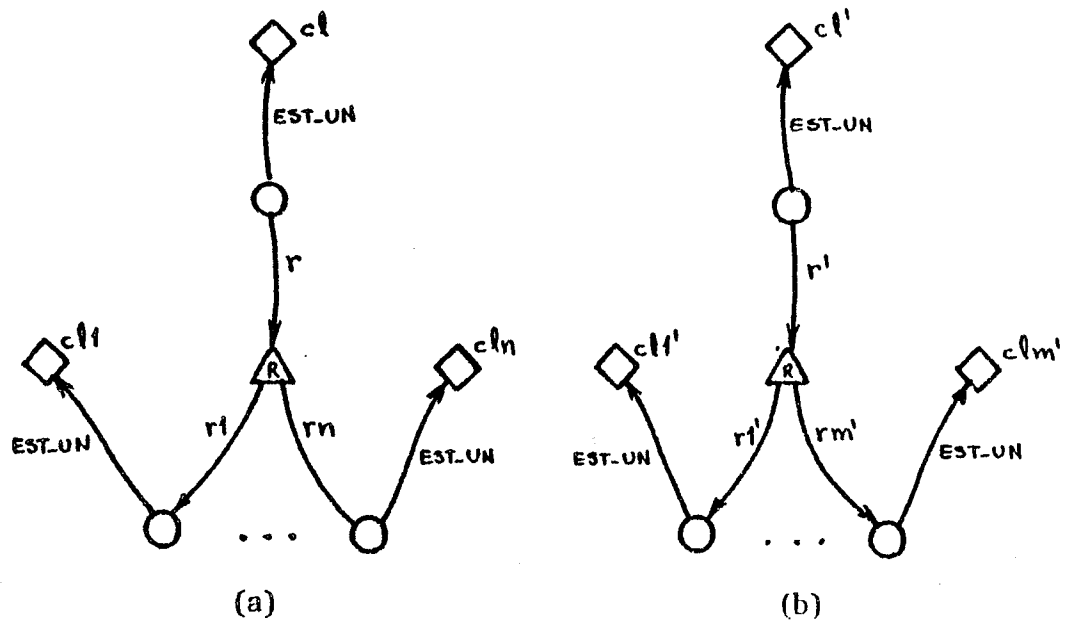


Figure 4.8.

Pour chaque partie <si> faire

Pour chaque figure F dans la partie <si> faire

- . On inclut dans L le résultat de la comparaison de F avec RM (cf. § 2.1. à 2.4.)
- . N:=N+1 ;

Pour chaque partie <alors> faire

Pour chaque figure F dans la partie <alors> faire

- . On inclut dans L le résultat de la comparaison de F avec RM (cf. § 2.1. à 2.4.).
- . M:=M+1 ;

fin.

2.6. Exemple 1

La comparaison du réseau de la figure 4.1. avec la partie de RSP montrée à la figure 4.9 donne les résultats suivants :

- LCOUPLE=(P1) ;
- LNONC1 = ((=,FREQUENCE,50) , (=,FREQUENCE,60));
- LNONC2 = ((≥,FREQUENCE,50)).

2.7. Exemple 2

La comparaison des sous-réseaux des figures 4.2 et 4.10 donne les résultats suivants :

(a) Pour la partie <si> :

- LCOUPLE=NIL
- LNONC1=((P,CM,JOINTS),(=,JOINTS,IMBRIQUES A 45 DEGRES))
- LNONC2=NIL.

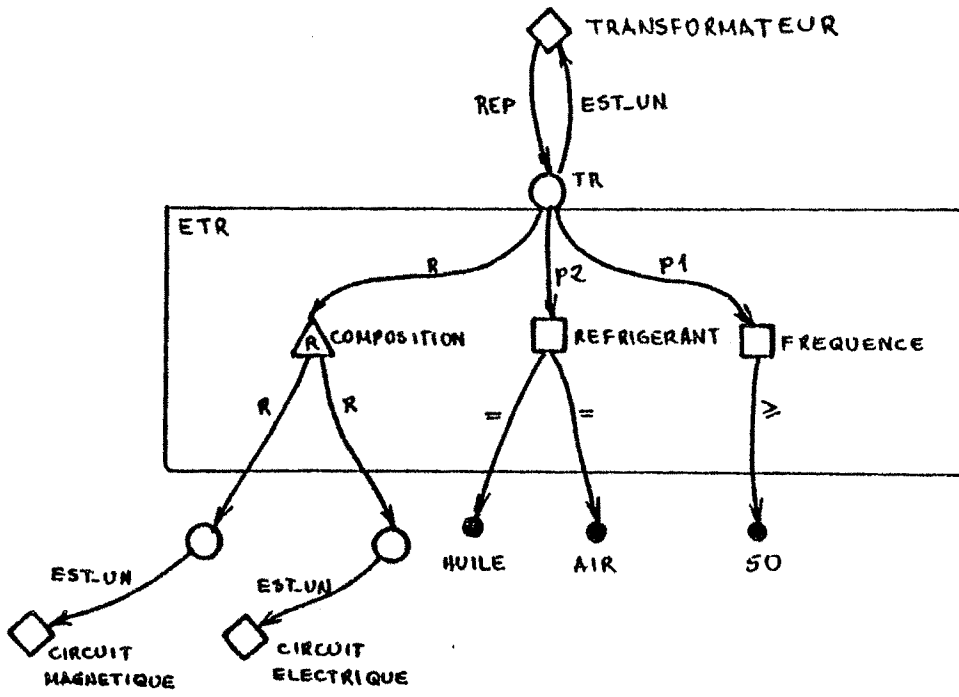


Figure 4.9.

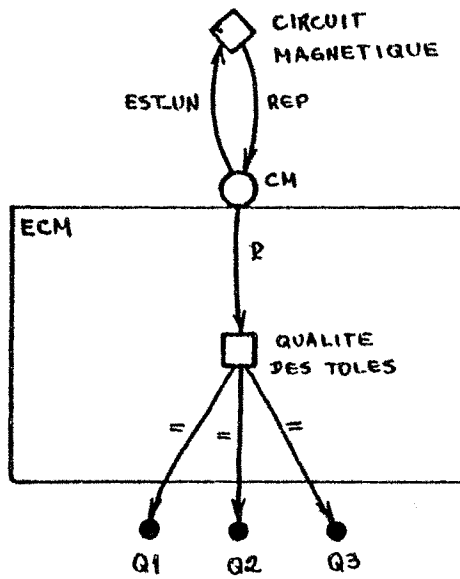


Figure 4.10.

(b) Pour la partie <alors> :

- LCOUPLE=((P,CM,QUALITE DES TOLES),(=,QUALITE DES TOLES,Q3)

- LNONC1=NIL

- LNONC2=((=,QUALITE DES TOLES,Q1),(=,QUALITE DES TOLES,Q2)).

3 - INSERTION DES NOUVELLES INFORMATIONS DANS LE RSP

Cette phase consiste à ajouter au RSP courant les sommets, arcs et espaces nécessaires pour représenter les nouvelles informations apportées par une phrase. Souvent, les sommets et les arcs sont insérés par simple transfert du contexte de la phrase (CP) au contexte de modification (CM). En parallèle, la cohérence des nouvelles informations avec les anciennes est partiellement vérifiée.

Reprenons l'exemple du paragraphe 2.6. Le résultat de la comparaison des réseaux des figures 4.1. et 4.9. est utilisé ainsi :

- La présence de P1 dans LCOUPLE indique que la propriété FREQUENCE de TR existe déjà dans le RSP,
- Les arcs (=,FREQUENCE,50) et (=,FREQUENCE,60) de LNONC1 constituent des nouvelles informations susceptibles d'être ajoutées au RSP.
- On vérifie que ces arcs sont compatibles avec l'arc (>,FREQUENCE,50) contenu dans LNONC2. On constate qu'ils sont plus restrictifs.
- L'arc (>,FREQUENCE,50) est remplacé par les arcs (=,FREQUENCE,50) et (=,FREQUENCE,60) qui sont transférés de CP à CM.
- Le contenu de CP est effacé.

Le remplacement précédent a pour but de maintenir la cohérence et de réduire la redondance dans la base de connaissances. Si on détecte une contradiction entre les nouvelles informations et les anciennes, le système interroge l'utilisateur qui doit alors choisir les informations à conserver.

N'ayant pas de capacités de déduction, le système réalise des vérifications très simples et localisées. En particulier, il n'est pas capable de déterminer si deux contraintes sont contradictoires ou si les valeurs possibles de deux propriétés deviennent incohérentes à cause d'une relation établie entre les propriétés. D'une certaine manière, ces vérifications sont à la charge de TROPIC.

Nous décrivons l'insertion de nouvelles informations plus complètement, ci-dessous.

3.1. Cas des figures de description de propriétés (valeurs possibles de propriété, relations n-aires et binaires entre propriétés)

Le résultat de la comparaison entre RP et RM est exploité comme suit :

- si LNONC1 et LNONC2 sont vides, aucune action n'est exécutée,
- si LCOUPLE et LNONC2 sont vides, tous les arcs de LNONC1 sont transférés de CP à CM⁽¹⁾,
- dans tous les autres cas, les arcs P de LCOUPLE sont utilisés pour déterminer où connecter dans le RSP les nouveaux éléments contenus dans LNONC1. On vérifie alors que les arcs de LNONC1 sont compatibles avec ceux de LCOUPLE et de LNONC2 :
- . si les premiers sont plus restrictifs que les seconds, ces derniers sont remplacés par les premiers,
- . si les arcs de LNONC1 sont moins restrictifs, ou s'ils sont contradictoires avec les arcs de LCOUPLE ou de LNONC2, l'utilisateur est interrogé pour décider quelles sont les informations à conserver.

Remarques

- Nous ne vérifions pas l'équivalence des relations entre propriétés; ceci peut donc donner lieu à des redondances dans le RSP.
- Les critères pour décider si des arcs sont plus restrictifs ou contradictoires par rapport à d'autres arcs ont été définis de façon arbitraire. Ils sont basés sur les relations d'ordre entre les entiers et entre les réels, et sur la supposition que la relation = est plus restrictive que les autres.

(1) Le transfert des arcs implique le transfert des sommets adjacents sauf pour les sommets ○ et ● qui se trouvent dans U. Par ailleurs, les sommets et les arcs déjà existants dans CM ne sont pas reproduits.

3.2. Cas des figures de relation entre objets

Le résultat de la comparaison entre RP et RM est exploité comme suit :

- si LNONC1 et LNONC2 sont vides, aucune action n'est exécutée,
- si LCOUPLE et LNONC2 sont vides, tous les arcs de LNONC1 sont transférés de CP à CM;
- dans tous les autres cas, les arcs (R, sommet \bigcirc , sommet \triangle) de LCOUPLE sont utilisés pour déterminer où connecter dans le RSP les nouveaux éléments contenus dans LNONC1 :
 - si LNONC2 est vide, les arcs de LNONC1 sont transférés à CM;
 - si LNONC2 n'est pas vide, ses arcs sont éliminés de CM et les arcs de LNONC1 insérés à leur place. Ceci veut dire que nous gardons dans le RSP seulement les informations apportées par la dernière phrase, choix qui résulte de ce que nous ne considérons pas la possibilité de relations entre objets alternatives.

3.3. Cas des contraintes et des conseils

Le traitement des contraintes et des conseils est légèrement plus élaboré. A titre d'exemple, considérons le résultat de la comparaison des sous-réseaux des figures 4.2. et 4.10 donné au § 2.7. Le réseau de la figure 4.11. est construit à partir de ce résultat de la façon suivante :

(a) Construction de la partie <si> :

- Le sommet-espace S-S1 est créé dans ECMAG (le contexte de modification) avec S1 \triangleleft (S1,ECMAG,U).
- Comme LCOUPLE et LNONC2 sont vides, toutes les informations de la partie <si> sont nouvelles. Ainsi, le sommet JOINTS et l'arc (P,CM,JOINTS) sont transférés de CP à ECMAG, et l'arc (=,JOINTS,IMBRIQUES A 45 DEGRES) est transféré de CP à ECMAG et S1 simultanément⁽¹⁾.

(b) Construction de la partie <alors> :

- Le sommet-espace S-S2 est créé dans ECMAG avec S2 \triangleleft (S2,ECMAG,U).
- Comme LNONC1 est vide, il n'y a pas de nouvel élément à ajouter et donc pas de possibilité de contradiction avec les éléments de LNONC2.
- D'après LCOUPLE, l'arc (=,QUALITE DES TOLES,Q3) existe déjà dans ECMAG. En conséquence, il est simplement inclus dans S2 (ce qui ne l'élimine pas de ECMAG)

 (1) Le transfert à ECMAG montre qu'une contrainte (ou un conseil) peut introduire une propriété et une ou plusieurs de ses valeurs possibles. De façon analogue, on peut introduire des relations entre objets

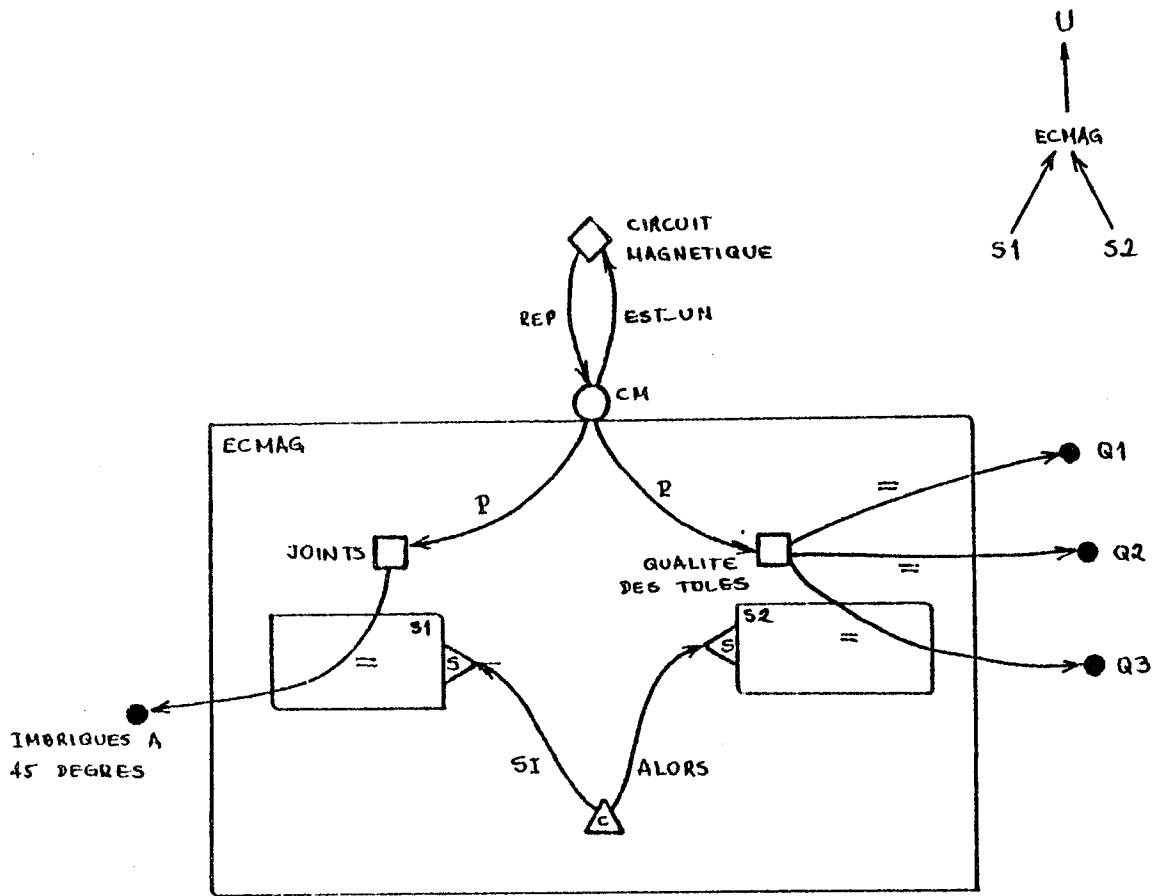


Figure 4.11.

(c) Construction de la contrainte :

- Le sommet \triangle et les arcs (SI,C,S-S1) et (ALORS,C,S-S2) sont créés dans ECMAG (le contexte de modification).

Note :

Les programmes que nous avons écrits ne sont pas capables de déterminer si deux contraintes ou deux conseils sont identiques.

4 - NORMALISATION DES FORMES REDUITES

Avant d'être interprétés (cf. §1), les formes réduites sont "normalisées". Cette normalisation facilite leur traitement ultérieur et contribue à assurer l'uniformité de la base de connaissances.

Nous effectuons quatre types de normalisation.

1/ Normalisation des relations entre objets

Les relations entre objets admettent généralement une correspondante inverse. Par exemple, l'inverse de la relation SUR est la relation SOUS, l'inverse de la relation A DROITE est la relation A GAUCHE etc... On peut alors exprimer une même situation différemment en utilisant une relation ou son inverse : ainsi, "le toit d'une tour est situé sur la colonne de la tour" est équivalent à "la colonne d'une tour est située sous le toit de la tour". Dans le RSP seulement une des relations, appelée la relation normale, est utilisés.

Nous disposons pour cela d'un dictionnaire dont chaque entrée a la forme /relation inverse/relation normale/, par exemple /sous/sur/. Ce dictionnaire permet de normaliser une forme réduite de relation entre objets comme

(relation inverse objet1 objet2),

ex.: (SOUS (COLONNE (DE TOUR)) (TOIT (DE TOUR)))

en

(relation normale objet2 objet1),

ex.: (SUR (TOIT (DE TOUR)) (COLONNE (DE TOUR))).

Evidemment, si une relation entre objets est exprimée par une relation normale, elle n'est pas transformée.

2/ Normalisation des situations dans les contraintes et les conseils

Lorsqu'une <situation> d'une partie <si> ou d'une partie <alors> (cf. §2 du chapitre 3) est donnée par :

(<propriété><objet1> (ou <relation primitive>₁...<relation primitive>_n)

elle est normalisée en :

(ou (<propriété><objet1><relation primitive>_n).

⋮

(<propriété><objet1><relation primitive>_n).

Par exemple,

(COND (FORME TOIT (OU (= PYRAMIDALE) (= CONIQUE)))
(HAUTEUR TOIT (> 3)))

est normalisée en

(COND (OU (FORME TOIT (= PYRAMIDALE))
(FORME TOIT (= CONIQUE)))
(HAUTEUR TOIT (> 3))).

3/ Normalisation des références anaphoriques

Les phrases "si un toit est pyramidal, alors il est vert" et "si un toit est pyramidal, sa couleur est verte" conduisent à la forme réduite :

(COND (FORME TOIT PYRAMIDAL)
(COULEUR QQUN VERT)),

laquelle est normalisée en

(COND (FORME TOIT PYRAMIDAL)
(COULEUR TOIT VERT))

Dans le cas général, on remplace l'élément QQUN par le premier objet trouvé dans la forme. L'utilisateur est informé du remplacement pour qu'il

puisse éventuellement le refuser.

4/ Normalisation de la référence à un objet particulier

La phrase "si le nombre de blocs d'une colonne est supérieur à 5, les blocs doivent avoir une hauteur inférieure à 1 mètre" conduit à la forme réduite :

```
(COND (CARD (BLOC (DE COLONNE)) (> 5))
      (HAUTEUR BLOC (> 1))).
```

Celle-ci ne contient pas explicitement que la restriction porte sur les blocs d'une colonne. Néanmoins, dans la phrase, l'article défini les indique que "blocs" fait référence à des objets déjà mentionnés et l'analyse sémantique conserve cette information en associant l'indicateur "defn" au mot BLOC (cf. §2.4 du chapitre 3). Cet indicateur est utilisé pour transformer l'élément (HAUTEUR BLOC (> 1)) de la forme réduite en (HAUTEUR (BLOC (DE COLONNE)) (> 1)).

Ainsi, dans les cas suffisamment simples comme celui-ci, la présence de l'indicateur "defn" associé à un objet conduit à ajouter à cet objet le modificateur (DE <objet>) où <objet> est le premier objet mentionné dans la phrase.

CHAPITRE V

COMMUNICATION RSP-TROPIC

1 - LE MODELE DE RESOLUTION DE PROBLEME UTILISE PAR TROPIC-----	113
2 - LES REGLES DE SYNTHESE -----	116
2.1. Définition -----	116
2.2. Utilisation par TROPIC -----	116
2.3. Génération des règles de synthèse à partir du RSP -----	117
2.3.1. Décomposition de la tâche (choisir-propriété-<cl> &x)---	118
2.3.2. Décomposition de la tâche (engendrer-composants-<cl>&x)	121
2.3.3. Cas des groupes -----	122
3 - LES REGLES D'ANALYSE -----	126
3.1. Définition -----	126
3.2. Utilisation par TROPIC -----	126
3.3. Génération des règles d'analyse à partir du RSP -----	127
3.3.1. Restrictions sur les valeurs de propriétés réelles ou entières -----	127
3.3.2. Relations entre propriétés -----	128
3.3.3. Contraintes -----	132
4 - LES CONSEILS -----	135
4.1. Définition et utilisation -----	135
4.2. Génération des conseils à partir du RSP -----	135

Dans ce chapitre nous établissons la correspondance entre les connaissances représentées dans le RSP et les règles de production directement utilisées par TROPIC.

TROPIC utilise quatre types de règles appelées règles de synthèse, règles d'analyse, conseils et diagnostics (Latombe, 1979). Les diagnostics n'étant pas traités par l'interface (cf. chapitre 2), les paragraphes suivants (§2,3,4) décrivent seulement la génération des trois premiers types de règles à partir du RSP. Cette génération est facilitée par le découpage du RSP en figures.

Le modèle de représentation des connaissances à base de RSP utilisé par l'interface est plus général que le modèle à base de règles de production utilisé par TROPIC. En conséquence, dans ce chapitre, nous ne considérons souvent qu'un sous-ensemble des possibilités offertes par le premier modèle.

Auparavant (§ 1), pour permettre une meilleure compréhension du chapitre, nous donnons une description générale du modèle de résolution de problème implanté dans TROPIC. Cette description est extraite de (Latombe, 1979).

Les programmes permettant de traduire le RSP en règles de production acceptables par TROPIC n'ont pas été implantés.

1 - LE MODELE DE RESOLUTION DE PROBLEME UTILISE PAR TROPIC

Traditionnellement, on considère la conception comme l'interaction de deux processus : la synthèse, qui consiste à engendrer un assemblage supposé remplir un cahier de charges donné, et l'analyse, qui consiste à observer le comportement de cet assemblage pour décider s'il est satisfaisant ou non. Le modèle de résolution de problème de TROPIC résulte d'une formalisation de cette considération.

Il est fondé sur l'exécution coordonnée d'un processus de résolution de problème, appelé "synthèse par réduction de tâches", et un processus de déduction, appelé "analyse par déduction de faits" :

1) La **synthèse par réduction de tâches** consiste à faire des hypothèses sur un assemblage potentiel (par exemple, le type de ses composants, les relations entre ces composants et les valeurs de certains paramètres), tout en décomposant une "tâche initiale" donnée en sous-tâches élémentaires. Par exemple, si il s'agit de concevoir un transformateur TR1, un pas du processus peut être de supposer que les composants de TR1 sont un circuit magnétique CM1 et un circuit électrique CE1, et réduire la tâche "concevoir transformateur TR1", "concevoir circuit électrique CE1" et "choisir le réfrigérant de TR1". Chaque pas peut exiger au système de choisir entre plusieurs façons de découper une même tâche, chacune d'elles correspondant à des hypothèses différentes. Par exemple, la tâche "choisir le réfrigérant de TR1" peut être réduite à l'ensemble vide de sous-tâches en supposant que le réfrigérant de TR1 est l'air ; elle peut aussi être réduite à la sous-tâche "concevoir cuve CU1" en faisant l'hypothèse que le réfrigérant de TR1 est l'huile et qu'il est contenu dans une cuve CU1. Comme nous allons le voir, certaines combinaisons de choix peuvent entraîner des contradictions.

2) L'**analyse par déduction de faits** consiste à déduire de nouveaux faits à partir des hypothèses faites et à contrôler la cohérence de l'ensemble des faits supposés (hypothèses) et des faits déduits. Les déductions sont effectuées en appliquant différents types de connaissances telles que des contraintes et des lois physiques. Par exemple, la contrainte "si le réfrigérant d'un transformateur est l'air, alors la tension aux bornes de toutes les galettes de ce transformateur doit être inférieure à 3kv ; sinon elle doit être inférieure à 5 kv" peut être utilisée pour déduire le fait "la tension aux bornes de GAL1 est inférieure à 5 kv" à partir des faits "le réfrigérant de TR1 est l'huile", "GAL1 est un composant de TR1" et "GAL1 est une galette". Des faits sont contradictoires lorsqu'ils ne peuvent pas être vrais simultanément ; par exemple

les faits "le nombre de tours de GAL1 est 350", "la tension par tour dans GAL1 est 18V", "la tension aux bornes de GAL1 est inférieure à 5 kv", et "la tension aux bornes de GAL1 est égale au nombre de tours de GAL1 multiplié par la tension par tour dans GAL1" sont des faits contradictoires. Les contradictions indiquent que certains buts ne sont pas atteints. Elles produisent des échecs qui conduisent le système à modifier certains choix en reprenant la synthèse par réduction de tâches à un point de décision antérieur.

Un problème est spécifié par une tâche initiale, un ensemble de règles de synthèse à utiliser pendant la synthèse par réduction de tâches, et un ensemble de règles d'analyse à utiliser pendant les déductions de faits par analyse. Il est considéré comme résolu lorsque la tâche initiale a été réduite à l'ensemble vide de telle façon qu'il n'y a pas de contradiction entre les faits engendrés par cette réduction. A ce moment, l'ensemble de faits supposés et de faits déduits constitue une solution.

