



HAL
open science

Une application de l'intelligence artificielle à la conception assistée par ordinateur (TROPIC)

Jean-Claude Latombe

► **To cite this version:**

Jean-Claude Latombe. Une application de l'intelligence artificielle à la conception assistée par ordinateur (TROPIC). Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1977. Français. NNT : . tel-00010582

HAL Id: tel-00010582

<https://theses.hal.science/tel-00010582>

Submitted on 12 Oct 2005

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

**Université Scientifique et Médicale de Grenoble
Institut National Polytechnique de Grenoble**

pour obtenir le grade de
DOCTEUR D'ETAT ES-SCIENCES
"Mathématiques"

par

Jean-Claude LATOMBE



**UNE APPLICATION DE L'INTELLIGENCE ARTIFICIELLE
A LA CONCEPTION ASSISTEE PAR ORDINATEUR**

(TROPIC)



Thèse soutenue le 25 novembre 1977 devant la Commission d'Examen :

Président : L. BOLLIET

Rapporteur : G. VEILLON

Examineurs : J.J. ALLAN

A. COLMERAUER

N.J. NILSSON

J.C. SABONNADIÈRE

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : M. TANCHE

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

(RANG A)

SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE

PROFESSEURS DE 1ère CLASSE

ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

.../...

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDO Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie

PROFESSEURS DE 2ème CLASSE

ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

.../...

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : D. BLOCH

**Vice-Président : René CARRE
Hervé CHERADAME
Marcel IVANES**

PROFESSEURS DES UNIVERSITES :

ANCEAU François	E.N.S.I.M.A.G.
BARRAUD Alain	E.N.S.I.E.G.
BAUDELET Bernard	E.N.S.I.E.G.
BESSON Jean	E.N.S.E.E.G.
BLIMAN Samuel	E.N.S.E.R.G.
BLOCH Daniel	E.N.S.I.E.G.
BOIS Philippe	E.N.S.H.G.
BONNETAIN Lucien	E.N.S.E.E.G.
BONNIER Etienne	E.N.S.E.E.G.
BOUVARD Maurice	E.N.S.H.G.
BRISSENEAU Pierre	E.N.S.I.E.G.
BUYLE BODIN Maurice	E.N.S.E.R.G.
CAVAIGNAC Jean-François	E.N.S.I.E.G.
CHARTIER Germain	E.N.S.I.E.G.
CHENEVIER Pierre	E.N.S.E.R.G.
CHERADAME Hervé	U.E.R.M.C.P.P.
CHERUY Arlette	E.N.S.I.E.G.
CHIAVERINA Jean	U.E.R.M.C.P.P.
COHEN Joseph	E.N.S.E.R.G.
COUMES André	E.N.S.E.R.G.
DURAND Francis	E.N.S.E.E.G.
DURAND Jean-Louis	E.N.S.I.E.G.
FELICI Noël	E.N.S.I.E.G.
FOULARD Claude	E.N.S.I.E.G.
GENTIL Pierre	E.N.S.E.R.G.
GUERIN Bernard	E.N.S.E.R.G.
GUYOT Pierre	E.N.S.E.E.G.
IVANES Marcel	E.N.S.I.E.G.
JAUSSAUD Pierre	E.N.S.I.E.G.
JOUBERT Jean-Claude	E.N.S.I.E.G.
JOURDAIN Geneviève	E.N.S.I.E.G.
LACOUME Jean-Louis	E.N.S.I.E.G.
LATOMBE Jean-Claude	E.N.S.I.M.A.G.

.../...

LESSIEUR Marcel	E.N.S.H.G.
LESPINARD Georges	E.N.S.H.G.
LONGEQUEUE Jean-Pierre	E.N.S.I.E.G.
MAZARE Guy	E.N.S.I.M.A.G.
MOREAU René	E.N.S.H.G.
MORET Roger	E.N.S.I.E.G.
MOSSIERE Jacques	E.N.S.I.M.A.G.
PARIAUD Jean-Charles	E.N.S.E.E.G.
PAUTHENET René	E.N.S.I.E.G.
PERRET René	E.N.S.I.E.G.
PERRET Robert	E.N.S.I.E.G.
PIAU Jean-Michel	E.N.S.H.G.
POLOUJADOFF Michel	E.N.S.I.E.G.
POUPOT Christian	E.N.S.E.R.G.
RAMEAU Jean-Jacques	E.N.S.E.E.G.
RENAUD Maurice	U.E.R.M.C.P.P.
ROBERT André	U.E.R.M.C.P.P.
ROBERT François	E.N.S.I.M.A.G.
SABONNADIÈRE Jean-Claude	E.N.S.I.E.G.
SAUCIER Gabrielle	E.N.S.I.M.A.G.
SCHLENKER Claire	E.N.S.I.E.G.
SCHLENKER Michel	E.N.S.I.E.G.
SERMET Pierre	E.N.S.E.R.G.
SILVY Jacques	U.E.R.M.C.P.P.
SOHM Jean-Claude	E.N.S.E.E.G.
SOUQUET Jean-Louis	E.N.S.E.E.G.
VEILLON Gérard	E.N.S.I.M.A.G.
ZADWORNY François	E.N.S.E.R.G.

PROFESSEURS ASSOCIES

BASTIN Georges	E.N.S.H.G.
BERRIL John	E.N.S.H.G.
CARREAU Pierre	E.N.S.H.G.
GANDINI Alessandro	U.E.R.M.C.P.P.
HAYASHI Hirashi	E.N.S.I.E.G.

PROFESSEURS UNIVERSITE DES SCIENCES SOCIALES (Grenoble II)

BOLLIET Louis
Chatelin Françoise

PROFESSEURS E.N.S. Mines de Saint-Etienne

RIEU Jean
SOUSTELLE Michel

CHERCHEURS DU C.N.R.S.

FRUCHART Robert
VACHAUD Georges

Directeur de Recherche
Directeur de Recherche

.../...

ALLIBERT Michel	Maître de Recherche
ANSARA Ibrahim	Maître de Recherche
ARMAND Michel	Maître de Recherche
BINDER Gilbert	
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DEPORTES Jacques	
DRIOLE Jean	Maître de Recherche
GIGNOUX Damien	
GIVORD Dominique	
GUELIN Pierre	
HOPFINGER Emil	Maître de Recherche
JOUD Jean-Charles	Maître de Recherche
KAMARINOS Georges	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Dore	Maître de Recherche
LASJAUNIAS J.C.	
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche
PIAU Monique	
PORTESEIL Jean-Louis	
THOLENCE Jean-Louis	
VERDILLON André	

CHERCHEURS du MINISTERE de la RECHERCHE et de la TECHNOLOGIE (Directeurs et Maîtres de Recherches, ENS Mines de St. Etienne)

LESBATS Pierre	Directeur de Recherche
BISCONDI Michel	Maître de Recherche
KOBYLANSKI André	Maître de Recherche
LE COZE Jean	Maître de Recherche
LALAUZE René	Maître de Recherche
LANCELOT Francis	Maître de Recherche
THEVENOT François	Maître de Recherche
TRAN MINH Canh	Maître de Recherche

PERSONNALITES HABILITEES à DIRIGER des TRAVAUX de RECHERCHE (Décision du Conseil Scientifique)

ALLIBERT Colette	E.N.S.E.E.G.
BERNARD Claude	E.N.S.E.E.G.
BONNET Rolland	E.N.S.E.E.G.
CAILLET Marcel	E.N.S.E.E.G.
CHATILLON Catherine	E.N.S.E.E.G.
CHATILLON Christian	E.N.S.E.E.G.
COULON Michel	E.N.S.E.E.G.
DIARD Jean-Paul	E.N.S.E.E.G.
EUSTAPOPOULOS Nicolas	E.N.S.E.E.G.
FOSTER Panayotis	E.N.S.E.E.G.

.../...

GALERIE Alain	E.N.S.E.E.G.
HAMMOU Abdelkader	E.N.S.E.E.G.
MALMEJAC Yves	E.N.S.E.E.G. (CENG)
MARTIN GARIN Régina	E.N.S.E.E.G.
NGUYEN TRUONG Bernadette	E.N.S.E.E.G.
RAVAINE Denis	E.N.S.E.E.G.
SAINFORT	E.N.S.E.E.G. (CENG)
SARRAZIN Pierre	E.N.S.E.E.G.
SIMON Jean-Paul	E.N.S.E.E.G.
TOUZAIN Philippe	E.N.S.E.E.G.
URBAIN Georges	E.N.S.E.E.G. (Laboratoire des ultra-réfractaires ODEILLON)
GUILHOT Bernard	E.N.S. Mines Saint Etienne
THOMAS Gérard	E.N.S. Mines Saint Etienne
DRIVER Julien	E.N.S. Mines Saint Etienne
BARIBAUD Michel	E.N.S.E.R.G.
BOREL Joseph	E.N.S.E.R.G.
CHOVET Alain	E.N.S.E.R.G.
CHEHIKIAN Alain	E.N.S.E.R.G.
DOLMAZON Jean-Marc	E.N.S.E.R.G.
HERAULT Jeanny	E.N.S.E.R.G.
MONLLOR Christian	E.N.S.E.R.G.
BORNARD Guy	E.N.S.I.E.G.
DESCHIZEAU Pierre	E.N.S.I.E.G.
GLANGEAUD François	E.N.S.I.E.G.
KOFMAN Walter	E.N.S.I.E.G.
LEJEUNE Gérard	E.N.S.I.E.G.
MAZUER Jean	E.N.S.I.E.G.
PERARD Jacques	E.N.S.I.E.G.
REINISCH Raymond	E.N.S.I.E.G.
ALEMANY Antoine	E.N.S.H.G.
BOIS Daniel	E.N.S.H.G.
DARVE Félix	E.N.S.H.G.
MICHEL Jean-Marie	E.N.S.H.G.
OBLED Charles	E.N.S.H.G.
ROWE Alain	E.N.S.H.G.
VAUCLIN Michel	E.N.S.H.G.
WACK Bernard	E.N.S.H.G.
BERT Didier	E.N.S.I.M.A.G.
CALMET Jacques	E.N.S.I.M.A.G.
COURTIN Jacques	E.N.S.I.M.A.G.
COURTOIS Bernard	E.N.S.I.M.A.G.
DELLA DORA Jean	E.N.S.I.M.A.G.
FONLUPT Jean	E.N.S.I.M.A.G.
SIFAKIS Joseph	E.N.S.I.M.A.G.
CHARUEL Robert	U.E.R.M.C.P.P.
CADET Jean	C.E.N.G.
COEURE Philippe	C.E.N.G. (LETI)

.../...

DELHAYE Jean-Marc	C.E.N.G. (STT)
DUPUY Michel	C.E.N.G. (LETI)
JOUBE Hubert	C.E.N.G. (LETI)
NICOLAU Yvan	C.E.N.G. (LETI)
NIFENECKER Hervé	C.E.N.G.
PERROUD Paul	C.E.N.G.
PEUZIN Jean-Claude	C.E.N.G. (LETI)
TAIEB Maurice	C.E.N.G.
VINCENDON Marc	C.E.N.G.

LABORATOIRES EXTERIEURS

DEMOULIN Eric	C.N.E.T.
DEVINE	C.N.E.T. (R.A.B.)
GERBER Roland	C.N.E.T.
MERCKEL Gérard	C.N.E.T.
PAULEAU Yves	C.N.E.T.
GAUBERT C.	I.N.S.A. Lyon

Je remercie

Monsieur L.BOLLIET, Professeur à l'IUT d'Informatique de Grenoble, qui m'a fait l'honneur de présider le jury de cette thèse,

Monsieur J.J.ALLAN, Professeur à l'Université Méthodiste de Dallas, Vice-Président du Groupe de Travail 5.2 de l'IFIP, qui a aidé et encouragé ce travail par de nombreuses suggestions,

Monsieur A.COLMERAUER, Professeur à l'Université d'Aix-Marseille, qui s'est intéressé à mon travail et qui a bien voulu le juger,

Monsieur N.J.NILSSON, Responsable de Recherche au Centre d'Intelligence Artificielle du Stanford Research Institute, qui par son enthousiasme et ses conseils m'a constamment encouragé lors des deux séjours que j'ai effectués dans ce Centre,

Monsieur J.C.SABONNADIÈRE, Maître de Conférences à l'INPG, qui m'a permis de commencer cette recherche dans son équipe,

Monsieur G.VEILLON, Professeur à l'INPG, qui m'a accueilli dans son équipe et qui a guidé ce travail par des suggestions et des critiques constructives.

Je remercie également

Earl SACERDOTI et Richard WALDINGER pour les nombreuses discussions que nous avons eues ensemble et pour tout le temps qu'ils ont passé à conseiller ce travail;

Augustin LUX pour ses nombreuses suggestions et pour son aide efficace lors de l'implantation de Tropic en LISP/CMS,

Mauricio LOPEZ pour la patience dont il a du faire preuve pour lire et commenter les premiers manuscrits de cette thèse,

Bertram RAPHAEL qui m'a accueilli par deux fois au Centre d'Intelligence Artificielle du SRI dont il était alors le directeur.

Je remercie Mademoiselle J.ARGENTO et Madame H.DIAZ qui ont assuré la dactylographie de ce document avec rapidité et une grande compétence, Monsieur RASOLONJATOVO pour la qualité de ses dessins, et le service de reproduction pour le soin qu'il a apporté à la réalisation matérielle de cette thèse.

Mon premier séjour au Centre d'Intelligence Artificielle du SRI a été subventionné par une bourse IRIA. L'implantation de Tropic sur IBM 360/67 a été réalisée dans le cadre du contrat SESCRI No. 77052.

Jean-Claude LATOMBE

SOMMAIRE

	Pages
INTRODUCTION	
Chapitre 1: PRESENTATION DE TROPIC	
1. A quoi sert Tropic ?.....	1.1
2. Les aspects importants de Tropic.....	1.4
2.1. L'hypothèse de décomposabilité et son exploitation.....	1.4
2.2. Le traitement des échecs.....	1.5
2.3. Application d'une stratégie hiérarchique.....	1.5
2.4. Coordination de deux programmes de résolution de problème.....	1.7
2.5. Utilisation d'heuristiques spécialisées.....	1.8
2.6. Résolution interactive des problèmes.....	1.9
2.7. Représentation des connaissances par des règles de production.....	1.9
3. Les programmes de Tropic et leur organisation.....	1.10
4. Autres applications de l'Intelligence Artificielle à la conception.....	1.12
5. Plans des chapitres 2 à 7.....	1.16
Chapitre 2: L'ENONCE D'UN PROBLEME	
1. Notions formelles préliminaires.....	2.1
1.1. Ensembles, listes, atomes, constantes, variables et expressions.....	2.1
1.2. Substitutions et instances.....	2.1
2. Les principaux concepts des modèles construits par Tropic et leur expression formelle.....	2.2
2.1. Les principaux concepts.....	2.2
2.2. Le formalisme.....	2.2
3. Les règles de construction de modèles.....	2.3
3.1. Définition des rcm:.....	2.4
3.2. Exemple.....	2.5
3.3. Notion de rcm conditionnelle.....	2.11
4. Les contraintes.....	2.12
4.1. Généralisation des littéraux.....	2.12
4.2. Interprétation logique des littéraux constants.....	2.13
4.3. Définition des contraintes.....	2.14

Chapitre 3: LA RECHERCHE D'UNE SOLUTION

1. Le cheminement avant.....	3.1
2. Les échecs.....	3.5
3. Les cheminements arrière.....	3.6
4. Algorithme simplifié de la recherche.....	3.10
5. Comparaison avec les modèles de transformation d'état et de décomposition de problème.....	3.10
5.1. Comparaison avec la transformation d'état.....	3.11
5.2. Comparaison avec la décomposition de problème.....	3.13

Chapitre 4: LES HEURISTIQUES DU CHEMINEMENT AVANT

1. Principe des heuristiques du cheminement avant.....	4.1
2. La stratégie hiérarchique.....	4.2
2.1. Les relations descriptives et associatives, et leur représentation.....	4.2
2.1.1. Les objets abstraits et les objets concrets..	4.2
2.1.2. Les relations descriptives et les relations associatives.....	4.3
2.1.3. Les D-littéraux, les A-littéraux, les A-termes et les C-termes.....	4.3
2.2. La structure hiérarchique d'un modèle.....	4.5
2.2.1. Les relations associatives hiérarchiques et transversales.....	4.5
2.2.2. Les A-H-littéraux et les A-T-littéraux.....	4.5
2.2.3. La relation \supset	4.6
2.3. Les niveaux de détail d'un modèle.....	4.7
2.3.1. Profondeur d'un atome de C-terme dans un atome de C-terme.....	4.7
2.3.2. Couple de dépendance d'un littéral.....	4.7
2.3.3. Description d'un atome de C-terme.....	4.8
2.3.4. Niveau de détail d'une description.....	4.9
2.3.5. Niveau de détail d'un modèle.....	4.9
2.4. La stratégie hiérarchique.....	4.9
2.4.1. Définition.....	4.9
2.4.2. Application.....	4.10
2.4.3. Implications sur la valeur logique des littéraux	4.18
2.4.4. Implications sur la satisfaction des contraintes	4.19

3. Le choix des ircm.....	4.20
3.1. Les conseils.....	4.20
3.2. Le choix des ircm.....	4.23

Chapitre 5: LE TRAITEMENT DES DONNEES NUMERIQUES

1. Les D-S-littéraux et les D-N-littéraux.....	5.1
2. Les restrictions symboliques et les restrictions numériques	5.2
3. Définition d'une solution.....	5.3
3.1. Compatibilité d'un ensemble de restrictions numériques	5.3
3.2. Définition d'une solution.....	5.4
4. Vérification de la compatibilité des restrictions numériques	5.5
4.1. Fréquence de cette vérification.....	5.5
4.2. Le graphe des restrictions numériques.....	5.6
4.3. Vérification de la compatibilité des restrictions numériques.....	5.8
4.3.1. Formulation du problème.....	5.8
4.3.2. Résolution du problème.....	5.11
4.3.3. Commentaires.....	5.12

Chapitre 6: LE TRAITEMENT DES ECHECS

1. Principe du traitement des échecs.....	6.2
1.1. Description simplifiée du traitement d'un échec.....	6.2
1.1.1. Exemple d'échec.....	6.2
1.1.2. Formation de l'expression causale.....	6.7
1.1.3. La détermination du point de retour.....	6.8
1.1.4. Le cheminement arrière et la reprise du cheminement avant.....	6.9
1.2. Cas de plusieurs échecs consécutifs.....	6.10
1.2.1. Principe de l'utilisation des indicateurs d'échec.....	6.11
1.2.2. Illustration sur un exemple.....	6.15
1.3. Les diagnostics et leur utilisation.....	6.29
1.3.1. Principe.....	6.29
1.3.2. Exemple.....	6.30