De plus, dans le but de réduire le nombre d'échecs rencontrés, TROPIC accepte des conseils qui permettent à l'utilisateur du système de lui communiquer son expérience et son intuition. Par exemple, le conseil "si la puissance d'un transformateur est supérieure à 100 kva, il est recommandé d'utiliser l'huile pour son réfrigérant" permet au système de déduire la recommandation "le réfrigérant de TR1 est l'huile" à partir des faits "la puissance de TR1 est supérieure à 100 kva" et "TR1 est un transformateur". Les conseils sont considérés par TROPIC comme des contraintes faibles qu'il n'est pas impératif de satisfaire.

L'utilisateur peut également aider le système à traiter les échecs rencontrés en lui fournissant des diagnostics. Ceux-ci conduisent TROPIC à éliminer ou à maintenir certains faits liés à l'échec rencontré, par exemple "si les pertes magnétiques dans un circuit magnétique dépassent largement (par exemple de plus de 30%) le seuil imposé par une contrainte, alors il est recommandé d'améliorer la qualité des tôles du circuit magnétique".

2 - LES REGLES DE SYNTHÈSE

2.1. Définition

Une règle de synthèse est une production

$$t \rightarrow (\{h_1, \dots, h_m\} \{t_1, \dots, t_n\})$$

où t, t_1, \dots, t_n ($n \geq 0$) sont des expressions représentant des tâches et h_1, \dots, h_m ($m \geq 0$) sont des expressions représentant des hypothèses. La partie droite peut aussi être une fonction qui s'évalue à une expression $(\{h_1, \dots, h_m\} \{t_1, \dots, t_n\})$. La production contient en général des variables qui sont instantiées lorsque la règle est appliquée.

Exemple :

(concevoir-transformateur &tr)

→ (I? { (= (type &tr) transformateur) }
 { (choisir-propriétés-transformateur &tr) }
 { engendrer-composants-transformateur &tr } }

2.2. Utilisation par TROPIC

Etant donnée une tâche initiale et un ensemble de règles de synthèse, TROPIC construit une arborescence ET/OU suivant une stratégie de recherche en profondeur. Les sommets-ET de l'arborescence sont étiquetés par des tâches instantiées et les sommets-OU sont étiquetés par des règles de synthèse instantiées. A chaque instant, cette arborescence représente la décomposition de la tâche initiale considérée par le système.

Initialement, l'arborescence de réduction de tâches consiste en un seul sommet auquel est associée la tâche initiale. Par la suite, chaque expansion de l'arborescence est effectuée en appliquant une règle de synthèse dont la

partie gauche coïncide avec la tâche associée à une feuille-ET de l'arborescence. Les deux actions suivantes sont alors exécutées :

- (1) Un sommet-OU S' est ajouté à l'arborescence comme un successeur de S, et la règle de synthèse instantiée est associée à S'.
- (2) Chaque tâche contenue dans la partie droite de la règle instantiée est associée à un sommet-ET qui est ajouté à l'arborescence comme successeur de S' ; si l'ensemble de tâches de la partie droite de la règle est vide , S' est un sommet dit terminal.

Simultanément, les hypothèses contenues dans la partie droite de la règle sont ajoutées dans une base de faits.

L'arborescence de réduction de tâches est dite terminale lorsque chacune de ses feuilles est un sommet terminal. Si à ce moment il n'y a pas de contradiction dans la base de faits, le problème est considéré comme résolu.

2.3. Génération des règles de synthèse à partir du RSP

Les règles de synthèse permettant de décrire à TROPIC les structures possibles d'un assemblage (classes des objets, relations entre ces objets) et les valeurs possibles des propriétés lorsque celles-ci n'ont qu'un nombre fini de valeurs possibles (cf. Latombe 1977).

Pour chaque <cl> présente dans le RSP, l'interface engendre une règle de synthèse de la forme :

```
(concevoir- <cl> &x)
→ (I?  {(= (type &x) <cl>)}
      {(choisir-propriété-<cl> &x)
       (engendrer-composants-<cl> &x)}))
```

où

- la tâche (choisir-propriétés-<cl>&x) permettra de définir les valeurs possibles des propriétés des objets de la classe <cl> (lorsque ces valeurs sont en nombre fini) (cf. §2.3.1.).
- la tâche (engendre-composants-<cl> &x) permettra de définir la structure de l'assemblage "&x" (cf. §2.3.2.),
- les atomes commençant par "&" sont des variables,
- I? est une fonction de TROPIC réalisant l'instantiation d'une expression.

Exemple :

```
(concevoir-transformateur &x)
  → (I?   {(= (type &x) transformateur)}
      {(choisir-propriétés-transformateur &x)
       (engendrer-composants-transformateur &x)}))
```

Remarque :

La règle ci-dessus suppose que la tâche initiale donnée à TROPIC est de la forme (concevoir-<cl> ASS), où <cl> est le nom de la classe de l'assemblage ASS à concevoir.

2.3.1. Décomposition de la tâche (choisir-propriété-<cl> &x)

Les valeurs des propriétés ayant un nombre fini de valeurs possibles sont spécifiées dans le RSP :

- soit par des arcs "=", l'ensemble déterminé par l'arc EB pouvant être : SYMBOLE, REEL ou ENTIER (cas de la figure 5.1.).
- soit par une borne supérieure (arc "<" ou "≤") et une borne inférieure (arc ">" ou "≥"), l'ensemble déterminé par EB étant ENTIER (cas de la figure 5.2.).

Pour chaque tâche (choisir-propriétés-<cl> &x), l'interface engendre une règle de synthèse de la forme

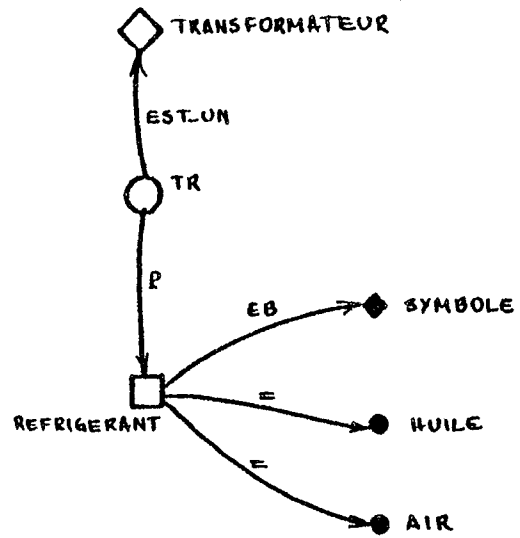


Figure 5.1.

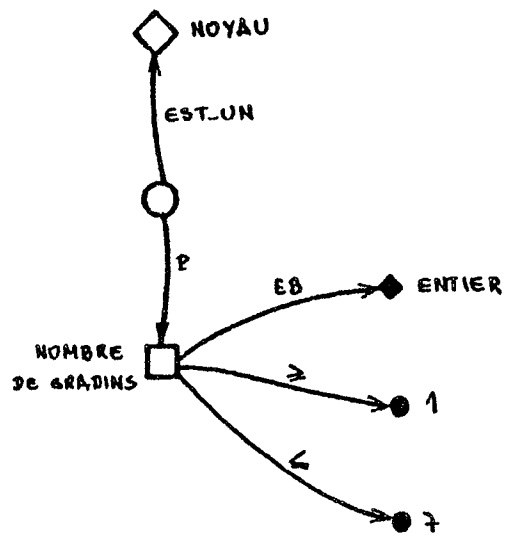


Figure 5.2.

```
(choisir-propriétés-<cl> &x)
→ (I?  NIL
    {(choisir-<prop1>-<cl> &x)
      (choisir-<prop2>-<cl> &x)
      ...
      (choisir-<propn>-<cl> &x)})
```

où <prop1>, <prop2>, ..., <propn> sont les étiquettes des sommets \square représentant des propriétés ayant un nombre fini de valeurs possibles.

Exemple :

```
(choisir-propriétés-transformateur &x)
→ (I?  NIL
    {(choisir-refrigérant-transformateur &x)
      (choisir-fréquence-transformateur &x)
      ...})
```

Si aucune propriété ayant un nombre fini de valeurs possibles n'est associée à la classe <cl>, la règle engendrée est

```
(choisir-propriétés-transformateur &x) → (NIL NIL)
```

Pour chaque tâche de la forme (choisir-<prop1>-<cl> &x), l'interface engendre autant de règles de synthèse que la propriété <prop1> a des valeurs possibles.

Chacune de ces règles a la forme

```
(choisir-<prop1>-<cl> &x)
→ (I?  {(= (<prop1> &x) v)} NIL)
```

où "v" est une valeur possible de la propriété <prop1> de <cl>.

Exemple :

Le RSP de la figure 5.1. permet d'engendrer les deux règles de synthèse suivantes :

- (choisir-réfrigérant-transformateur &x)
 - (I? {(= (réfrigérant &x) air)} NIL)
- (choisir-réfrigérant-transformateur &x)
 - (I? {(= (réfrigérant &x) huile)} NIL)

2.3.2. Décomposition de la tâche (engendrer-composants-<cl> &x)

Si le représentant générique de la classe <cl> n'est l'objet primaire d'aucune relation de composition, alors l'interface engendre la règle de synthèse

```
(engendrer-composants-<cl> &x) → (NIL NIL)
```

Dans le cas contraire, soient <cl1>, <cl2>, ..., <cln> les classes. Les représentants génériques figurant dans une relation de composition ayant le représentant générique de <cl> pour objet primaire (figure 5.3.).

L'interface engendre alors la règle de synthèse suivante :

```
(engendrer-composants-<cl> &x)
→ (PROG
  (SETQ &y1 (GENCONS))
  (SETQ &y2 (GENCONS))
  ...
  (SETQ &yn (GENCONS))
  (I? {(composant &x &y1)
      (composant &x &y2)
      ...
      (composant &x &yn)
      (<assoc1> &y11 &ym11)
      ...
      (<assucp> &y1p ... &ym11)}
    {(concevoir-<cl1>&y1)
     (concevoir-<cl2> &y2)
     ...
     (concevoir-<cln> &yn )}))
```


où :

- GENCONS est une fonction de TROPIC qui engendre des noms d'objets,
- $\langle \text{associ} \rangle$, $\forall i=1, \dots, p$ est le nom d'une relation associative entre des objets $\&y_j^i$ ($j=1, \dots, m_i$) de l'ensemble $\{\&y_1, \dots, \&y_n\}$

Exemple :

Le réseau de la figure 5.3. conduit à engendrer la règle de synthèse suivante :

```
(engendrer-composants-tour &x)
→ (PROG)
    (SETQ &y1 (GENCONS))
    (SETQ &y2 (GENCONS))
    (I? {(composant &x &y1)
        (composant &x &y2)
        (sur &y1 &y2)}
        {(concevoir-colonne &y1)
         (concevoir-toit &y2)}))
```

Remarque :

A ce niveau l'interface ne considère que les relations associatives entre des objets ayant le même père. En effet, les relations associatives entre des objets n'ayant pas le même père (mais ayant bien sûr un autre ancêtre commun) sont difficiles à décrire à l'aide des règles de synthèse acceptables par TROPIC (cf. les "règles conditionnelles" dans Latombe 1977).

2.3.3. Cas des groupes

Lorsque le représentant générique d'une classe $\langle cl \rangle$ est composé d'un groupe d'objets de la classe $\langle cl1 \rangle$, le traitement est légèrement différent.

Considérons le réseau de la figure 5.4. Il conduit à produire les règles de synthèse suivantes :

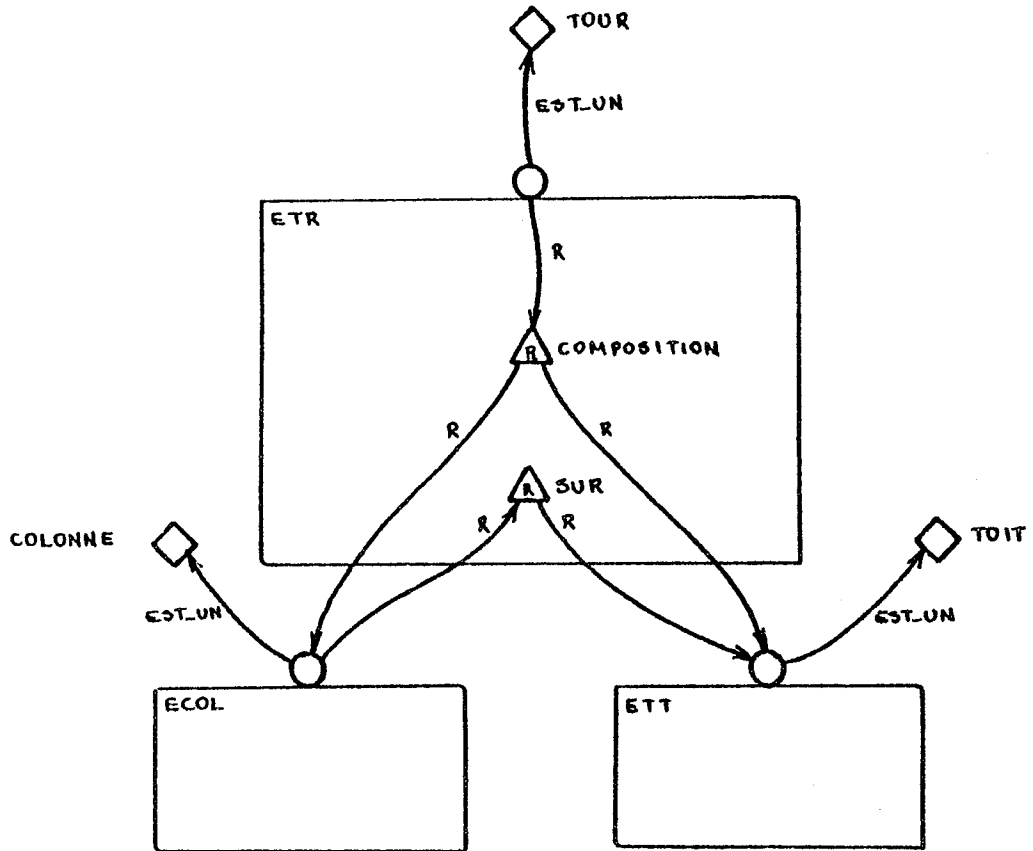


Figure 5.3.

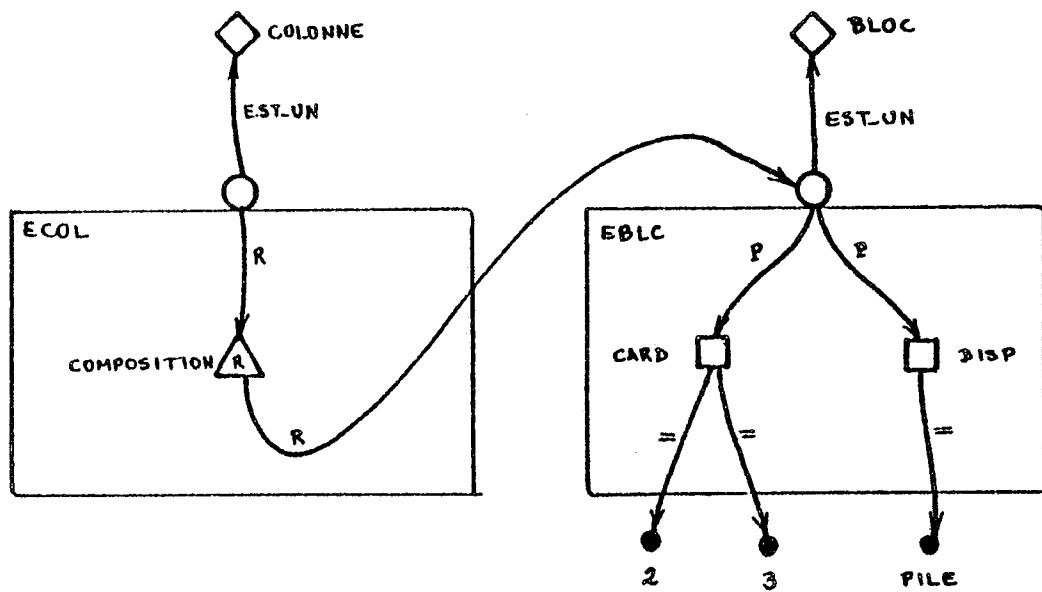


Figure 5.4.

```

- (engendrer-composants-colonne &x)
  → (PROG
      (SETQ &y1 (GENCONS))
      (SETQ &y2 (GENCONS))
      (I? {(= (nombre-de-blocs &x) 2)
           (composant &x &y1)
           (composant &x &y2)
           (sur &y1 &y2)}
          {(concevoir-bloc &y1)
           (concevoir-bloc &y2)})))

```

```

- (engendrer-composants-colonne &x)
  → (PROG
      (SETQ &y1 (GENCONS))
      (SETQ &y2 (GENCONS))
      (SETQ &y3 (GENCONS))
      (I? {(= (nombre-de-blocs &x) 3)
           (composant &x &y1)
           (composant &x &y2)
           (composant &x &y3)
           (sur &y1 &y2)
           (sur &y2 &y3)}
          {(concevoir-bloc &y1)
           (concevoir-bloc &y2)
           (concevoir-bloc &y3)})))

```

La génération de ces règles est fait ainsi :

- une règle est engendrée pour chaque valeur possible de la cardinalité du groupe ;
 - la propriété "nombre-de-<cl1>s" est associée à la classe <cl>; on note que les valeurs de cette propriété ne sont pas définies par les règles de synth
- synthèse suivant la méthode du § 2.1.1.

- les relations (ici SUR) entre les objets du groupe provenant de la disposition (DISP) interne du groupe sont exprimées dans ces règles.

Remarques :

- . Lorsque le représentant générique d'une classe d'objets <cl> est composé de n objets de classes <cl1>, ..., <cln> et de p groupes d'objets de classes <cl1'>, ..., <clp'>, l'interface engendre la règle de synthèse suivante :

```
(engendrer-composants-<cl> &x)
  → (I?  NIL
      {(engendrer-composants0-<cl> &x)
       (engendrer-composants1-<cl> &x)
       ...
       (engendrer-composantsp-<cl> &x)})
```

où :

- la réduction de la tâche (engendrer-composants0-<cl> &x) permet d'engendrer les n composants simples (non-groupes) du représentant de <cl>.
- la réduction des tâches (engendrer-composantsi-<cl> &x), i=1, ..., p, permet d'engendrer les composants "groupes" du représentant de <cl>.

. D'une manière générale, nous n'avons envisagé que des groupes simples formés avec des objets d'une même classe, car les règles de synthèse de TROPIC sont mal adaptées à la représentation de groupes complexes. Par ailleurs, nous n'avons pas envisagé le cas où un groupe dans son ensemble, ou un objet d'un groupe, participe à des relations associatives avec des objets n'appartenant pas à ce groupe.

3 - LES REGLES D'ANALYSE

3.1. Définition

Une règle d'analyse est une production :

$$\{f_1, \dots, f_m\} \rightarrow \{f_1', \dots, f_n'\}$$

où $f_1, \dots, f_m, f_1', \dots, f_n'$ ($m \geq 0, n \geq 0$) sont des expressions représentant des faits. La partie droite peut être une fonction dont l'évaluation donne $\{f_1', \dots, f_n'\}$ ($n \geq 0$).

Exemple :

```
{(= (type &x) bloc)
 (> (hauteur &x) 1)}
→ (I? {(= (section &x) carré)})
```

3.2. Utilisation par TROPIC

Chaque fois qu'une règle de synthèse est appliquée, TROPIC active toutes les règles d'analyse activables.

Une règle d'analyse est activable lorsque les faits qui constituent sa partie gauche coïncident avec des faits dans la base de faits courante.

Lorsqu'une règle d'analyse est activée, sa partie droite est instantiée ou évaluée selon la substitution donnée par la coïncidence de la partie gauche. Les faits ainsi obtenus sont ajoutés à la base de faits où ils peuvent à leur tour contribuer à rendre d'autres règles d'analyse activables.

Lorsqu'il n'y a plus de règle d'analyse activable, TROPIC vérifie la cohérence de la base de faits. S'il ne détecte pas de contradiction, il applique une nouvelle règle de synthèse, sinon il fait appel à la procédure de traitement d'erreur.

3.3. Génération des règles d'analyse à partir du RSP

Les règles d'analyse permettent de décrire à TROPIC les restrictions sur les valeurs de propriétés réelles ou entières (lorsque ces valeurs sont de nombre infini), les relations entre propriétés et les contraintes.

3.3.1. Restrictions sur les valeurs de propriétés réelles ou entières

Soit $\langle cl \rangle$ une classe d'objet. Soit $\langle prop \rangle$ l'étiquette d'un sommet Π du RSP dont l'arc EB pointe sur REEL ou ENTIER. Si des restrictions ($\leq, \geq, <, >, \neq$) sont appliquées à la propriété $prop$ (indépendamment de contraintes), l'interface engendre une règle d'analyse de la forme :

$$\begin{aligned} & \{ (= (type \&x) \langle cl \rangle) \} \\ \rightarrow & (I? \{ (r_1 (\langle prop \rangle \&x) v_1) \\ & \dots \\ & (r_n (\langle prop \rangle \&x) v_n) \} \}) \end{aligned}$$

où chaque fait $(r_i (\langle prop \rangle \&x) v_i)$ traduit une restriction :

- r_i est l'étiquette de l'arc du RSP qui spécifie la restriction,
- v_i est la valeur pointée par cet arc.

Exemple :

Le RSP de la figure 5.5. permet d'engendrer la règle d'analyse :

$$\begin{aligned} & \{ (= (type \&x) tour) \} \\ \rightarrow & (I? \{ (> (hauteur \&x) 10) \\ & (< (hauteur \&x) 20) \} \}) \end{aligned}$$

Remarques :

- Lorsque les restrictions appliquées à une propriété entière ne permettent qu'un nombre fini de valeurs pour cette propriété, les valeurs sont spécifiées par des règles de synthèse (cf. § 2.3.) et non par des règles d'analyse.

- Pour n'importe quel ensemble de base, à chaque nom de propriété l'interface associe un indicateur spécifiant l'ensemble de base des valeurs. TROPIC s'en sert lors de la vérification de la cohérence de la base de faits.

3.3.2. Relations entre propriétés

Nous ne considérons ici que les relations n-aires car les règles de production acceptées par TROPIC ne permettent pas de représenter de façon particulière les relations binaires.

Pour chaque figure de relation n-aire, l'interface engendre une règle d'analyse. Il y a trois cas possibles correspondant à la forme générale de la figure de relation n-aire montrée à la figure 5.6. :

(a) <cl0>=<cl1>=...=<cln> et la relation est placée dans le contexte de la classe <cli> ($i \in \{0, 1, \dots, n\}$). La règle engendrée est de la forme :

```
{ (= (type &x0) <cl0>)}
  →(I? {(r (<prop0> &x0) <expr>)})
```

où <expr> est le corps de l'expression lambda associée au sommet F, dans lequel les paramètres ont été remplacés par des références de la forme (<prop*i*> & aux propriétés correspondantes ($i=1, \dots, n$).

(b)] i et j tels que <cli>≠<clj> et la relation est placée dans le contexte d'une des classes <clk> ($k \in \{0, \dots, n\}$). La règle engendrée est de la forme :

```
{ (= (type &x0) <cl0>)
  (= (type &x1) <cl1>)
  ...
  (= (type &xn) <cln>)
  (composant &xi &xj)
  ...}
  → (I? {(r (<prop0> &x0) <expr>)}).
```

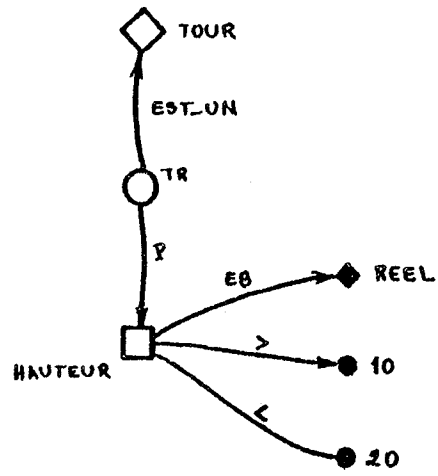


Figure 5.5.

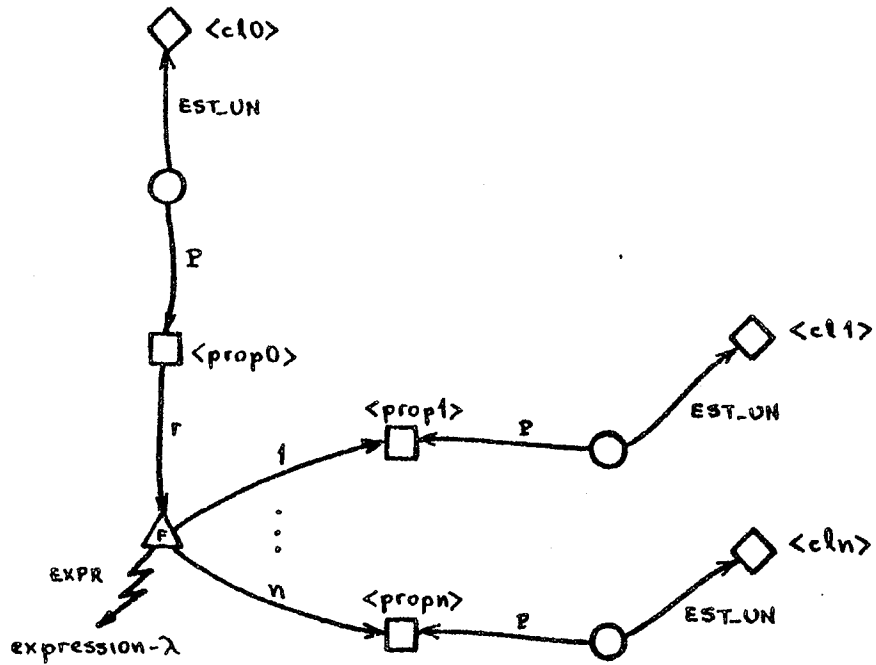


Figure 5.6.