	Pages
2. Description formelle simplifiée du traitement des échecs	6.31
2.1. Première étape.....	
2.1.1. La recherche d'une solution.....	6.32
2.1.2. La détermination du point de retour.....	6.33
2.1.3. Complétude du traitement.....	6.35
2.2. Deuxième étape.....	6.40
2.3. Troisième étape.....	6.43
2.4. Discussion des hypothèses 1', 2' et 3'.....	6.45
2.4.1. Hypothèse 1'.....	6.45
2.4.2. Hypothèse 2'.....	6.46
2.4.3. Hypothèse 3'.....	6.46
2.4.4. Conclusion.....	6.47
3. Description des algorithmes.....	6.48
3.1. Formation de l'expression causale initiale.....	6.48
3.1.1. L'expression causale initiale en l'absence de diagnostic.....	6.48
3.1.2. L'expression causale initiale en présence de diagnostics.....	6.53
3.2. La détermination du point de retour.....	6.55
3.3. Le cheminement arrière.....	6.57
3.4. La reprise du cheminement avant.....	6.58
 Chapitre 7: ACTIVATION DES CONTRAINTES, DES CONSEILS ET DES DIAGNOSTICS	
1. Activation des contraintes, des conseils et des diagnostics pendant le cheminement avant.....	7.1
1.1. Hypothèses de travail.....	7.1
1.2. L'algorithme d'activation.....	7.2
1.3. La fonction INSTANCES.....	7.10
2. Mises à jour lors d'un cheminement arrière.....	7.12
2.1. Compléments sur l'algorithme d'activation.....	7.12
2.2. L'algorithme des mises à jour lors d'un cheminement arrière.....	7.12
2.3. Compléments sur l'algorithme des mises à jour.....	7.15

Chapitre 8: IMPLANTATION ET EXPERIMENTATION DE TROPIC ;
QUELQUES NOUVELLES DIRECTIONS DE RECHERCHE

1. Implantation de Tropic.....	8.1
1.1. Historique.....	8.1
1.1.1. L'écriture de Tropic dans le langage QLISP...	8.1
1.1.2. Le passage de QLISP à INTERLISP.....	8.2
1.1.3. Le transport des programmes sur IBM 360/67...	8.2
1.2. Description générale des programmes.....	8.3
1.3. Utilisation de l'implantation de Tropic sur IBM 360/67	8.3
2. Expérimentation de Tropic.....	8.4
2.1. L'exemple du transformateur.....	8.5
2.1.1. La description des problèmes.....	8.5
2.1.2. La résolution des problèmes.....	8.8
2.2. Les aspects positifs et les aspects négatifs mis en évidence par cette expérimentation.....	8.9
2.2.1. Le fonctionnement de Tropic.....	8.9
2.2.2. La description des problèmes.....	8.10
2.2.3. La résolution des problèmes.....	8.11
3. Quelques nouvelles directions de recherche.....	8.12
3.1. Réalisation d'un interface de communication en langue naturelle avec Tropic.....	8.13
3.2. Mise en oeuvre d'autres modèles de résolution de problème.....	8.13
3.3. Définition d'un modèle de représentation des relations de causes à effets.....	8.14
3.4. Définition d'un modèle d'apprentissage.....	8.14
3.5. Modification interactive de la description d'un problème.....	8.15
3.6. Utilisation de rcm conditionnelles.....	8.15
3.7. Utilisation de plusieurs quantifications de variables	8.16
3.8. Réalisation d'un logiciel général de traitement d'échec.....	8.16
CONCLUSION.....	

Appendice A: UN EXEMPLE DE CONCEPTION DE TOUR

1. Le fichier f.....	A.1
2. La trace de la résolution.....	A.6

Appendice B: LES RCM DE L'EXEMPLE DU TRANSFORMATEUR

REFERENCES

LISTES DES FIGURES

	Pages
Figure 1.1.....	1.3
Figure 1.2.....	1.11
Figure 3.1.....	3.2
Figure 3.2.....	3.4
Figure 3.3.....	3.8
Figure 3.4.....	3.9
Figure 3.5.....	3.12
Figure 4.1.....	4.5
Figure 4.2.....	4.13
Figure 4.3.....	4.16
Figure 4.4.....	4.17
Figure 5.1.....	5.7
Figure 6.1.....	6.5
Figure 6.2.....	6.6
Figure 6.3.....	6.11
Figure 6.4.....	6.17
Figure 6.5.....	6.18
Figure 6.6.....	6.19
Figure 6.7.....	6.21
Figure 6.8.....	6.24
Figure 6.9.....	6.25
Figure 6.10.....	6.28
Figure 6.11.....	6.52
Figure 7.1.....	7.5

INTRODUCTION

On considère souvent la conception comme une activité de résolution de problème dont le but est de définir un objet répondant à un besoin, par exemple une machine, un mécanisme, un bâtiment. On s'attendrait donc à trouver la résolution de problème ("problem solving") parmi les thèmes essentiels de la Conception Assistée par Ordinateur (CAO).

Pourtant, un examen de la littérature spécialisée (Vlietstra 1973, Allan 1977, Lang 1974, CAD 74 et 76, Design Automation Conferences 1974-1977, Computer Aided Design Journal) montre que les principaux axes du développement de la CAO ont été la communication homme-machine (graphique et non graphique), les bases de données, l'analyse numérique et la construction de banques de programmes spécifiques. Si les modèles informels de l'activité de conception existent en abondance, les références à l'emploi de méthodes générales de résolution de problèmes sont relativement rares et concernent essentiellement l'optimisation numérique. Ainsi, la plupart des systèmes de CAO existants offrent des outils pour décrire, manipuler, analyser et éventuellement optimiser (numériquement) des modèles d'objets (cf. notamment (Jacquart 1975)), mais l'emploi de ces outils est entièrement à la charge du concepteur. On constate avec un certain étonnement que la CAO actuelle ignore presque complètement les méthodes non numériques de résolution de problème qui peuvent sélectionner et modifier la "structure" de l'objet à concevoir en fonction des objectifs à atteindre.

Au contraire, la résolution de problème a toujours été un thème majeur de l'Intelligence Artificielle. Plusieurs modèles généraux de résolution fondés sur le concept de recherche heuristique dans un espace de possibilités y ont été élaborés: transformation d'état, décomposition de problème, démonstration de théorème (Nilsson 1971). Pendant les années 60, le succès de ces modèles a été limité à la résolution de problèmes parfois très combinatoires, mais de représentations simples (problèmes de jeux par exemple). Plus récemment, les progrès effectués dans les domaines de la "représentation

des connaissances" (Hendrix 1975, Davis 1975) et des langages très évolués du type PLANNER (Bobrow 1974) ont permis de maîtriser des représentations beaucoup plus complexes, et de réaliser des systèmes où les modèles heuristiques sont appliqués à la résolution de problèmes réels: analyse de spectres de masse (Feigenbaum 1971), diagnostic et thérapie d'infection bactériennes (Shortliffe 1976), réparations d'appareils électromécaniques (Hart 1975), analyse de circuits électroniques (Stallman 1976). Ces systèmes sont caractéristiques d'une tendance importante de l'évolution actuelle de l'Intelligence Artificielle. Ils montrent que pour résoudre un problème réel une organisation appropriée des connaissances est souvent plus importante qu'une heuristique puissante.

L'expérience de la CAO que nous avons acquise en réalisant et en utilisant le système ESPACE (Latombe 1972, Bolopion 1975), et au contact d'autres équipes (notamment au sein de l'AF.MICADO* et du Groupe de Travail 5.2. de l'IFIP**) nous a convaincus de l'intérêt que présenterait la mise en oeuvre d'outils de résolution de problèmes dans un système de CAO (Latombe 1976a, Latombe 1976b). Elle nous a conduits à adapter à la conception des modèles de résolution de problème et de représentation des connaissances issus de l'Intelligence Artificielle. Nous avons ainsi conçu et réalisé progressivement le système Tropic qui est décrit dans ce document.

Tropic est donc le produit d'une recherche qui intéresse à la fois la Conception Assistée par Ordinateur et l'Intelligence Artificielle. Nous résumons ci-dessous les aspects les plus intéressants du système, relativement à ces deux disciplines:

a/ Point de vue de la CAO

Tropic constitue un environnement de travail très souple qui permet au concepteur d'exprimer progressivement les objectifs à atteindre et ses connaissances sur les moyens de les atteindre. Ces objectifs et ces connaissances

* Association Française ayant pour Mission la Conception Assistée et le Dessin par Ordinateur.

** Groupe de Travail sur la CAO de l'"International Federation for Information Processing".

déterminent un problème que le système a pour tâche de résoudre. Ils sont exprimés sous la forme d'items modulaires (appelés "règles de production") qui peuvent être facilement ajoutés, retirés ou modifiés. Le concepteur peut aussi communiquer à Tropic son expérience et son intuition sur la façon de résoudre le problème, et participer de manière interactive à sa résolution. L'utilisation de Tropic n'est pas restreinte à une discipline technique particulière.

b/ Point de vue de l'Intelligence Artificielle

Tropic met en oeuvre un modèle de résolution de problème qui exploite la décomposabilité des problèmes de conception, tout en tenant compte des interdépendances entre les sous-problèmes engendrés. Ce modèle incorpore une méthode de traitement d'échec sophistiquée qui est fondée sur l'analyse et la mémorisation des conditions des échecs subis. Il peut utiliser des heuristiques spécialisées fournies par le concepteur qui l'aident non seulement à guider la génération d'une solution, mais aussi à corriger ses erreurs. Parmi les autres aspects intéressants de Tropic, nous citerons la coordination de deux programmes de résolution de problèmes fondés sur des principes différents et l'application d'une stratégie hiérarchique. Bien que ces méthodes aient été développées pour résoudre des problèmes de conception, leur principe est plus général.

Le chapitre 1 donne une vue d'ensemble de Tropic. Les chapitres 2, 3, 4, 5, 6 et 7 analysent le système en détail. Le chapitre 8 décrit son implantation et son expérimentation ; il propose quelques nouvelles directions de recherche.

Chapitre 1: PRESENTATION DE TROPIC

1. A quoi sert Tropic?

Un problème de conception est un problème ouvert qui n'admet ni formulation, ni solution définitive (Rittel 1972, Bazjanac 1974): en effet, on ne peut jamais, ni exprimer intégralement le besoin que l'objet à concevoir a pour objectif de satisfaire, ni établir une liste exhaustive des moyens qui peuvent servir à construire un modèle de cet objet. Pour un tel problème, on doit se contenter d'une solution acceptable obtenue avec des ressources (temps, ressource financière) nécessairement limitées (Simon, 1969).

Il s'en suit que le processus de conception est un processus au cours duquel le problème de conception est formulé et résolu plusieurs fois (Bazjana 1974):

- . Chaque formulation consiste à définir l'énoncé d'un problème fermé (i.e., qui peut être résolu par une personne ou un système qualifié, sans autres informations que celles qui sont contenues dans l'énoncé). Cet énoncé détermine donc implicitement les solutions qui peuvent être engendrées. Il reflète la compréhension et la connaissance qu'a le concepteur, à un instant donné, du besoin à satisfaire et des moyens de le satisfaire.

- . Chaque résolution consiste à rendre explicite une (ou plusieurs) solution d'un problème fermé. Elle a pour but d'aider le concepteur à découvrir de nouveaux aspects du problème de conception, qui le conduiront éventuellement à formuler l'énoncé d'un nouveau problème fermé.

Tropic est un système de résolution de problème dont la tâche est de résoudre les problèmes fermés énoncés par le concepteur. Son objectif est de permettre à celui-ci de mieux se consacrer à l'analyse du besoin à satisfaire et à la détermination des moyens potentiels de le satisfaire. Son domaine d'application est la construction de modèles de systèmes physiques dans lesquels l'aspect graphique n'est pas essentiel. A titre d'exemple, nous avons

appliqué Tropic à la conception de transformateurs triphasés de moyenne puissance (entre 10 et 2000 KVA). Ce domaine n'est pas restreint à une discipline technique particulière comme l'Electrotechnique ou la Mécanique.

L'énoncé d'un problème doit être présenté à Tropic sous la forme de règles de construction de modèles, qui définissent implicitement l'ensemble de tous les modèles que Tropic pourra construire pendant la résolution du problème, et de contraintes, qui définissent les conditions qu'un modèle devra satisfaire pour être une solution.

a/ Les règles de construction de modèles, ou rcm, expriment des connaissances telles que:

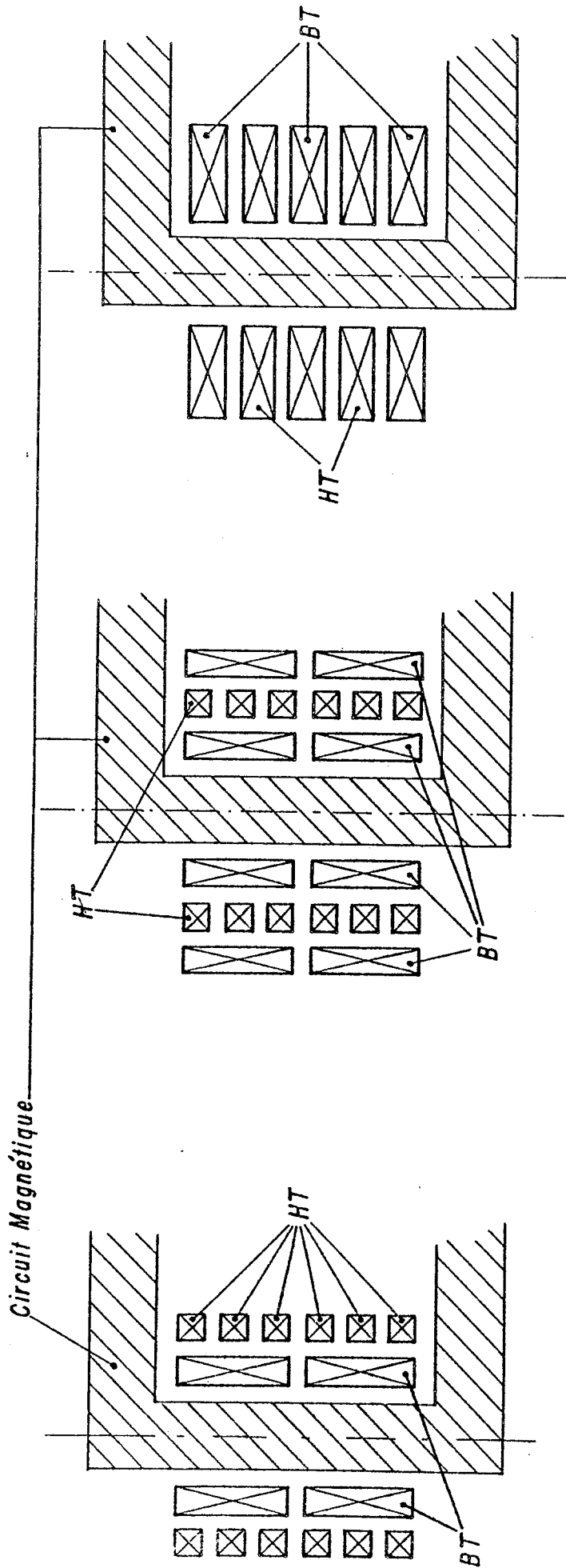
- . "Le réfrigérant d'un transformateur peut être l'huile ou l'air",
- . "Un transformateur est composé d'un circuit magnétique, d'un bobinage et, si l'huile est le réfrigérant, d'une cuve",
- . "Les joints d'un circuit magnétique peuvent être plans, imbriqués à 90° ou imbriqués à 45°",
- . "Le circuit primaire (ou secondaire) d'un bobinage peut être monté en étoile, en triangle ou en zig-zag".

Il est important de noter que les rcm permettent de construire des modèles qui diffèrent non seulement par la valeur de certains paramètres (par exemple la nature du réfrigérant, la forme des joints d'un circuit magnétique), mais aussi par leurs structures, par exemple:

- . un transformateur peut être composé ou non d'une cuve,
- . les enroulements qui composent un bobinage peuvent être concentriques, bi-concentriques ou alternés (cf. figure 1.1.).

b/ Les contraintes expriment des conditions binaires qui sont soit satisfaites, soit contredites par un modèle, par exemple:

- . "La tension d'alimentation primaire du transformateur à concevoir doit être égale à 380V",
- . "Les circuits primaire et secondaire du bobinage de ce transformateur doivent être montés respectivement en triangle et en étoile",



Un enroulement concentrique est composé de deux empilages de galettes séparés par un cylindre isolant. Les galettes de l'empilage intérieur (resp. extérieur) appartiennent toutes au circuit Basse Tension (resp. Haute Tension).

a. Enroulement concentrique

Un enroulement bi-concentrique est composé de trois empilages de galettes séparés par deux cylindres isolants. Les galettes des empilages intérieur et extérieur (resp. de l'empilage intermédiaire) appartiennent toutes au circuit Basse Tension (resp. Haute Tension).

b. Enroulement bi-concentrique

Un enroulement alterné est composé d'un empilage de galettes. Ces galettes appartiennent alternativement au circuit Basse Tension et au circuit Haute Tension. Les deux galettes extrêmes appartiennent au circuit Basse Tension.

c. Enroulement alterné

figure 1.1.

- . "La tension de court-circuit du transformateur doit être égale à 4 pour cent de sa tension nominale", °
- . "Si l'air est le réfrigérant, alors les tensions primaire et secondaire du transformateur doivent être inférieures à 5 KV".

Les contraintes servent aussi à exprimer les équations d'analyse, par exemple: $\phi = BS$, $P = \sqrt{3} UI \cos\phi$, $\Delta\theta = K\left(\frac{W}{S}\right)^{0,8}$. Certaines équations peuvent avoir plusieurs formes correspondant à des structures différentes de modèle.

Soit μ l'ensemble de modèles qui est implicitement défini par les rcm. La tâche de Tropic consiste à rendre explicite un modèle de μ qui satisfait les contraintes du problème. Ce modèle est une solution du problème.

2. Les aspects importants de Tropic

L'absence de système de résolution de problème ayant des objectifs comparables à ceux de Tropic, le volume des données qu'il est nécessaire de traiter pour résoudre un problème de conception, et la variété de ces données nous ont conduits à concevoir Tropic suivant une approche pragmatique et évolutive que reflètent les deux aspects suivants du système:

- . Tropic ne met pas en oeuvre un module de traitement unique. Au contraire, il coordonne plusieurs modules qui exploitent différentes propriétés des problèmes de conception, par exemple la décomposabilité de ces problèmes et la structure hiérarchique des systèmes techniques.
- . Tropic n'est pas un système de résolution de problème entièrement autonome, bien qu'il puisse être utilisé comme tel dans le cas de problèmes simples. Au contraire, il offre au concepteur plusieurs possibilités (heuristiques spécialisées, interaction) de participer à la résolution d'un problème.

Ces deux aspects de Tropic apparaissent nettement dans les paragraphes ci-dessous

2.1. L'hypothèse de décomposabilité et son exploitation

Les rcm sont des opérateurs dont l'application permet de construire un

modèle en décomposant le problème défini par l'énoncé présenté à Tropic en sous-problèmes de plus en plus élémentaires (cf. chapitre 2). Par exemple, une rcm peut décomposer le problème "concevoir un transformateur" en trois sous-problèmes, "concevoir un circuit magnétique", "concevoir un bobinage" et "concevoir une cuve", après avoir ajouté les données suivantes au modèle: "le réfrigérant du transformateur est l'huile ; ses composants sont un circuit magnétique, un bobinage et une cuve". Ces données peuvent être vues comme une "solution" du problème décomposé, sous réserve que les sous-problèmes engendrés puissent être résolus à leur tour. Le principe des rcm repose ainsi sur l'hypothèse de la décomposabilité des problèmes de conception. Cette hypothèse est vérifiée par une classe importante de problèmes de conception (Simon 1969).

Toutefois, les sous-problèmes successivement engendrés peuvent ne pas être indépendants. En effet, les contraintes peuvent lier des données qui résultent de la décomposition de sous-problèmes différents. Ces sous-problèmes ne peuvent donc pas être résolus indépendamment les uns des autres. Par exemple, la contrainte "si les enroulements d'un transformateur sont alternés et si ce transformateur est composé d'une cuve, alors cette cuve ne doit pas avoir une forme en cloche" lie les solutions des sous-problèmes "concevoir un bobinage" et "concevoir une cuve" qui résultent de la décomposition du problème "concevoir un transformateur".

Le modèle de résolution de problème de Tropic est un modèle heuristique hybride qui combine les modèles classiques de transformation d'état ("state-space model") et de décomposition de problème ("problem-reduction model") (Nilsson 1971). Il exploite la décomposabilité des problèmes pour contrôler et limiter la propagation des modifications qui sont effectuées sur la partie de modèle déjà construite, lorsque celle-ci contredit une contrainte du problème. Il permet simultanément de tenir compte des interdépendances entre les sous-problèmes (cf. chapitre 3).

2.2. Le traitement des échecs

Nous disons que Tropic est en échec lorsqu'il constate que la partie de modèle qu'il a engendrée contredit une (ou plusieurs) contrainte du problème.

Il analyse alors les conditions de cette contradiction pour déterminer les modifications à apporter à la solution courante. En empêchant de modifier des parties de modèle qui ne peuvent être responsables de la contradiction, cette analyse évite souvent l'exécution successive de cheminements arrière ("backtracking") aveugles aux points de décision les plus récents, comme cela est le cas dans de nombreux systèmes de résolution de problème (cf. chapitre 6).

Considérons par exemple le cas simple où la contrainte suivante est contredite: "si les enroulements du bobinage d'un transformateur sont alternés et si ce transformateur est composé d'une cuve, alors cette cuve ne doit pas avoir une forme en cloche". L'analyse de la contradiction conduira Tropic à modifier soit la forme de la cuve, soit la disposition des enroulements, soit la nature du réfrigérant. Elle lui évitera par exemple de modifier la forme des joints du circuit magnétique, qui ne peut être tenue pour responsable de la contradiction.

De plus, Tropic mémorise les conditions des échecs qu'il subit dans des expressions que nous appelons indicateurs d'échec. Ces expressions ont plusieurs applications intéressantes (cf. chapitre 6):

- . elles peuvent éviter à Tropic de subir deux fois le même échec,
- . elles lui permettent de reprendre en considération les conditions d'anciens échecs, pour les corriger différemment si cela est nécessaire,
- . elles facilitent la vérification de la compatibilité des corrections effectuées après un échec avec les parties de modèle qui sont restées inchangées après cet échec.

L'analyse des conditions des échecs et la mémorisation des indicateurs d'échec sont deux des moyens utilisés par Tropic pour faire face au caractère combinatoire des problèmes.

2.3. Application d'une stratégie hiérarchique

Tropic applique une stratégie hiérarchique descendante qui consiste à construire un modèle à des "niveaux de détail" successifs qui décrivent des

objets de plus en plus primitifs. Par exemple, dans le cas du transformateur, la stratégie hiérarchique peut consister à engendrer d'abord les propriétés globales du transformateur, puis celles de ses composants (le circuit magnétique, le bobinage et éventuellement la cuve), puis celles des composants de ces composants, ...

La stratégie hiérarchique appliquée par Tropic est déterminée par des informations qui sont fournies par le concepteur dans l'énoncé du problème. Elle a dans Tropic un fonction méthodologique et une fonction heuristique:

- . Comme méthodologie (cf. notamment les chapitres 4, 5 et 7), elle sert à organiser et à coordonner les activités de Tropic, en permettant de rechercher une solution dans une succession d'espaces ayant des dimensions plus réduites que l'espace qui est déterminé par l'énoncé du problème (Newell 1972 a).
- . Comme heuristique (cf. chapitre 4), elle est un moyen de faire face au caractère combinatoire des problèmes. Son principe repose alors sur l'hypothèse suivant laquelle il est possible de discriminer les données d'un modèle d'après leurs importances relatives de telle sorte que la génération d'un niveau de détail remette peu en question les niveaux de détail déjà engendrés (Newell 1972 a, Sacerdoti 1973, Sacerdoti 1975). Cette hypothèse est en général vérifiée par les problèmes de conception (Alexander 1964, Simon 1969, Eastman 1978).

2.4. Coordination de deux programmes de résolution de problème

Tropic utilise et coordonne deux programmes de résolution de problème pour traiter des données de natures différentes (cf. chapitre 5):

- . L'un de ces programmes applique une méthode de recherche heuristique dans une arborescence ET/OU (Nilsson 1971). Il sert à traiter les données que nous qualifions de symboliques, par exemple: la forme des joints d'un circuit magnétique, le montage des circuits primaire et secondaire d'un bobinage triphasé, la disposition concentrique, bi-concentrique ou alternée d'un enroulement. Ces données appartiennent à des ensembles discrets.

- . L'autre programme est fondé sur l'application d'une méthode d'optimisation numérique. Il sert à traiter les données que nous qualifions de numériques (nombres réels), par exemple: la tension d'alimentation primaire d'un transformateur, les pertes magnétiques dans un circuit magnétique, l'échauffement du réfrigérant. Ces données appartiennent à des continuum de nombres. Le programme mis en oeuvre dans Tropic s'inspire de méthodes utilisées dans CADSYS (Whitney 1974).

2.5. Utilisation d'heuristiques spécialisées

Il est rare qu'un problème de conception soit tout à fait nouveau. En général, une certaine expérience a déjà été acquise lors de la résolution de problèmes antérieurs appartenant à la même discipline technique. Tropic permet au concepteur de formuler son expérience et son intuition sous la forme de conseils et de diagnostics:

- . Les conseils expriment des informations susceptibles d'aider Tropic lorsqu'il choisit une rcm, par exemple: "si la puissance apparente du transformateur à concevoir doit être supérieure à 50 KVA, alors il est recommandé de choisir l'huile pour réfrigérant" (cf. chapitre 4).
- . Les diagnostics expriment des informations susceptibles d'aider Tropic lorsqu'il détermine les modifications à effectuer, après avoir constaté la contradiction d'une contrainte. Une telle information est par exemple: "si les pertes magnétiques dans un circuit magnétique dépassent nettement le seuil imposé par une contrainte (par exemple $\frac{\text{pertes magnétiques}}{\text{seuil imposé}} > 0.2$), alors il faut essayer d'améliorer la qualité (normalisée) des tôles qui constituent le circuit magnétique". Les diagnostics du concepteur complètent ceux qui sont automatiquement effectués par Tropic (cf. chapitre 6)

Les conseils et les diagnostics constituent des heuristiques que nous qualifions de spécialisées, par opposition aux heuristiques générales qui sont contenues dans les programmes de Tropic. Ils permettent au concepteur de contribuer à la résolution du problème. Leur principe peut être rapproché de celui des "méta-règles" décrites par Davis (1977).

Les conseils sont aussi un moyen d'exprimer des souhaits sur la solution. Le conseil suivant peut par exemple exprimer le goût d'un architecte: "Si les murs d'une maison sont blancs, alors il est souhaité que son toit soit rouge".

2.6. Résolution interactive des problèmes

Lors de la définition d'une rcm, d'une contrainte, d'un conseil ou d'un diagnostic, le concepteur peut prévoir que des questions lui soient posées sur son terminal pendant la résolution du problème. Par exemple, un conseil peut être formulé de la façon suivante: "dès qu'un conducteur est utilisé, demander quelles sont ses dimensions (normalisées) conseillées". Tropic offre ainsi la possibilité de résoudre les problèmes de manière interactive. Le concepteur, qui connaît la partie de modèle engendrée lorsque les questions lui sont posées, peut fournir des informations plus judicieuses au système (Tropic imprime une trace qui informe le concepteur de l'avancement de la résolution).

2.7. Représentation des connaissances par des règles de production

Les rcm, les contraintes, les conseils et les diagnostics sont énoncés comme des règles de production de la forme "Conditions → Déclarations" (Waterman 1970, Newell 1972b, Davis 1975). Les avantages qui résultent de la modularité des connaissances ainsi décrites ont déjà été mis en évidence dans d'autres systèmes, notamment MYCIN (Shortliffe 1976). Ils sont particulièrement intéressants dans Tropic:

- . Un problème de conception peut être formulé plusieurs fois (cf. début du §1). En général, chaque formulation diffère assez peu de la précédente. La modularité des règles de production facilite ces mises à jour.
- . Tropic n'a pas de connaissances à priori dans telle ou telle discipline technique. Il s'en suit que l'énoncé d'un problème contient souvent un grand

nombre d'informations qui décrivent ces connaissances et qui peuvent donc être utiles à la formulation d'autres problèmes (c'est en général le cas des rcm et des contraintes qui expriment les équations d'analyse). Grâce à leur modularité, les règles de production peuvent être mémorisées dans des fichiers qui peuvent être combinés à volonté. Pour énoncer un problème, le concepteur peut ainsi utiliser des connaissances qui ont été décrites à l'occasion d'un problème antérieur. Cela explique que les rcm d'un problème puissent exprimer une connaissance générale telle que "le circuit primaire du bobinage d'un transformateur peut être monté en étoile, en triangle ou en zig-zag", alors qu'une contrainte du même problème impose que ce circuit soit monté en triangle (cf. §1).

3. Les programmes de Tropic et leur organisation

Tropic est opérationnel sur l'ordinateur IBM 360/67 du CICG* sous le système conversationnel CP/CMS. Il est écrit en LISP/CMS, une version du langage LISP (McCarthy 1965) développée par Lux (1975). La figure 1.2. montre ses principaux programmes et leur organisation fonctionnelle approximative. La stratégie hiérarchique est appliquée par le coordonnateur qui commande l'exécution des programmes de résolution de problème (prp) symbolique et numérique. Si l'un de ces programmes constate la contradiction d'une contrainte, il fait appel au programme de traitement d'échec, qui détermine les modifications à effectuer et qui les effectue. Tous ces programmes travaillent sur une base de données commune qui représente l'état courant de la résolution du problème. Le programme d'activation des connaissances a pour fonction d'enregistrer dans cette base les règles de production C → D dont les conditions d'utilisation sont satisfaites.

* Centre Interuniversitaire de Calcul de Grenoble

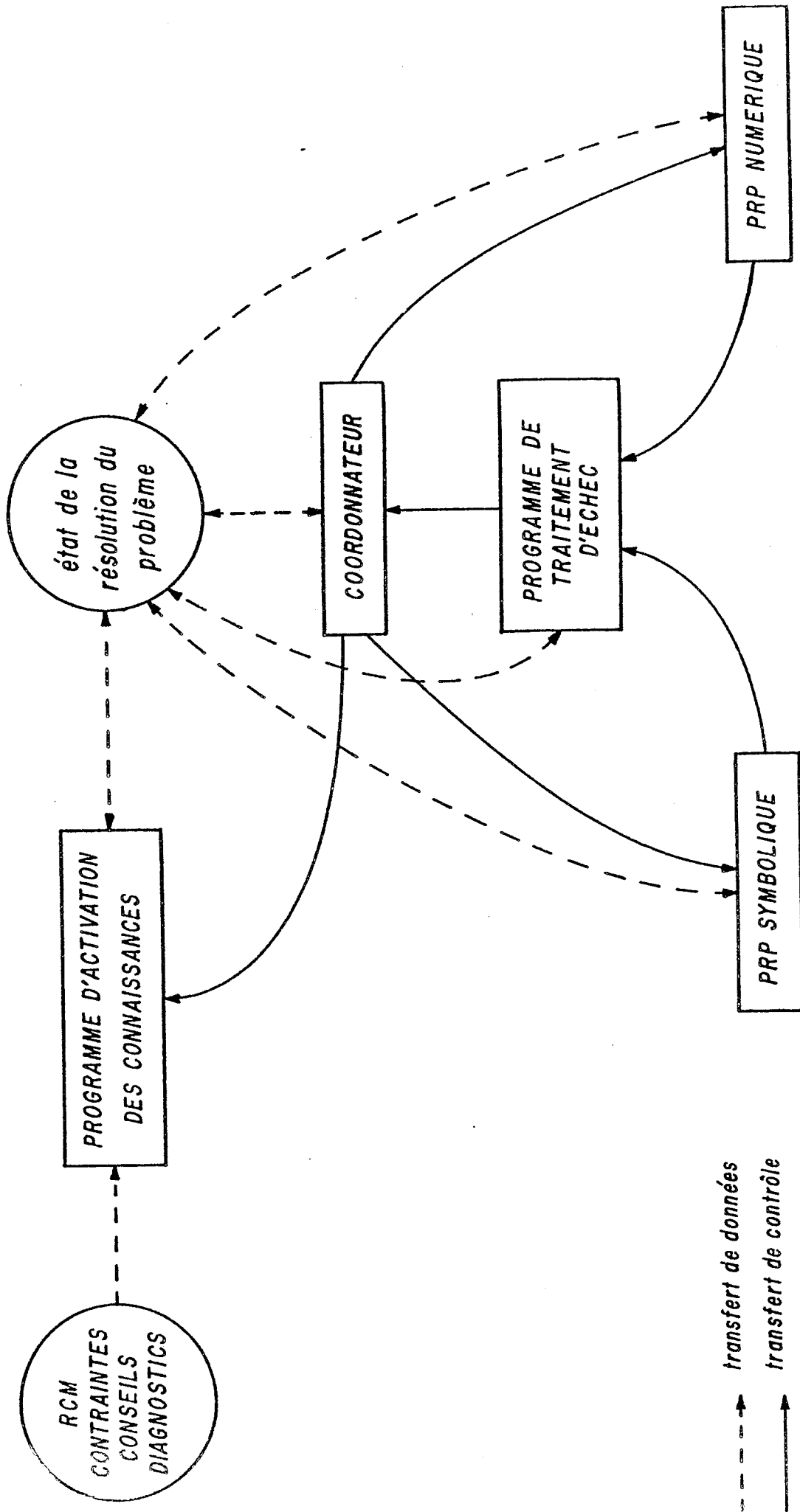


Figure : 1.2

4. Autres applications de l'Intelligence Artificielle à la conception

Avant de décrire Tropic en détail, nous passons en revue les principales publications que nous connaissons et qui décrivent des travaux appliquant des méthodes d'intelligence artificielle à la conception. Nous présentons cette revue suivant un ordre approximativement chronologique, en regroupant les publications qui relèvent d'un même thème.

a) JIRAUGH 1966

C'est à notre connaissance la plus ancienne publication qui fait référence à la fois à l'Intelligence Artificielle et à l'Automatisation de la Conception. Elle décrit notamment un programme qui utilise une méthode heuristique simple de transformation d'état pour engendrer des schémas de câblage, en essayant de minimiser la longueur des fils utilisés.

b) FREEMAN et NEWELL 1971

Freeman et Newell présentent un intéressant modèle pour concevoir en terme de fonctions. Les éléments de ce modèle sont les structures et les fonctions. Chaque structure fournit un ensemble de fonctions (par exemple, la structure "lame" fournit la fonction "coupe"), et pour chaque fonction qu'elle fournit, une structure requiert qu'on lui fournisse un ensemble de fonctions (par exemple, pour couper, une lame doit être tenue). Une connexion fonctionnelle peut lier deux structures si l'une d'elles fournit une fonction qui est requise par l'autre (par exemple, un manche permet de tenir une lame). Une structure composée est un ensemble de structures élémentaires liées par des connexions fonctionnelles (par exemple, un couteau qui est composé d'un manche et d'une lame).

Dans ce modèle, un problème consiste à construire une structure composée ayant des spécifications fonctionnelles imposées, à l'aide de structures élémentaires données. Pour résoudre un tel problème, les auteurs proposent deux méthodes de transformation d'état, dites ascendante (on part des structures élémentaires données et on essaye de construire progressivement la structure demandée) et descendante (on part des spécifications fonctionnelles de la structure demandée). Ils décrivent un exemple relatif à la conception d'une table de symboles symétrique pour un langage de programmation.

c) EASTMAN 1971 et 1973, PFEFFERKORN 1972 et 1975

Eastman et Pfefferkorn décrivent deux programmes, GSP ("General Spade Planner") et DPS ("Design Problem Solver"), qui sont capables de résoudre automatiquement des problèmes dits d'allocation spatiale. Ces problèmes consistent à placer des objets dans une salle, tout en satisfaisant des contraintes données (adjacence, distance, orientation, accès...). Par exemple, la salle peut être une salle d'ordinateur, les objets à placer étant l'unité centrale, l'imprimante, le lecteur de cartes, les dérouleurs de bandes, ... ; les contraintes peuvent alors être: chaque objet doit être accessible, le lecteur de cartes doit être situé à moins de 2 mètres de la porte d'entrée, ... Les programmes d'Eastman (GSP) et de Pfefferkorn (DPS) appliquent des méthodes heuristiques de transformation d'état. La stratégie de DPS est sensiblement plus riche que celle de GSP. DPS peut notamment diagnostiquer les causes des difficultés qu'il rencontre, puis choisir des actions appropriées à ces causes. Toutefois, la description que Pfefferkorn donne de ce traitement (dans les documents que nous possédons) est très superficielle, et semble être spécifique des problèmes d'allocation spatiale.

d) SIMON 1973

Cette publication ne s'adresse que partiellement à la conception et ne décrit aucune réalisation pratique. Nous la citons, car nous pensons qu'elle a influencé notre travail. Simon y définit ce qu'il appelle les problèmes mal structurés (en les opposant aux problèmes bien structurés), et présente la conception d'une maison comme un exemple d'un tel problème. Il montre que les programmes de résolution de problème développés en Intelligence Artificielle, tels que GPS (Ernst 1969), peuvent être utiles pour résoudre ces problèmes, à condition de leur adjoindre un mécanisme d'évocation de la connaissance. Cela le conduit à proposer le schéma d'un système de résolution de problème dont nous nous sommes inspirés dans la figure 1.2. Il suggère aussi que la connaissance soit organisée en un "système de production" (ensemble de règles de production).

e) POWERS 1973, POWERS et RUDD 1974

Ces deux publications (semblables) passent en revue les trois principaux modèles de résolution de problèmes étudiés en Intelligence Artificielle: la

transformation d'état, la décomposition de problème et la démonstration de théorème. Elles présentent des applications possibles de chacun de ces modèles à la chimie (synthèses de processus et de molécules). Mais la naïveté de ces applications les rend peu crédibles (cf. notamment l'application de GP à la synthèse d'un processus chimique). La publication de Powers (1973) a cependant le mérite d'avoir été la première à présenter les modèles de résolution de problème développés en Intelligence Artificielle à une conférence de CAO.

f) IWATA et al. 1975

Cette publication décrit un programme, CAAD-I ("Computer-Aided-Alloy-Designing-I"), capable de concevoir des alliages ayant des propriétés imposées. Le programme commence par sélectionner un matériau initial dans une base de matériaux, en fonction des propriétés imposées. Il améliore ensuite le matériau choisi en appliquant des "tactiques" ; ces tactiques sont des règles d'inférence qui dérivent de l'expérience des métallurgistes, par exemple: l'ajout de molybdène au nickel permet d'augmenter la résistance à la chaleur de ce dernier. Ces tactiques sont rangées dans une base de données et peuvent être combinées de façon à définir des "stratégies".

g) STALLMAN et SUSSMAN 1976, McDERMOTT 1977, SUSSMAN 1977

Ces publications décrivent les principaux travaux effectués au Laboratoire d'Intelligence Artificielle du MIT dans le domaine de l'analyse et de la synthèse des circuits électroniques.