Chaque relation de composition existant dans le RSP entre deux objets de classes $\langle cl_i \rangle$ et $\langle cl_j \rangle$ ($i=0,1,\dots,n$; $j=0,1,\dots,n$; $i \neq j$) est représenté par un fait (composant $\&x_i \ \&x_j$) dans la partie gauche de la règle.

(c) La relation est placée dans le contexte de $\langle cl \rangle$, la classe d'un ancêtre commun des objets des classes $\langle cl_0 \rangle, \langle cl_1 \rangle, \dots, \langle cl_n \rangle$ ($\langle cl \rangle \neq \langle cl_i \rangle, \forall i=0,1,\dots,n$)
 Dans ce cas, l'interface engendre une règle d'analyse de la forme :

```
{ (= (type &x) <cl> )
  (= (type &x0) <cl0>)
  ...
  (= (type &xn) <cln>)
  (composant &x &xi)
  ...
  (composant &xi &xj)
  ... }
→ (I? {(r (<prop0> &x0) <expr>)}).
```

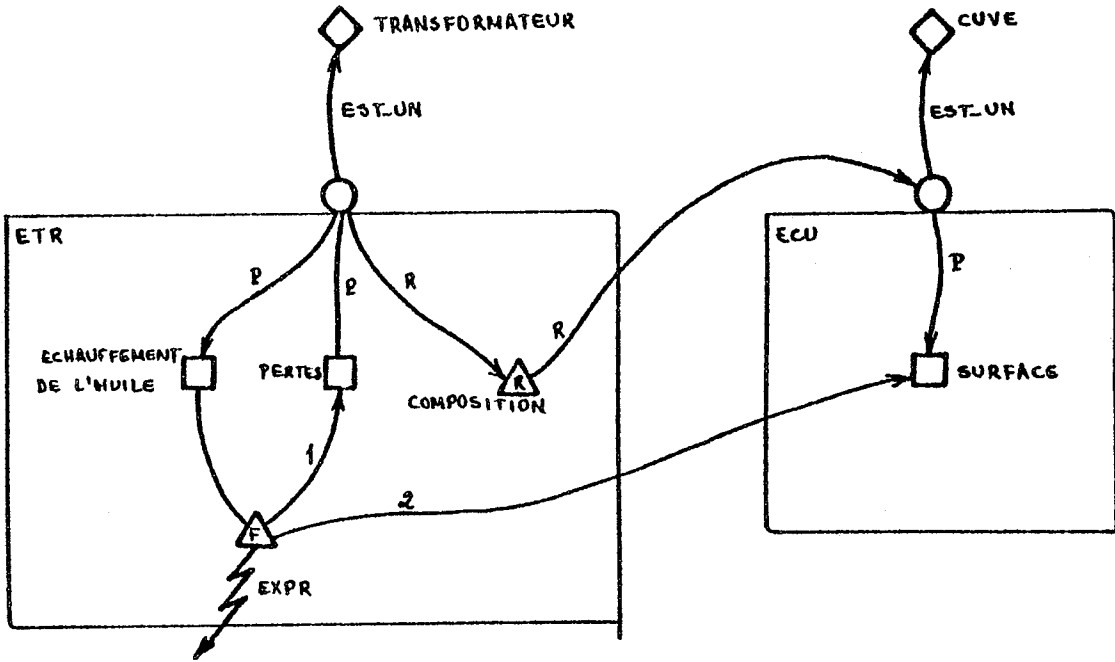
L'existence de l'ancêtre commun est représentée par le fait (= (type &x)<cl>) et par les faits (composant &x &xi) dans la partie gauche de la règle ($i=0,1,\dots$).

Exemple

Le RSP de la figure 5.7 permet d'engendrer la règle d'analyse

```
{(= (type &x0) transformateur)
 (= (type &x1) cuve)
 (composant &x0 &x1)}
(I? {(= (échauffement_de_l'huile &x0) (TIMES 445
      (EXP (QUOTIENT (pertes &x0)(surface &x1)) 0.8))))})
```

qui correspond au cas (b) énoncé ci-dessus.



(LAMBDA (P S) (TIMES 445 (EXP (QUOTIENT P S) 0.8))))

Figure 5.7.

3.3.3. Contraintes

Considérons d'abord le cas où les contraintes ont toutes une seule partie <si> et une seule partie <alors>. Pour chaque contrainte dans le RSP, soient :

- <cl1>, ..., <clp> les classes d'objets concernés par la contrainte,
- <p1>, ..., <pn> les étiquettes des arcs (<p&>, <prop*i*>, <val*i*>) contenus dans l'espace <si> de la contrainte, où <prop*i*> est l'étiquette du sommet □ relié au sommet . d'étiquette <val*i*> (i=1, ..., n).
- <rel1>, ..., <relq> les étiquettes des sommets \triangle_R adjacents aux arcs R contenus dans l'espace <si> de la contrainte.
- <p'1>, ..., <p'n> les étiquettes des arcs (<p'j>, <prop'j>, <val'j>) contenus dans l'espace <alors> de la contrainte, où <prop'j> est l'étiquette du sommet □ relié au sommet . d'étiquette <val'j> (j=1, ..., m),
- <rel'1>, ..., <rel's> les étiquettes des sommets R adjacents aux arcs R contenus dans l'espace <alors> de la contrainte.

De façon analogue au traitement des relations entre propriétés, il existe trois cas possibles de règle d'analyse à engendrer :

- (a) La contrainte concerne une seule classe d'objets <cl> et elle est placée dans le contexte de <cl>: Dans ce cas la règle engendrée est de la forme :

```
{(= (type &x) <cl>)
  (<pi> (<propi> &x) <vali>)
  ...}
  → (I? {(<p'j> (<prop'j> &xk) <val'j>)
    ... })
```

- (b) La contrainte concerne plusieurs classes d'objets <cl1>, ..., <cln> et elle est placée dans le contexte d'une de ces classes. Dans ce cas la règle d'analyse engendrée a la forme :

```
{( (= (type &x1) <cl1>)
  ...
  (= (type &xp) <clp>)
```

```

(composant &xk &x1)
...
(<pi> (<propi> &xk) <val&>)
...
(<rel1> &xk &x1)
...
(<relq> &xk &x1) }
  + (I? {(<p'j> (<prop'j> &xk) <val'j>)
    ...
    (<rel'1> &xk &x1)
    ...
    (<rel's> &xk &x1)})

```

Chaque relation de composition existant dans le RSP entre deux objets concernés par la contrainte est représentée par un fait (composant &xk &x1) dans la partie gauche de la règle d'analyse ($k \leq p$, $1 \leq q$).

(c) La contrainte concerne plusieurs classes d'objets <c1>, ..., <cn> et elle est placée dans le contexte d'un ancêtre commun <c1> de ces classes. Dans ce cas, la règle engendrée a la même forme que dans le cas précédent. L'existence de l'ancêtre commun est représentée par le fait (= (type &x) <c1>) et par les faits (composants &x &xi) dans la partie gauche de la règle ($i=1, \dots, n$).

Exemple :

1 Le RSP de la figure 5.8 permet d'engendrer la règle d'analyse :

```

{ (= (type &x1) transformateur)
  (= (type &x2) circuit-magnétique)
  (composant &x1 &x2)
  (= (réfrigérant &x1) huile) }
  + { (= (qualité-des-toiles &x2) q3) }

```

Dans le cas général, l'interface engendre $m \times n$ règles d'analyse par contrainte contenue dans le RSP, où m est le nombre de parties <si> et n est le nombre de parties <alors> de la contrainte. Chaque règle d'analyse résulte de la combinaison d'une partie <si> et d'une partie <alors>.

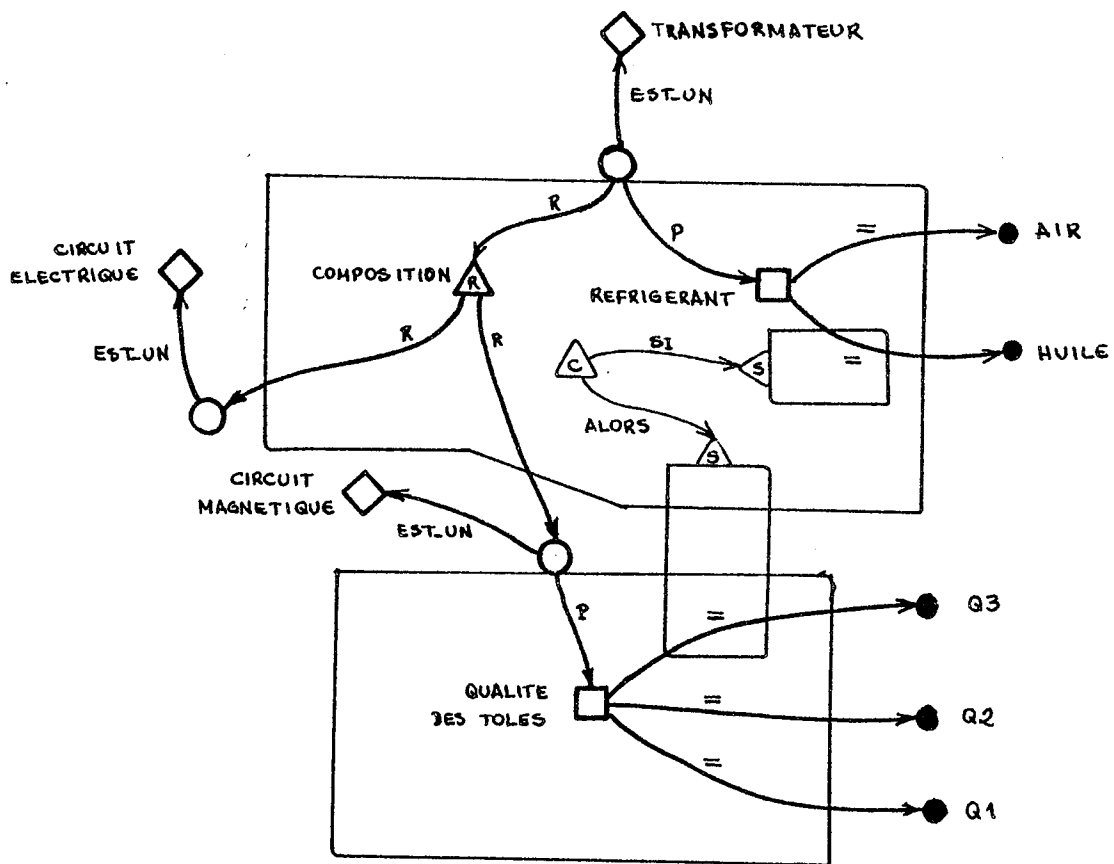


Figure 5.8.

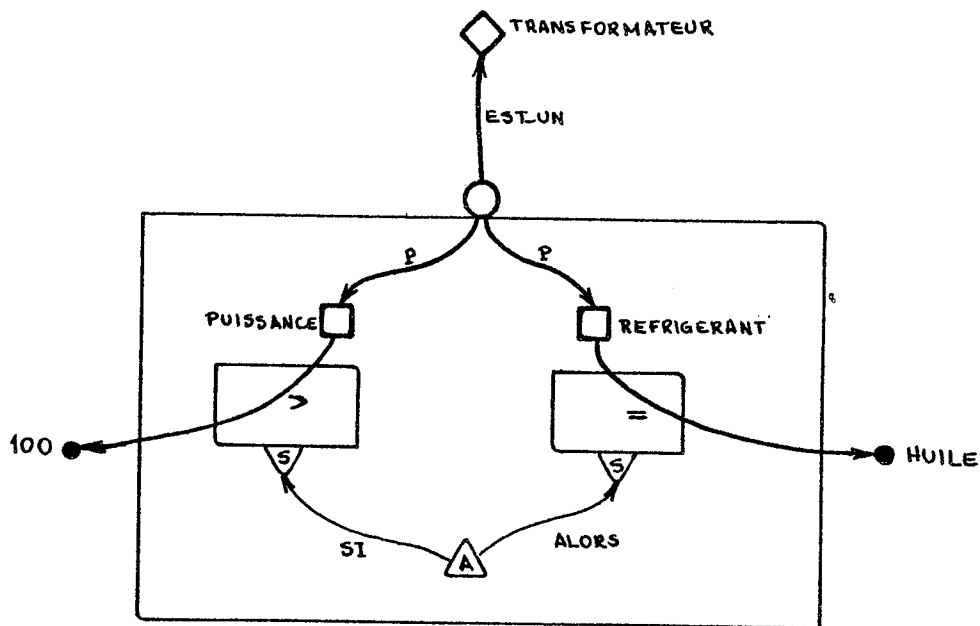


Figure 5.9.

4 - LES CONSEILS

4.1. Définition et utilisation

Un conseil a la même forme syntaxique qu'une contrainte. TROPIC la traite comme une contrainte faible dont la satisfaction n'est pas impérative.

4.2. Génération des conseils à partir du RSP

Les conseils sont traités par l'interface d'une façon analogue aux contraintes (cf. §3.3.3.) pour engendrer des productions semblables aux règles d'analyse.

Exemple :

Le RSP de la figure 5.9. conduit à engendrer le conseil :

```
{ (= (type &x) transformateur)
  ( > (puissance &x) 100) }
→ (I? {(= (réfrigérant &x) huile)})
```



Chapitre VI

DISCUSSION, IMPLANTATION, EXPERIMENTATION

1 - DISCUSSION -----	139
1.1. La généralité du programme PIAF -----	139
1.2. Pourquoi un réseau sémantique? -----	140
1.3. Généralité de l'interface -----	141
1.4. Limitations -----	142
2 - IMPLANTATION -----	143
2.1. Généralités -----	143
2.2. Implantation du RSP -----	146
2.3. Le constructeur du RSP -----	150
3 - EXPERIMENTATION -----	152
3.1. Domaine d'expérimentation -----	152
3.2. Quelques exemples -----	153
4 - TROIS PROBLEMES DE REPRESENTATION DANS LE RSP -----	165
4.1. Relations mutuellement exclusives -----	165
4.2. Le problème du composant partagé -----	165
4.3. Combinaisons ET/OU de restrictions sur les valeurs des propriétés	167

Ce chapitre regroupe : une discussion de quelques aspects généraux de l'interface (§ 1), une présentation de l'implantation sur ordinateur (§ 2) et une description de l'expérimentation effectuée (§ 3, complété par les appendices A, B et C). Il introduit en fin trois problèmes de représentation dans le RSP que nous n'avons pas résolus (§ 4).

1 - DISCUSSION

1.1. La généralité du programme PIAF

PIAF a été conçu pour l'analyse syntaxique de textes où le langage utilisé est très libre. Ainsi, il est muni d'une grammaire de dépendances assez générale qui le conduit à délivrer plusieurs arborescences de dépendances pour la plupart des phrases que nous lui soumettons.

Le langage utilisé pour décrire des connaissances sur les assemblages électromécaniques est plus restreint. PIAF a alors tendance à considérer trop de cas syntaxiques. De plus, les arborescences qu'il délivre ont une structure trop riche pour représenter utilement l'information renfermée dans ces phrases.

Dans la mesure où cela n'entraînait pas trop de complication, nous avons réduit cet excès de généralité de PIAF (pour notre application) en simplifiant sa grammaire de façon à diminuer le nombre d'arborescences en général délivrées. Mais nous avons quand même dû ajouter des mécanismes afin de filtrer et de réduire les arborescences engendrées.

Une alternative possible à PIAF serait d'utiliser une grammaire de transformations (Colmerauer 1975, Hendrix 1977) restreinte à notre domaine d'application. Un ensemble d'actions est associé à chaque production d'une telle grammaire, ce qui permet de mêler analyse syntaxique et analyse sémantique. De plus, la grammaire peut être définie progressivement.

Toutefois, bien que le langage utilisé pour décrire des connaissances électromécaniques soit restreint, il est difficile de caractériser précisément cette restriction. Aussi, la structure de l'interface fonctionnant de façon satisfaisante, la généralité de PIAF peut devenir un avantage au cours de recherches à venir car elle permet d'envisager une communication plus riche avec TROPIC.

1.2. Pourquoi un réseau sémantique?

Plutôt que de construire une base de connaissances sous la forme d'un RSP, nous aurions pu essayer de traduire directement les phrases soumises à l'interface dans le langage des règles de production accepté par TROPIC, comme cela a déjà été fait dans MYCIN (Davis 1977). Toutefois, il n'y a en général pas de correspondance simple entre une phrase et une règle acceptable par TROPIC. Un tel procédé serait donc soit très difficile à réaliser, soit très restrictif quant au langage accepté par l'interface.

Le formalisme des RSP permet de représenter les connaissances sans qu'il soit nécessaire de les décomposer en petits modules et il permet aussi un accès commode aux connaissances déjà mémorisées. De plus, le formalisme des RSP, notamment la facilité qu'il offre de délimiter des sous-ensembles d'informations dans des espaces, nous permet de structurer les RSP par des figures construites d'après les descripteurs sémantiques attachés aux mots des formes réduites. Ces figures sont suffisantes au programme de liaison RSP-TROPIC pour traduire le RSP dans le langage accepté par TROPIC.

Par ailleurs, le RSP peut être utilisé autrement que comme base de connaissances intermédiaire. En effet, la correspondance RSP-TROPIC (cf. chapitre 5) a été définie pour ne pas modifier l'implantation de TROPIC. Néanmoins, nous pouvons imaginer que celui-ci travaille directement sur un RSP

pour y puiser des connaissances, et pour y construire (en termes de sommets, arcs et espaces) les solutions aux problèmes. Dans ce cas, au cours de la résolution d'un problème, on peut utiliser la partition en espaces pour distinguer des solutions partielles, des solutions alternatives ou différents aspects d'une solution.

Le formalisme des RSP permet aussi d'envisager de compléter les méthodes de résolution de problèmes de TROPIC, en particulier par l'application d'un processus de propagation de contraintes analogue à celui utilisé par Laurière (1976). Ce processus consiste à effectuer des déductions sur le réseau de façon à réduire le nombre de possibilités susceptibles d'être pris en considération par TROPIC pendant la résolution d'un problème.

1.3. Généralité de l'interface

De la même façon que PIAF et le programme de transformation d'arborescences sont des outils indépendants d'une application linguistique donnée, le modèle de représentation d'assemblages physiques a été conçu pour être indépendant d'une discipline technique particulière. Ainsi, pour utiliser l'interface sur une application différente de la conception d'assemblages électromécaniques, il suffit de modifier le dictionnaire (et éventuellement la grammaire de PIAF), les règles de traitement post-syntaxique, et les descripteurs sémantiques, sans toucher aux programmes. Ceci assure aussi la possibilité d'augmenter progressivement le nombre de cas traités et la complexité de ceux-ci, dans le cadre d'une application technique donnée.

Les figures définies sont toutefois propres aux assemblages physiques et leur modification nécessiterait une mise à jour partielle des programmes de construction du RSP.

Ces capacités d'évolution et d'adaptation témoignent, nous l'espérons, d'une certaine généralité des concepts de base de l'interface.

1.4. Limitations

Les limitations de l'interface concernent la complexité des phrases qu'il peut traiter et le type de connaissances qu'il peut représenter :

a/ Les phrases traitées par l'interface sont assez simples comme le montrent les exemples donnés dans le texte et notamment au chapitre 6.

En particulier, nous ne traitons pas les relations n-aires entre propriétés, qui doivent être exprimées directement par des formes réduites.

En outre, nous n'avons pas de mécanisme pour traiter des références entre phrases, si bien que chacune d'elles doit contenir tous les renseignements nécessaires à son interprétation ; il en résulte parfois des énoncés assez lourds.

Finalement, l'interface dispose de connaissances trop limitées sur le domaine d'application pour résoudre de façon satisfaisante les ambiguïtés.

b/ Le formalisme de représentation basé sur les figures permet la description de connaissances sur les assemblages physiques tels qu'ils sont considérés dans TROPIC. Ces assemblages sont décrits notamment par leur composition, par les propriétés qui leur sont associées et par des relations entre ces propriétés, tandis que les relations entre sous-assemblages occupent une place beaucoup moins importante. C'est pourquoi nous ne traitons pas des relations géométriques complexes.

En particulier, la notion de distance entre deux objets est impossible à représenter avec les éléments que nous avons proposés.

Il serait intéressant d'étudier une extension du formalisme pour représenter des relations spatiales complexes et permettre son application dans des systèmes utilisant des techniques graphiques par exemple.

Par ailleurs, le contrôle de la cohérence dans la base de connaissances est rudimentaire.

2. IMPLANTATION

2.1. Généralités

La figure 6.1. montre la structure globale de la partie de l'interface qui concerne la construction du RSP à partir de phrases en français. Le moniteur appelle le programme PIAF qui lit une phrase et en fait l'analyse syntaxique. Le résultat est une(ou plusieurs)arborescence de dépendances présentée sous la forme d'une liste, laquelle est fournie au transformateur d'arborescences. Celui-ci produit une forme réduite unique qui à son tour est transmise au constructeur du RSP. Ce dernier incorpore dans le RSP les nouvelles connaissances apportées. Pour l'expérimentation, il est aussi possible de fournir des formes réduites directement au constructeur du RSP, sans passer par les modules précédents.

Tous les programmes sont écrits en LISP/360 à l'exception du programme PIAF, écrit en LP/360. Pour permettre la communication entre PIAF et le moniteur, il existe un interface LP/360-LISP/360 qui réalise l'activation de PIAF et le transfert des arborescences de dépendances à LISP (Joloboff,1978). Nous rappelons que le programme permettant la communication RSP-TROPIC n'a pas été implanté.

Le programme PIAF occupe 50k-octets de mémoire centrale. Le transformateur d'arborescences compte 63 fonctions LISP. Le constructeur du RSP en compte 252 réparties ainsi : 138 fonctions de construction et exploitation de RSP, 76 fonctions de construction de figures, 9 fonctions de "pattern-matching", 16 fonctions d'incorporation de nouvelles connaissances et 13 fonctions d'impression.

Le système est opérationnel sur un IBM 360/67 sous CP/CMS. Voici l'organisation d'une séance d'utilisation :

- (1) L'interface PIAF-LISP exige que la séance commence par une activation de PIAF :

```
sous CMS...
piaf1          /* nom du module PIAF-LISP */
EXECUTION BEGINS ...
FORME DU RESULTAT? (STRUCTURE OU CHAINE OU LISP)
lisp
MODE? (PAS A PAS OU CONTINU)
cont
ALGORITHMES? (DEP OU DEPCVAR OU DEP+CVAR)
dep
>              /* A ce moment, LISP a le contrôle */
```

- (2) On charge les programmes de transformation d'arborescences et de construction du RSP :

```
>deflec(transfrb nil)
>deflec(consrsp nil)
>phrase ( )
?              /* le caractère ? indique la demande d'une phrase */
```

- (3) Les informations données par chaque phrase sont insérées dans le RSP :

```
? un transformateur est composé d'un circuit magnétique et d'un
  circuit électrique.
D'AC
? la puissance d'un ...
  :
? fin.              /* fin de la séance */
```

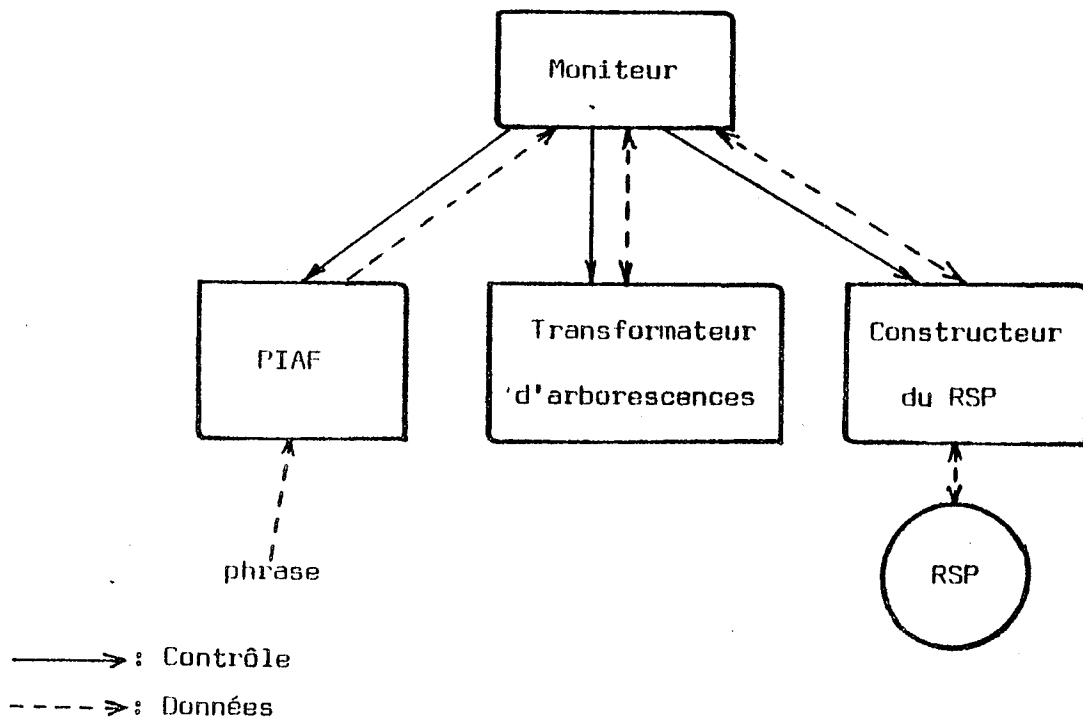


Figure 6.1.