Le programme décrit par Stallman et Sussman (1976) est un programme d'analyse de circuits électroniques. Des règles de production codent les lois de la physique (lois d'Ohm, de Kirchoff) et les modèles des composants électroniques non linéaires (transistors par exemple). L'application de chacune de ces règles permet au programme de faire une déduction simple sur le circuit à analyser, par exemple: la connaissance de la tension aux deux extrémités d'une résistance et de la valeur de cette résistance lui permet de déduire l'intensité du courant dans la résistance (une telle déduction est possible parce que le programme inclut un module pour la manipulation formelle d'équations algébriques). Le programme effectue l'analyse d'un circuit en appliquant une

méthode dite de propagation de contraintes. Cette méthode consiste à effectuer des déductions le long de plusieurs branches du circuit tant qu'une contradiction (deux valeurs de la tension à un même noeud, obtenues par des déductions différentes, ne sont pas égales) n'a pas été mise en évidence. Pendant l'application de cette méthode, le programme remplace chaque composant non linéaire par un circuit linéaire équivalent, après avoir fait l'hypothèse d'un point de fonctionnement pour le composant. Les contradictions qui sont constatées par la méthode de propagation de contraintes résultent de ce que certaines hypothèses de ce type sont fausses. Le programme effectue alors un cheminement arrière "intelligent" qui consiste d'abord à déterminer les hypothèses qui peuvent être responsables de la contradiction, puis à modifier l'une d'elles. L'analyse d'un circuit est terminée lorsque la tension à chaque noeud et l'intensité dans chaque branche sont connues.

McDermott (1977) décrit un programme capable de concevoir des circuits électroniques simples. Ce programme dérive partiellement des travaux de Freeman et Newell (1971), et applique une méthode de conception descendante. Sussman (1977) présente la conception des circuits électroniques comme un objectif stimulant pour la recherche en Intelligence Artificielle. Il décrit un exemple dans lequel il applique la méthode dite de mise au point de plans presque corrects ("debugging almost right plans"), développée dans HACKER (Sussman 1975), à la synthèse d'un quadripole dont le gain en tension est donné en fonction de la fréquence. Toutefois, à notre connaissance, le programme capable de traiter un tel exemple n'a pas été réalisé.

A l'exception des idées exposées par Simon (1973), les travaux cités dans ce paragraphe ont peu influencé notre travail, soit parce que les méthodes décrites sont trop simples en regard des problèmes que nous nous proposons d'aborder (cas de a, b et e), soit parce qu'elles sont trop spécifiques d'un type de problème (cas de c et f), soit parce qu'elles sont trop récentes (cas de g). Nous verrons toutefois au chapitre 8 que les méthodes développées par Sussman et son groupe de travail peuvent stimuler de nouvelles recherches sur Tropic.

5. Plan des chapitres 2 à 7

Les chapitres 2 à 7 décrivent Tropic en détail:

- . Le chapitre 2 définit les rcm et les contraintes qui composent l'énoncé d'un problème.
- . Le chapitre 3 décrit le principe du modèle de résolution de problème mis en oeuvre. Il définit la représentation qui permet d'exploiter la décomposabilité des problèmes.
- . Le chapitre 4 décrit les heuristiques utilisées pour construire un modèle solution. Il définit la stratégie hiérarchique et les conseils.
- . Le chapitre 5 introduit et décrit le traitement des données numériques. Jusqu'à ce chapitre, nous ignorerons donc complètement les aspects numériques des problèmes à résoudre.
- . Le chapitre 6 décrit le traitement des échecs et définit les diagnostics.
- . Le chapitre 7 décrit le programme d'activation des règles de production.

Afin d'éviter des complications techniques propres à l'exemple du transformateur, nous utiliserons dans ces chapitres un exemple d'illustration très simple. Le problème de cet exemple consiste en la construction d'un modèle représentant une tour de hauteur donnée. Cette tour est composée d'une colonne et d'un toit ; la colonne est elle-même composée d'un ou de plusieurs blocs ; plusieurs formes et plusieurs couleurs sont possibles pour le toit et les blocs.

Chapitre 2: L'ENONCE D'UN PROBLEME

L'énoncé d'un problème soumis à Tropic est composé de rcm^* et de contraintes que nous définissons aux paragraphes 3 et 4 de ce chapitre. Auparavant, nous introduisons quelques notions formelles (§1). Nous décrivons ensuite les principaux concepts qui caractérisent les modèles que Tropic peut construire, et nous définissons le formalisme qui est utilisé pour les exprimer (§2).

1. Notions formelles préliminaires

1.1. Ensembles, listes, atomes, constantes, variables et expressions

Pour décrire les informations sur lesquelles travaille Tropic, nous utilisons des ensembles finis et des listes (ces dernières étant définies comme des suites finies d'éléments avec des duplications possibles). Par convention, nous notons les ensembles entre deux accolades, par exemple {b a c}, et les listes entre deux parenthèses, par exemple (a b c b). L'ensemble vide et la liste vide sont tous les deux notés NIL.

Les éléments des ensembles et des listes qui ne sont ni des ensembles, ni des listes sont appelés atomes (par exemple: TOUR, TR1, CARRE, 2). Par définition, si le premier caractère d'un atome est "\$" (par exemple: \$TT, \$BLC), alors l'atome est une variable, sinon il est une constante.

Une expression est soit un ensemble, soit une liste, soit un atome. Elle est dite constante si elle ne contient aucune variable.

1.2. Substitutions et instances

Une substitution σ est un ensemble $\{(v_1 c_1)(v_2 c_2)\dots(v_n c_n)\}$, éventuellement vide, où v_i et c_i ($i=1$ à n) désignent respectivement une variable et une constante. De plus, il n'y a pas deux couples $(v_i c_i)$ et $(v_j c_j)$ de σ tels que v_i et v_j désignent la même variable ($i \neq j$).

Si le couple $(v c)$ appartient à une substitution σ , c est appelé le substitut de v dans σ .

* règles de construction de modèles

L'application d'une substitution $\sigma = \{(v_1 c_1)(v_2 c_2) \dots (v_n c_n)\}$ à une expression E est appelée instanciation de E par σ . Elle consiste à remplacer dans E toutes les occurrences de v_i ($i=1$ à n) par c_i . L'expression obtenue est notée $E:\sigma$ et est appelée instance de E par σ .

Exemple:

L'instance de $E = \{(TYPE \$TT TOIT)(FORME \$FT \$F)\}$ par $\sigma = \{(\$TT TT1)(\$F PRISMATIQUE)\}$ est $E:\sigma = \{(TYPE TT1 TOIT)(FORME TT1 PRISMATIQUE)\}$.

2. Les principaux concepts des modèles construits par Tropic et leur expression formelle

2.1. Les principaux concepts

Le domaine d'application de Tropic est la construction de modèles de systèmes physiques. Parmi les données qui constituent ces modèles, nous distinguons les objets et les relations:

- un objet peut désigner une entité physique (par exemple, un transformateur, un bobinage, une tour, un bloc), une couleur, une forme, un nombre etc.. En fait, il n'y a rien qui ne puisse être considéré a priori comme un objet.

- une relation est un prédicat dont les arguments sont des objets, par exemple: "(l'objet) BLC1 repose sur (l'objet) BLC2", "(l'objet) TT1 est un composant de (l'objet) TR1", "la forme de (l'objet) TT1 est (l'objet) PYRAMIDALE", "la couleur de (l'objet) TT1 est (l'objet) ROUGE".

Commentaires:

a/ Les objets et les relations d'un modèle construit par Tropic forment deux ensembles disjoints, ce qui correspond implicitement à travailler dans la logique du 1er ordre.

b/ Les objets et les relations ne sont pas la réalité, mais un point de vue de celle-ci, souvent appelé le "réel perçu".

2.2. Le formalisme

Un modèle est exprimé sous la forme d'un ensemble $M = \{l_1 l_2 \dots l_m\}$ de littéraux. Chaque littéral l_i ($i=1$ à m) de M est une liste $(Pred_i t_{i,1} t_{i,2} \dots t_{i,n_i})$, où $Pred_i, t_{i,1}, t_{i,2}, \dots, t_{i,n_i}$ désignent des constantes.

- . Pred_i est appelé le prédicat de l_i ,
- . $t_{i,j}$, $\forall j \in [1, n_i]$, est appelé le terme de rang j de l_i .

Nous interprétons chaque littéral de M comme la représentation d'une relation, et chaque terme de ce littéral comme la représentation d'un objet figurant dans cette relation.

Exemple:

L'ensemble M ci-dessous représente une tour composée d'une colonne et d'un toit ; la colonne est composée de deux blocs de section ronde et de couleur blanche ; le toit est de forme pyramidale et de couleur rouge.

$M = \{ (\text{TYPE TR1 TOUR})(\text{COMPOSANT TR1 COL1})(\text{COMPOSANT TR1 TT1})(\text{SUR COL1 TT1})$
 $(\text{TYPE COL1 COLONNE})(\text{TYPE TT1 TOIT})(\text{NOMBRE-DE-BLOCS COL1 2})(\text{COMPOSANT COL1 BLC1})(\text{COMPOSANT COL1 BLC2})(\text{SUR BLC1 BLC2})(\text{TYPE BLC1 BLOC})(\text{TYPE BLC2 BLOC})$
 $(\text{SECTION BLC1 RONDE})(\text{COULEUR BLC1 BLANCHE})(\text{SECTION BLC2 RONDE})(\text{COULEUR BLC2 BLANCHE})(\text{FORME TT1 PYRAMIDALE})(\text{COULEUR TT1 ROUGE}) \}$.

3. Les règles de construction de modèles

Les rcm contenues dans l'énoncé d'un problème déterminent implicitement l'ensemble de tous les modèles que Tropic pourra contruire pendant la résolution du problème. Chaque rcm est un opérateur dont l'application permet:

- . d'ajouter de nouvelles informations à la partie de modèle déjà construite,
- . de décomposer le problème défini par l'énoncé, ou un de ses sous-problèmes, en un ensemble de sous-problèmes plus élémentaires.

Un modèle est construit en appliquant des rcm successives jusqu'à ce que le problème défini par l'énoncé soit réduit à l'ensemble vide de sous-problème.

Pour éviter de possibles confusions nous appellerons désormais :

- . problème original, le problème qui est déterminé par l'énoncé (rcm et contraintes) fourni par le concepteur,
- . problèmes formels, les expressions qui sont utilisées dans les rcm pour représenter le problème original et ses sous-problèmes,
- . problème initial, le problème formel (unique) qui représente le problème original.

Un problème formel est une liste $(Pb \ t_1 \ t_2 \ \dots \ t_n)$, $n \geq 0$, où Pb est une constante appelée le nom du problème formel et t_i , $\forall i \in [1, n]$, est un atome (constante ou variable).

Exemples : (CONCEVOIR-TOUR), (ENGENDRER-COMPOSANTS-TOUR \$TR) (DETERMINER-FORME-TOIT \$TT).

Le problème initial doit être spécifié par l'utilisateur. Dans les exemples à venir ce sera (CONCEVOIR-TOUR).

3.1. Définition des rcm

Une rcm est une expression de la forme

$$(Prob \rightarrow (Littx \ Décomp))$$

où:

- . *Prob* est un problème formel,
- . *Littx* est un ensemble de littéraux,
- . *Décomp* est un ensemble (éventuellement vide) de problèmes formels.

Les expressions *Prob* et $(Littx \ Décomp)$ sont respectivement appelées partie gauche et partie droite de la rcm.

Les rcm contenues dans l'énoncé d'un problème déterminent un espace Π de possibilités que nous définissons ainsi:

- une possibilité de Π est un couple $(m \ d)$ où m est un ensemble de littéraux et d un ensemble de problèmes formels.

- une possibilité $(m' \ d')$ dérive directement d'une possibilité $(m \ d)$ si et seulement si il existe une instance de rcm, ou ircm, $(Prob:\sigma \rightarrow (Littx:\sigma \ Décomp:\sigma))$ telle que:

- . $Prob:\sigma \in d$,
- . $m' = m \cup Littx:\sigma$,
- . $d' = (d - \{Prob:\sigma\}) \cup Décomp:\sigma$.

- une possibilité $(m' \ d')$ dérive d'une possibilité $(m \ d)$ si et seulement si il existe une suite de possibilités $(m_1 \ d_1)$, $(m_2 \ d_2)$, ..., $(m_n \ d_n)$, $n \geq 1$ telles que:

- . $(m_1 \ d_1) = (m \ d)$,
- . $(m_n \ d_n) = (m' \ d')$,
- . $\forall i \in [2, n]$, $(m_i \ d_i)$ dérive directement de $(m_{i-1} \ d_{i-1})$.

- $\Pi = \{(m \ d) \ / \ (m \ d) \text{ dérive de } (NIL \ \{po\})\}$ où po désigne le problème initial.

L'ensemble μ de modèles implicitement déterminé par les rcm est $\mu = \{M / (M \text{ NIL}) \in \Pi\}$.

Commentaires:

a/ si $(m \ d)$ et $(m' \ d')$ sont deux possibilités de Π telles que $(m' \ d')$ dérive de $(m \ d)$, alors $m \subseteq m'$.

b/ les possibilités de la forme $(M \ \text{NIL})$ que l'on peut dériver d'une possibilité $(m \ d)$ de Π ne dépendent pas de l'ordre de décomposition des problèmes formels contenus dans d .

c/ une rcm peut être de la forme $(\text{Prob} \rightarrow \text{Fct})$, où Fct est une fonction. La partie droite de l'instance par σ d'une telle rcm est obtenue en évaluant la fonction après avoir donné la valeur c à v , $\forall (v \ c) \in \sigma$. Dans la suite, nous noterons $(\text{Littx}:\sigma \ \text{Décomp}:\sigma)$ la partie droite de l'instance par σ d'une rcm, qu'elle ait été obtenue par instanciation d'une expression ou par évaluation d'une fonction.

3.2. Exemple

Les 16 rcm énoncées ci-dessous permettent de construire des modèles de tour. Les commentaires suivants devraient faciliter leur lecture:

- Chaque modèle qu'il est possible de construire à l'aide de ces rcm représente une tour composée d'une colonne et d'un toit ; la colonne est composée d'un ou de deux blocs ; chaque bloc a une section carrée ou ronde, et est de couleur blanche ou verte ; la forme du toit est pyramidale ou prismatique, et sa couleur est rouge ou noire.

- Les rcm 1, 2 et 6 sont de la forme $(\text{Prob} \rightarrow \text{Fct})$, où Fct est une fonction.

- $!$ est une fonction de Tropic qui engendre de nouvelles constantes en concaténant une "racine" (par exemple: TR, COL, BLC) et un nombre entier ; ainsi l'évaluation répétée de $! \ \text{BLC}$ produira successivement BLC1, BLC2, ...

- $I?$ est une fonction de Tropic qui effectue l'instanciation de l'expression qu'elle prend pour argument ; ainsi, si $\$TR$, $\$COL$ et $\$TT$ ont respectivement TR1, COL1 et TT1 pour valeur, alors $I? \{(COMPOSANT \ \$TR \ \$COL) (COMPOSANT \ \$TR \ \$TT)(SUR \ \$COL \ \$TT)\}$ s'évalue à $\{(COMPOSANT \ TR1 \ COL1) (COMPOSANT \ TR1 \ TT1)(SUR \ COL1 \ TT1)\}$.

- Dans le tableau 2.1., nous utilisons ces rcm pour construire le modèle M donné en exemple au paragraphe 2.2.

rcm 1

((CONCEVOIR-TOUR)

→ Fonct: \$TR ← ! TR

I?({(TYPE \$TR TOUR)}

{(ENGENDRER-COMPOSANTS-TOUR \$TR)}))

rcm 2

((ENGENDRER-COMPOSANTS-TOUR \$TR)

→ Fonct: \$COL ← ! COL

\$TT ← ! TT

I? ({(COMPOSANT \$TR \$COL)(COMPOSANT \$TR \$TT)
(SUR \$COL \$TT)}

{(CONCEVOIR-COLONNE \$COL)(CONCEVOIR-TOIT \$TT)}))

rcm 3

((CONCEVOIR-COLONNE \$COL)

→ ({(TYPE \$COL COLONNE)}

{(DETERMINER-NOMBRE-DE-BLOCS \$COL)}))

rcm 4

((DETERMINER-NOMBRE-DE-BLOCS \$COL)

→ ({(NOMBRE-DE-BLOCS \$COL 1)}

{(ENGENDRER-COMPOSANTS-COLONNE \$COL 1)}))

rcm 5

((DETERMINER-NOMBRE-DE-BLOCS \$COL)

→ ({(NOMBRE-DE-BLOCS \$COL 2)}

{(ENGENDRER-COMPOSANTS-COLONNE \$COL 2)}))

rcm 6

((ENGENDRER-COMPOSANTS-COLONNE \$COL \$N)

→ Fonct: si \$N = 1

alors \$BLC ← ! BLC

```

I? {{(COMPOSANT $COL $BLC)}
    {(CONCEVOIR-BLOC $BLC)}}
sinon $BLC1 ← ! BLC
      $BLC2 ← ! BLC
I? {{(COMPOSANT $COL $BLC1){COMPOSANT $COL $BLC2}
    (SUR $BLC1 $BLC2)}}
    {(CONCEVOIR-BLOC $BLC1){CONCEVOIR-BLOC $BLC2}}))

```

rcm 7

```

((CONCEVOIR-BLOC $BLC)
 → {{(TYPE $BLC BLOC)}
    {(DETERMINER-SECTION-BLOC $BLC)
    (DETERMINER-COULEUR-BLOC $BLC)}}))

```

rcm 8

```

((DETERMINER-SECTION-BLOC $BLC)
 → {{(SECTION $BLC CARREE)} NIL}))

```

rcm 9

```

((DETERMINER-SECTION-BLOC $BLC)
 → {{(SECTION $BLC RONDE)} NIL}))

```

rcm 10

```

((DETERMINER-COULEUR-BLOC $BLC)
 → {{(COULEUR $BLC BLANCHE)} NIL}))

```

rcm 11

```

((DETERMINER-COULEUR-BLOC $BLC)
 → {{(COULEUR $BLC VERTE)} NIL}))

```

rcm 12

```

((CONCEVOIR-TOIT $TT)
 → {{(TYPE $TT TOIT)}
    {(DETERMINER-FORME-TOIT$TT)(DETERMINER-COULEUR-TOIT $TT)}}))

```


rcm 13

```
((DETERMINER-FORME-TOIT $TT)
  → ({{(FORME $TT PYRAMIDALE)}} NIL))
```

rcm 14

```
((DETERMINER-FORME-TOIT $TT)
  → ({{(FORME $TT PRISMATIQUE)}} NIL))
```

rcm 15

```
((DETERMINER-COULEUR-TOIT $TT)
  → ({{(COULEUR $TT ROUGE)}} NIL))
```

rcm 16

```
((DETERMINER-COULEUR-TOIT $TT)
  → ({{(COULEUR $TT NOIRE)}} NIL))
```

Tableau 2.1.

IRCM APPLIQUEE	m	POSSIBILITE ENGENDEE d
	NIL	{ (CONCEVOIR-TOUR)}
rcm 1 : NIL	{ (TYPE TR1 TOUR)}	{ (ENGENDRER-COMPOSANTS-TOUR TR1)}
rcm 2 : { (\$TR TR1)}	{ (TYPE TR1 TOUR){COMPOSANT TR1 COL1} (COMPOSANT TR1 TT1){SUR COL1 TT1}}	{ (CONCEVOIR-COLONNE COL1) (CONCEVOIR-TOIT TT1)}
rcm 3 : { (\$COL COL1)}	{ (TYPE TR1 TOUR)...(SUR COL1 TT1) (TYPE COL1 COLONNE)}	{ (DETERMINER-NOMBRE-DE-BLOCS COL1) (CONCEVOIR-TOIT TT1)}
rcm 5 : { (\$COL COL1)}	{ (TYPE TR1 TOUR)...(TYPE COL1 COLONNE) (NOMBRE-DE-BLOCS COL1 2)}	{ (ENGENDRER-COMPOSANTS-COLONNE COL1 2) (CONCEVOIR-TOIT TT1)}
rcm 12 : { (\$TT TT1)}	{ (TYPE TR1 TOUR)...(NOMBRE-DE-BLOCS COL1 2) (TYPE TT1 TOIT)}	{ (ENGENDRER-COMPOSANTS-COLONNE COL1 2) (DETERMINER-FORME-TOIT TT1) (DETERMINER-COULEUR-TOIT TT1)}
rcm 13 : { (\$TT TT1)}	{ (TYPE TR1 TOUR)...(TYPE TT1 TOIT) (FORME TT1 PYRAMIDALE)}	{ (ENGENDRER-COMPOSANTS-COLONNE COL1 2) (DETERMINER-COULEUR-TOIT TT1)}
rcm 15 : { (\$TT TT1)}	{ (TYPE TR1 TOUR)...(FORME TT1 PYRAMIDALE) (COULEUR TT1 ROUGE)}	{ (ENGENDRER-COMPOSANTS-COLONNE COL1 2)}
rcm 6 : { (\$COL COL1) (\$N 2)}	{ (TYPE TR1 TOUR)...(COULEUR TT1 ROUGE) (COMPOSANT COL1 BLC1) (COMPOSANT COL1 BLC2) (SUR BLC1 BLC2)}	{ (CONCEVOIR-BLOC BLC1) (CONCEVOIR-BLOC BLC2)}

Tableau 2.1.(suite)

IRCM APPLIQUEE	m	POSSIBILITE ENGENDREE d
rcm 7 : {(\$BLC BLC1)}	{(TYPE TR1 TOUR)...(SUR BLC1 BLC2) (TYPE BLC1 BLOC)}	{(DETERMINER-SECTION-BLOC BLC1) (DETERMINER-COULEUR-BLOC BLC1) (CONCEVOIR-BLOC BLC2)}
rcm 9 : {(\$BLC BLC1)}	{(TYPE TR1 TOUR)...(TYPE BLC1 BLOC) (SECTION BLC1 RONDE)}	{(DETERMINER-COULEUR-BLOC BLC1) (CONCEVOIR-BLOC BLC2)}
rcm 10 : {(\$BLC BLC1)}	{(TYPE TR1 TOUR)...(SECTION BLC1 CARREE) (COULEUR BLC1 BLANCHE)}	{(CONCEVOIR-BLOC BLC2)}
rcm 7 : {(\$BLC BLC2)}	{(TYPE TR1 TOUR)...(COULEUR BLC1 BLANCHE) (TYPE BLC2 BLOC)}	{(DETERMINER-SECTION-BLOC BLC2) (DETERMINER-COULEUR-BLOC BLC2)}
rcm 9 : {(\$BLC BLC2)}	{(TYPE TR1 TOUR)...(TYPE BLC2 BLOC) (SECTION BLC2 RONDE)}	{(DETERMINER-COULEUR-BLOC BLC2)}
rcm 10 : {(\$BLC BLC2)}	{(TYPE TR1 TOUR)(COMPOSANT TR1 COL1) (COMPOSANT TR1 TT1)(SUR COL1 TT1) (TYPE COL1 COLONNE)(NOMBRE-DE-BLOCS COL1 2) (TYPE TT1 TOIT)(FORME TT1 PYRAMIDALE) (COULEUR TT1 ROUGE)(COMPOSANT COL1 BLC1) (COMPOSANT COL1 BLC2)(SUR BLC1 BLC2) (TYPE BLC1 BLOC)(SECTION BLC1 RONDE) (COULEUR BLC1 BLANCHE)(TYPE BLC2 BLOC) (SECTION BLC2 RONDE)(COULEUR BLC2 BLANCHE)}	NIL

3.3. Notion de rcm conditionnelle

Une rcm conditionnelle est une expression

$$((\text{Prob Cond}) \rightarrow (\text{Littx D\`ecomp}))$$

où *Cond* est un ensemble de littéraux qui décrivent les conditions d'applicabilité de la rcm (cf. §4.1. et 4.2.).

Exemple :

Les 4 rcm énoncées ci-dessous se rapportent à l'exemple du transformateur. La quatrième rcm est une rcm conditionnelle dont l'application dépend du réfrigérant choisi.

rcm 1.

```
((CONCEVOIR-TRANSFORMATEUR)
  → Fonct: $TRSF ← ! TRSF
      I? ({(TYPE $TRSF TRANSFORMATEUR)}
          {(DETERMINER-REFRIGERANT $TRSF)}))
```

rcm 2

```
((DETERMINER-REFRIGERANT $TRSF)
  → ({(REFRIGERANT $TRSF AIR)} NIL))
```

rcm 3

```
((DETERMINER-REFRIGERANT $TRSF)
  → ({(REFRIGERANT $TRSF HUILE)} NIL))
```

rcm 4

```
((ENGENDRER-COMPOSANTS-TRANSFORMATEUR $TRSF)
  {(REFRIGERANT $TRSF $REF)})
  → Fonct: $CM ← ! CM
          $BOB ← ! BOB
          Si $REF = AIR
              alors I? ({(COMPOSANT $TRSF $CM)
                          (COMPOSANT $TRSF $BOB)}
                        {(CONCEVOIR-CIRCUIT-MAGNETIQUE $CM)
                          (CONCEVOIR-BOBINAGE $BOB)})
```

```

sinon $CUVE ← ! CUVE
I? ({(COMPOSANT $TRSF $CM)
      (COMPOSANT $TRSF $BOB)
      (COMPOSANT $TRSF $CUVE)})
  {(CONCEVOIR-CIRCUIT-MAGNETIQUE $CM)
    (CONCEVOIR-BOBINAGE $BOB)
    (CONCEVOIR-CUVE $CUVE)}})

```

Les rcm conditionnelles sont dans certains cas un moyen d'expression plus naturel que les rcm non conditionnelles. Toutefois, leur traitement présente des difficultés dues à ce que la condition énoncée dans le commentaire b du paragraphe 3.1. peut ne plus être vérifiée. Comme la version actuelle de Tropic ne permet de les utiliser qu'avec de nombreuses précautions, nous ignorerons les rcm conditionnelles jusqu'au chapitre 8.

4. Les contraintes

Les contraintes décrivent les conditions qu'un modèle doit satisfaire pour être une solution du problème original. Elles sont exprimées à l'aide de littéraux plus généraux que ceux qui servent à exprimer les modèles. Ces littéraux constituent les éléments d'un langage de discours sur les modèles que les rcm permettent de construire.

4.1. Généralisation des littéraux

Un littéral est une liste $(Pred\ t_1\ t_2 \dots t_n)$ ou $(Pfx\ Pred\ t_1\ t_2 \dots t_n)$, où:

- . Pfx est l'un des atomes suivants: NON, PG (pour "Plus Grand") et PP (pour "Plus Petit") ; il est appelé le préfixe de l ;
- . $Pred$ est une constante différente de NON, PG et PP ; il est appelé le prédicat de l ;
- . $t_i, \forall i \in [1, n]$, est un atome ; il est appelé le terme de rang i de l.

Exemples:

- . (NON FORME \$TT PRISMATIQUE) [*lire: "la forme de \$TT n'est pas prismatique"*].
- . (PG NOMBRE-DE-BLOCS \$COL 1) [*lire: "le nombre de blocs de \$COL est plus grand que 1"*].

Nous disons qu'un littéral est:

- . positif, s'il n'a pas de préfixe,
- . négatif, si son préfixe est NON,
- . supérieur, si son préfixe est PG,
- . inférieur, si son préfixe est PP.

Les littéraux supérieurs et inférieurs doivent avoir deux termes et deux seulement. Leurs termes de rang 2, lorsqu'ils sont notés par des constantes, doivent être des nombres.

4.2. Interprétation logique des littéraux constants

Tropic interprète un littéral l constant comme une formule logique atomique ayant la valeur VRAI ou FAUX dans un modèle M donné. Cette valeur est définie ainsi:

- . Si l est positif et si $l \in M$, alors l a la valeur VRAI dans M .

Exemple: (COULEUR BLC1 BLANCHE) a la valeur VRAI dans le modèle M donné en exemple au paragraphe 2.2.

- . Si l est positif et si $l \notin M$, alors l a la valeur FAUX dans M .

Exemple: (FORME TT1 PRISMATIQUE) a la valeur FAUX dans M .

- . Si l est un littéral négatif (NON Pred $t_1 t_2 \dots t_n$) et si (Pred $t_1 t_2 \dots t_n$) $\notin M$, alors l a la valeur VRAI dans M .

Exemple: (NON COULEUR BLC1 VERTE) a la valeur VRAI dans M .

- . Si l est un littéral négatif (NON Pred $t_1 t_2 \dots t_n$) et si (Pred $t_1 t_2 \dots t_n$) $\in M$, alors l a la valeur FAUX dans M .

Exemple: (NON FORME TT1 PYRAMIDALE) a la valeur FAUX dans M .

. Si l est un littéral supérieur (PG Pred $t_1 t_2$) et si il existe dans M un littéral (Pred $t_1 t'_2$) tel que $t'_2 > t_2$, alors l a la valeur VRAI dans M .

Exemple: (PG NOMBRE-DE-BLOCS COL1 1) a la valeur VRAI dans M .

. Si l est un littéral supérieur (PG Pred $t_1 t_2$) et si il n'existe pas dans M un littéral (Pred $t_1 t'_2$) tel que $t'_2 > t_2$, alors l a la valeur FAUX dans M .

Exemple: (PG NOMBRE-DE-BLOCS COL1 5) a la valeur FAUX dans M .

. Si l est un littéral inférieur (PP Pred $t_1 t_2$) et si il existe dans M un littéral (Pred $t_1 t'_2$) tel que $t'_2 < t_2$, alors l a la valeur VRAI dans M .

Exemple: (PP NOMBRE-DE-BLOCS COL1 5) a la valeur VRAI dans M .

. Si l est un littéral inférieur (PP Pred $t_1 t_2$) et si il n'existe pas dans M un littéral (Pred $t_1 t'_2$) tel que $t'_2 < t_2$, alors l a la valeur FAUX dans M .

Exemple: (PP NOMBRE-DE-BLOCS COL1 2) a la valeur FAUX dans M .

4.3. Définition des contraintes

Une contrainte est une expression de la forme

où $(Activ \rightarrow Restrict)$

- . *Activ* est un ensemble de littéraux qui représentent les conditions d'activation de la contrainte,
- . *Restrict* est un ensemble de littéraux qui représentent les restrictions imposées par la contrainte, ou une fonction qui s'évalue à un ensemble de littéraux représentant ces restrictions.

Les expressions *Activ* et *Restrict* sont respectivement appelées partie gauche et partie droite de la contrainte.

L'instance par σ d'une contrainte $(Activ \rightarrow Restrict)$ est dite active dans un modèle M si et seulement si tous les littéraux de $Activ:\sigma$ sont des littéraux constants ayant la valeur VRAI dans M (i.e.: M vérifie toutes les conditions que ces littéraux représentent).

Un modèle M satisfait l'instance par σ d'une contrainte (Activ \rightarrow Restrict) si et seulement si l'une des deux propositions suivantes est vraie:

- . (Activ: $\sigma \rightarrow$ Restrict: σ) n'est pas active dans M ,
- . tous les littéraux de Restrict: σ ont la valeur VRAI dans M (i.e.: M satisfait les restrictions que ces littéraux représentent).

[Note: la partie droite d'une instance de contrainte qui est active dans un modèle ne doit contenir que des littéraux constants. Tropic ignore la restriction représentée par un littéral qui ne vérifie pas cette condition, et en avertit le concepteur.]

Un modèle M satisfait une contrainte si et seulement si il satisfait toutes les instances de cette contrainte.

Exemples:

- . (((TYPE \$TR TOUR) (COMPOSANT \$TR \$COL) (COMPOSANT \$TR \$TT) (TYPE \$COL COLONNE) (TYPE \$TT TOIT) (COMPOSANT \$COL \$BLC) (TYPE \$BLC BLOC) (FORME \$TT \$F))
 → Fonct: si \$F = PYRAMIDALE alors I? { (SECTION \$BLC CARREE) }
 sinon I? { (SECTION \$BLC RONDE) })

[Lire: "si la forme du toit d'une tour est pyramidale, alors la section de chaque bloc de la colonne de cette tour doit être carrée, sinon elle doit être ronde"].

Seules les instances de cette contrainte par { (\$TR TR1) (\$COL COL1) (\$TT TT1) (\$BLC BLC1) (\$F PYRAMIDALE) } et par { (\$TR TR1) (\$COL COL1) (\$TT TT1) (\$BLC BLC2) (\$F PYRAMIDALE) } sont actives dans le modèle M donné en exemple au paragraphe 2.2. ; elles sont toutes les deux satisfaites par M . La contrainte est donc satisfaite par M .

- . (((TYPE \$COL COLONNE) (COMPOSANT \$COL \$BLC1) (COMPOSANT \$COL \$BLC2) (SUR \$BLC1 \$BLC2) (TYPE \$BLC1 BLOC) (TYPE \$BLC2 BLOC) (SECTION \$BLC1 \$S))
 → { (NON SECTION \$BLC2 \$S) })

[Lire: "si deux blocs sont composants d'une même colonne, l'un reposant sur l'autre, ils doivent avoir des sections différentes".]

Seule l'instance de cette contrainte par { (\$COL COL1) (\$BLC1 BLC1) (\$BLC2 BLC2) (\$S CARREE) } est active dans M ; elle n'est pas satisfaite par M . La contrainte n'est donc pas satisfaite par M .

Un modèle M est une solution du problème original si et seulement si il satisfait toutes les contraintes qui sont contenues dans l'énoncé de ce problème.

Chapitre 3: LA RECHERCHE D'UNE SOLUTION

La résolution d'un problème consiste pour Tropic à rechercher une solution dans l'espace des possibilités qui est déterminé par les rcm. Les composantes de cette recherche, décrites aux paragraphes 1, 2 et 3, sont le cheminement avant, les échecs et les cheminements arrière:

- . le cheminement avant consiste à appliquer des ircm successives jusqu'à ce qu'une solution ait été engendrée,
- . les échecs sont des situations dans lesquelles Tropic reconnaît que le cheminement avant est engagé dans une impasse,
- . Les cheminements arrière ont pour but de sortir le cheminement avant des impasses dans lesquelles il est engagé.

Au paragraphe 4, nous énonçons l'algorithme simplifié de la recherche effectuée par Tropic. Au paragraphe 5, nous comparons le modèle de résolution mis en oeuvre avec les modèles de transformation d'état et de décomposition de problèmes.

1. Le cheminement avant

Le cheminement avant procède par expansions successives d'une arborescence ET/OU (Nilsson 1971), dite arborescence de construction de modèles, ou plus brièvement acm. A chaque instant, cette arborescence représente la décomposition courante du problème initial effectuée par Tropic à l'aide des rcm. Les problèmes formels engendrés par cette décomposition sont associés aux sommets-ET de l'acm et les ircm appliquées sont associées à ses sommets-OU. La figure 3.1 montre l'acm qui correspond à la décomposition du tableau 2.1. (l'ordre de décomposition des problèmes formels est indiqué entre parenthèses à côté des sommets-ET).