Il est aussi possible d'utiliser directement le constructeur du RSP, qui accepte des formes réduites données sur le terminal ou trouvées dans un fichier :

```
>deflec(consrsp nil)
>formred( )
FORME ←          /* A ce moment on peut énoncer une forme réduite */
                  (FREQUENCE TRANSFORMATEUR (OU (= 50)(= 60)))
D'AC
FORME ← fin      /* fin de la séance */
>formredf(f1 nil) /* Ceci permet de lire des formes réduites du fichier f1 *,
```

Finalement, il est possible de sauvegarder le RSP créé pendant la séance dans un fichier LISP et de le récupérer plus tard :

```
>saversp(f1)      /* Le RSP est sauvegardé dans le fichier f1 */
:
>getrsp(f1)       /* Le RSP du fichier f1 est chargé en mémoire */
```

2.2. Implantation du RSP

Le RSP est implanté comme une collection de sommets, arcs et espaces. Chacun d'eux est constitué par un atome LISP auquel sont associés un type et une liste qui sert à le relier à d'autres éléments du RSP. La figure 6.2. montre la représentation des sommets, arcs et espaces :

- Pour les sommets :

```
<arcsin> est la liste d'arcs incidents au sommet
<arcsor> est la liste d'arcs sortant du sommet
<espass> est l'espace éventuel associé au sommet pour former un sommet-espace
<escont> est la liste des espaces qui contiennent le sommet
<lprop> est la liste des propriétés du sommet.
```

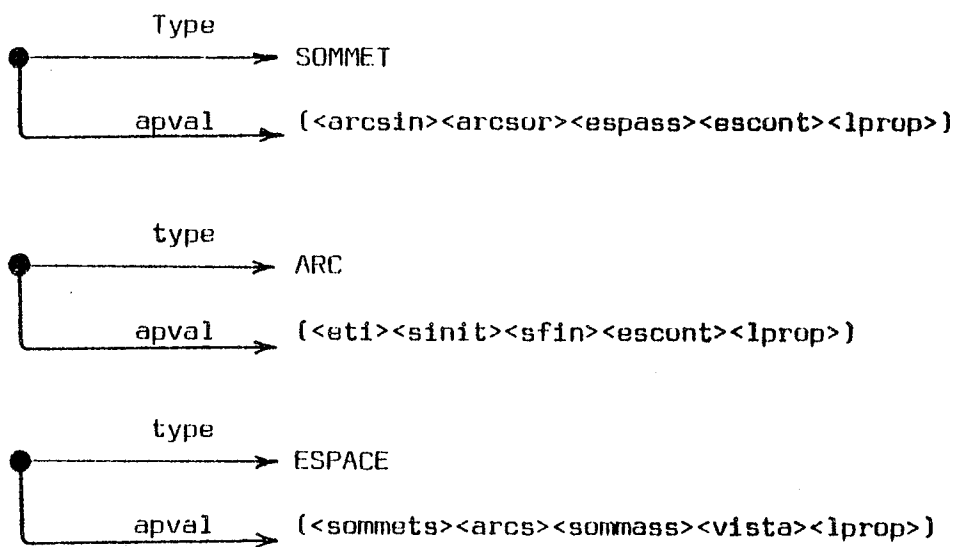


Figure 6.2.

- Pour les arcs :

<eti> est l'étiquette de l'arc
 <sinit> est le sommet initial de l'arc
 <sfinit> est le sommet terminal de l'arc
 <escont> est la liste des espaces qui contiennent l'arc
 <lprop> est la liste des propriétés de l'arc.

- Pour les espaces :

<sommets> est la liste des sommets contenus dans l'espace
 <arcs> est la liste des arcs contenus dans l'espace
 <sommass> est le sommet éventuel associé à l'espace pour former un
 sommet-espace.
 <vista> est la liste des espaces qui constituent la vista de l'espace
 <lprop> est la liste des propriétés de l'espace.

Par exemple, les éléments de la figure 6.3. constituent le réseau de la figure 6.4.

Pour construire et manipuler des RSP, il existe un ensemble de fonctions de base. Les éléments de la figure 6.3. ont été ainsi construits à l'aide des instructions suivantes :

```
E1 := cresptype(NIL) ; /* création de l'espace universel */
spput(E1, 'NOM, 'U);
S1 := crenode(list E1); /*un sommet est créé dans E1 */
gnodeput(S1, 'NOM, 'TOUR);
gnodeput(S1, 'TYPE, 'CLASSE);
E2 := crespace(list E1); /* création d'un espace dont la vista est composée
des espaces passés en paramètre (E1) et de l'espace
créé. La vista de E2 est donc (E2,E1) */
spput(E2, 'NOM, 'ETOUR);
S2 := cresnode(E2, list E1); /* un sommet est créé dans l'espace E1 et associé
à l'espace E2 pour former un sommet-espace(S2-E2) */
gnodeput(S2, 'TYPE, 'OBJET);
A1 := crearc('EST_UN, S2, S1, list E1); /* l'arc est créé dans E1 */
A2 := crearc('REP, S1, S2, list E1);
```

La création de E2 montre comment la vista d'un espace est déterminé au moment de sa création en indiquant les espaces qui la composent. Pour obtenir

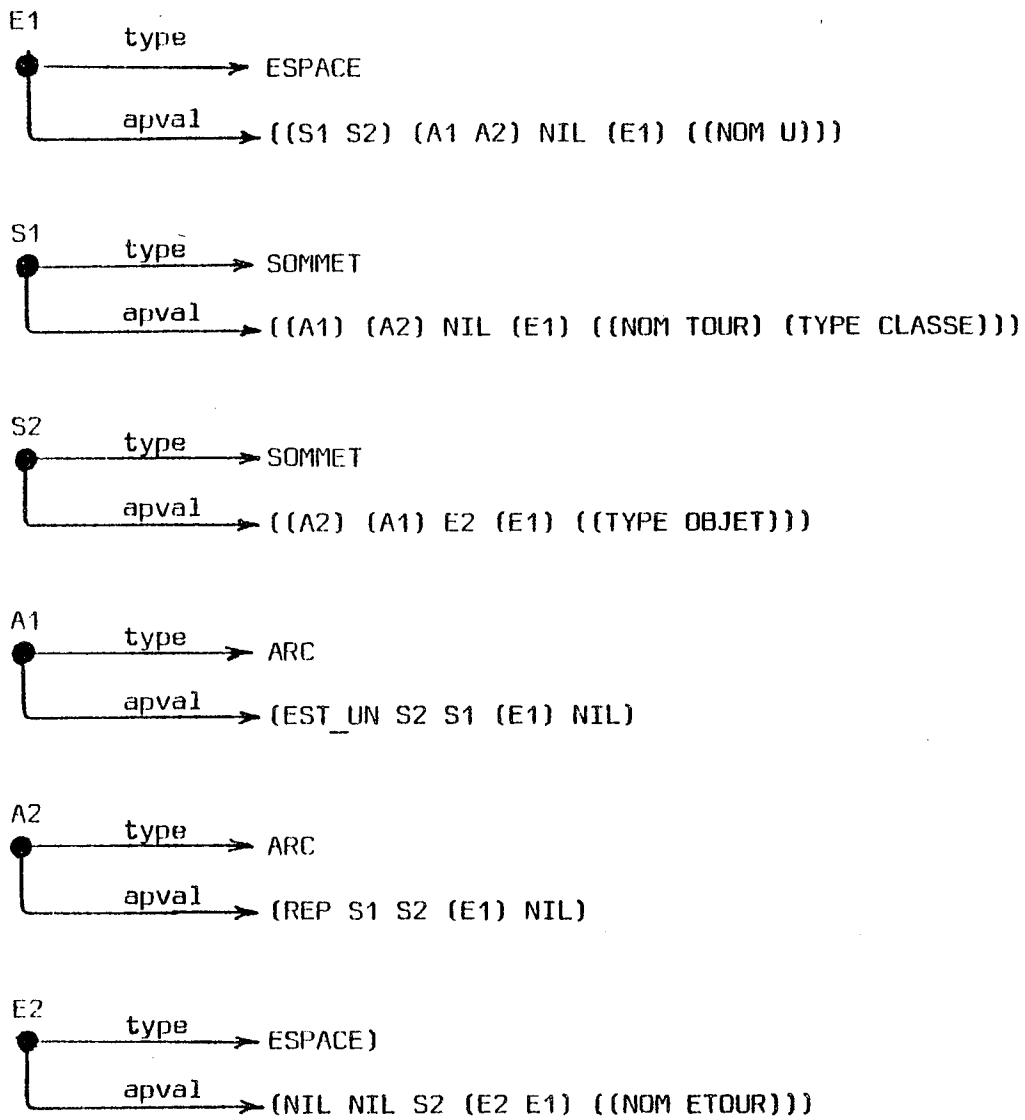


Figure 6.3.

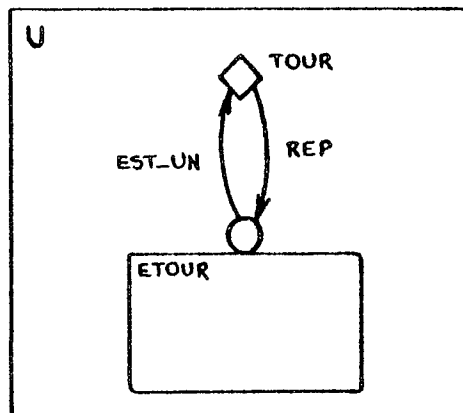
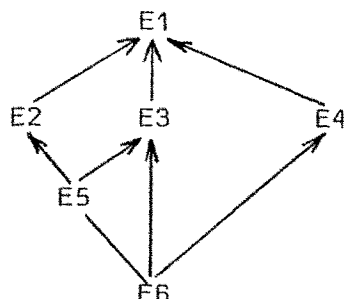


Figure 6.4.

l'organisation d'espaces



il faut ajouter les actions suivantes:

```

E3 := crespacel(list E1);
E4 := crespacel(list E1);
E5 := crespacel(list(E2 E3 E1));
E6 := crespacel(list(E5 E4 E2 E3 E1));
  
```

2.3. Le constructeur du RSP

La structure du constructeur du RSP est montrée à la figure 6.5. :

- Construction et exploitation de RSP :

Ce module est composé d'un ensemble de fonctions pour créer des sommets, arcs et espaces, leur associer des propriétés, les manipuler de façons diverses (transfert des sommets et des arcs d'un espace à un autre, élimination de sommets et arcs), et consulter le contenu du réseau (trouver les arcs et sommets d'un espace, trouver les sommets accessibles à partir d'un sommet donné, trouver la vista d'un espace, etc...).

- Construction des figures :

Ce module s'occupe de construire à partir des formes réduites, les figures correspondantes. Comme les figures sont composées de sommets, arcs et espaces, ce module fait appel aux fonctions de base pour la construction des RSP.

- Pattern-matching :

Ce module s'occupe de comparer le sous-réseau créé par le module précédent avec le reste du RSP courant. Le but est de déterminer les nouvelles informations qui devront être incorporées au RSP. Ce module fait appel aux fonctions de base pour consulter le contenu du RSP.

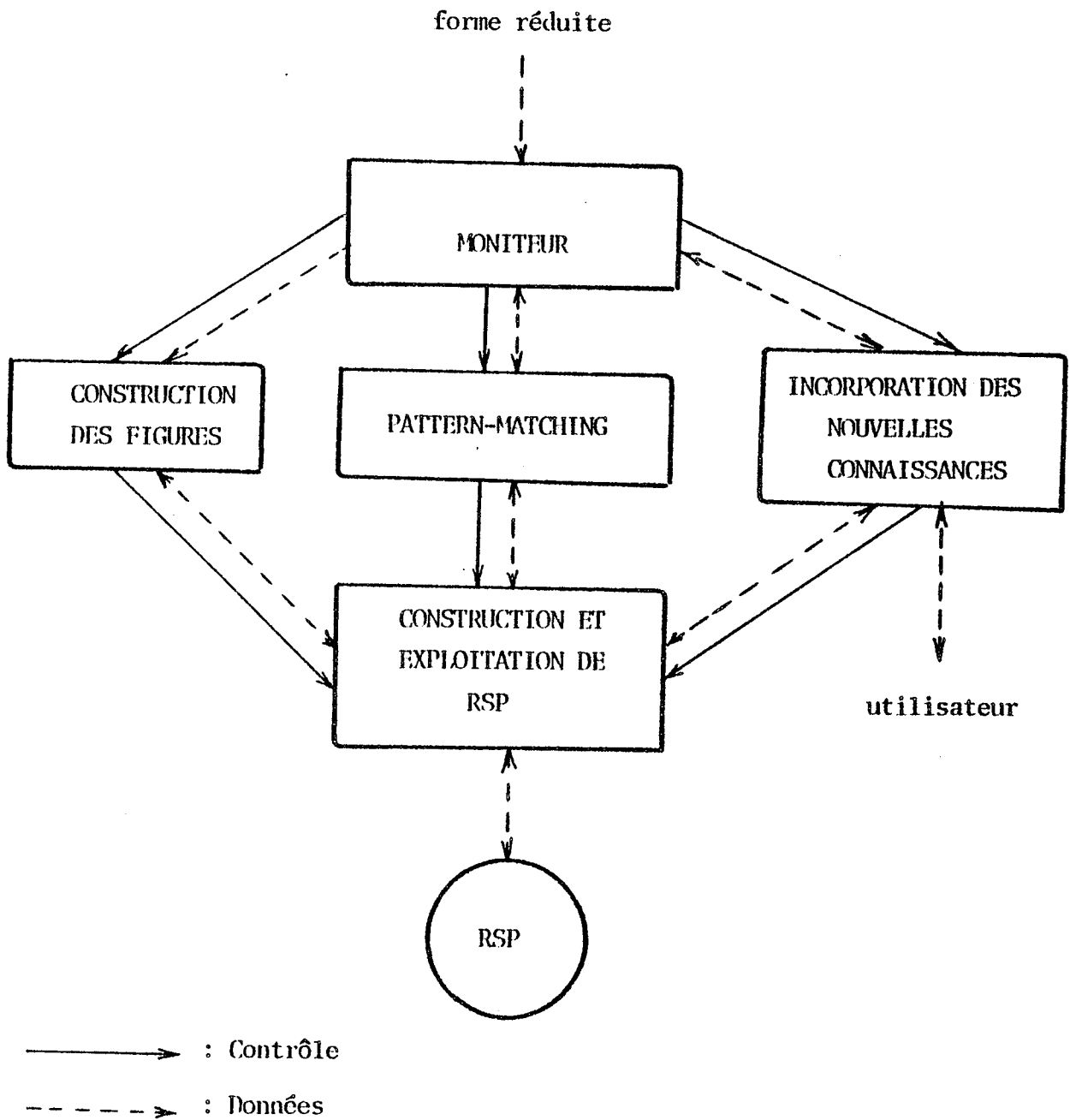


Figure 6.5.

- Incorporation des nouvelles connaissances :

Ce module s'occupe de vérifier la cohérence des nouvelles informations avec celles déjà existantes, d'interroger l'utilisateur en cas d'incompatibilité, et d'ajouter au RSP les nouveaux éléments d'information. Pour cela, il utilise des fonctions de base pour créer des éléments nouveaux et pour consulter et manipuler le RSP courant.

- Moniteur :

Ce module reçoit les formes réduites et appelle les fonctions nécessaires pour leur traitement.

3 - EXPERIMENTATION

3.1. Domaine d'expérimentation

L'expérimentation de l'interface a été réalisée dans le contexte des assemblages physiques. Plus particulièrement, nous avons utilisé deux exemples déjà traités par TROPIC : la description d'une tour et la description d'un transformateur électrique.

Nous avons donc augmenté le dictionnaire de PIAF avec un vocabulaire adapté à nos besoins, en particulier en ce qui concerne les assemblages électromécaniques. D'autre part, nous avons fourni les descripteurs sémantiques nécessaires pour le traitement des phrases et la construction du RSP. Nous considérons que les règles de transformation définies (cf. appendice A) sont assez générales pour être appliquées dans le traitement de la plupart des descriptions d'assemblages physiques.

Nous avons utilisé des phrases relativement simples où le rôle de chaque groupe de mots est facilement déterminé. Dans l'état actuel du système, le constructeur du RSP est capable de traiter des formes réduites plus complexes que celles que les règles de transformation définies permettent d'obtenir. En effet, le nombre de règles dont nous disposons est assez réduit, si bien

que nous ne pouvons pas traiter certains types de phrases. En particulier, le traitement des références anaphoriques est très limité et celui des relations n-aires entre propriétés est inexistant (les relations n-aires sont rentrées à partir des formes réduites).

3.2. Quelques exemples

Les appendices B et C montrent le traitement de la description d'une tour et celui de la description d'un transformateur électrique respectivement. Ils montrent aussi les RSP obtenus. Ci-dessous nous illustrons seulement quelques aspects du traitement des phrases.

Exemple 1 : Paraphrases

Le toit d'une tour est situé sur la colonne de la tour.

```
(EST (TOIT LE * (D' * (TOUR UNE *))) * (SITU * (SUR * (COLONNE LA * (DE * (TOUR LA *))))))
R001
(EST (TOIT * (D' * (TOUR UNE *))) * (SITU * (SUR * (COLONNE LA * (DE * (TOUR LA *))))))
R 002
(EST (TOIT * (D' * TOUR)) * (SITU * (SUR * (COLONNE LA * (DE * (TOUR LA *))))))
R045
(EST (TOIT * (DE * TOUR)) * (SITU * (SUR * (COLONNE LA * (DE * (TOUR LA *))))))
R 001
(EST (TOIT * (DE * TOUR)) * (SITU * (SUR * (COLONNE * (DE * (TOUR LA *))))))
R001
(EST (TOIT * (DE * TOUR)) * (SITU * (SUR * (COLONNE * (DE * TOUR))))
R045
(EST (TOIT * (DE * TOUR)) * (SITU * (SUR * (COLONNE * (DE * TOUR))))
R062
(EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * TOUR)))
R061
(SUR (TOIT * (DE * TOUR)) * (COLONNE * (DE * TOUR)))
```

(SUR (TOIT (DE TOUR)) (COLONNE (DE TOUR)))

La règle R062 identifie et élimine le mot superflu "situé". Pour cela, cette phrase produit la même forme réduite que la phrase suivante :

Le toit d'une tour est sur la colonne de la tour.

(EST (TOIT LE * (D' * (TOUR UNE *))) * (SUR * (COLONNE LA * (DE * (TOUR LA *))))
 R001
 (EST (TOIT * (D' * (TOUR UNE *))) * (SUR * (COLONNE LA * (DE * (TOUR LA *))))
 R002
 (EST (TOIT * (D' * TOUR)) * (SUR * (COLONNE LA * (DE * (TOUR LA *))))
 R045
 (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE LA * (DE * (TOUR LA *))))
 R001
 (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * (TOUR LA *))))
 R001
 (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * TOUR))))
 R 045
 (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * TOUR))))
 R 061
 (SUR (TOIT * (DE * TOUR)) * (COLONNE * (DE * TOUR)))

(SUR (TOIT (DE TOUR)) (COLONNE (DE TOUR)))

Par ailleurs, les règles R041 et R044 permettent d'arriver à la forme réduite

(COMPOSITION TOUR (ET COLONNE TOIT))

à partir des phrases "une tour est composée d'une colonne et d'un toit"
 et "les composants d'une tour sont une colonne et un toit" (cf. appendice B).

Exemple 2 : Détermination des propriétés implicites

La section d'un bloc est carrée ou ronde.

(EST (SECTION LA * (D' * (BLOC UN *))) * (CARRE * (OU * ROND)))
 R003
 (EST (SECTION LA * (D' * (BLOC UN *))) * (OU CARRE ROND))
 R001
 (EST (SECTION * (D' * (BLOC UN *))) * (OU CARRE ROND))
 R002
 (EST (SECTION * (D' * BLOC)) * (OU CARRE ROND))
 R026
 (EST (SECTION * BLOC) * (OU CARRE ROND))
 R 008
 (EST (SECTION * BLOC) * (OU (*EG CARRE) ROND))

R006
 (EST (SECTION * BLOC) * (OU (*EG CARRE) (*EG ROND)))
 R022
 (SECTION * BLOC (OU (*EG CARRE) (*EG ROND)))

(SECTION BLOC (OU (*EG CARRE) (*EG ROND)))

un toit est prismatique ou pyramidal.

(EST (TOIT UN *) * (PRISMATIQUE * (OU * PYRAMID)))
 L05
 (EST (TOIT UN *) * (OU PRISMATIQUE PYRAMID))
 R002
 (EST TOIT * (OU PRISMATIQUE PYRAMID))
 R003
 (EST TOIT * (OU (*EG PRISMATIQUE) PYRAMID))
 R006
 (EST TOIT * (OU (*EG PRISMATIQUE) (*EG PYRAMID)))
 R027
 (FORME TOIT (OU (*EG PRISMATIQUE) (*EG PYRAMID)) *)

(FORME TOIT (OU (*EG PRISMATIQUE) (*EG PYRAMID)))

Commentaire :

Dans la première phrase, la propriété SECTION est explicitement spécifiée.
 Dans la deuxième phrase, la règle R027 ajoute la propriété FORME grâce à un
 indicateur sémantique qui indique que PRISMATIQUE et PYRAMIDAL sont des
 valeurs possibles de cette propriété.

Exemple 3 - Relation binaire entre propriétés

La section du premier bloc d'une colonne doit être égale à
 ? la section du deuxième bloc de la colonne.

```

(D (SECTION LA * (DU * (BLOC PREMIER * (D' * (COLONNE UNE *)))) * (ET * (*EG * (SECTION LA * (DU * (BLOC
(BLOC DEUXIEM * (DE * (COLONNE LA *)))))))
R 001
(D (SECTION * (DU * (BLOC PREMIER * (D' * (COLONNE UNE *)))) * (ET * (*EG * (SECTION LA * (DU * (BLOC
DEUXIEM * (DE * (COLONNE LA *)))))))
R 126
(D (SECTION * (DU * (BLOC (IDENT PREMIER) * (D' * (COLONNE UNE *)))) * (ET * (*EG * (SECTION LA *
(DU * (BLOC DEUXIEM * (DE * (COLONNE LA *)))))))
R 002
(D (SECTION * (DU * (BLOC (IDENT PREMIER) * (D' * COLONNE))) * (ET * (*EG * (SECTION LA * (DU * (BLOC
DEUXIEM * (DE * (COLONNE LA *)))))))
R 045
(D (SECTION * (DU * (BLOC (IDENT PREMIER) * (DE * COLONNE))) * (ET * (*EG * (SECTION LA * (DU * (BLOC
DEUXIEM * (DE * (COLONNE LA *)))))))
R028
(D (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE))) * (ET * (*EG * (SECTION LA * (DU * (BLOC DEUXIEM
* (DE * (COLONNE LA *)))))))
R 001
(D (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE))) * (ET * (*EG * (SECTION * (DU * (BLOC DEUXIEM
* (DE * (COLONNE LA *)))))))
R 126
(D (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE))) * (ET * (*EG * (SECTION * (DU * (BLOC (IDENT
DEUXIEM * (DE * (COLONNE LA *)))))))
R 001
(D (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE))) * (ET * (*EG * (SECTION * (DU * (BLOC (IDENT
DEUXIEM * (DE * COLONNE))))))
R045
(D (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE))) * (ET * (*EG * (SECTION * (DU * (BLOC (IDENT
DEUXIEM * (DE * COLONNE))))))

```

R 028
 (D (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE)) * (ET * (*EG * (SECTION: * (BLOC (IDENT DEUXIEM)
 * (DE * COLONNE))))))
 R004
 (EST (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE)) * (*EG * (SECTION * (BLOC (IDENT DEUXIEM) *
 (DE * COLONNE))))))
 R022
 (SECTION * (BLOC (IDENT PREMIER) * (DE * COLONNE)) (*EG * (SECTION * (BLOC (IDENT DEUXIEM) * (DE * COLONNE)
))))

(SECTION (BLOC (IDENT PREMIER) (DE COLONNE)) (*EG (SECTION (BLOC (IDENT DEUXIEM) (DE COLONNE))))

Commentaire :

Les formes réduites qui expriment les valeurs possibles d'une propriété et les relations binaires entre propriétés ont des structures voisines. Pour cela, la règle R022 est appliquée dans les deux cas (voir première phrase de l'exemple 2).

Il faut remarquer aussi comment la règle R126 identifie les membres spéciaux d'un groupe.

Exemple 4 : Identification des groupes

Un circuit magnétique est composé de 3 noyaux et de 2 culasses.

```
(EST (CIRCUIT MAGNETIQUE UN *) * (COMPOS * (DE * (NOYAU 3 *) (ET * (DE * (CULASSE 2 *))))))
R002
(EST (CIRCUIT MAGNETIQUE UN *) * (COMPOS * (DE * (ET (NOYAU 3 *) (CULASSE 2 *))))))
R009
(EST CIRCUIT MAGNETIQUE * (COMPOS * (DE * (ET (NOYAU 3 *) (CULASSE 2 *))))))
(EST CIRCUIT MAGNETIQUE * (COMPOS * (DE * (ET (NOYAU * (CARD * (*EG * 3))) (CULASSE 2 *))))))
R009
(EST CIRCUIT MAGNETIQUE * (COMPOS * (DE * (ET (NOYAU * (CARD * (*EG * 3))) (CULASSE * (CARD * (*EG *
2))))))
R 041
(COMPOSITION CIRCUIT MAGNETIQUE (ET (NOYAU * (CARD * (*EG * 3))) (CULASSE * (CARD * (*EG * 2)))) *)
```

```
(COMPOSITION CIRCUIT MAGNETIQUE (ET (NOYAU (CARD (*EG 3))) (CULASSE (CARD (*EG 2))))
```

Commentaire :

La règle R009 identifie un groupe d'objets par la présence d'un modificateur numérique différent de 1.

Exemple 5 : Traitement des contraintes

si le refrigerant d'un transformateur est l'huile ou le pyralene ,

il est compose d'une cuve.

(a) (EST SI (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (HUILE L' * (OU * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R E U J
S T R U C T U R E F A U S S E

N I L

(b) (EST SI (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (HUILE L' * (OU * (PYRALENE LE *))) (, * (EST IL * (COMPOS) (D' * (CUVE UNE *))))
R E U J
S T R U C T U R E F A U S S E

159

N I L

T R O P D E D E P E N D A N T S ?