Au début de la résolution, l'acm ne contient que sa racine (un sommet-ET à laquelle est associé le problème initial. Ensuite, chaque expansion de l'acm consiste à décomposer un problème Pb associé à un sommet-ET N pendant (i.e.: sans successeur) de l'acm en lui appliquant une ircm $i = (Prob: \sigma \rightarrow (Littx: \sigma \text{ Décomp: } \sigma))$ telle que $Prob: \sigma = Pb$. Les opérations suivantes sont alors effectuées

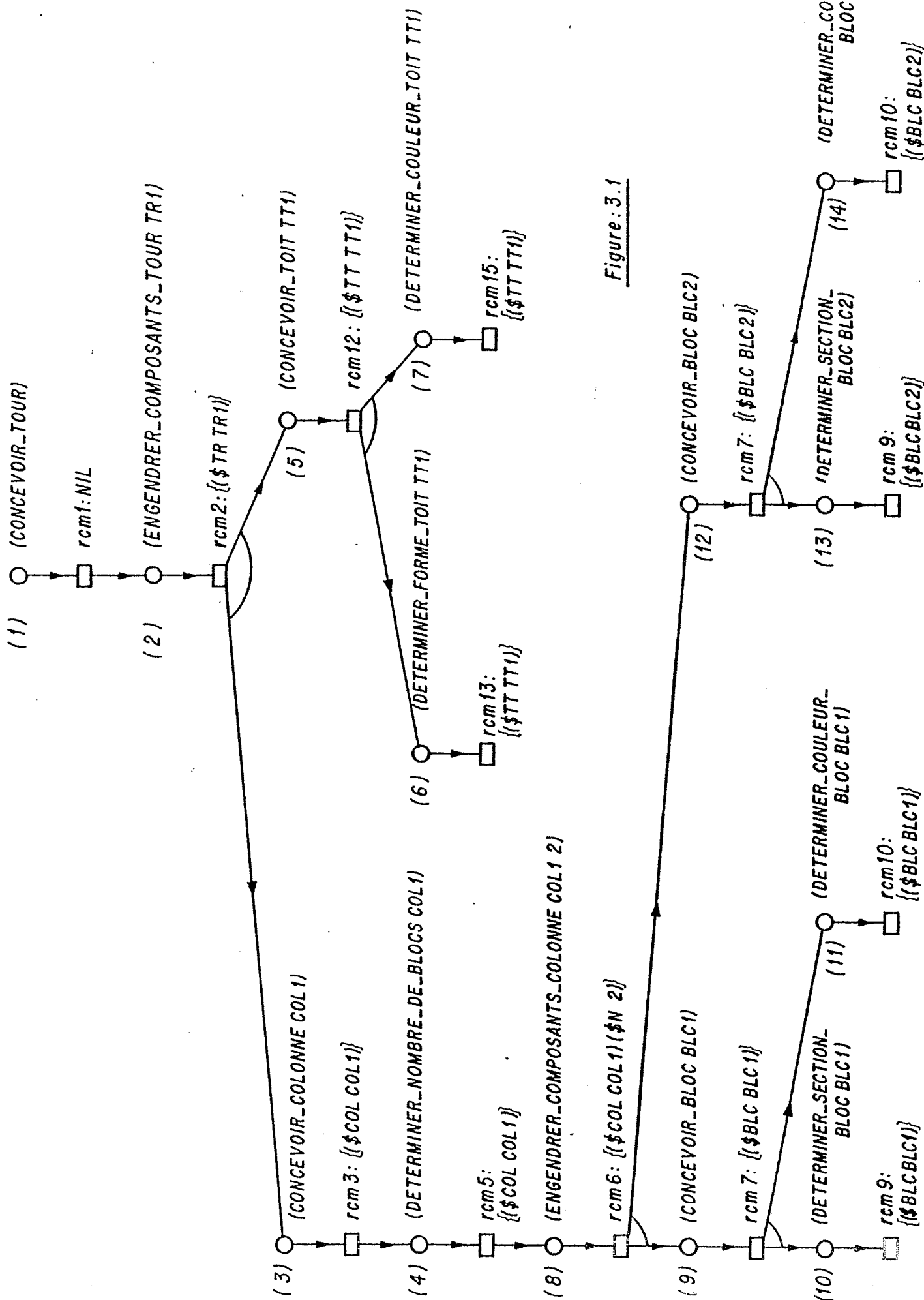


Figure: 3.1

- . l'ircm i est associée à un sommet-OU N' qui est ajouté à l'acm comme successeur immédiat du sommet N ,
- . chaque problème formel contenu dans $\text{Décomp}:\sigma$ est associé à un sommet-ET qui est ajouté à l'acm comme successeur immédiat de N' (si $\text{Décomp}:\sigma = \text{NIL}$, N' reste sans successeur).

De plus, à chaque expansion de l'acm, les littéraux contenus dans la partie droite de l'ircm appliquée sont rangés comme nouveaux éléments d'un ensemble initialement vide, appelé état.

L'acm est dite finale lorsque tous ses sommets pendants sont des sommets-OU (cas de la figure 3.1). L'état courant est alors un modèle (le modèle M donné en exemple au paragraphe 2.2. du chapitre 2, dans le cas de l'acm de la figure 3.1.). C'est une solution s'il satisfait toutes les contraintes du problème.

Commentaires

a/ Nous décrirons au chapitre 4 les heuristiques utilisées par Tropic à chaque expansion de l'acm pour choisir le problème formel à décomposer et l'ircm à appliquer.

b/ Chaque ircm i , qui est applicable à un problème formel P_b associé à un sommet-ET N de l'acm (i.e., i est de la forme $(P_b \rightarrow \dots)$), est associée par Tropic à un sommet-OU N' successeur immédiat de N . Par convention, si N' ne fait pas partie de l'acm courante, nous traçons l'arc qui relie N à N' en pointillé (cf. figure 3.2, où l'acm est celle de la figure 3.1.). Tropic utilise ces sommets-OU pour conserver des informations sur les échecs déjà subis (cf. les indicateurs d'échec, chapitre 6).

c/ Soit m l'état courant à un instant quelconque du cheminement avant. Nous imposons que chaque littéral de m ait été engendré par l'application d'une ircm unique, ce qui implique qu'il n'existe pas deux sommets-ET distincts de l'acm courante auxquels est associé le même problème formel. Si, pendant la résolution d'un problème, Tropic constate que cette condition n'est pas satisfaite, il en avertit l'utilisateur par un message.

d/ Soit α l'acm courante à un instant quelconque du cheminement avant. Nous imposons qu'il n'y ait pas deux problèmes formels de même nom associés à deux sommets-ET situés sur un même chemin de α . Cette condition a pour but d'empêcher Tropic de s'engager dans la construction d'une acm infinie. Toutefois, il pourrait être intéressant lors d'une recherche future d'ajouter à Tropic un module de traitement qui permette d'éviter cette condition.

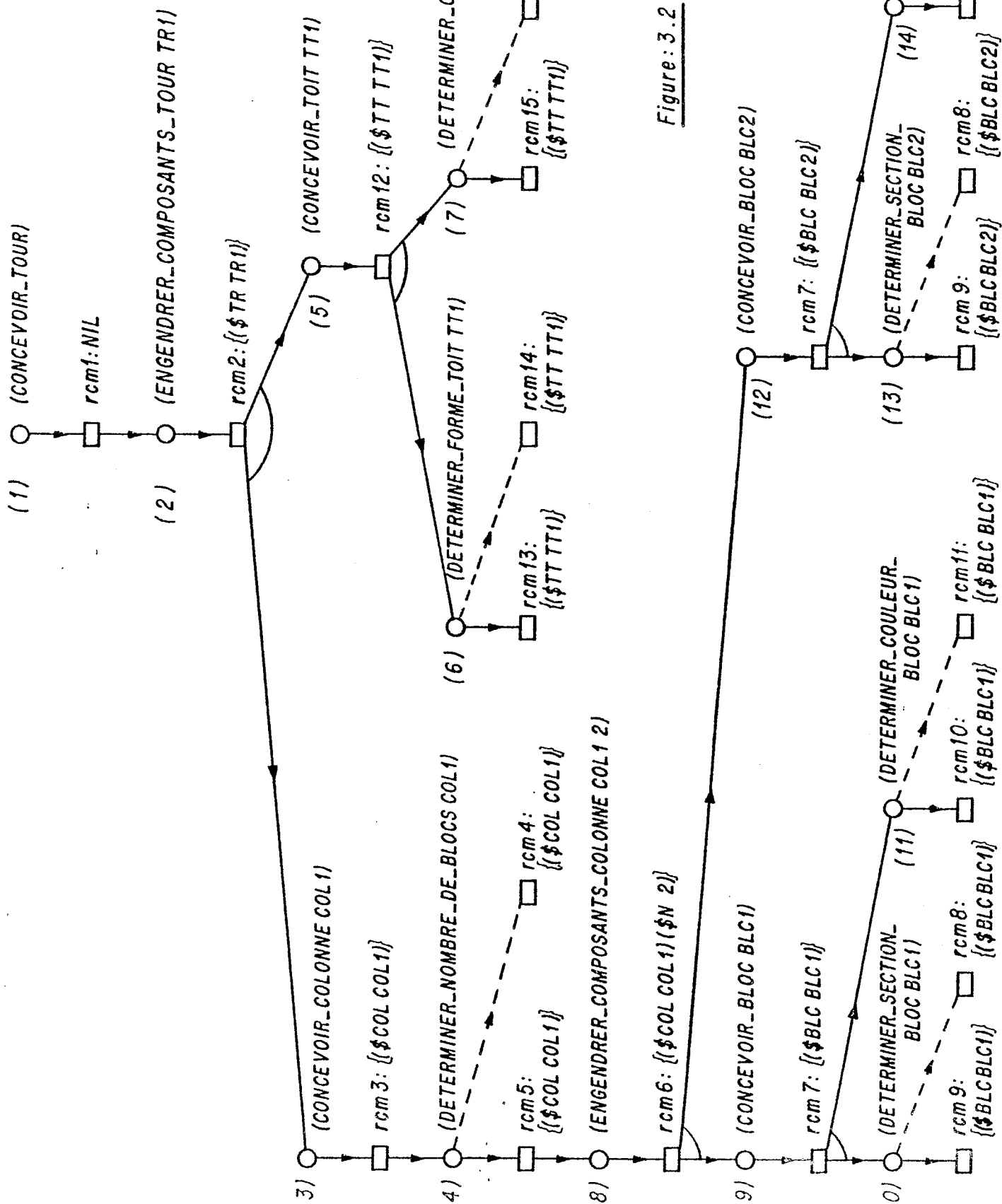


Figure : 3.2

2. Les échecs

Soient α et m l'acm et l'état courants à un instant quelconque du cheminement avant. Nous notons $\mu(\alpha)$ l'ensemble des modèles qu'il est possible de construire en poursuivant le cheminement avant jusqu'à l'obtention d'une acm finale. $\forall M \in \mu(\alpha)$, on a: $m \subseteq M$ (cf. commentaire a, §3.1., chapitre 2).

Nous définissons la valeur logique d'un littéral constant l dans l'état m de telle sorte que si l a la valeur VRAI (resp. FAUX) dans m , alors l aura la valeur VRAI (resp. FAUX) dans M , $\forall M \in \mu(\alpha)$. Ainsi:

- . si l est positif et si $l \in m$, alors l a la valeur VRAI dans m ,
 - . si l est un littéral négatif (NON Pred $t_1 t_2 \dots t_n$) et si (Pred $t_1 t_2 \dots t_n$) $\in m$, alors l a la valeur FAUX dans m ,
 - . si l est un littéral supérieur (PG Pred $t_1 t_2$) et si il existe dans m un littéral (Pred $t_1 t'_2$) tel que $t'_2 > t_2$, alors l a la valeur VRAI dans m ,
 - . si l est un littéral inférieur (PP Pred $t_1 t_2$) et si il existe dans m un littéral (Pred $t_1 t'_2$) tel que $t'_2 < t_2$, alors l a la valeur VRAI dans m .
- Dans tous les autres cas, nous considérerons provisoirement que la valeur de l dans m est indéterminée (cf. commentaire b).

L'instance par σ d'une contrainte (Activ \rightarrow Restrict) est dite active dans l'état m si et seulement si tous les littéraux de Activ: σ sont des littéraux constants ayant la valeur VRAI dans m . Si l'instance de contrainte (Activ: $\sigma \rightarrow$ Restrict: σ) est active dans l'état m , alors les restrictions représentées par les littéraux de Restrict: σ sont dites actives dans m .

L'état m contredit l'instance par σ d'une contrainte (activ \rightarrow Restrict) si et seulement si les deux propositions suivantes sont vraies simultanément:

- . (Activ: $\sigma \rightarrow$ Restrict: σ) est active dans m ,
- . au moins un littéral de Restrict: σ a la valeur FAUX dans m .

L'état m contredit une contrainte si et seulement si il contredit au moins une instance de cette contrainte.

Exemple:

Considérons le tableau 2.1. et la contrainte suivante:

*{(TYPE \$TR TOUR)(COMPOSANT \$TR \$COL)(COMPOSANT \$TR \$TT)(TYPE \$COL COLONNE)
 (TYPE \$TT TOIT)(COMPOSANT \$COL \$BLC)(TYPE \$BLC BLOC)(SECTION \$BLC RONDE)}
 → {(NON FORME \$TT PYRAMIDALE)}.*

*L'état obtenu immédiatement après l'application de l'ircm rem9:{\$BLC BLC1}
 contredit cette contrainte. En effet, l'instance de cette contrainte par
 {\$TR TR1}{\$COL COL1}{\$TT TT1}{\$BLC BLC1} est alors active dans l'état cou-
 rant, et le littéral (NON FORME TT1 PYRAMIDALE) a la valeur FAUX dans cet état*

Si l'état m contredit une contrainte du problème original, aucun des modèles de $\mu(\alpha)$ ne pourra satisfaire cette contrainte. Aucun de ces modèles ne sera donc une solution. Tropic reconnaît alors que le cheminement avant est engagé dans une impasse. Nous disons qu'il est en échec.

Commentaires

a/ Dans les chapitres 4 et 5, nous identifierons plusieurs cas d'échec. Tropic n'attend pas toujours la contradiction effective d'une contrainte pour reconnaître que le cheminement avant est engagé dans une impasse.

b/ Dans le chapitre 4 (§ 2.1.3. et § 2.4.3.), nous énoncerons d'autres règles permettant à Tropic de déterminer la valeur logique d'un littéral constant dans un état donné.

c/ Dans le chapitre 7, nous décrirons l'algorithme qui détermine à chaque instant les instances de contraintes qui sont actives dans l'état courant.

d/ Lorsqu'il n'y aura pas d'ambiguïté, il nous arrivera de dire que la valeur d'un littéral est VRAI, FAUX ou indéterminée, qu'une instance de contrainte est active, ..., sans préciser l'état de référence. En général, cet état sera l'état courant.

3. Les cheminements arrière

Tropic interrompt le cheminement avant chaque fois qu'il est en échec. Il reconsidère alors les choix d'ircm enregistrés dans l'acm et détermine un

sommet-ET N non pendant, dit point de retour, à partir duquel le cheminement avant sera repris avec une autre ircm.

La reprise du cheminement avant est précédée d'un cheminement arrière au point de retour N. Ce cheminement arrière consiste à effacer les effets des ircm associées aux sommets-OU successeurs de N dans l'acm:

- . les littéraux engendrés par les ircm associées aux sommets-OU successeurs de N sont retirés de l'état courant,
- . les sommets-ET et OU successeurs de N et les arcs correspondants sont retirés de l'acm courante.

Si, à la suite d'un échec, aucun point de retour ne peut être déterminé, Tropic considère que l'échec est irrémédiable, et il arrête la résolution du problème (cf. chapitre 6).

Exemples:

Considérons l'acm (figure 3.2.) et l'état M (exemple du § 2.2. du chapitre 2) obtenus à la fin du tableau 2.1., et supposons que Tropic soit alors en échec.

. Supposons que Tropic décide de reprendre le cheminement avant à partir du sommet-ET auquel est associé le problème formel (DETERMINER-FORME-TOIT TT1). Après le cheminement arrière à ce sommet, l'acm est celle de la figure 3.3. et l'état courant est M-{(FORME TT1 PYRAMIDALE)}. Il suffit alors à Tropic d'appliquer l'ircm rcm 14: {(\$TT TT1)} pour obtenir un nouveau modèle.

[Note: Au paragraphe 1.1. du chapitre 6, nous décrirons le traitement d'un échec au cours duquel ce cheminement arrière sera effectué.]

. Supposons que Tropic décide de reprendre le cheminement avant à partir du sommet-ET auquel est associé le problème (DETERMINER-NOMBRE-DE-BLOCS COL1). Après le cheminement arrière à ce sommet, l'acm est celle de la figure 3.4. et l'état courant est M-{(NOMBRE-DE-BLOCS COL1 2)(COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2)(SUR BLC1 BLC2)(TYPE BLC1 BLOC)(SECTION BLC1 RONDE)(COULEUR BLC1 BLANCHE)(TYPE BLC2 BLOC)(SECTION BLC2 RONDE)(COULEUR BLC2 BLANCHE)}.