(c) (EST (SI * (EST (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (HUILE L' * (OU * (PYRALENE LE *))) ,)) IL * (COMPOS * (D' * (CUVE UNE *))))
R E U J
S T R U C T U R E F A U S S E

N I L

(d) (EST SI (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (HUILE L' * (OU * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
C 03
(EST SI (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R 001
(EST SI (REFRIGERANT * (D' * (TRANSFORMATEUR UN *))) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R002
(EST SI (REFRIGERANT * (D' * TRANSFORMATEUR)) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R 023
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R 001
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU HUILE (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R 001
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU HUILE (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R 001
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU HUILE PYRALENE) (, * (EST IL * (COMPOS * (D' * (CUVE UNE *))))))
R002
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU HUILE PYRALENE) (, * (EST IL * (COMPOS * (D' * CUVE))))
)
R041
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU HUILE PYRALENE) (, * (COMPOSITION QQUN CUVE *)))
R008
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG HUILE) PYRALENE) (, * (COMPOSITION QQUN CUVE *)))
R008
(EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG HUILE) (*EG PYRALENE)) (, * (COMPOSITION QQUN CUVE *)))
R 022
(REFRIGERANT SI * TRANSFORMATEUR (OU (*EG HUILE) (*EG PYRALENE)) (, * (COMPOSITION QQUN CUVE *)))
R081
(COND (REFRIGERANT * TRANSFORMATEUR (OU (*EG HUILE) (*EG PYRALENE))) (COMPOSITION QQUN CUVE *)))

(CUID (REFRIGERANT TRANSFORMATEUR (OU (*EG HUILE) (*EG PYRALENE))) (COMPOSITION QQUN CUVE))

Commentaires :

- Les règles REJ1 et REJ3 ont été appliquées pour rejeter les arborescences a, b et c.
- Les règles R022 et R041 ont servi à identifier les formes réduites correspondant aux parties SI et ALORS (cf. § 3).
- Le pronom personnel IL a été remplacé par l'indicateur QQUN. Au moment de résoudre la référence, l'indicateur QQUN sera remplacé par TRANSFORMATEUR.

Exemple 6 : Vérification de la cohérence

Voici un dialogue avec le système pour établir les valeurs possibles de la puissance d'un transformateur :

→ 1 a puissance d'un transformateur est 100 kva.

(PUISSANCE TRANSFORMATEUR (*EG 100))
D 'AC

→ 1 a puissance d'un transformateur est superieur ou egal a 20 kva
? et inferieur ou egal a 2000 kva.

(PUISSANCE TRANSFORMATEUR (ET (*SE 20) (*IE 2000)))

C ES DEUX VALEURS SONT INCOMPATIBLES
(1) (*IE 2000)
(2) (*EG 100)

L AQUELLE VOULEZ-VOUS GARDER?(1 OU 2)

1
D 'AC

→ 1 a puissance d'un transformateur est inferieur a 1500 kva.

(PUISSANCE TRANSFORMATEUR (*PP 1500))

J E REMPLACE
(*IE 2000)
P AR
(*PP 1500)
D 'AC

Commentaires :

- La première phrase fixe à 100 la valeur de la propriété.
- La deuxième phrase propose des valeurs incompatibles avec celles déjà

existantes (100). Le système considère d'abord la valeur 2000 et interroge l'utilisateur, qui décide d'éliminer la valeur 100. Le système garde alors la valeur (≤ 2000) et vérifie sa compatibilité avec la valeur (≥ 20) avant de garder aussi cette dernière.

- La troisième phrase propose des valeurs plus restrictives que (≤ 2000). Pour cela, le système décide d'éliminer (≤ 2000) et garder seulement (< 1500) et (≥ 20). Le RSP obtenu est :

*** DEFINITION DE TRANSFORMATEUR ***

DESCRIPTION DE TRANSFORMATEUR
DANS LE CONTEXTE DE TRANSFORMATEUR

- PROPRIETES:
P U I S S A N C E (*SE 20)
P U I S S A N C E (*PP 1500)

Exemple 7 : Relation n-aire entre propriétés

```
1) (ECHAUFFEMENT_DE_L'HUILE TRANSFORMATEUR (*EG (*FONCT EH (PERTES TRANSFORMATEUR) (SURFACE_DE_RADIATIC
(CUVE (DE TRANSFORMATEUR))))))
D'AC
2) (ECHAUFFEMENT_DE_L'HUILE TRANSFORMATEUR (*IE (*FONCT EH (PERTES TRANSFORMATEUR) (SURFACE_DF_RADIATIC
(CUVE (DE TRANSFORMATEUR))))))
```

CES DEUX VALEURS SONT INCOMPATIBLES

```
(1) (*IE (FONCTION EH (PERTES (TRANSFORMATEUR TRAN7)) (SURFACE_DE_RADIATION (CUVE CUVE39))))
(2) (*EG (FONCTION EH (PERTES (TRANSFORMATEUR TRAN7)) (SURFACE_DE_RADIATION (CUVE CUVE39))))
```

LAQUELLE VOULEZ-VOUS GARDER?(1 OU 2)

2
D'AC

-Commentaire : Le système détecte que la deuxième forme réduite fait référence à la même fonction EH avec les mêmes paramètres. Les relations ≤ et = sont donc jugées incompatibles et l'utilisateur interrogé. Il choisit de garder la deuxième, ce qui produit le RSP :

*** DEFINITION DE TRANSFORMATEUR ***

DESCRIPTION DE TRANSFORMATEUR TRAN7
DANS LE CONTEXTE DE TRANSFORMATEUR

-PROPRIETES:
ECHAUFFEMENT_DE_L'HUILE (*EG (FONCTION EH (SURFACE_DE_RADIATION (CUVE CUVE39)) (PERTES (TRANSFORMATEUR
TRAN7))))

-RELATIONS AVEC D'AUTRES OBJETS:
COMPOSITION ((CUVE CUVE39) (CIRCUIT_ELECTRIQUE CIRC24) (CIRCUIT_MAGNETIQUE CIRC16))

N.B.- TRAN7, CUVE39, CIRC24 et CIRC16 sont des noms internes du système.

4 - TROIS PROBLEMES DE REPRESENTATION DANS LE RSP

Nous introduisons ci-dessous trois problèmes concernant la représentation d'assemblages physiques dans le cadre du modèle proposé au chapitre 2. Nous ne les avons pas résolus au niveau de l'implantation du RSP.

4.1. Relations mutuellement exclusives

Dans la description d'un assemblage, il peut arriver que deux relations entre objets soient mutuellement exclusives. Par exemple, nous pouvons avoir: "soit x est à droite de y, soit x est sur y". Ceci peut être représenté en incluant les relations concernées dans un espace d'un type spécial qui indique qu'elles sont mutuellement exclusives (cf. figure 6.6.).

Cette représentation a été laissée à l'écart dans notre modèle car, dans la plupart des cas, il est possible de faire autrement. En effet, l'existence de relations mutuellement exclusives est en général accompagnée par une indication des situations dans lesquelles chaque relation est applicable. Ceci est exprimé par des expressions conditionnelles (contraintes) comme

SI condition1 ALORS relation1 et

SI condition2 ALORS relation2 ,

où condition1 et condition2 sont telles qu'elles ne sont jamais satisfaites simultanément.⁽¹⁾ Ainsi, le réseau de la figure 6.6. est remplacé par celui de la figure 6.7. ; les arcs R1 et R2 sont contenus dans S1 et S2 respectivement et ne sont pas visibles de Ex.

4.2. Le problème du composant partagé

Dans notre modèle, si un objet a une description particulière lorsqu'il est composant d'un assemblage A, cette description est placée dans le contexte de A. Par exemple, la description particulière d'un bloc d'une colonne est contenue dans le contexte de COLONNE. Ceci permet de distinguer des descriptions particulières d'objets d'une même classe.

(1) De plus, condition1 et condition2 doivent être les seules conditions portant sur relation1 et relation2.

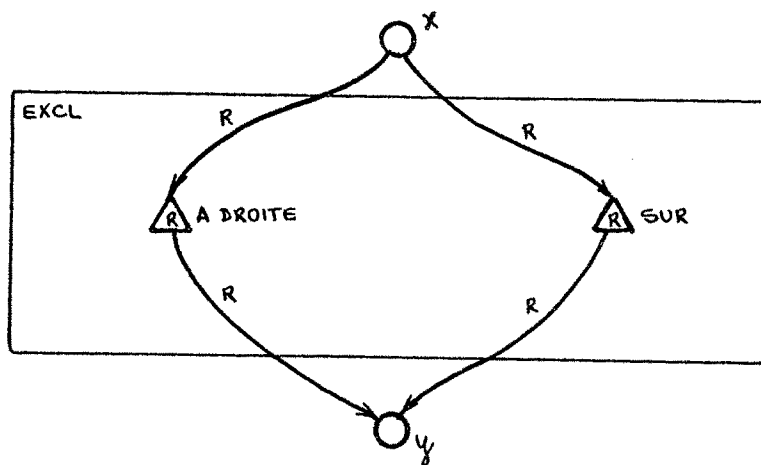


Figure 6.6.

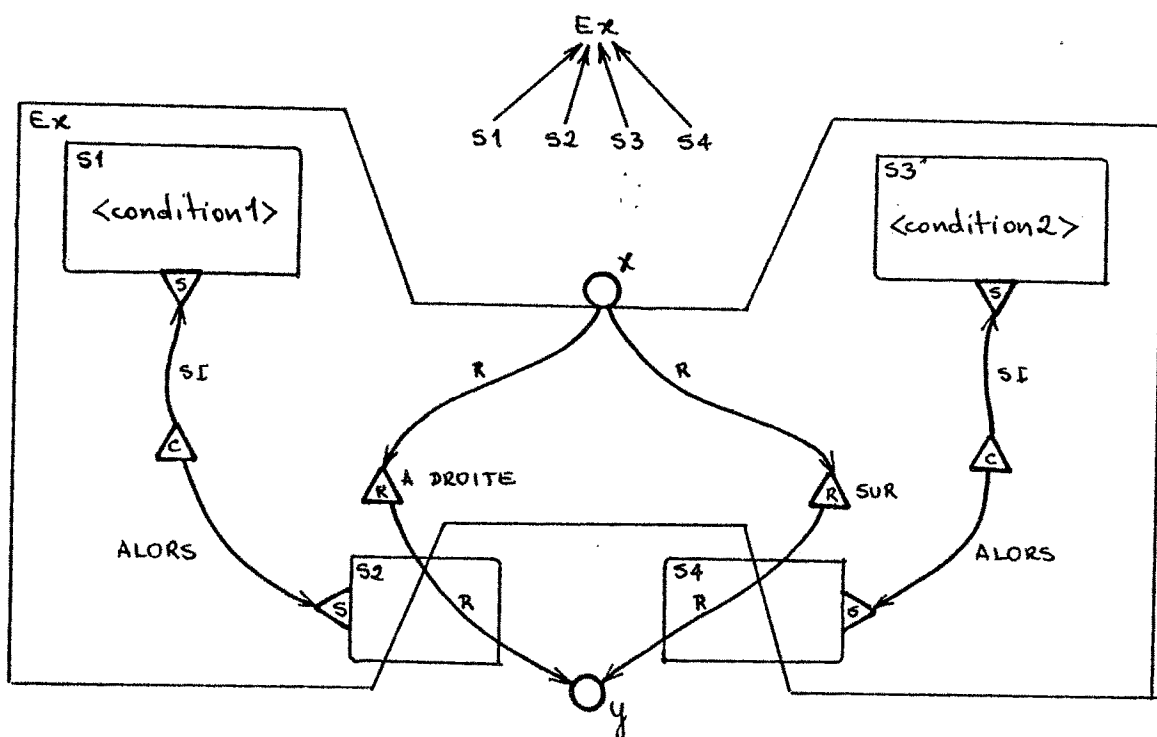


Figure 6.7.

Si nous voulons donner une description particulière à "un bloc d'une colonne d'une tour" il faut placer cette description dans les contextes de COLONNE et de TOUR simultanément pour pouvoir la distinguer de la description "d'un bloc d'un toit d'une tour", dans le cas où le toit d'une tour est lui aussi composé de blocs (cf. figure 6.8.).

Une solution alternative consiste à utiliser des expressions conditionnelles (contraintes) pour délimiter la description et les conditions dans lesquelles elle est valable. Par exemple, le réseau de la figure 6.8. peut être remplacé par celui de la figure 6.9. Cette représentation a l'avantage d'être uniforme mais elle est aussi assez lourde.

4.3. Combinaisons ET/OU de restrictions sur les valeurs des propriétés

Le modèle proposé au § 5.1. du chapitre 2 pour représenter les valeurs possibles d'une propriété d'objet est insuffisant lorsque ces valeurs sont spécifiées par une combinaison ET/OU de restrictions. Par exemple, l'interface ne peut ni traiter, ni représenter, la phrase "l'épaisseur d'une tôle doit être égale à 0,5mm, ou comprise entre 0,7mm et 1,0mm, ou comprise entre 1,2mm et 1,5mm".

Pour généraliser notre modèle, on pourrait réduire le type de phrase précédent à une expression composée d'une disjonction de restrictions, où chaque restriction est donnée

- soit par un élément de la forme ($= v_i$) où v_i est une constante réelle ou entière,
- soit par une conjonction de la forme (ET ($r_1 v_1$)($r_2 v_2$)) où r_1 est une des relations $>$ ou \geq , r_2 est une des relations $<$ ou \leq (chaque conjonction correspond donc à un intervalle).

La phrase donnée en exemple ci-dessus serait ainsi transformée dans la forme réduite suivante :

(EPAISSEUR TOLE(OU ($= 0,5$)) (ET ($\geq 0,7$)($\leq 1,0$))(ET ($\geq 1,2$)($\leq 1,5$))))),

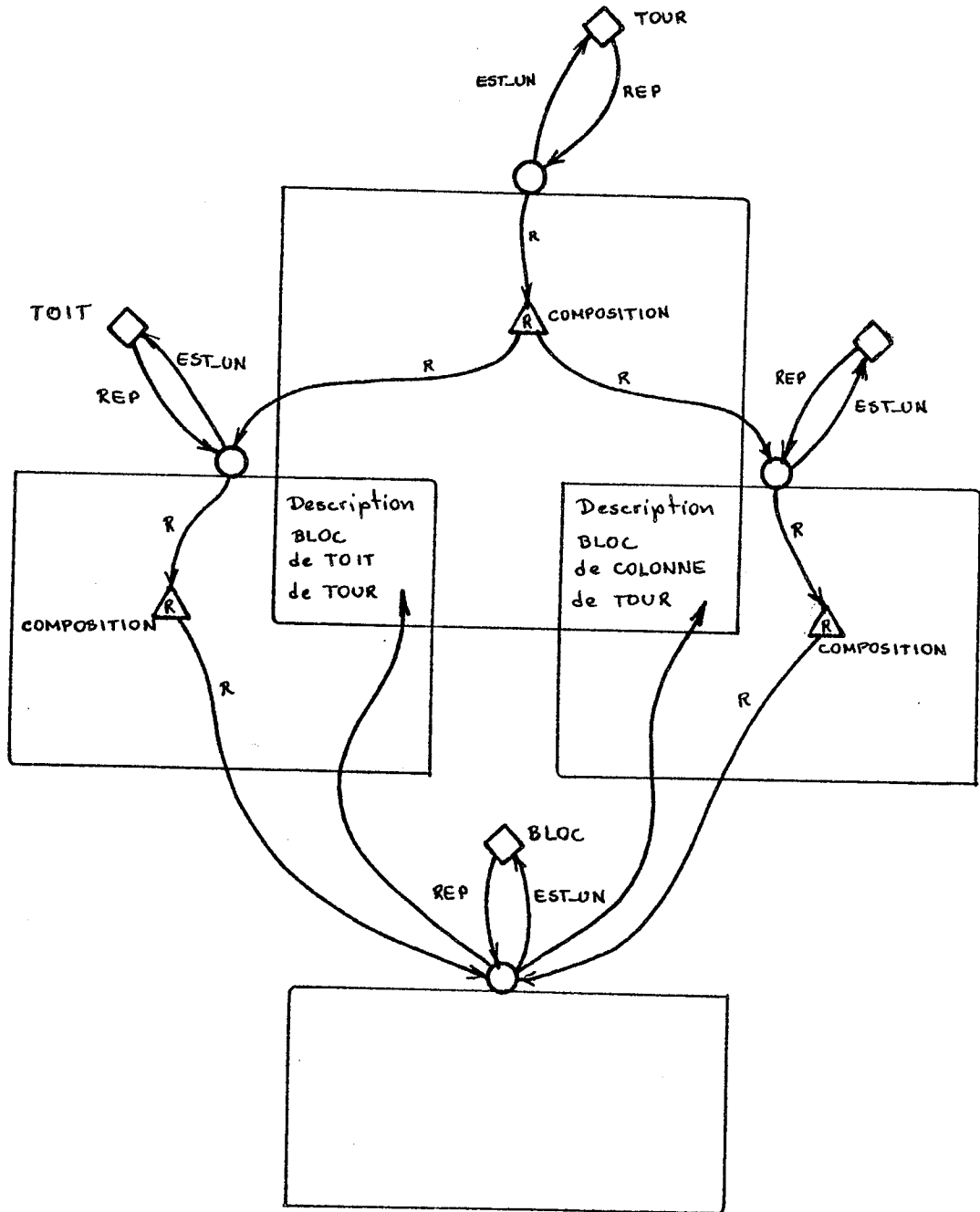


Figure 6.8.

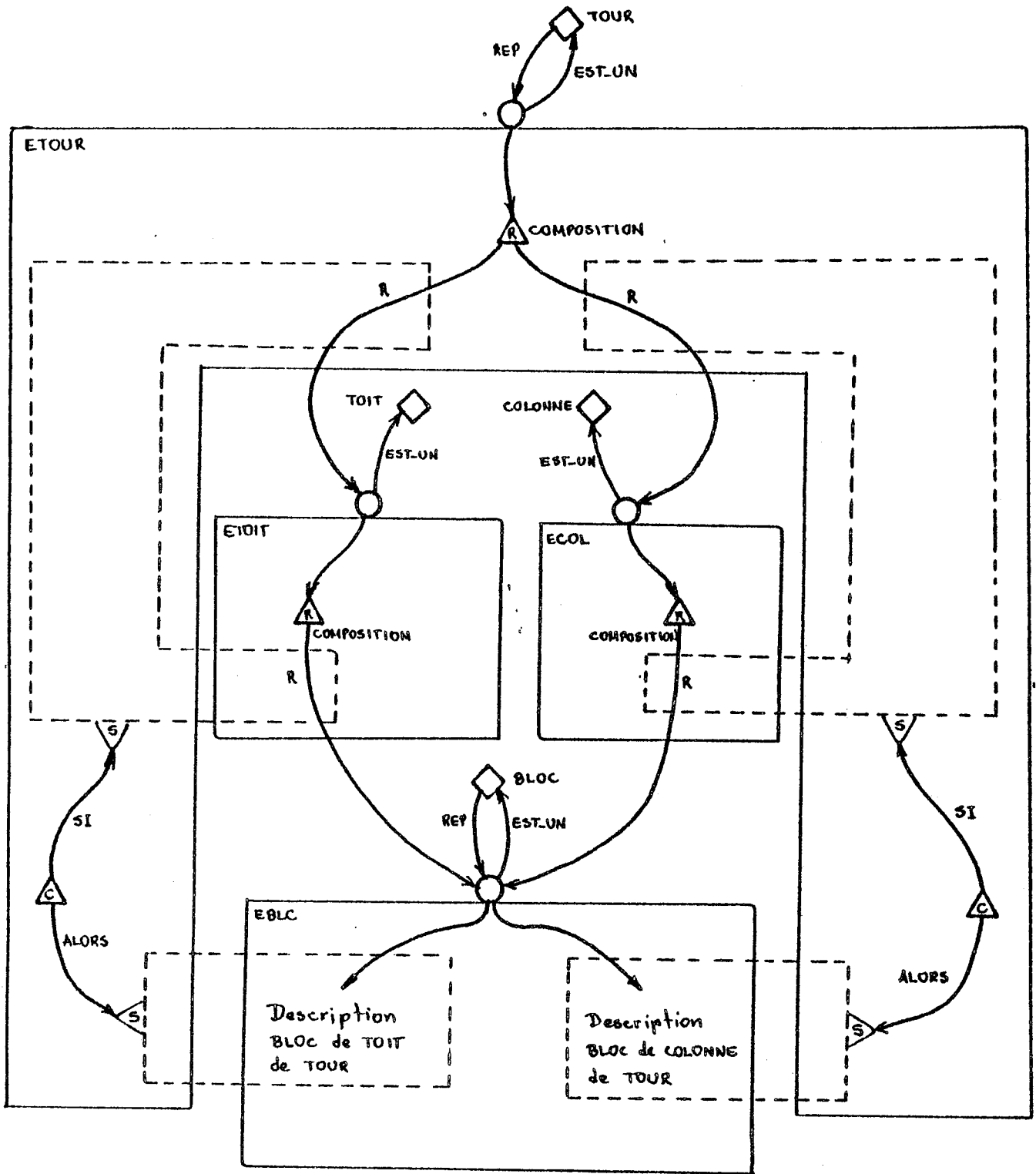


Figure 6.9.

puis représentée par le réseau de la figure 6.10, où

- chaque sommet $\triangle I$ symbolise un intervalle défini par les arcs sortants,
- chaque arc \in indique que la valeur d'une propriété doit être contenue dans un intervalle.

Toutefois, nous avons préféré éviter cette représentation générale afin de simplifier les traitements effectués par les programmes de l'interface.

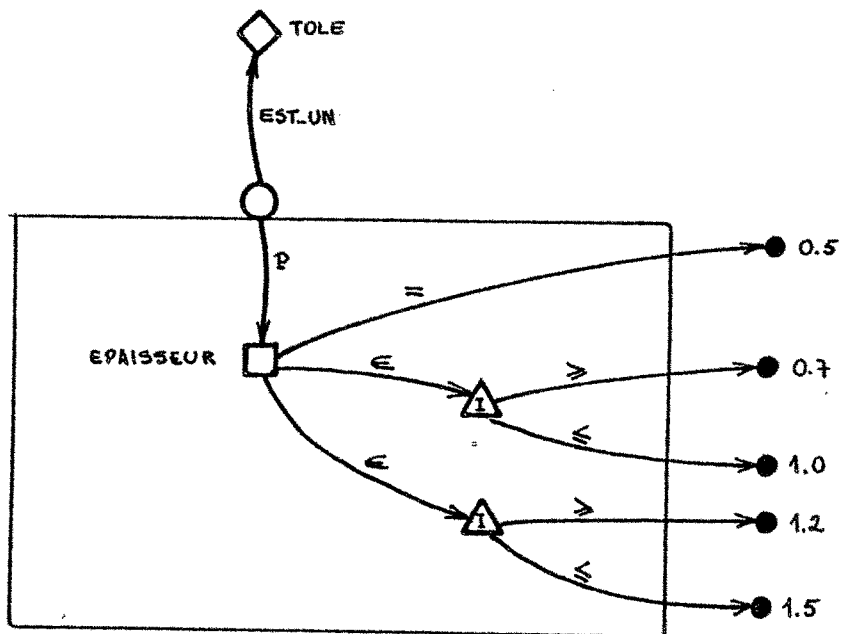


Figure 6.10.

CONCLUSION

Nous avons réalisé un interface de communication en langue naturelle avec le système TROPIC. Cet interface présente quatre caractéristiques principales :

(1) Il intègre des programmes déjà réalisés dans l'équipe d'Intelligence Artificielle de l'IMAG. Ceci nous a permis d'observer et de critiquer leur adaptabilité et leur comportement dans une application comme la nôtre. En outre, la structure modulaire qui en résulte a facilité l'isolement des problèmes pratiques propres à chaque aspect du traitement.

(2) Les connaissances nécessaires à TROPIC sont mémorisées dans un réseau sémantique partitionné qui fournit une structure intermédiaire entre la langue naturelle et la structure de données de TROPIC. Cette représentation permet d'avoir une vue d'ensemble des informations à un instant donné. Cette possibilité pourrait être utilisée par un programme de résolution de problèmes travaillant directement sur le réseau.

(3) Les programmes de l'interface ont été conçus pour être aussi indépendants que possible des connaissances spécifiques du domaine d'application. Ainsi, PIAF utilise un dictionnaire et des règles grammaticales indépendants des programmes, le traitement post-syntaxique est guidé par des règles de transformation définies selon l'application et le constructeur du réseau sémantique se sert de descripteurs sémantiques qui interviennent aussi pendant l'analyse sémantique. L'interface présente donc un certain niveau de généralité permettant de l'adapter à des applications différentes.

(4) Dans le cadre d'une application donnée, la séparation programmes/données permet le développement et la mise au point progressifs de l'interface.

En effet, il est toujours possible d'accroître le nombre de données linguistiques et de règles de transformation, ou de changer leur contenu pour augmenter le nombre de cas traités.

Par ailleurs, nous pensons que le modèle de représentation de connaissances et les méthodes de comparaison de réseaux utilisés par l'interface sont suffisamment généraux pour être étendus à d'autres programmes de CAO, par exemple au traitement des relations géométriques et topologiques dans un système utilisant des techniques graphiques. Une telle application exigerait bien sûr un enrichissement du modèle de représentation qu'il serait intéressant d'étudier.

Finalement, la communication avec TROPIC étant facilitée par l'interface, on peut envisager l'utilisation de ce système par des non spécialistes.

Appendice A : LES REGLES DE TRANSFORMATION

Nous donnons ci-dessous l'ensemble des règles de transformation utilisées pour obtenir les formes réduites des phrases traitées en exemples dans les appendices B et C.

Nous rappelons que chaque règle a quatre parties : partie gauche, partie droite, conditions d'application et effets secondaires.