Commentaires

a/ Le traitement d'un échec comporte deux parties: la détermination d'un point de retour et le cheminement arrière à ce point. La détermination du point de retour résulte d'une analyse des conditions de l'échec, notamment des

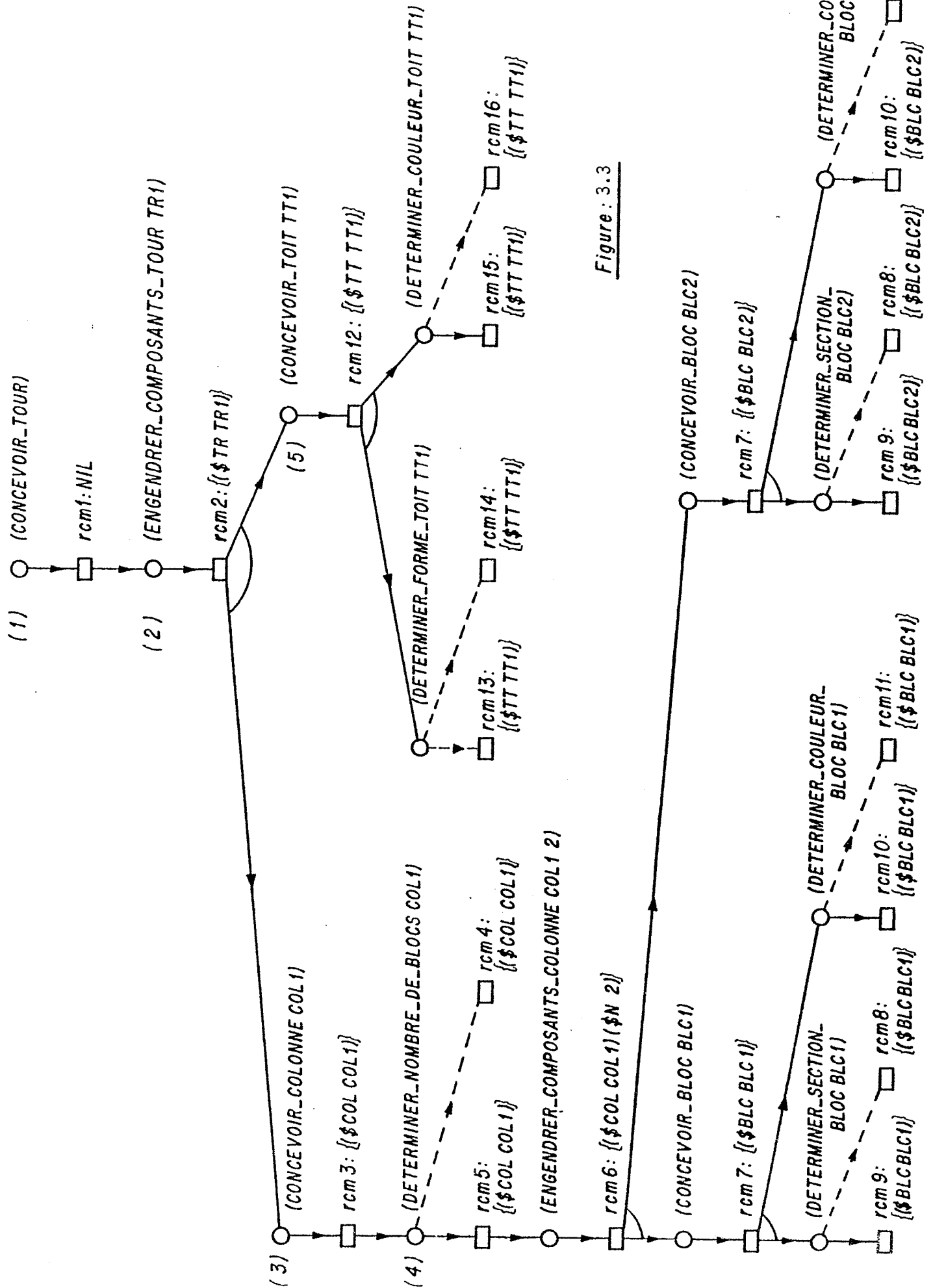


Figure: 3.3

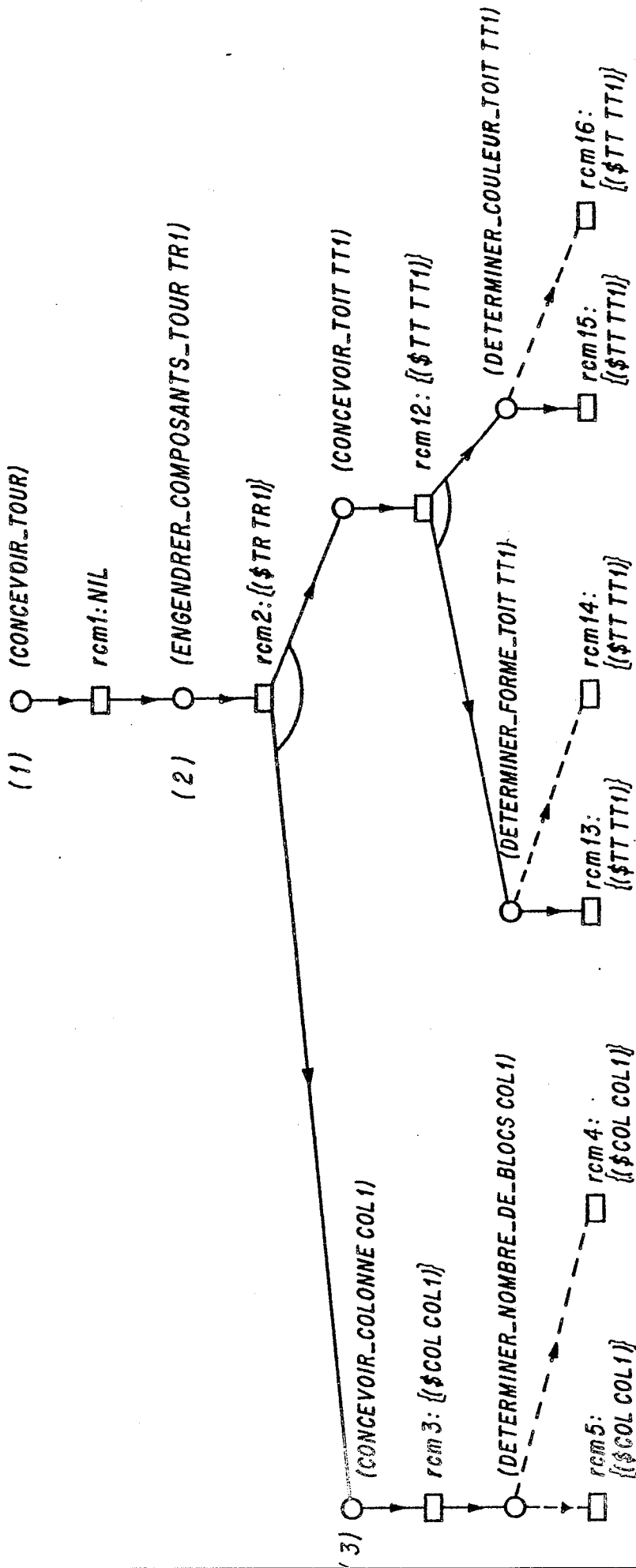


Figure : 3.4

conditions d'activation de l'instance de contrainte qui est contredite (on peut en effet éviter l'échec en rendant cette instance inactive). Une description complète de ce traitement sera faite au chapitre 6.

b/ Après un cheminement arrière, Tropic associe un indicateur d'échec au sommet-OU correspondant à l'ircm dont l'application vient d'être abandonnée (l'ircm rcm 13:{\$TT TT1}) dans le premier exemple donné ci-dessus). Cet indicateur décrit les conditions dans lesquelles cette ircm a conduit à un échec. Tropic pourra l'utiliser plus tard, notamment pour éviter de rencontrer à nouveau cet échec (cf. chapitre 6).

4. Algorithme simplifié de la recherche

A - La racine de l'acm à laquelle est associé le problème initial est engendrée. L'état courant est initialisé à NIL.

B - Le cheminement avant est effectué jusqu'à ce que l'on ait engendré une solution ou rencontré un échec:

- . dans le premier cas, la résolution du problème original s'achève sur un succès,
- . dans le deuxième cas, on exécute C.

C - Si l'analyse des conditions de l'échec permet de déterminer un point de retour, on effectue un cheminement arrière à ce point, puis on reprend l'exécution de l'algorithme en B ; sinon, la résolution du problème original s'achève sur un échec.

5. Comparaison avec les modèles de transformation d'état et de décomposition de problème

Le modèle de résolution de problème mis en oeuvre dans Tropic est un modèle hybride combinant les modèles classiques de transformation d'état et de décomposition de problème. En le comparant à ces deux modèles, nous montrons comment il exploite la décomposabilité des problèmes, tout en tenant compte des interdépendances entre les sous-problèmes engendrés.

5.1. Comparaison avec la transformation d'état

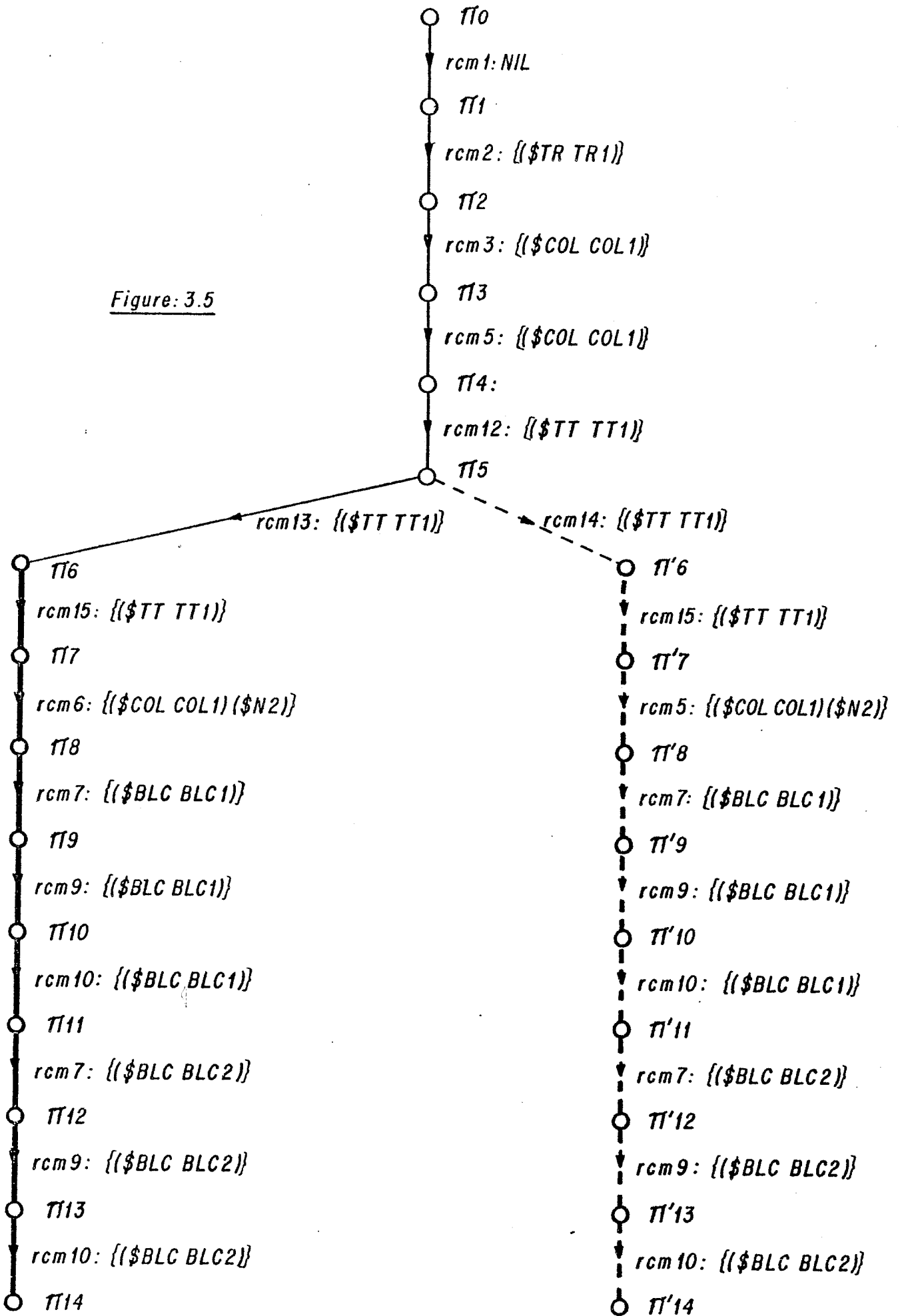
Dans le modèle de transformation d'état, la résolution d'un problème consiste à rechercher un état vérifiant un ensemble X de critères dans un espace E d'états déterminés par un état initial ϵ_0 et un ensemble T d'opérateurs de transformation d'état. Chaque opérateur $\tau \in T$ est une fonction partielle de E dans E . Les choix d'opérateurs effectués pendant la recherche sont enregistrés dans un graphe orienté. A chaque sommet de ce graphe est associé un état et un seul. A chaque arc est associé l'opérateur qui a transformé l'état associé à l'origine de l'arc en l'état associé à l'extrémité de l'arc. Initialement, le graphe ne contient donc qu'un seul sommet auquel est associé l'état initial ϵ_0 . Le problème est résolu lorsque le graphe de recherche contient un sommet auquel est associé un état vérifiant les critères de X .

Soient po le problème initial, R l'ensemble des rcm et C l'ensemble des contraintes contenues dans l'énoncé d'un problème soumis à Tropic. Ce problème peut être résolu suivant le modèle de transformation d'état en considérant que $E = \Pi$ (espace des possibilités déterminé par R), $\epsilon_0 = (NIL\{po\})$, $T = R$ et $X = C$. L'exécution de l'algorithme du paragraphe 4 correspond alors à l'application d'une méthode de recherche du type "profondeur d'abord" ("depht-first method", Nilsson 1971). Toutefois, le graphe engendré ne reflète pas la décomposabilité du problème, ce qui conduit à des redondances dans la recherche.

Un exemple de redondance est illustré par le graphe de la figure 3.5., dans lequel la même séquence de décisions figure deux fois (traits épais). Ce graphe est engendré par un cheminement avant interrompu par un cheminement arrière:

- . le chemin en trait plein correspond à la première partie du cheminement avant décrite par le tableau 2.1.;
- . le chemin en pointillé correspond à la deuxième partie du cheminement avant effectuée après un cheminement arrière au sommet-ET étiqueté par le problème (DETERMINER-FORME-TOIT TT1) ; bien que seule la forme du toit TT1 soit modifiée, les autres choix doivent être effectués à nouveau, car $\forall i \in [6,14]$
 $\pi_i \neq \pi'_i$.

Figure: 3.5



Dans l'acm de la figure 3.2., les choix des ircm rcm 15: {(\$TT TT1)}, rcm 6: {(\$COL COL1)(\$N 2)}, rcm 7: {(\$BLC BLC1)}, rcm 9: {(\$BLC BLC1)}, rcm 10: {(\$BLC BLC1)}, rcm 7: {(\$BLC BLC2)}, rcm 9: {(\$BLC BLC2)} et rcm 10: {(\$BLC BLC2)} sont enregistrés le long des chemins indépendants du problème (DETERMINER-FORME-TOIT TT1). Cette représentation, qui reflète la décomposabilité du problème original, évite à Tropic de refaire ces choix après le cheminement arrière du problème (DETERMINER-FORME-TOIT TT1) (cf. le premier exemple du paragraphe 3).

5.2. Comparaison avec la décomposition de problème

Dans le modèle de décomposition de problème, la résolution d'un problème consiste à rechercher une décomposition d'un problème initial po en un ensemble de problèmes primitifs caractérisés par un ensemble ψ de critères. Cette recherche est effectuée dans un espace P de problèmes formels déterminé par po et un ensemble Δ d'opérateurs de décomposition de problème. Chaque opérateur $\delta \in \Delta$ est une fonction partielle de P dans 2^P . Les choix d'opérateurs effectués pendant la recherche sont enregistrés dans un graphe ET/OU. A chaque sommet-ET de ce graphe est associé un problème formel et un seul. A chaque sommet-OU est associé l'opérateur qui a décomposé le problème associé au prédécesseur immédiat de N en l'ensemble des problèmes associés aux successeurs immédiats de N . Initialement, ce graphe ne contient donc qu'un seul sommet (un sommet-ET) auquel est associé le problème initial po . Ensuite, on définit de façon récursive un sommet résolu:

- . Un sommet-ET est résolu si et seulement si un problème primitif est associé à ce sommet ou si ce sommet a un successeur immédiat (un sommet-OU) résolu.
- . Un sommet-OU est résolu si et seulement si tous ses successeurs immédiats (des sommets-ET) sont résolus.

Le problème original est résolu lorsque le graphe ET/OU engendré contient un sous-graphe de sommets résolus qui permet de vérifier que le sommet auquel est associé le problème initial est résolu.

En l'absence de contrainte, le modèle de résolution de problème mis en oeuvre dans Tropic serait essentiellement un modèle de décomposition de problème appliquant une méthode de recherche du type "profondeur d'abord". Pour plus de rigueur, on pourrait alors considérer qu'il existe un problème primitif unique engendré par les ircm de la forme (Prob: $\sigma \rightarrow$ (Littx: σ NIL)).

En interdisant certains états, les contraintes interdisent certaines décompositions du problème initial. Ainsi, contrairement au modèle de décomposition de problème, le modèle utilisé par Tropic permet de tenir compte d'interdépendances entre des problèmes formels situés sur des chemins distincts de l'acm. La prise en compte de cette interdépendance complique toutefois le traitement des échecs. En effet, un échec subi par Tropic peut être la conséquence d'une mauvaise combinaison de choix enregistrés sur des chemins distincts de l'acm, et plusieurs cheminement arrière sont alors possibles. Il est donc nécessaire de mémoriser les causes d'un échec qui n'ont pas été utilisées lors d'un premier traitement, pour pouvoir les prendre à nouveau en considération en cas de nouvel échec (cf. la deuxième fonction d'un indicateur d'échec, § 1.2.1., chapitre 6).

Chapitre 4: LES HEURISTIQUES DU CHEMINEMENT AVANT

Dans ce chapitre, nous décrivons les heuristiques utilisées par Tropic à chaque expansion de l'acm pour choisir le problème formel à décomposer, puis l'ircm à appliquer.

1. Principe des heuristiques du cheminement avant

a/ Sauf si un cheminement arrière est effectué, tous les problèmes formels pendants (i.e. associés à des sommets-ET pendants) de l'acm devront être décomposés tôt ou tard. L'ordre de leur décomposition étant sans influence sur les modèles que l'on peut construire (cf. commentaire b, § 3.1., chapitre 2), une heuristique adaptée au choix du problème à décomposer est de supposer que certains choix d'ircm déjà effectués sont mauvais, et de favoriser l'apparition d'échecs dus à ces choix (Nilsson 1971, p.136). Cette heuristique peut éviter de décomposer inutilement certains problèmes formels issus de choix d'ircm erronés, et favoriser l'apparition d'un échec irrémédiable dans le cas où le problème original n'a pas de solution.

Dans Tropic, cette heuristique prend la forme d'une stratégie hiérarchique descendante. Cette stratégie, qui consiste à engendrer un modèle à des niveaux de détail de plus en plus élémentaires, détermine un ordre partiel de décomposition sur les problèmes pendants de l'acm. Elle repose sur l'hypothèse suivant laquelle il est possible de discriminer les données susceptibles de constituer un modèle d'après leurs importances relatives, de telle sorte que la génération d'un niveau de détail remette peu (ou mieux, pas du tout) en question les niveaux de détail précédemment engendrés. La discrimination des données est faite par Tropic suivant des informations fournies par le concepteur (cf. §2.3.).

b/ Chaque problème formel est décomposé par l'application d'une seule ircm. On doit donc choisir l'ircm qui a le plus grand "mérite", c'est-à-dire

celle dont l'application est la plus susceptible de contribuer à la construction d'une solution. Pour ce faire, Tropic utilise d'une part l'heuristique générale qui consiste à choisir l'icrom dont l'application permet de satisfaire immédiatement le plus grand nombre de restrictions de contraintes et d'autre part les heuristiques spécialisées qui sont définies par les conseils du concepteur (cf. §3).

2. La stratégie hiérarchique

La définition de cette stratégie nécessite de décrire les caractéristiques des modèles que Tropic peut construire avec plus de détail que nous ne l'avons fait au paragraphe 2 du chapitre 2. Ceci nous conduit à définir successivement les relations descriptives et associatives qui composent un modèle (§ 2.1.), la structure hiérarchique de ce modèle (§ 2.2.), les niveaux de détail qui sont déterminés par cette structure (§ 2.3.) et enfin la stratégie hiérarchique qui est appliquée par Tropic (§ 2.4.).

2.1. Les relations descriptives et associatives, et leur représentation

2.1.1. Les objets abstraits et les objets concrets

Parmi les objets d'un modèle, nous distinguons les objets abstraits et les objets concrets (Abrial 1974):

. Un objet abstrait est un objet qui admet une définition indépendante des modèles où il apparaît, par exemple: le nombre 100, la couleur ROUGE, la forme PRISMATIQUE. Cette définition est soit connue a priori de Tropic (cas des nombres), soit implicite dans l'énoncé d'un problème (cas de la couleur ROUGE et de la forme PRISMATIQUE). Ainsi, un objet abstrait a pour Tropic une "existence" indépendante de son apparition, ou non apparition, dans un modèle.