1 - Règles d'arrêt ou de rejet (leur partie droite est NIL)

- REJ1
 ((&X1 VERB) SI * (&X2 , COCO))
 N IL
 N IL
 N IL
- REJ2
 ((&X1 (VERB "XET "EST)) * (&X2 ADJQ))
 N IL
 N IL
 N IL
- REJ3
 ((&X1 VERB) (SI * ((&X2 VERB) * (&X3 , VIRG))))
 N IL
 N IL
 N IL
- REJ4
 ((&X1 VERB) SI * (ET * ((&X2 VERB) SI * (&X3 ,))))
 N IL
 N IL
 N IL

2 - Règles pour le traitement de la conjonction

- C 01
 ((&&X1 VERB) SI * ((&X2 ET | OU) * ((&&X3 VERB) SI * (&&X4 VIRG))))
 ((?IND &X2 (CVERB)) SI &&X1 * &&X3 &&X4)
 NIL
 N IL
- C 02
 ((&DE DE | D') * (&&X1 SUBC) ((&Y COCO) * ((&DE2 DE | D') * (&&X2 SUBC))
))
 (&DE * ((?IND &Y (SUBC)) &&X1 &&X2))
 N IL
 N IL
- C 03
 ((&&X1 SUBC | ADJC | ADJQ | NUM) * ((&Y ET | OU) * (&&X2 SUBC | ADJC | A
 DJQ | NUM)))
 (&Y &&X1 &&X2)
 (CL (&&X1) = CL (&&X2))
 (PLUSIEURS (CL (&Y) = CL (&&X1)))
- C 04
 ((&&X1 SUBC | ADJC | ADJQ | NUM) * ((&CONJ , COCO) * (&&X2 OU | ET)))
 (&&X2 &&X1)
 N IL
 N IL

3 - Règles générales

- R 001
 ((&X1 SUBC) (&X2 ARTD))
 ((?IND &X1 (DEFN)))
 N IL
 N IL
- R 002
 ((&X1 SUBC) (&X2 UN | UNE))
 ((?IND &X1 (INDEF)))
 N IL
 N IL
- R 003
 ((&&X2 SUBC) (&X1 ADJP))
 (&&X2 * (DE * (?IND QJUN (SUBC))))
 N IL
 N IL

- R 004
 ((&X3 VERB) (&X1 SUBC | POPL) * (E⁷¹ * (&X2 ADJQ | PPAS | RELATION | N
 U⁷¹)))
 ((?IND EST (VERB)) &X1 * &X2)
 N IL
 N IL
- R005
 ((&X1 SUBC UNITES) (&X2 NUM))
 (&X2)
 N IL
 N IL
- R 006
 (COMPR * (ENTRE * (ET &X1 &X2)))
 ((?IND ET (ADJC)) (*SE &X1) (*IE &X2))
 N IL
 N IL
- R 007
 ((&X1 VERB XET) * (&X2 NUM | ADJQ | SUBC))
 (&X1 * ((?IND *EG (ADJC)) * &X2))
 N IL
 N IL
- R008
 ((&X2 OU) (&X1 NUM | ADJQ | SUBC))
 ((?IND &X2 (ADJC)) (*EG &X1))
 N IL
 N IL
- R 009
 ((&X1 SUBC UNITES) (&X2 NUM OU UN UNE ⁷¹))
 (&X1 * (CARD * (*EG * &X2)))
 N IL
 N IL
- R 010
 ((&X1 SUBC) (&X2 OU NUM))
 (&X1 * (CARD * &X2))
 N IL
 N IL
- R 011
 ((&X1 ADJC) * ((&X2 UNITES) &X3))
 (&X1 * &X2)
 N IL
 N IL
- R012
 ((&X1 ATTRIBUT) * (&X2 ADJQ))
 (&X1 * ((?IND *EG (ADJC)) * &X2))
 N IL
 N IL
- R013
 ((&X1 SUBC ATTRIBUT) * (&D PEPC DANS))
 (&X1 * &D)
 N IL
 (PLUSIEURS (NOM (&D) = DE))
- R 014
 ((&X2 VERB) (&X1 SUBC DEFH ATTRIBUT))
 (&X2 (&X1 * (DE * QUUN)))
 N IL
 N IL

4 - Règles pour le traitement des propriétés (attributs)

- R021
 ((EST (&X1 SUBC ~ATTRIBUT) * ((DE DE | D') * ((&X2 SUBC ATTRIBUT) * (&X3 ADJC))))
 ((?IND &X2 (CVERB *FREDUITE)) &X1 &X3)
 N IL
 N IL
- R 022
 ((&X1 VERB) ((&X2 SUBC ATTRIBUT) * (&X4 SUBC)) * (&X3 ADJC))
 ((?IND &X2 (CVERB *FREDUITE)) * &X4 &X3)
 N IL
 N IL
- R 023
 (A (&X1 SUBC) * ((&X2 SUBC ATTRIBUT) * (&X3 ADJC)))
 ((?IND &X2 (CVERB *FREDUITE)) &X1 &X3)
 N IL
 N IL
- R 024
 ((&X3 VERB XET) (&X1 (SUBC | POPL) ~COULEUR) * (*EG * (&X2 VCOULEUR)))

 ((?IND COULEUR (CVERB *FREDUITE)) &X1 (*EG &X2))
 N IL
 (SI (CL (&X1) = POPL) ALORS (PLUSIEURS (NOM (&X1) = QQUN) (CL (&X1) = SUBC)))
- R 025
 ((&V VERB XET) (&X1 (SUBC | POPL) ~COULEUR) * (&X2 (ET | OU) VCOULEUR)
)
 ((?IND COULEUR (CVERB *FREDUITE)) &X1 &X2)
 N IL
 N IL
- R 026
 ((&X3 VERB XET) (&X1 (SUBC | POPL) ~FORME) * (*EG * (&X2 VFORME)))
 ((?IND FORME (CVERB *FREDUITE)) &X1 * (*EG &X2))
 N IL
 (SI (CL (&X1) = POPL) ALORS (PLUSIEURS (NOM (&X1) = QQUN) (CL (&X1) = SUBC)))
- R 027
 ((&V VERB XET) (&X1 (SUBC | POPL) ~FORME) * (&X2 (ET | OU) VFORME))
 ((?IND FORME (CVERB *FREDUITE)) &X1 &X2)
 N IL
 N IL
- R 028
 ((&X1 SUBC ATTRIBUT) * ((DE DE | D' | DU) * (&X2 SUBC)))
 (&X1 * &X2)
 N IL
 N IL
- R029
 ((&X1 SUBC ATTRIBUT) NE * (&X2 SUBC) (&X3 *EG) PAS)
 (&X1 * &X2 &X3)
 N IL
 (PLUSIEURS (NOM (&X3) = *NE))

5 - Règles pour le traitement des relations entre objets

- R041
 (EST (&X1 SUBC) * ((&CO COMPOS PPAS) * ((&DE DE | D') * (&X2 SUBC))))
 ((?IND COMPOSITION (CVERB *FREDUITE)) &X1 &X2)
 N IL
 N IL
- R042
 (EST (COMPOSITION * ((&DE DE | D') * (&X1 SUBC))) * (&X2 SUBC))
 ((?IND COMPOSITION (CVERB *FREDUITE)) &X1 &X2)
 N IL
 N IL
- R 043
 ((&CO COMPOS PRE IND) (&X1 SUBC) * (&X2 SUBC))
 ((?IND COMPOSITION (CVERB *FREDUITE)) &X2 &X1)
 N IL
 N IL
- R 044
 (SONT (COMPOSANT * ((&DE DE | D') * (&X1 SUBC))) * (&X2 SUBC))
 ((?IND COMPOSITION (CVERB *FREDUITE)) &X1 &X2)
 N IL
 N IL
- R 045
 ((&X1 SUBC ~COMP ~ATTRIBUT ~GROUPE) * ((&DE DE | D') * (&X2 SUBC ~QUIN
)))
 ((?IND &X1 (COMP)) * (DE * &X2))
 N IL
 N IL
- R 061
 ((&X1 VERB) (&X4 SUBC) * ((&X2 RELATION) * (&X3 SUBC)))
 ((?IND &X2 (CVERB *FREDUITE)) &X4 * &X3)
 N IL
 N IL
- R062
 ((&X1 VERB XET) * ((&X2 PPAS) * (&X3 PEPC RELATION)))
 (&X1 * &X3)
 N IL
 N IL

6 - Règles pour le traitement des contraintes et des conseils

- R081
 ((&X1 CVERB) S) * ((&VIRG, VIRG) * (&X2 CVERB))
 ((?IND COND (*FREDUITE)) &X1 &X2)
 N IL
 N IL

- R 082
 ((&X1 CVERB) * (SI * (&X2 CVERB)))
 ((?IND CONDU (*FREDUITE)) &X2 &X1)
 N IL
 N IL
- R 102
 ((&X1 CVERB) SI * ((&COMA VIRG) * (CONS * (&X2 CVERB)))
 ((?IND CONS (*FREDUITE)) &X1 &X2)
 NIL
 N IL

7 - Règles pour le traitement des groupes

- R121
 ((&V PPAS COMPOS) * ((&DE DE | D') * (&X1 SUBC PLU)))
 (&V * (DE * (&X1 * (CARD * 1))))
 N IL
 N IL
- R 122
 ((&V VERB PRES IND) (&X1 SUBC) * (&X2 DGROUPE))
 ((?IND DISP (CVERB *FREDUITE)) &X1 (*EG &X2))
 N IL
 N IL
- R 123
 ((&V VERB XET) (&X1 SUBC) * ((&PPAS PPAS) * ((&P PEPC EN) * (&X2 SUBC D
 G ROUPE))))
 ((?IND DISP (CVERB *FREDUITE)) &X1 (*EG &X2))
 N IL
 N IL
- R 124
 ((&X1 SUBC) ((&X2 NUM) * ((&A PEPC A) * (&X3 NUM))))
 (&X1 * (CARD * (ET (*SE &X2) (*IE &X3))))
 N IL
 N IL
- R 125
 ((&X1 SUBC DGROUPE) * ((&DE DE | D') * (&X2 SUBC PLU)))
 (&X2 (CARD (*PG 1)) (DISP (*EG &X1)))
 NIL
 N IL
- R126
 ((&X1 SUBC) (&X2 ADJQ))
 (&X1 (IDENT &X2))
 N IL
 N IL

Appendice B : EXEMPLE DE DESCRIPTION D'UNE TOUR

Pour chaque phrase nous montrons l'arborescence de dépendances délivrée par PIAF, les transformations qu'elle subit et la forme réduite obtenue. Nous ne montrons pas les arborescences rejetées. La deuxième partie de l'appendice montre le RSP construit à partir de quelques unes des phrases données.

1 - Traitement des phrases

une tour est composée d'une colonne et d'un toit.

```
(EST (TOUR UNE *) * (COMPOS * (D' * (COLONNE UNE *) (ET * (D' * (TOIT UN *))))))
• C 02
  (EST (TOUR UNE *) * (COMPOS * (D' * (ET (COLONNE UNE *) (TOIT UN *))))
• R 002
  (EST TOUR * (COMPOS * (D' * (ET (COLONNE UNE *) (TOIT UN *))))
• R002
  (EST TOUR * (COMPOS * (D' * (ET COLONNE (TOIT UN *))))
• R 002
  (EST TOUR * (COMPOS * (D' * (ET COLONNE TOIT))))
• R 041
  (COMPOSITION TOUR (ET COLONNE TOIT) *)

(COMPOSITION TOUR (ET COLONNE TOIT))
```

Le toit d'une tour est sur la colonne de la tour.

- (EST (TOIT LE * (D' * (TOUR UNE *))) * (SUR * (COLONNE LA * (DE * (TOUR LA * *))))
- R001
- (EST (TOIT * (D' * (TOUR UNE *))) * (SUR * (COLONNE LA * (DE * (TOUR LA *)))));
- R002
- (EST (TOIT * (D' * TOUR)) * (SUR * (COLONNE LA * (DE * (TOUR LA *))))))
- R045
- (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE LA * (DE * (TOUR LA *))))))
- R001
- (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * (TOUR LA *))))))
- R001
- (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * TOUR))))
- R 045
- (EST (TOIT * (DE * TOUR)) * (SUR * (COLONNE * (DE * TOUR))))
- R 061
- (SUR (TOIT * (DE * TOUR)) * (COLONNE * (DE * TOUR)))

(SUR (TOIT (DE TOUR)) (COLONNE (DE TOUR)))

une colonne est composee de 1 ou 2 blocs.

- (EST (COLONNE UNE *) * (COMPOS * (DE * (BLOC (1 * (OU * 2)) *))))
- C03
- (EST (COLONNE UNE *) * (COMPOS * (DE * (BLOC (OU 1 2) *))))
- R002
- (EST COLONNE * (COMPOS * (DE * (BLOC (OU 1 2) *))))
- R008
- (EST COLONNE * (COMPOS * (DE * (BLOC (OU (*EG 1) 2) *))))
- R008
- (EST COLONNE * (COMPOS * (DE * (BLOC (OU (*EG 1) (*EG 2)) *))))
- R010
- (EST COLONNE * (COMPOS * (DE * (BLOC * (CARD * (OU (*EG 1) (*EG 2)))))))
- R041
- (COMPOSITION COLONNE (BLOC * (CARD * (OU (*EG 1) (*EG 2)))) *)

(COMPOSITION COLONNE (BLOC (CARD (OU (*EG 1) (*EG 2))))))

la section d'un bloc est carrée ou ronde.

- C03 (EST (SECTION LA * (D¹ * (BLOC UN *))) * (CARRE * (OU * ROND)))
- R001 (EST (SECTION LA * (D¹ * (BLOC UN *))) * (OU CARRE ROND))
- R002 (EST (SECTION * (D¹ * (BLOC UN *))) * (OU CARRE ROND))
- R020 (EST (SECTION * (D¹ * BLOC)) * (OU CARRE ROND))
- R003 (EST (SECTION * BLOC) * (OU CARRE ROND))
- R008 (EST (SECTION * BLOC) * (OU (*EG CARRE) ROND))
- R022 (EST (SECTION * BLOC) * (OU (*EG CARRE) (*EG ROND)))
- (SECTION * BLOC (OU (*EG CARRE) (*EG ROND)))

la couleur d'un bloc est blanche ou verte.

- C03 (EST (COULEUR LA * (D¹ * (BLOC UN *))) * (BLAN * (OU * VERT)))
- R001 (EST (COULEUR LA * (D¹ * (BLOC UN *))) * (OU BLAN VERT))
- R002 (EST (COULEUR * (D¹ * (BLOC UN *))) * (OU BLAN VERT))
- R023 (EST (COULEUR * (D¹ * BLOC)) * (OU BLAN VERT))
- R006 (EST (COULEUR * BLOC) * (OU BLAN VERT))
- R008 (EST (COULEUR * BLOC) * (OU (*EG BLAN) VERT))
- R022 (EST (COULEUR * BLOC) * (OU (*EG BLAN) (*EG VERT)))
- (COULEUR * BLOC (OU (*EG BLAN) (*EG VERT)))
- (COULEUR BLOC (OU (*EG BLAN) (*EG VERT)))

un toit est rouge ou noir.

- (EST (TOIT UN *) * (ROUG * (OU * NOIR)))
- C 03
- (EST (TOIT UN *) * (OU ROUG NOIR))
- R002
- (EST TOIT * (OU ROUG NOIR))
- R 008
- (EST TOIT * (OU (*EG ROUG) NOIR))
- R 008
- (EST TOIT * (OU (*EG ROUG) (*EG NOIR)))
- R 025
- (COULEUR TOIT (OU (*EG ROUG) (*EG NOIR)) *)

(COULEUR TOIT (OU (*EG ROUG) (*EG NOIR)))

un toit est prismatique ou pyramidal.

- (EST (TOIT UN *) * (PRISMATIQUE * (OU * PYRAMID)))
- C03
- (EST (TOIT UN *) * (OU PRISMATIQUE PYRAMID))
- R002
- (EST TOIT * (OU PRISMATIQUE PYRAMID))
- R003
- (EST TOIT * (OU (*EG PRISMATIQUE) PYRAMID))
- R006
- (EST TOIT * (OU (*EG PRISMATIQUE) (*EG PYRAMID)))
- R027
- (FORME TOIT (OU (*EG PRISMATIQUE) (*EG PYRAMID)) *)

(FORME TOIT (OU (*EG PRISMATIQUE) (*EG PYRAMID)))

si une colonne est composee de 2 blocs , les blocs sont blancs.

```

(EST SI ( COLOMNE UHE *) * (COMPOS * (DE * (BLOC 2 *))) (, * (SONT (BLOC LES *) * BLAN)))
. R 002
(EST SI COLOMNE * (COMPOS * (DE * (BLOC 2 *))) (, * (SONT (BLOC LES *) * BLAN)))
. R 003
(EST SI COLOMNE * (COMPOS * (DE * (BLOC * (CARD * (*EG * 2)))) (, * (SONT (BLOC LES *) * BLAN)))
. R 041
(COMPOSITION SI COLOMNE (BLOC * (CARD * (*EG * 2))) * (, * (SONT (BLOC LES *) * BLAN)))
. R 001
(COMPOSITION SI COLOMNE (BLOC * (CARD * (*EG * 2))) * (, * (SONT BLOC * BLAN)))
. R 007
(COMPOSITION SI COLOMNE (BLOC * (CARD * (*EG * 2))) * (, * (SONT BLOC * (*EG * BLAN))))
. R 024
(COMPOSITION SI COLOMNE (BLOC * (CARD * (*EG * 2))) * (, * (SONT BLOC * (*EG * BLAN))))
. R 081
(COUD ( COMPOSITION COLOMNE (BLOC * (CARD * (*EG * 2))) * (COULEUR BLOC (*EG BLAN) *)))
(COUD ( COMPOSITION COLOMNE (BLOC * (CARD * (*EG * 2))) * (COULEUR BLOC (*EG BLAN) *)))
(COUD ( COMPOSITION COLOMNE (BLOC (CARD (*EG 2)))) (COULEUR BLOC (*EG BLAN)))

```

si la section d'un bloc est carree , il est vert.

```

(EST SI (SECTION LA * (D' * (BLOC UN *))) * CARRE (, * (EST IL * VERT)))
. R001
(EST SI (SECTION * (D' * (BLOC UN *))) * CARRE (, * (EST IL * VERT)))
. R002
(EST SI (SECTION * (D' * BLOC)) * CARRE (, * (EST IL * VERT)))
. R028
(EST SI (SECTION * BLOC) * CARRE (, * (EST IL * VERT)))
. R007
(EST SI (SECTION * BLOC) * (*EG * CARRE) (, * (EST IL * VERT)))
. R007
(EST SI (SECTION * BLOC) * (*EG * CARRE) (, * (EST IL * (*EG * VERT))))

```

```

R022 (SECTION SI * BLOC (*EG * CARRE) (, * (EST IL * (*EG * VERT))))
R024
(SSECTION SI * BLOC (*EG * CARRE) (, * (COULEUR QQUN (*EG VERT *)))
R061 (COND (SECTION * BLOC (*EG * CARRE)) (COULEUR QQUN (*EG VERT *)))

(COND (SECTION BLOC (*EG CARRE)) (COULEUR QQUN (*EG VERT)))

si un toit est prismatique , sa hauteur est 2 metres.

(EST SI (TOIT UN *) * PRISMATIQUE (, * (EST (HAUTEUR SA *) * (METRE 2 *))))
R002
(EST SI TOIT * PRISMATIQUE (, * (EST (HAUTEUR SA *) * (METRE 2 *))))
R007
(EST SI TOIT * (*EG * PRISMATIQUE) (, * (EST (HAUTEUR SA *) * (METRE 2 *))))
R005
(EST SI TOIT * (*EG * PRISMATIQUE) (, * (EST (HAUTEUR * (DE * QQUN)) * (METRE 2 *))))
R005
(EST SI TOIT * (*EG * PRISMATIQUE) (, * (EST (HAUTEUR * (DE * QQUN)) * 2)))
R026
(FORME SI TOIT * (*EG PRISMATIQUE) (, * (EST (HAUTEUR * (DE * QQUN)) * 2)))
R028
(FORME SI TOIT * (*EG PRISMATIQUE) (, * (EST (HAUTEUR * QQUN) * 2)))
R007
(FORME SI TOIT * (*EG PRISMATIQUE) (, * (EST (HAUTEUR * QQUN) * (*EG * 2))))
R022
(FORME SI TOIT * (*EG PRISMATIQUE) (, * (HAUTEUR * QQUN (*EG * 2))))
R061 (COND (FORME TOIT * (*EG PRISMATIQUE)) (HAUTEUR * QQUN (*EG * 2)))

(COND (FORME TOIT (*EG PRISMATIQUE)) (HAUTEUR QQUN (*EG 2)))

```

si la forme d'un toit ne est pas prismatique , sa hauteur est 3 metres.

- R001 (EST SI (FORME LA * (D' * (TOIT UN *))) NE * PAS PRISMATIQUE (, * (EST (HAUTEUR SA *) * (METRE 3 *)))
- R002 (EST SI (FORME * (D' * (TOIT UN *))) NE * PAS PRISMATIQUE (, * (EST (HAUTEUR SA *) * (METRE 3 *)))
- R 023 (EST SI (FORME * (D' * TOIT)) NE * PAS PRISMATIQUE (, * (EST (HAUTEUR SA *) * (METRE 3 *)))
- R007 (EST SI (FORME * TOIT) NE * PAS PRISMATIQUE (, * (EST (HAUTEUR SA *) * (METRE 3 *)))
- R005 (EST SI (FORME * TOIT) NE * PAS (, * (EST (HAUTEUR SA *) * (METRE 3 *))) * (*EG * PRISMATIQUE))
- R005 (EST SI (FORME * TOIT) NE * PAS (, * (EST (HAUTEUR * (DE * QQUN)) * (METRE 3 *))) * (*EG * PRISMATIQUE))
- R 005 (EST SI (FORME * TOIT) NE * PAS (, * (EST (HAUTEUR * (DE * QQUN)) * 3)) * (*EG * PRISMATIQUE))
- R028 (EST SI (FORME * TOIT) NE * PAS (, * (EST (HAUTEUR * QQUN)) * 3)) * (*EG * PRISMATIQUE))
- R007 (EST SI (FORME * TOIT) NE * PAS (, * (EST (HAUTEUR * QQUN)) * (*EG * 3))) * (*EG * PRISMATIQUE))
- R022 (FORME SI NE * TOIT (*EG * PRISMATIQUE) PAS (, * (EST (HAUTEUR * QQUN) * (*EG * 3))) *)
- R029 (FORME SI * TOIT (*NE * PRISMATIQUE) (, * (EST (HAUTEUR * QQUN) * (*EG * 3))) *)
- R022 (FORME SI * TOIT (*NE * PRISMATIQUE) (, * (HAUTEUR * QQUN (*EG * 3))) *)
- R081 (COND (FORME * TOIT (*NE * PRISMATIQUE) *) (HAUTEUR * QQUN (*EG * 3)))
- (COND (FORME TOIT (*NE PRISMATIQUE)) (HAUTEUR QQUN (*EG 3)))

si la section d'un bloc est ronde, sa hauteur est supérieur ou égal a 4 metres

et inférieur ou égal a 8 metres.

- (EST SI (SECTION LA * (D' * (BLOC UN *))) * ROND (, * (EST (HAUTEUR SA *)) * (*SE * (METRE 4 *)) (ET * (*IE * (METRE 3 *))))))
- C 05
- (EST SI (SECTION LA * (D' * (BLOC UN *))) * ROND (, * (EST (HAUTEUR SA *)) * (ET (*SE * (METRE 4 *)))
- R001
- (EST SI (SECTION * (D' * (BLOC UN *))) * ROND (, * (EST (HAUTEUR SA *)) * (ET (*SE * (METRE 4 *))) (*IE * (METRE 3 *))))))
- R 002
- (EST SI (SECTION * (D' * BLOC)) * ROND (, * (EST (HAUTEUR SA *)) * (ET (*SE * (METRE 4 *))) (*IE * (METRE 3 *))))))
- R028
- (EST SI (SECTION * BLOC) * ROND (, * (EST (HAUTEUR SA *)) * (ET (*SE * (METRE 4 *))) (*IE * (METRE 2 *))))))
- R007
- (EST SI (SECTION * BLOC) * (*EG * ROND) (, * (EST (HAUTEUR SA *)) * (ET (*SE * (METRE 4 *))) (*IE * (METRE 8 *))))))
- R003
- (EST SI (SECTION * BLOC) * (*EG * ROND) (, * (EST (HAUTEUR * (DE * QQUN)) * (ET (*SE * (METRE 4 *))) (*IE * (METRE 8 *))))))
- R 005
- (EST SI (SECTION * BLOC) * (*EG * ROND) (, * (EST (HAUTEUR * (DE * QQUN)) * (ET (*SE * 4) (*IE * (METRE 3 *))))))
- R005
- (EST SI (SECTION * BLOC) * (*EG * ROND) (, * (EST (HAUTEUR * (DE * QQUN)) * (ET (*SE * 4) (*IE * 8))))))
- R022
- (SECTION SI * BLOC (*EG * ROND) (, * (EST (HAUTEUR * (DE * QQUN)) * (ET (*SE * 4) (*IE * 8))))))
- R 028
- (SECTION SI * BLOC (*EG * ROND) (, * (EST (HAUTEUR * QQUN) * (ET (*SE * 4) (*IE * 3))))))
- R 022
- (SECTION SI * BLOC (*EG * ROND) (, * (HAUTEUR * QQUN (ET (*SE * 4) (*IE * 3))))))
- R081
- (COND (SECTION * BLOC (*EG * ROND)) (HAUTEUR * QQUN (ET (*SE * 4) (*IE * 8))))
- (COND (SECTION BLOC (*EG ROND)) (HAUTEUR QQUN (ET (*SE 4) (*IE 8))))

si une colonne est composee de 2 blocs ,
le premier bloc est situe sur le deuxieme bloc.

(EST SI (COLONNE UNE *) * (COMPOS * (DE * (BLOC 2 *))) (, * (EST (BLOC LE PREMIER *) * (SITU * (SUR * (BLOC LE DEUXIEM *))))))
 • R002
 (EST SI COLONNE * (COMPOS * (DE * (BLOC 2 *))) (, * (EST (BLOC LE PREMIER *) * (SITU * (SUR * (BLOC LE DEUXIEM *))))))
 • R009
 (EST SI COLONNE * (COMPOS * (DE * (BLOC * (CARD * (*EG * 2)))) (, * (EST (BLOC LE PREMIER *) * (SITU * (SUR * (BLOC LE DEUXIEM *))))))
 • R041
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (EST (BLOC LE PREMIER *) * (SITU * (SUR * (BLOC LE DEUXIEM *))))))
 • R041
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (EST (BLOC PREMIER *) * (SITU * (SUR * (BLOC LE DEUXIEM *))))))
 • R126
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (EST (BLOC (IDENT PREMIER) *) * (SITU * (SUR * (BLOC LE DEUXIEM *))))))
 • R001
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (EST (BLOC (IDENT PREMIER) *) * (SITU * (SUR * (BLOC DEUXIEM *))))))
 • R126
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (EST (BLOC (IDENT PREMIER) *) * (SITU * (SUR * (BLOC (IDENT DEUXIEM *))))))
 • R062
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (EST (BLOC (IDENT PREMIER) *) * (SUR * (BLOC (IDENT DEUXIEM *))))))
 • R061
 (COMPOSITION SI COLONNE (BLOC * (CARD * (*EG * 2))) * (, * (SUR (BLOC (IDENT PREMIER) *) * (BLOC (IDENT DEUXIEM *))))))
 • R081
 (COND (COMPOSITION COLONNE (BLOC * (CARD * (*EG * 2))) *) (SUR (BLOC (IDENT PREMIER) *) * (BLOC (IDENT DEUXIEM *)))
 (COND (COMPOSITION COLONNE (BLOC (CARD (*EG 2)))) (SUR (BLOC (IDENT PREMIER) (BLOC (IDENT DEUXIEM))

2. Le RSP construit à partir des phrases

Nous montrons ci-dessous l'image du RSP telle qu'elle est imprimée par le système. La figure B.1. montre partiellement son équivalent graphique.

*** DEFINITION DE TOUR ***

DESCRIPTION DE TOUR
DANS LE CONTEXTE DE TOUR

- RELATIONS AVEC D'AUTRES OBJETS:
COMPOSITION (TOIT COLONNE)

DESCRIPTION DE TOIT
DANS LE CONTEXTE DE TOUR

- RELATIONS AVEC D'AUTRES OBJETS:
SUR (COLONNE)

*** DEFINITION DE TOIT ***

DESCRIPTION DE TOIT
DANS LE CONTEXTE DE TOIT

- PROPRIETES:

FORME (*EG PYRAMID)

FORME (*EG PRISMATIQUE)

C OULEUR (*EG NOIR)

C OULEUR (*EG ROUG)

- CONTRAINTES:

SI

(FORME TOIT (*EG PRISMATIQUE))

A LORS

(HAUTEUR TOIT (*EG 2))

SI

(FORME TOIT (*NE PRISMATIQUE))

A LORS

(HAUTEUR TOIT (*EG 3))

DESCRIPTION DE TOIT
DANS LE CONTEXTE DE TOUR

- RELATIONS AVEC D'AUTRES OBJETS:

SUR (COLONNE)

*** DEFINITION DE COLONNE ***

DESCRIPTION DE COLONNE
DANS LE CONTEXTE DE COLONNE

- RELATIONS AVEC D'AUTRES OBJETS:
COMPOSITION ((UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44))

- CONTRAINTES:

SI
(ET (CARD (UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44) (*EG 2))
(COMPOSITION COLO11 (UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44)))
A LORS
(COULEUR (UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44) (*EG BLAN))

SI
(ET (CARD (UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44) (*EG 2))
(COMPOSITION COLO11 (UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44)))
A LORS
(DISP (UN GROUPE DE ((*EG 2) (*EG 1)) BLOC BLOC44) (*EG PILE))

DESCRIPTION DE BLOC
DANS LE CONTEXTE DE COLONNE

- PROPRIETES:
CARD (*EG 2)
CARD (*EG 1)

*** ** DEFINITION DE BLOC *****

**D ESCRIPTION DE BLOC
D ANS LE CONTEXTE DE BLOC**

- PROPRIETES:
S ECTION (*EG ROND)
S ECTION (*EG CARRE)
C OULEUR (*EG VERT)
C OULEUR (*EG BLAN)

- CONTRAINTES:

SI
(SECTION BLOC (*EG ROND))
A LORS
(ET (HAUTEUR BLOC (*IE 8)) (HAUTEUR BLOC (*SE 4)))

SI
(SECTION BLOC (*NE ROND))
A LORS
(ET (HAUTEUR BLOC (*IE 9)) (HAUTEUR BLOC (*SE 5)))

SI
(SECTION BLOC (*EG CARRE))
A LORS
(COULEUR B L OC (*EG VERT))

**D ESCRIPTION DE BLOC
D ANS LE CONTEXTE DE COLONNE**

- PROPRIETES:
C ARD (*EG 2)
C A RD (*EG 1)

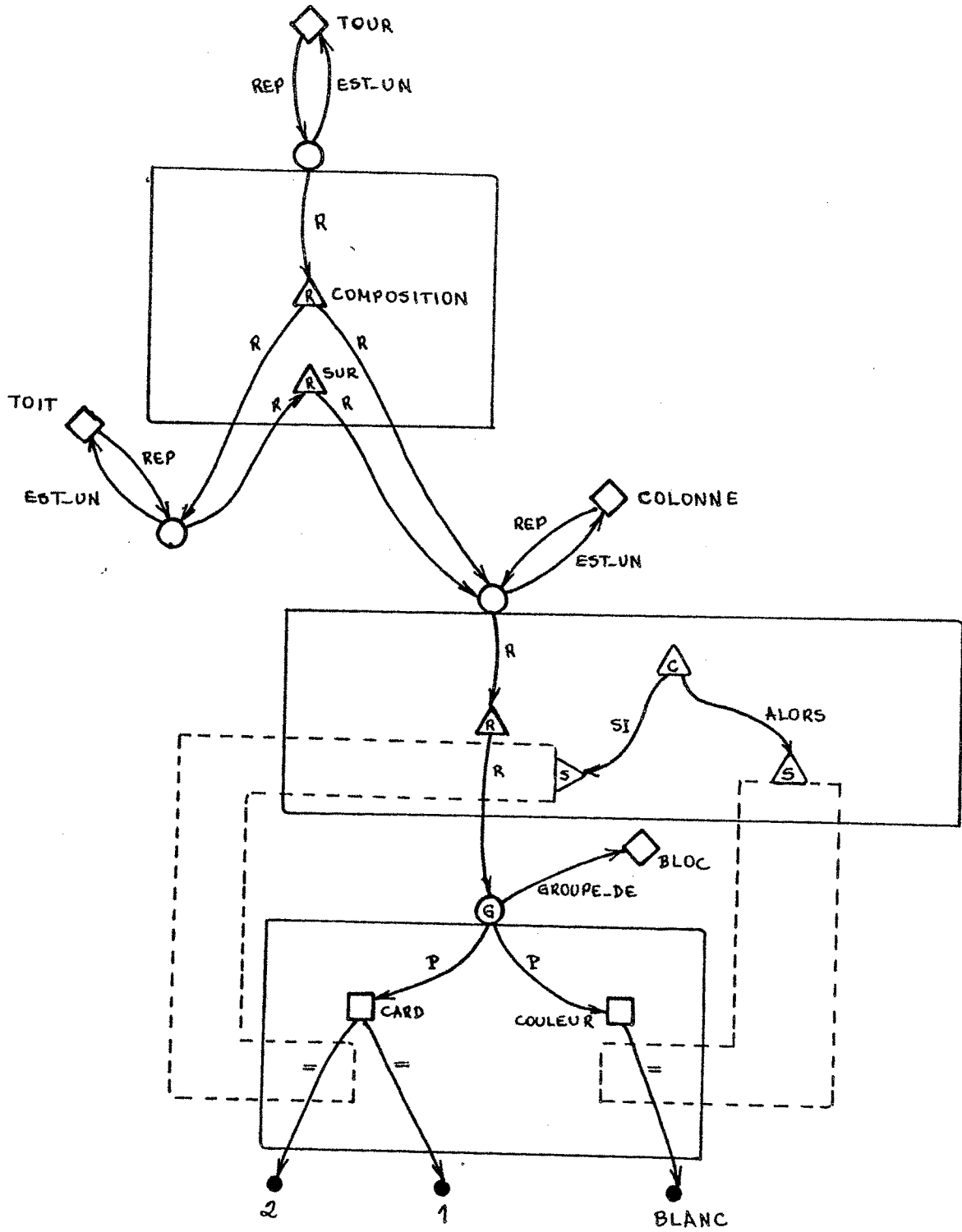


Figure B.1. - Une partie du RSP

Appendice C : EXEMPLE DE DESCRIPTION D'UN TRANSFORMATEUR

Pour chaque phrase nous montrons l'arborescence de dépendances délivrée par PIAF, les transformations qu'elle subit et la forme réduite obtenue. Nous ne montrons pas les arborescences rejetées. Les mutilations des mots sont dues à une limitation sur leur longueur dans la version de PIAF dont nous disposions. La mutilation des nombres est due à une erreur dans l'interface PIAF-LISP.

La deuxième partie de l'appendice montre le RSP construit à partir de quelques unes des phrases données.

1 - Traitement des phrases

La puissance d'un transformateur est supérieure à 20 kva
?

et inférieur à 2000 kva.

- C03 (EST (PUISSANCE LA * (D' * (TRANSFORMATEUR UN *))) * (*PG * (KVA 2 *)) (ET * (*PP * (KVA 2 *))))
- R001 (EST (PUISSANCE LA * (D' * (TRANSFORMATEUR UN *))) * (ET (*PG * (KVA 2 *))) (*PP * (KVA 2 *))))
- R002 (EST (PUISSANCE * (D' * (TRANSFORMATEUR UN *))) * (ET (*PG * (KVA 2 *))) (*PP * (KVA 2 *))))
- R026 (EST (PUISSANCE * (D' * TRANSFORMATEUR)) * (ET (*PG * (KVA 2 *))) (*PP * (KVA 2 *))))
- R 005 (EST (PUISSANCE * TRANSFORMATEUR) * (ET (*PG * (KVA 2 *))) (*PP * (KVA 2 *))))
- R005 (EST (PUISSANCE * T RANSFORMATEUR) * (ET (*PG * 2)) (*PP * (KVA 2 *))))
- R022 (EST (PUISSANCE * TRANSFORMATEUR) * (ET (*PG * 2)) (*PP * 2))
- (PUISSANCE * TRANSFORMATEUR (ET (*PG * 2)) (*PP * 2))

(PUISSANCE T RANSFORMATEUR (ET (*PG 2) (*PP 2)))

le réfrigérant d'un transformateur est l'air , l'huile ou le pyralène.

- C03 (EST (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (AIR L' * C, * (HUILE L' * (OU * (PYRALENE LE *))))))
- C 04 (EST (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (AIR L' * C, * (OU (HUILE L' * (PYRALENE LE *))))
- (EST (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (OU (AIR L' * (HUILE L' * (PYRALENE LE *))))

• R001 (EST (REFRIGERANT * (D' * (TRANSFORMATEUR UN *))) * (OU (AIR L' *) (HUILE L' *) (PYRALENE LE *)))
 • R002 (EST (REFRIGERANT * (D' * TRANSFORMATEUR)) * (OU (AIR L' *) (HUILE L' *) (PYRALENE LE *)))
 • R020 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU (AIR L' *) (HUILE L' *) (PYRALENE LE *)))
 • R 001 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU AIR (HUILE L' *) (PYRALENE LE *)))
 • R 001 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU AIR HUILE (PYRALENE LE *)))
 • R001 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU AIR HUILE PYRALENE))
 • R008 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG AIR) HUILE PYRALENE))
 • R006 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG AIR) (*EG HUILE) PYRALENE))
 • R008 (EST (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG AIR) (*EG HUILE) (*EG PYRALENE)))
 • R022 (REFRIGERANT * TRANSFORMATEUR (OU (*EG AIR) (*EG HUILE) (*EG PYRALENE)))

 (REFRIGERANT TRANSFORMATEUR (OU (*EG AIR) (*EG HUILE) (*EG PYRALENE)))

un transformateur est compose d'un circuit electrique et

d'un circuit magnetique.

- (EST (TRANSFORMATEUR UN *) * (COMPOS * (D' * (CIRCUIT ELECTRIQUE UN *) (ET * (D' * (CIRCUIT MAGNETIQUE UN *))))))
- C 02
- (EST (TRANSFORMATEUR UN *) * (COMPOS * (D' * (ET (CIRCUIT ELECTRIQUE UN *) (CIRCUIT MAGNETIQUE UN *))))))
- R 002
- (EST TRANSFORMATEUR * (COMPOS * (D' * (ET (CIRCUIT ELECTRIQUE UN *) (CIRCUIT MAGNETIQUE UN *))))))
- R002
- (EST TRANSFORMATEUR * (COMPOS * (D' * (ET CIRCUIT ELECTRIQUE (CIRCUIT MAGNETIQUE UN *))))))
- R002
- (EST TRANSFORMATEUR * (COMPOS * (D' * (ET CIRCUIT ELECTRIQUE CIRCUIT MAGNETIQUE))))
- R041
- (COMPOSITION TRANSFORMATEUR (ET CIRCUIT ELECTRIQUE CIRCUIT MAGNETIQUE) *)

(COMPOSITION TRANSFORMATEUR (ET CIRCUIT ELECTRIQUE CIRCUIT MAGNETIQUE))

si le réfrigérant d'un transformateur est l'huile ou le pyralène,

il est classé d'une cuve.

- (EST SI (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (HUILE L' * (OU * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- C 03
- (EST SI (REFRIGERANT LE * (D' * (TRANSFORMATEUR UN *))) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- R 001
- (EST SI (REFRIGERANT * (D' * (TRANSFORMATEUR UN *))) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- R002
- (EST SI (REFRIGERANT * (D' * (TRANSFORMATEUR)) * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- R 020
- (EST SI (REFRIGERANT * TRANSFORMATEUR * (OU (HUILE L' * (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- R 001
- (EST SI (REFRIGERANT * TRANSFORMATEUR * (OU HUILE (PYRALENE LE *))) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- R 002
- (EST SI (REFRIGERANT * TRANSFORMATEUR * (OU HUILE PYRALENE) (, * (EST IL * (COMPOS * (D' * (CUVE UME *))))))
- R041
- (EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU HUILE PYRALENE) (, * (COMPOSITION QQUIN CUVE *)))
- R008
- (EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG HUILE) PYRALENE) (, * (COMPOSITION QQUIN CUVE *)))
- R008
- (EST SI (REFRIGERANT * TRANSFORMATEUR) * (OU (*EG HUILE) (*EG PYRALENE)) (, * (COMPOSITION QQUIN CUVE *)))
- R 022
- (REFRIGERANT SI * TRANSFORMATEUR (OU (*EG HUILE) (*EG PYRALENE)) (, * (COMPOSITION QQUIN CUVE *)))
- R001
- (COND (REFRIGERANT * TRANSFORMATEUR (OU (*EG HUILE) (*EG PYRALENE))) (COMPOSITION QQUIN CUVE *)))
- (COND (REFRIGERANT TRANSFORMATEUR (OU (*EG HUILE) (*EG PYRALENE))) (COMPOSITION QQUIN CUVE))

La qualite des toles J'un circuit magnetique est q1 ou q2.

- C 05 (EST (QUALITE DES TOLES LA * (D' * (CIRCUIT MAGNETIQUE UN *))) * (Q1 * (OU * Q2)))
- R 001 (EST (QUALITE DES TOLES LA * (D' * (CIRCUIT MAGNETIQUE UN *))) * (OU Q1 Q2))
- R 002 (EST (QUALITE DES TOLES * (D' * (CIRCUIT MAGNETIQUE UN *))) * (OU Q1 Q2))
- R 026 (EST (QUALITE DES TOLES * (D' * CIRCUIT MAGNETIQUE)) * (OU Q1 Q2))
- R 008 (EST (QUALITE DES TOLES * CIRCUIT MAGNETIQUE) * (OU Q1 Q2))
- R 006 (EST (QUALITE DES TOLES * CIRCUIT MAGNETIQUE) * (OU (*EG Q1) Q2))
- R 022 (EST (QUALITE DES TOLES * CIRCUIT MAGNETIQUE) * (OU (*EG Q1) (*EG Q2)))
- (QUALITE DES TOLES * CIRCUIT MAGNETIQUE (OU (*EG Q1) (*EG Q2)))

(QUALITE DES TOLES CIRCUIT MAGNETIQUE (OU (*EG Q1) (*EG Q2)))

un circuit magnetique est compose de 3 noyaux et de 2 culasses.

- C 02 (EST (CIRCUIT MAGNETIQUE UN * * (COMPOS * (DE * (NOYAU 3 *)) (ET * (DE * (CULASSE 2 *))))))
- R 002 (EST (CIRCUIT MAGNETIQUE UN * * (COMPOS * (DE * (ET (NOYAU 3 *)) (CULASSE 2 *))))))
- R 009 (EST CIRCUIT MAGNETIQUE * (COMPOS * (DE * (ET (NOYAU 3 *)) (CULASSE 2 *))))))
- R 009 (EST CIRCUIT MAGNETIQUE * (COMPOS * (DE * (ET (NOYAU * (CARD * (*EG * 3))) (CULASSE 2 *))))))
- R 041 (EST CIRCUIT MAGNETIQUE * (COMPOS * (DE * (ET (NOYAU * (CARD * (*EG * 3))) (CULASSE * (CARD * (*EG * 2))))))
- (COMPOSITION: CIRCUIT MAGNETIQUE (ET (NOYAU * (CARD * (*EG * 3))) (CULASSE * (CARD * (*EG * 2)))) *)
- (COMPOSITION CIRCUIT MAGNETIQUE (ET (NOYAU (CARD (*EG 3))) (CULASSE (CARD (*EG 2))))))

Le montage primaire d'un circuit électrique est triangle ou étoile.

- (EST (MONTAGE PRIMAIRE LE * (D' * (CIRCUIT ELECTRIQUE UN *))) * (TRIANGLE * (OU * ETOILE)))
- C 03
- (EST (MONTAGE PRIMAIRE LE * (D' * (CIRCUIT ELECTRIQUE UN *))) * (OU TRIANGLE ETOILE))
- R 001
- (EST (MONTAGE PRIMAIRE * (D' * (CIRCUIT ELECTRIQUE UN *))) * (OU TRIANGLE ETOILE))
- R002
- (EST (MONTAGE PRIMAIRE * (D' * CIRCUIT ELECTRIQUE)) * (OU TRIANGLE ETOILE))
- R 028
- (EST (MONTAGE PRIMAIRE * CIRCUIT ELECTRIQUE) * (OU TRIANGLE ETOILE))
- R 003
- (EST (MONTAGE PRIMAIRE * CIRCUIT ELECTRIQUE) * (OU (*EG TRIANGLE) ETOILE))
- R008
- (EST (MONTAGE PRIMAIRE * CIRCUIT ELECTRIQUE) * (OU (*EG TRIANGLE) (*EG ETOILE)))
- R022
- (MONTAGE PRIMAIRE * CIRCUIT ELECTRIQUE (OU (*EG TRIANGLE) (*EG ETOILE)))

(MONTAGE PRIMAIRE CIRCUIT ELECTRIQUE (OU (*EG TRIANGLE) (*EG ETOILE)))

un circuit électrique est composé de 3 enroulements.

- (EST (CIRCUIT ELECTRIQUE UN *) * (COMPOS * (DE * (ENROULEMENT 3 *))))
- R002
- (EST CIRCUIT ELECTRIQUE * (COMPOS * (DE * (ENROULEMENT 3 *))))
- R009
- (EST CIRCUIT ELECTRIQUE * (COMPOS * (DE * (ENROULEMENT * (CARD * (*EG * 3))))))
- R041
- (COMPOSITION CIRCUIT ELECTRIQUE (ENROULEMENT * (CARD * (*EG * 3)) *)
- (COMPOSITION CIRCUIT ELECTRIQUE (ENROULEMENT (CARD (*EG 3))))

l'echauff de l'huile dans un transformateur doit eˆtre inferieur a 45 degres.

- R 001 (D (ECHAUFF DE L'HUILE L' * (DANS * (TRANSFORMATEUR UN *))) * (ETT * (*PP * (DEGRE 4 *))))
- R 002 (D (ECHAUFF DE L'HUILE * (DANS * (TRANSFORMATEUR UN: *))) * (ETT * (*PP * (DEGRE 4 *))))
- R 013 (D (ECHAUFF DE L'HUILE * (DANS * TRANSFORMATEUR)) * (ETT * (*PP * (DEGRE 4 *))))
- R 005 (D (ECHAUFF DE L'HUILE * (DE * TRANSFORMATEUR)) * (ETT * (*PP * (DEGRE 4 *))))
- R 004 (D (ECHAUFF DE L'HUILE * (DE * TRANSFORMATEUR)) * (ETT * (*PP * 4)))
- R 028 (EST (ECHAUFF DE L'HUILE * (DE * TRANSFORMATEUR)) * (*PP * 4))
- R 022 (EST (ECHAUFF DE L'HUILE * TRANSFORMATEUR) * (*PP * 4))
- (ECHAUFF DE L'HUILE * TRANSFORMATEUR (*PP * 4))

(ECHAUFF DE L'HUILE TRANSFORMATEUR (*PP 4))

un noyau est composeˆ de gradins.

- R 002 (EST (NOYAU UN *) * (COMPOS * (DE * GRADIN)))
- R 041 (EST NOYAU * (COMPOS * (DE * GRADIN)))
- (COMPOSITION NOYAU GRADIN *)

(COMPOSITION NOYAU GRADIN)

l'organisation d'un enroulement est concentrique, biconcentrique ou alterne.

- (EST (ORGANISATION L' * (D' * (ENROULEMENT UN *))) * (CONCENTRIQUE * (, * (BICONCENTRIQUE * (OU * ALTERNE)))
- C 03
- (EST (ORGANISATION L' * (D' * (ENROULEMENT UN *))) * (CONCENTRIQUE * (, * (OU BICONCENTRIQUE ALTERNE)))
- C 04
- (EST (ORGANISATION L' * (D' * (ENROULEMENT UN *))) * (OU CONCENTRIQUE BICONCENTRIQUE ALTERNE))
- R001
- (EST (ORGANISATION * (D' * (ENROULEMENT UN *))) * (OU CONCENTRIQUE BICONCENTRIQUE ALTERNE))
- R 002
- (EST (ORGANISATION * (D' * ENROULEMENT)) * (OU CONCENTRIQUE BICONCENTRIQUE ALTERNE))
- R 028
- (EST (ORGANISATION * ENROULEMENT) * (OU CONCENTRIQUE BICONCENTRIQUE ALTERNE))
- R 008
- (EST (ORGANISATION * ENROULEMENT) * (OU (*EG CONCENTRIQUE) BICONCENTRIQUE ALTERNE))
- R 008
- (EST (ORGANISATION * ENROULEMENT) * (OU (*EG CONCENTRIQUE) (*EG BICONCENTRIQUE) ALTERNE))
- R 008
- (EST (ORGANISATION * ENROULEMENT) * (OU (*EG CONCENTRIQUE) (*EG BICONCENTRIQUE) (*EG ALTERNE)))
- R022
- (ORGANISATION * ENROULEMENT (OU (*EG CONCENTRIQUE) (*EG BICONCENTRIQUE) (*EG ALTERNE)))
- (ORGANISATION ENROULEMENT (OU (*EG CONCENTRIQUE) (*EG BICONCENTRIQUE) (*EG ALTERNE)))

?

2. Le RSP construit à partir des phrases

* * * DEFINITION DE TRANSFORMATEUR * * *

DESCRIPTION DE TRANSFORMATEUR
DANS LE CONTEXTE DE TRANSFORMATEUR

- PROPRIETES:

FREQUENCE (*EG 00)

FREQUENCE (*EG 50)

REFRIGERANT (*EG PYRALENE)

REFRIGERANT (*EG HUILE)

REFRIGERANT (*EG AIR)

ECHAUFFEMENT DE L'HUILE (*PP 45)

- RELATIONS AVEC D'AUTRES OBJETS:

COMPOSITION (CUVE CIRCUIT ELECTRIQUE CIRCUIT MAGNETIQUE)

- CONTRAINTES:

S I

(REFRIGERANT TRANSFORMATEUR (*EG PYRALENE))

A LORS

(COMPOSITION TRANS CUVE)

* * * DEFINITION DE CIRCUIT ELECTRIQUE * * *

DESCRIPTION DE CIRCUIT ELECTRIQUE
DANS LE CONTEXTE DE CIRCUIT ELECTRIQUE

- PROPRIETES:

MONTAGE PRIMAIRE (*EG ETOILE)

MONTAGE PRIMAIRE (*EG TRIANGLE)

MONTAGE SECONDAIRE (*EG ETOILE)

MONTAGE SECONDAIRE (*EG TRIANGLE)

- RELATIONS AVEC D'AUTRES OBJETS:

COMPOSITION ((UN GROUPE DE ((*EG 3)) ENROULEMENT ENRO189))

DESCRIPTION DE ENROULEMENT

DANS LE CONTEXTE DE CIRCUIT ELECTRIQUE

- PROPRIETES:

CARD (*EG 3)

*** DEFINITION DE CIRCUIT MAGNETIQUE ***

DESCRIPTION DE CIRCUIT MAGNETIQUE
DANS LE CONTEXTE DE CIRCUIT MAGNETIQUE

- PROPRIETES:

JOINT (*EG HIBRIQUES A 45 DEGRES)

JOINT (*EG HIBRIQUES A 90 DEGRES)

JOINT (*EG PLANS)

QUALITE DES TOLES (*EG Q2)

QUALITE DES TOLES (*EG Q1)

- RELATIONS AVEC D'AUTRES OBJETS:

COMPOSITION ((UN GROUPE DE ((*EG 2)) CULASSE CULA110)

(UN GROUPE DE ((*EG 3)) NOYAU NOYA108))

DESCRIPTION DE NOYAU
DANS LE CONTEXTE DE CIRCUIT MAGNETIQUE

- PROPRIETES:

CARD (*EG 3)

DESCRIPTION DE CULASSE
DANS LE CONTEXTE DE CIRCUIT MAGNETIQUE

- PROPRIETES:

CARD (*EG 2)

*** DEFINITION DE NOYAU ***

DESCRIPTION DE NOYAU
DANS LE CONTEXTE DE NOYAU

- RELATIONS AVEC D'AUTRES OBJETS:

COMPOSITION ((UN GROUPE DE ((*PG 1)) GRADIN GRAD145))

DESCRIPTION DE GRADIN
DANS LE CONTEXTE DE NOYAU

- PROPRIETES:

CARD (*PG 1)

DISP (*EG OIGNON)

DESCRIPTION DE NOYAU
DANS LE CONTEXTE DE CIRCUIT MAGNETIQUE

- PROPRIETES:
CARD (*EG 3)

*** DEFINITION DE CULASSE ***

DESCRIPTION DE CULASSE
DANS LE CONTEXTE DE CULASSE

- PROPRIETES:
COEFFICIENT DE SATURATION (*PP 2)
COEFFICIENT DE SATURATION (*PG 1)

DESCRIPTION DE CULASSE
DANS LE CONTEXTE DE CIRCUIT MAGNETIQUE

-PROPRIETES:
CARD (*EG 2)

*** DEFINITION DE ENROULEMENT ***

DESCRIPTION DE ENROULEMENT
DANS LE CONTEXTE DE ENROULEMENT

- PROPRIETES:
ORGANISATION (*EG ALTERNE)
ORGANISATION (*EG BICONCENTRIQUE)
ORGANISATION (*EG CONCENTRIQUE)

DESCRIPTION DE ENROULEMENT
DANS LE CONTEXTE DE CIRCUIT ELECTRIQUE

- PROPRIETES:
CARD (*EG 3)

Appendice D : Algorithmes pour la construction du sous-réseau correspondant à une phrase

Chaque algorithme est décrit par une fonction. Les symboles précédés d'un "&" sont des variables. Pour faciliter la correspondance avec la description BNF des formes réduites (cf. § 2.1. du chapitre 3) nous utilisons les éléments entre crochets (<propriété>, <objet>, ...) comme des variables dans les fonctions. Les symboles en majuscules sont des noms de fonctions.

Le premier pas de la construction du sous-réseau d'une phrase consiste à créer l'espace EP de la phrase. Le sous-réseau est construit dans EP en utilisant des fonctions spécialisées pour le traitement de chaque forme réduite lesquelles appellent à leur tour des fonctions pour la construction de parties de figure. La figure A.1 montre le schéma des appels.

- GENPROP construit les figures de valeurs possibles de propriété, de relation n-aire entre propriétés et de relation binaire entre propriétés selon le modèle :
<fréduite de propriété> ::= (<propriété> <objet1> <description de propriété>)
- GENREL construit les figures de relation entre objets selon le modèle :
<fréduite de relation> ::= (<relation> <objet primaire> <objet secondaire>)
- GENCONT construit les figures de contrainte selon le modèle :
<fréduite de contrainte> ::= (COND <si> <alors>).
- GENCONS construit les figures de conseil selon le modèle :
<fréduite de conseil> ::= (CONS <si> <alors>)
- GENCLASSE localise ou crée une classe donnée et son représentant générique (sommet \diamond et \circ).

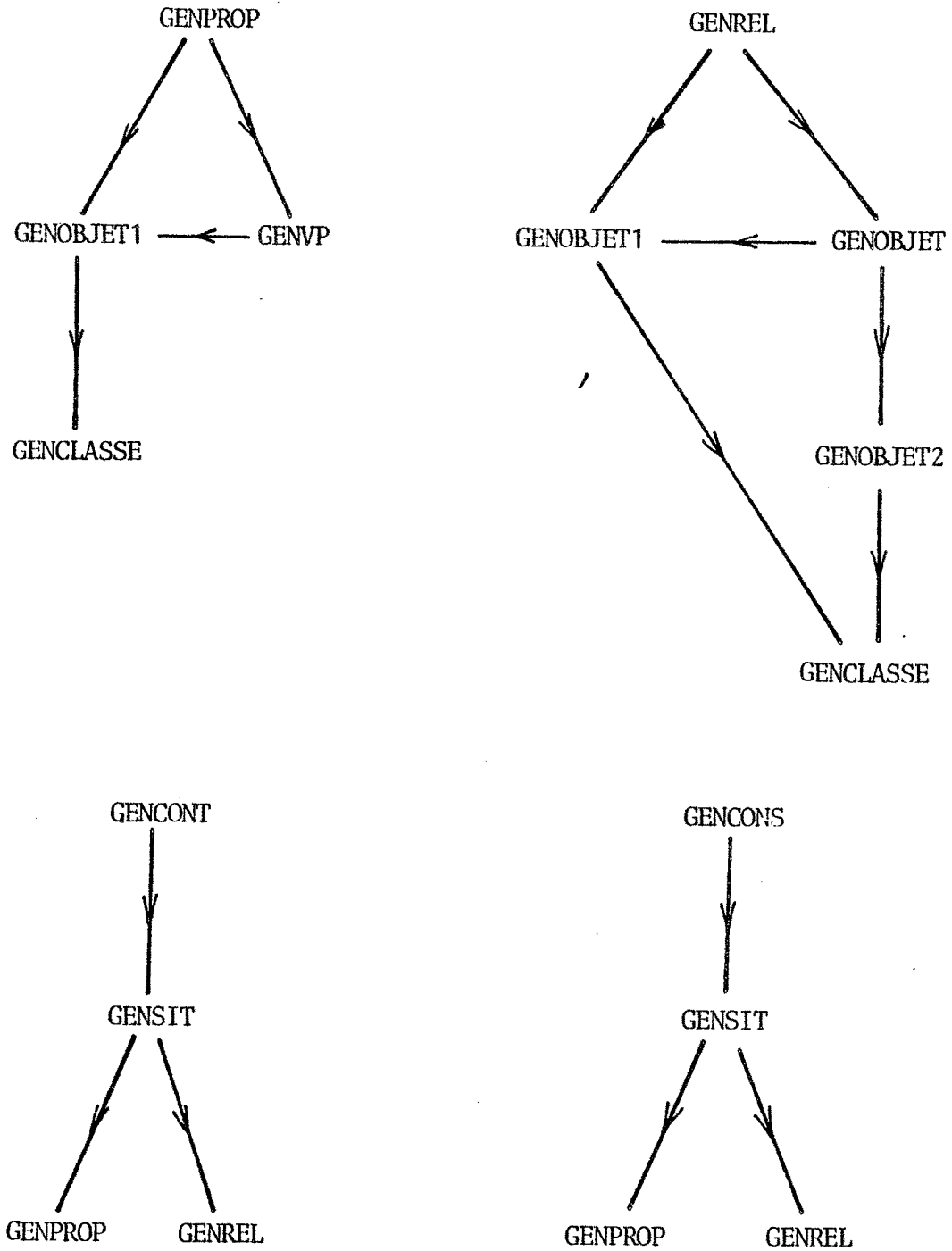


Figure D.1.

- GENOBJET1 crée des sommets \bigcirc correspondant au modèle :
 $\langle \text{objet1} \rangle ::= \langle \text{classe} \rangle | (\langle \text{classe} \rangle \langle \text{modificateur 1} \rangle)$
- GENOBJET2 engendre des sommets \textcircled{G} selon le modèle :
 $\langle \text{objet} \rangle ::= (\langle \text{classe} \rangle \langle \text{modificateur2} \rangle)$
- GENOBJET engendre des sommets \bigcirc ou \textcircled{G} en utilisant GENOBJET1 ou GENOBJET2 selon le cas.
- GENVP engendre des sommets \bullet , \square ou \triangle selon le modèle :
 $\langle \text{valeur de rep} \rangle ::= \langle \text{constante} \rangle | \langle \text{reference à propriété} \rangle |$
 $\langle \text{expression fonctionnelle} \rangle$
- GENSIT construit les figures de situation selon le modèle :
 $\langle \text{situation} \rangle ::= \langle \text{freuite1} \rangle |$
 $(\text{OU } \langle \text{freuite1} \rangle \langle \text{fréuite1} \rangle^+) |$
 $(\text{ET } \langle \text{fréuite1} \rangle \langle \text{fréuite1} \rangle^+)$

Nous définissons ci-dessous chacune de ces fonctions :

GENPROP ($\langle \text{propriété} \rangle \langle \text{objet1} \rangle \langle \text{description de propriété} \rangle$)

début

&obj := GENOBJET1($\langle \text{objet1} \rangle$);

On engendre dans EP un sommet \square étiqueté par $\langle \text{propriété} \rangle$; soit &prop ce sommet ;

On relie &obj à &prop par un arc d'étiquette P dans EP ;

si $\langle \text{description de propriété} \rangle \equiv \langle \text{relation primitive} \rangle$

alors &vp := GENVP($\langle \text{valeur de rep} \rangle$) ;

On relie &prop à &vp par un arc étiqueté $\langle \text{nom de rep} \rangle$;

sinon si $\langle \text{description de propriété} \rangle \equiv (\text{OU } \langle \text{relation primitive} \rangle$
 $\langle \text{relation primitive} \rangle^+)$

alors pour chaque $\langle \text{relation primitive} \rangle$ faire

début

&vp := GENVP($\langle \text{valeur de rep} \rangle$) ;

On relie &prop à &vp par un arc étiqueté $\langle \text{nom de rep} \rangle$;

fin ;

```

sinon si <description de propriété>≡
    (ET <relation primitive><relation primitive>+)
    alors pourchaque <relation primitive> faire
        début
            &vp:=GENVP(<valeur de rep>);
            On relie &prop à &vp par un arc
            étiqueté <nom de rep>;
        fin ;
    sinon ERREUR ;
    fsi ;
fsi ;
fsi ;
fsi ;
fin.

GENREL (<relation><objet primaire><objet secondaire>)
début
    &obj1:=GENOBJET1(<objet primaire>);
    On crée un sommet  $\triangle R$  étiqueté par <relation> ; soit &rel ce sommet ;
    On relie &obj1 à &rel par un arc étiqueté R ;
    si <objet secondaire>≡<objet>
        alors &obj2:=GENOBJET(<objet secondaire>);
        On relie &rel à &obj2 par un arc R
    sinon si <objet secondaire>≡ (ET <objet><objet>+)
        alors pourchaque <objet> faire
            début
                &obj2:=GENOBJET(<objet secondaire>);
                On relie &rel à &obj2 par un arc étiqueté R
            fin ;
        sinon ERREUR ;
    fsi ;
fsi ;
fin.

```

GENCONT (COND <si><alors>)

début

On crée un sommet $\triangle C$;

GENSIT(<si>);

GENSIT(<alors>);

pour chaque sommet $\triangle S$ produit par "GENSIT(<si>)" faire

On relie le sommet $\triangle C$ au sommet $\triangle S$ par un arc étiqueté SI ;

pour chaque sommet $\triangle S$ produit par "GENSIT(<alors>)" faire

On relie le sommet $\triangle C$ au sommet $\triangle S$ par un arc étiqueté ALORS ;

fin.

Remarque :

La fonction GENCONS est analogue à la fonction GENCONT. Nous ne la décrivons pas ici.

GENSIT(<situation>)

début

si <situation> \equiv <fréduite1>

alors On crée un sommet $\triangle S$ et son espace associé &esps ;

On utilise GENPROP ou GENREL pour construire dans &esps le sous-réseau correspondant à <fréduite1>;

sinon si <situation> \equiv (OU <fréduite1><fréduite1>⁺)

alors pour chaque <fréduite1> faire

début

On crée un sommet $\triangle S$ et son espace associé &esps ;

On utilise GENPROP ou GENREL pour construire dans &esps

le sous-réseau correspondant à <fréduite1> ;

fin ;

sinon si <situation> \equiv (ET<fréduite1><fréduite1>⁺)

alors on crée un sommet $\triangle S$ et son espace associé &esps ;

pour chaque <fréduite1> faire

On utilise GENPROP ou GENREL pour construire dans

&esps le sous-réseau correspondant à <fréduite1> ;

sinon ERREUR ;

fsi ;

fsi ;

fsi ;

fin .

GENVP(<valeur de rep>)

début

si <valeur de rep>≡<constante>

alors s'il existe dans U un sommet ● étiqueté par <constante> , ce sommet est retourné comme valeur de la fonction ; sinon ce sommet est créé dans U et retourné comme valeur de la fonction

sinon si <valeur de rep>≡<référence à propriété>

alors début

&obj := GENOBJET1(<objet1>);

s'il existe un sommet □ étiqueté par <propriété> relié à &obj par un arc d'étiquette P, alors ce sommet est retourné comme valeur de la fonction

fin ;

sinon si <valeur de rep>≡<expression fonctionnelle>

alors Pour chaque paramètre on détermine le sommet □ qui le représente. On crée un sommet △ auquel on associe comme propriété EXPR le corps de la fonction indiquée par <nom de fonction>. On relie le sommet △ à chacun des sommets □ par des arcs indiquant l'ordre des paramètres. On retourne le sommet △ comme valeur de la fonction

sinon ERREUR ;

fsi ;

fsi ;

fsi ;

fin.

GENOBJET (<objet>)

début

si (&obj:=GENOBJET1(<objet>))≠ERREUR

alors &obj

sinon GENOBJET2(<objet>)

fsi ;

fin.

GENOBJET1 (<objet1>)

début

si <objet1>≡<classe>

alors GENCLASSE(<classe>)

sinon si <objet1> (<classe> (IDENT (= &id))(DE &c1))

alors On recherche le sommet ○ représentant le membre spécial dont la propriété IDENT est &id dans le groupe d'objets de la classe <classe> qui compose le représentant générique de la classe &c1. Si ce sommet existe, il est retourné comme valeur de la fonction ; sinon il est créé et retourné comme valeur de la fonction

sinon si <objet> (<classe> (DE &c1))

alors On retourne le sommet ○ représentant générique de la classe <classe>

sinon ERREUR ;

fsi ;

fsi ;

fsi ;

fin.

GENOBJET2 (<objet>)

début

si <objet> (<classe> (CARD (<nom de rep><entier>)))

alors &c1:= le sommet ◇ relié par un arc EST_UN au sommet ○ obtenu par GENCLASSE(<classe>) ;

On crée un sommet Ⓞ ; soit &obj ce sommet ;

On relie &obj à &c1 par un arc étiqueté GROUPE_DE ;

On crée un sommet □ étiqueté CARD ; soit &prop ce sommet ;

&vp:=GENVP(<entier>);

On relie &prop à &rp par un arc étiqueté <nom de rep> ;

On relie &obj à &prop par un arc étiqueté P

```

sinon si <objet> (<classe> (DISP (= <disposition>)))
  alors &cl:= le sommet ◊ relié par un arc EST_UN au sommet ○
    obtenu par GENCLASSE(<classe>) ;
    On crée un sommet ⊙ ; soit &obj ce sommet ;
    On relie &obj à &cl pour un arc étiqueté GROUPE_DE ;
    On crée un sommet □ étiqueté DISP ; soit &prop ce sommet ;
    &vp:=GENVP(<disposition>) ;
    On relie &prop à &vp par un arc étiqueté = ;
    On relie &obj à &prop par un arc étiqueté P
  sinon si <objet>≡(<classe> (CARD (<nom de rep><entier>))
    (DISP (= <disposition>)))
    alors On crée un sommet ⊙ et on lui associe les
      propriétés CARD et DISP par le procédé montré
      ci-dessus
    sinon ERREUR ;
  fsi ;
fsi ;
fsi ;
fin.

```

GENCLASSE (<classe>)

début

```

si un sommet ◊ étiqueté <classe> existe dans le RSP
  alors On retourne le sommet ○ auquel il permet d'accéder par l'arc d'étiquette
  sinon Un sommet ◊ d'étiquette classe et un sommet ○ sont créés dans U.
    Le sommet ○ est relié au sommet ◊ par un arc EST_UN et le sommet ◊ e
    relié au sommet ○ par un arc REP (les deux arcs sont créés dans U).
    Le sommet ○ est retourné comme valeur de la fonction.

```

fsi ;

fin.

BIBLIOGRAPHIE

ABRIAL J.R.

"Data semantics"

Cours donné à l'école d'hiver de la CEE à l'Alpe d'Huez, 1973

BACH Emmon

"An introduction to transformational grammars"

Holt, Rinehart and Winston Inc, 1964

BERGE C.

"Théorie de graphes et ses applications"

Dunod, 1963

BERRY-ROGGIE G.L., WULZ H.

"THE DESIGN OF PLIDIS, a problem solving information system
with german as query language"

Institut für deutsche sprache, Mannheim (GFR), december 1976

BOBROW D.

"New programming languages for artificial intelligence research"

Artificial intelligence, vol.6, n°3, 1974

BOBROW, DANIEL G.- WINOGRAD, TERRY

"An overview of KRL, a knowledge representation language"

Stanford AI Labo, Memo AIM-293, Computer science dept.,

Stanford University, november 1976.

BRUCE B.

"Case systems for natural language"

Artificial intelligence, vol.6, 1975.

CARBONELL J.R. and COLLINS A.M.

"Natural semantics in artificial intelligence"

Proceedings of the third IJCAI, Stanford, California, 1973, pp.344/351.

CHAUCHE Jacques

"Transducteurs et arborescences. Etudes et réalisations de systèmes appliquées aux grammaires transformationnelles"
 Université scientifique et médicale de Grenoble
 Thèse d'Etat, décembre 1974.

CHOMSKY Noam

"Transformational analysis"
 Dissertation, University of Pennsylvania, 1955

CHOMSKY Noam

"Syntactic structures"
 The Hague, 1957

COLMERAUER Alain

"Les grammaires de métamorphose"
 Groupe d'I.A., Marseille-Luminy, novembre 1975

COURTIN Jacques

"Algorithmes pour le traitement interactif des langues naturelles"
 Université de Grenoble, thèse d'Etat
 octobre 1977

COURTIN Jacques, DUJARDIN Danièle

"Paramètres linguistiques de la morphologie française dans le système PIAF"
 Université de Grenoble, Laboratoire d'informatique,
 décembre 1976

DAVIS Randall, KING Jonathan

"An overview of production systems".
 Stanford University, 1976

DELIYANNI A., KOWALSKI R.

"Logic and semantic networks"
 Workshop on logic and data bases, Toulouse, France, novembre 1977.

DUDA R., HART P., NILSSON N. and SUTHERLAND G.

"Semantic network representations in rule-based inference systems"

S.R.I., Technical Note 136

Menlo Park, California, 1977

FIKES R. and HENDRIX G.G.

"A network-based knowledge representation and its natural deduction system"

I.J.C.A.I.

Cambridge, U.S.A., August 1977

FILLMORE C.J.

"The case for case" dans "Universals in linguistic theory"

Holt, Rinehart and Wilson, 1968

GRANDJEAN Ernest

"Conception et réalisation d'un dictionnaire pour un analyseur interactif de langues naturelles"

Thèse Ingénieur C.N.A.M., Grenoble, Février 1975

HENDRIX G.G.

"Partitioned networks for the mathematical modeling of natural language semantics"

TR NL-28, Dept. of Computer Sciences

The University of Texas at Austin, 1975

HENDRIX G.G.

"Expanding the utility of semantic networks through partitioning"

I.J.C.A.I.

Tbilisi, USSR, September 1975

HENDRIX G.G.

"The LIFER manual. A guide to building practical natural language interfaces"

SRI Artificial Intelligence Center,

Technical Note 138, 1977

IOMDINE L.L., MEL'CHUK I.A., PERTSAV N.V.

"Fragment de modèle de syntaxe russe de surface dans analyse et validation dans l'étude des données textuelles"

Edition du CNRS, 1977

JOLOBOFF Vania

"Unification d'arborescences. Evaluation sémantique d'énoncés en langue naturelle"

Thèse de Docteur-Ingénieur, INPG, Septembre 1978

KOWALSKI Robert A.

"Logic for problem solving"

Memo N°75, University of Edinburgh, March 1974

KOWALSKI Robert A.

"General laws in data description"

Workshop on logic and data bases, Toulouse, France, Novembre 1977

LATOMBE Jean-Claude

"Une application de l'intelligence artificielle à la conception assistée par ordinateur (TROPIC)"

USMG, INPG; Thèse d'Etat, Novembre 1977

LATOMBE Jean-Claude

"Failure processing in a system for designing complex assemblies"

Sixth IJCAI, Tokyo, 1979

LAURIERE Jean-Louis

"Un langage et un programme pour énoncer et résoudre des problèmes combinatoires"

Thèse de Doctorat, Paris 1976

LEVESQUE Hector J.

"A procedural approach to semantic networks"

Technical Report N°105

Dept. of Computer Sciences, University of Toronto, April 1977

LOPEZ Julio

"Systèmes de transformation de ramifications paramétrées.
Définitions et applications"

Institut National Polytechnique de Grenoble

Thèse de Docteur-Ingénieur, Juin 1979

MAC CARTHY J.

"LISP 1.5 Programmer's manual"

M.I.T. Computer Center and Research Laboratory of Electronics
Cambridge, Mass., 1962

MAC DERMOTT Drew V.

"Very large planner type data bases"

M.I.T., AI Memo 339, September 1975

MAC DERMOTT Drew V. and SUSSMAN Gerald J.

"The conniver reference manual"

M.I.T., AI Memo 259a, May 1972, updated January 1974

MINSKY Marvin (Editor)

"Semantic information processing"

The M.I.T. Press, Cambridge, Mass., 1968

NEWELL Allen and SIMON Herbert A.

"Human problem solving"

Prentice-Hall, INC., Englewood Cliffs, New Jersey, 1972

NOVAK Gordon

"Computer Understanding of physics problems stated in natural
language"

Dept. of Computer Sciences, The University of Texas at Austin
Technical Report NL-30, 1976

QUILLIAN M.R.

"Semantic memory" dans "Semantic information processing"

M. Minsky, The M.I.T. Press, Cambridge, Mass., 1968

QUILLIAN Ross M.

"The teachable language comprehender : a simulation program
and theory of language"

CACM, V.12, N°8, August 1969

RIEGER Chuck

"Conceptual overlays: a mechanism for the interpretation of
sentence meaning in context"

Dept. of Computer Sciences, University of Maryland, 1976

RIEGER Chuck and GRINBERG Milt

"The causal representation and simulation of physical mechanisms"

Dept. of Computer Sciences, University of Maryland, 1977

RUMELHART D.E. and NORMAN D.A.

"Active semantic networks as a model of human memory"

Advance Paper, Third IJCAI, Standford, California, 1973, pp. 450-457

SHANK R.C.

"Identification of conceptualisations underlying natural language"

In Shank and Colby (eds.), "Computer models of thought and language"

W.H. Freeman and Co., San Francisco, California, 1973

SHANK R.C. (Ed.)

"Conceptual information processing"

Amsterdam, North Holland, 1975

SHAPIRO S.C.

"A net structure for semantic information storage. Deduction and
retrieval"

Advance Papers, Second International Joint Conference in Artificial
Intelligence,

London, 1971, pp. 512-523

SHORTLIFFE E. et al.

"Computer based consultations in clinical therapeutics:
Explanation and rule acquisition capabilities of the MYCIN system"
Stanford University, 1974

SIMMONS Robert F., BRUCE Bertran C.

"Some relations between predicate calculus and semantic
net representation of discourse"
Department of Computer sciences, University of Texas, Austin,
Texas, USA, in IJCAI, Londres, 1971

SIMMONS R.F.

"Semantic networks: their computation and use for understanding
english sentences"
In Shank and Colby (Eds), Computer models of thought and language,
W.H.Freeman and Co, San Francisco, California, 1973

SUSSMAN G.J., McDERMOTT D.V.

"Why conniving is better than planning"
A.I. memo n° 255A, MIT, AI Lab., April 1972

TESNIERE L.

"Eléments de syntaxe structurale"
Editions Klincksieck, 1959

TURING Alan M.

"Intelligent machinery"
1948

WINOGRAD Terry

"Procedures as a representation for data in a computer program
for understanding natural language"
Ph.D thesis, MIT, 1971

WINOGRAD Terry

"Five lectures on artificial intelligence"

Stanford University, Computer sciences department, AIM n°246
and STAN CS 74-459, Stanford, September 1974.

WINOGRAD Terry

"Frames and the declarative procedural controversey"

In DG Bobrow and A. Collins (Eds), "Representation and understanding"
New-York Academic Press, 1975

WOODS W.A.

"Transition netwok grammars for natural language analysis"

CACM, vol.13, n.10, October 1970

WOODS W.A.

"What's in a link?"

In D.G.Bobrow and A.Collins(Eds) "Representation and understanding"
New-York Academic Press, 1975

WINSTON P.H.

"Learning structural descriptions from examples"

Ph.D.thesis, Dept. of electrical engineering, MIT, Cambridge,
Mass., 1970.

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de Messieurs :

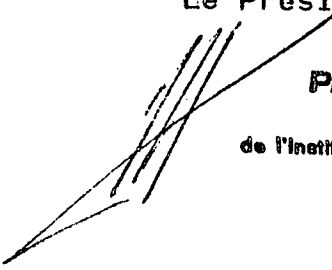
- J.Cl. LATOMBE, Maître-Assistant à l'Institut National Polytechnique de GRENOBLE
- A. MASSABO, Responsable de l'équipe CFAO au Centre de Recherches de BATELLE-GENEVE-

Monsieur Mauricio LOPEZ IZQUIERDO

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Grenoble, le 18 Septembre 1979

Le Président de l'I.N.P.G.


Ph. TRAYNARD
Président
de l'Institut National Polytechnique