. Un objet concret est défini dans un modèle par ses liens avec les autres objets. La définition d'un objet concret est donc altérée par la modification

de ces liens. Ainsi, un objet concret n'a d'"existence" pour Tropic que s'il figure dans le modèle considéré.

2.1.2. Les relations descriptives et les relations associatives

Parmi les relations d'un modèle, nous distinguons les relations descriptives et les relations associatives:

- . Une relation descriptive établit un lien direct entre un objet concret et un objet abstrait, par exemple: "la couleur de (l'objet concret) TT1 est (l'objet abstrait) ROUGE.

- . Une relation associative établit un lien direct entre plusieurs objets concrets, par exemple: "(l'objet concret) BLC2 repose sur (l'objet concret) BLC1", "(l'objet concret) TT1 est un composant de (l'objet concret) TR1".

2.1.3. Les D-littéraux, les A-littéraux, les A-termes et les C-termes

Tropic classe les littéraux et les termes qui apparaissent dans un modèle parmi plusieurs catégories de littéraux et de termes. L'appartenance d'un littéral l et de ses termes à une de ces catégories est déterminée par la valeur d'une propriété qui est associée à l'atome Pred qui note le prédicat de l . Nous appelons cette propriété la propriété structurante de Pred. Sa valeur, que nous notons Struct(Pred), doit être fournie par l'utilisateur dans l'énoncé du problème. C'est une liste de la forme (D...) ou (A...) [i.e., une liste dont le premier élément est soit l'atome D, soit l'atome A]:

- . Si Struct(Pred) = (D...), nous disons que l est un D-littéral, et nous imposons qu'il ait deux termes et deux seulement. Nous l'interprétons comme la représentation d'une relation descriptive, ses termes de rang 1 et de rang 2 représentant respectivement l'objet concret et l'objet abstrait associés par cette relation.

- . Si Struct(Pred) = (A...), nous disons que l est un A-littéral. Nous l'interprétons comme la représentation d'une relation associative, ses termes représentant les objets concrets liés par cette relation.

Nous disons qu'un terme est:

- . un A-terme si et seulement si il est de rang 2 dans un D-littéral,
- . un C-terme si et seulement si ce n'est pas un A-terme.

Exemples:

Dans les exemples à venir, nous supposons que Struct(TYPE)=(D...), Struct(NOM-BRE-DE-BLOCS)=(D...), Struct(SECTION)=(D...), Struct(FORME)=(D...), Struct(COULEUR)=(D...), Struct(COMPOSANT)=(A...) et Struct(SUR)=(A...). Ainsi, (TYPE TR1 TOUR) et (SECTION BLC1 CARREE) sont des D-littéraux ; (COMPOSANT TR1 TT1) et (SUR BLC1 BLC2) sont des A-littéraux. Dans ces littéraux, TOUR et CARREE sont des A-termes ; TR1, TT1, BLC1 et BLC2 sont des C-termes.

Généralisation

Soit l un littéral quelconque de prédicat Pred (i.e.: l peut être positif, négatif, supérieur ou inférieur, et ses termes peuvent être notés par des constantes ou des variables):

- . si Struct(Pred)=(D...), nous disons que l est un D-littéral et que ses termes de rangs 1 et 2 sont respectivement un C-terme et un A-terme,
- . si Struct(Pred)=(A...), nous disons que l est un A-littéral et que ses termes sont des C-termes.

Exemple:

(COULEUR \$BLC \$C) est un D-littéral dans lequel les variables \$BLC et \$C notent respectivement un C-terme et un A-terme.

Remarque:

Dans la version actuelle de Tropic, nous imposons qu'un modèle ne contienne pas deux D-littéraux ayant même prédicat et même terme de rang 1. Nous pouvons ainsi énoncer de nouvelles règles permettant de déterminer la valeur logique d'un D-littéral constant l dans un état m quelconque engendré par Tropic. [Ces règles qui s'ajoutent à celles que nous avons énoncées au paragraphe 2 du chapitre 3, ont pour unique fonction d'accélérer éventuellement la reconnaissance d'un échec. La condition imposée sur les D-littéraux n'est donc pas indispensable].

- . Si l est un D-littéral positif ($\text{Pred } t_1 t_2$) et s'il existe dans m un littéral ($\text{Pred } t_1 t'_2$) tel que $t'_2 \neq t_2$, alors l a la valeur FAUX dans m ;
- . Si l est un D-littéral négatif ($\text{NON Pred } t_1 t_2$) et s'il existe dans m un littéral ($\text{Pred } t_1 t'_2$) tel que $t'_2 \neq t_2$, alors l a la valeur VRAI dans m ;
- . Si l est un D-littéral supérieur ($\text{PG Pred } t_1 t_2$) et s'il existe dans m un littéral ($\text{Pred } t_1 t'_2$) tel que $t'_2 \leq t_2$, alors l a la valeur FAUX dans m ,
- . Si l est un D-littéral inférieur ($\text{PP Pred } t_1 t_2$) et s'il existe dans m un littéral ($\text{Pred } t_1 t'_2$) tel que $t'_2 \geq t_2$, alors l a la valeur FAUX dans m .

2.2. La structure hiérarchique d'un modèle

2.2.1. Les relations associatives hiérarchiques et transversales

Parmi les relations associatives d'un modèle, nous distinguons les relations associatives hiérarchiques et les relations associatives transversales:

. Les relations associatives hiérarchiques servent à définir des objets concrets en termes d'objets considérés comme plus élémentaires. Chacune de ces relations associe deux objets, par exemple: "TT1 est un composant de TR1".
 [Note: une relation associative hiérarchique ne décrit pas nécessairement la composition physique de deux objets].

. Les relations associatives transversales servent à exprimer les autres relations entre objets concrets, par exemple: "BLC2 repose sur BLC1" , "(l'enroulement) ENR1 entoure (le noyau) NOY1".

2.2.2. Les A-H-littéraux et les A-T-littéraux

La valeur de la propriété structurante du prédicat d'un A-littéral est soit (A H), soit (A T):

. Dans le premier cas, nous disons que l est un A-H-littéral, et nous imposons qu'il ait deux termes et deux seulement. Nous l'interprétons comme la représentation d'une relation associative hiérarchique, son terme de rang 2 représentant l'objet le plus élémentaire figurant dans cette relation.

. Dans le deuxième cas, nous disons que l est un A-T-littéral. Nous l'interprétons comme la représentation d'une relation associative transversale.

Exemples:

Dans les exemples à venir, nous supposons que $\text{Struct}(\text{COMPOSANT})=(A\ H)$ et $\text{Struct}(\text{SUR})=(A\ T)$. Ainsi, $(\text{COMPOSANT}\ \text{TR1}\ \text{TT1})$ est un A-H-littéral, et $(\text{SUR}\ \text{BLC1}\ \text{BLC2})$ est un A-T-littéral.

2.2.3. La relation \succ

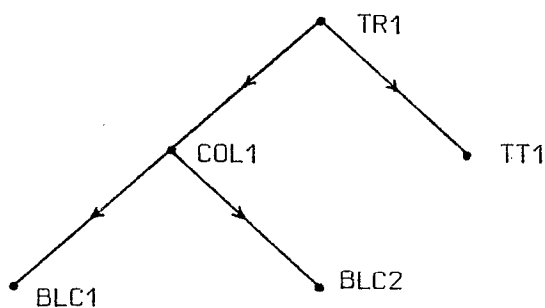
Soient un modèle M construit par Tropic et l'ensemble Ω des atomes qui servent à noter les C-termes de ce modèle.

Nous définissons dans M la relation \succ (lire "est ascendant direct de") incluse dans $\Omega \times \Omega$ par: $\forall c_1$ et $c_2 \in \Omega$, $c_1 \succ c_2$ si et seulement si il existe dans M un A-H-littéral $(\text{Pred}\ c_1\ c_2)$.

Nous imposons que le graphe de cette relation soit une arborescence. Nous appelons racine de M l'atome de Ω qui correspond à la racine de cette arborescence.

Exemple:

Le graphe de la relation \succ dans le modèle M (exemple du §2.2., chapitre 2) est l'arborescence de la figure 4.1. La racine de M est donc TR1.



,figure 4.1.

2.3. Les niveaux de détail d'un modèle

Dans les paragraphes ci-dessous, M est un modèle quelconque engendré par Tropic ; Ω est l'ensemble des atomes qui notent les C-termes de ce modèle. Dans les exemples, M est le modèle donné en exemple au paragraphe 2.2. du chapitre 2 ; dans M , $\Omega = \{TR1 \text{ COL1 } TT1 \text{ BLC1 } BLC2\}$.

2.3.1. Profondeur d'un atome de C-terme dans un atome de C-terme

Soient A l'arborescence de la relation \succ dans M , et c_1 et c_2 deux éléments de Ω . S'il existe un chemin orienté de A , qui part du sommet correspondant à c_1 et qui arrive au sommet correspondant à c_2 , alors nous définissons la profondeur de c_2 dans c_1 comme la longueur de ce chemin (0, si $c_1=c_2$) ; s'il n'existe pas un tel chemin, la profondeur de c_2 dans c_1 n'est pas définie.

Nous notons $\text{Prof}(c_1, c_2)$ la profondeur de c_2 dans c_1 , si elle est définie et $\Omega(c, q)$ l'ensemble de tous les éléments de Ω qui sont à la profondeur q dans c .

Exemples:

Dans M :

- . $\text{Prof}(TR1, TR1) = 0$, $\text{Prof}(TR1, BLC2) = 2$
- . $\Omega(TR1, 0) = \{TR1\}$, $\Omega(TR1, 1) = \{COL1 \text{ } TT1\}$, $\Omega(TR1, 2) = \{BLC1 \text{ } BLC2\}$,
 $\Omega(TT1, 1) = \text{NIL}$.

2.3.2. Couple de dépendance d'un littéral

A chaque littéral l de M nous associons un couple $(c \ q)$ unique appelé le couple de dépendance de l dans M ; $c \in \Omega$ et est appelé l'atome de dépendance de l dans M ; q est un nombre entier positif ou nul, et est appelé la profondeur de dépendance de l dans M .

. Le couple de dépendance d'un D-littéral $(\text{Pred } c \ a)$ dans M est $(c \ 0)$.

Exemple: le couple de dépendance de $(\text{COULEUR } TT1 \text{ } ROUDE)$ dans M est $(TT1 \ 0)$.

. Soient $\overset{*}{\succ}$ (lire "est ascendant de") la fermeture transitive de \succ dans $\Omega \times \Omega$ ($\overset{*}{\succ}$ est une relation d'ordre partiel strict) et $\underline{\overset{*}{\succ}}$ la relation d'ordre partiel faible associée à $\overset{*}{\succ}$ dans $\Omega \times \Omega$. Le couple de dépendance d'un A-littéral $(\text{Pred } c_1 \ c_2 \ \dots \ c_n)$ dans M est $(c \ q)$, où c est le majorant minimum (unique) de $\{c_1 \ c_2 \ \dots \ c_n\}$ pour $\underline{\overset{*}{\succ}}$, et $q = \underset{i \in [1, n]}{\text{Max}} (\text{Prof}(c, c_i))$.

Exemple: Le couple de dépendance du A-H-littéral (COMPOSANT TR1 COL1) dans M est (TR1 1), celui du A-T-littéral (SUR BLC1 BLC2) est (COL1 1).

Nous notons $Dep(c,q)$ l'ensemble des littéraux de M ayant (c q) pour couple de dépendance dans M.

Exemple:

Dans M, $Dep(COL1,1) = \{(COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2)(SUR BLC1 BLC2)\}$.

Generalisation

Soit l un littéral quelconque:

. Si l est un D-littéral et si l'atome c qui note son C-terme appartient à Ω , alors le couple de dépendance de l dans M est (c 0).

Exemple: le couple de dépendance des littéraux (FORME TT1 \$F) et (NON COULEUR TT1 ROUGE) dans M est (TT1 0).

. Si l est un A-H-littéral et si l'atome c_1 qui note son terme de rang 1 appartient à Ω , alors le couple de dépendance de l dans M est (c_1 1).

Exemple: le couple de dépendance de (COMPOSANT COL1 \$BLC) dans M est (COL1 1).

. Si l est un A-T-littéral et si les atomes c_1, c_2, \dots, c_n qui notent ses termes appartiennent tous à Ω , alors le couple de dépendance de l dans M est (c q), où c est le majorant minimum de $\{c_1 c_2 \dots c_n\}$ pour \underline{x} et $q = \text{Max}_{i \in [1,n]} (\text{Prof}(c, c_i))$.

Exemple: le couple de dépendance de (NON SUR BLC1 TT1) dans M est (TR1 2).

Dans tous les autres cas, le couple de dépendance de l dans M n'est pas défini ; c'est par exemple le cas du littéral (SECTION \$BLC CARREE).

2.3.3. Description d'un atome de C-terme

Soient c un élément quelconque de Ω et $\Omega(c,*) = \{c'/c \underline{x} c'\}$. Nous appelons description de c dans M l'ensemble de tous les littéraux de M tels que l'atome de dépendance dans M de chacun d'eux appartienne à $\Omega(c,*)$.

Nous notons Desc(c) cette description. Si r est la racine de M, alors Desc(r) = M.

Exemple:

Dans M, $Desc(COL1) = \{(TYPE COL1 COLONNE)(NOMBRE-DE-BLOCS COL1 2)(COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2)(SUR BLC1 BLC2)(TYPE BLC1 BLOC)(SECTION BLC1 RONDE)(COULEUR BLC1 BLANCHE)(TYPE BLC2 BLOC)(SECTION BLC2 RONDE)(COULEUR BLC2 BLANCHE)\}$.

2.3.4. Niveau de détail d'une description

Soient c un élément de Ω , l un littéral de $\text{Desc}(c)$, $(c'q')$ le couple de dépendance de l dans M et q'' la profondeur de c' dans c . Nous définissons la profondeur de l dans $\text{Desc}(c)$ comme étant $q = q' + q''$.

Exemple: (SUR BLC1 BLC2) a la profondeur 2 (= 1+1) dans $\text{Desc}(TR1)$.

Nous définissons le niveau de détail q de $\text{Desc}(c)$ comme l'ensemble de tous les littéraux de M à la profondeur q dans $\text{Desc}(c)$; nous le notons $\text{Niv}(c,q)$.

2.3.5. Niveau de détail d'un modèle

Soit r la racine de M . Nous définissons le niveau de détail q de M comme $\text{Niv}(r,q)$.

Exemple:

Dans M , $\text{NIV}(TR1,2) = \{(COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2)(SUR BLC1 BLC2)(TYPE BLC1 BLOC)(SECTION BLC1 RONDE)(COULEUR BLC1 BLANCHE)(TYPE BLC2 BLOC)(SECTION BLC2 RONDE)(COULEUR BLC2 BLANCHE)\}$.

2.4. La stratégie hiérarchique

Au paragraphe 2.4.1., nous définissons la stratégie hiérarchique. Cette définition ne fait pas intervenir l'acm. Elle concerne uniquement l'évolution de l'état courant pendant la résolution du problème original.

Au paragraphe 2.4.2., nous décrivons l'application de la stratégie hiérarchique. Nous montrons comment la définition du paragraphe 2.4.1. détermine un ordre partiel de décomposition sur les problèmes pendants de l'acm.

Au paragraphe 2.4.3., nous décrivons les implications de la stratégie hiérarchique sur l'interprétation logique des littéraux constants. Au paragraphe 2.4.4., nous décrivons ses implications sur le traitement des restrictions imposées par les instances actives de contraintes.

2.4.1. Définition

La stratégie hiérarchique de Tropic consiste à construire un modèle M en engendrant successivement ses niveaux de détail $0,1,\dots,i,i+1,\dots$

La génération de chaque niveau de détail est subdivisée en étapes plus fines appelées étapes de la stratégie hiérarchique. Chaque étape consiste à

engendrer les littéraux de M ayant un couple de dépendance donné. Nous notons $ETAPE(c,q)$ l'étape qui correspond à la génération du sous-ensemble $Dep(c,q)$ de M .

L'enchaînement des étapes qui permet à Tropic d'engendrer le sous-ensemble $Niv(c,q)$ de M (niveau de détail q de la description de c dans M) est décrit par l'algorithme $NIVEAU(c,q)$ suivant:

$NIVEAU(c,q)$: . Si $q=0$, alors on exécute $ETAPE(c,0)$.

. Si $q=1$, alors:

- on exécute $ETAPE(c,1)$, ce qui a pour effet d'engendrer $\Omega(c,1)$ (ensemble des C-termes ayant la profondeur 1 dans c),
- si $\Omega(c,1) = NIL$, on note que la description de c est complète, sinon pour tout c' dans $\Omega(c,1)$ on exécute $ETAPE(c',0)$.

. Si $q > 1$, alors:

- pour tout c' dans $\Omega(c,1)$ dont la description n'est pas complète, on exécute $NIVEAU(c',q-1)$,
- si la description de tout c' dans $\Omega(c,1)$ est complète, on note que la description de c est complète, sinon on exécute $ETAPE(c,q)$.

La stratégie hiérarchique de Tropic est définie par l'exécution itérative de cet algorithme avec c égal à la racine de M et q successivement égal à 0, 1, 2, ..., tant que la description de la racine de M n'est pas complète.

Nous illustrons cette stratégie sur un exemple au paragraphe suivant.

Commentaire

Soit m l'état courant à un instant quelconque du cheminement avant. Il résulte de la définition ci-dessus que le graphe de la relation \supset définie dans m est une arborescence. Nous pouvons ainsi définir dans m le couple de dépendance d'un littéral l comme nous l'avons fait dans M au paragraphe 2.3.2.

2.4.2. Application

Pour décrire l'application de la stratégie hiérarchique, nous supposons que les deux conditions suivantes sont satisfaites:

. Si une ircm $(Prob:\sigma \rightarrow (Littx:\sigma \text{ Décomp}:\sigma))$ est applicable à un problème pendant de l'acm courante, alors tous les littéraux de $Littx:\sigma$ ont le même couple

de dépendance dans l'état qui serait obtenu en ajoutant ces littéraux à l'état courant. Nous appelons ce couple le couple de dépendance de l'ircm. Toutes les ircm applicables à un même problème de l'acm ont le même couple de dépendance. Nous appelons ce couple le couple de dépendance du problème formel.

Nous verrons toutefois que la vérification de ces deux conditions n'est pas indispensable à l'application de la stratégie hiérarchique (cf. commentaire b).

Tropic applique la stratégie hiérarchique ainsi:

- . Il exécute itérativement l'algorithme NIVEAU(r,q) avec r égal à l'atome du couple de dépendance du problème initial et q successivement égal à 0, 1, 2, ..., jusqu'à ce que la description de r soit complète.
- . Il exécute chaque étape ETAPE(c,q) en décomposant successivement tous les problèmes pendants ayant (c q) pour couple de dépendance (cf. commentaire a).

La stratégie hiérarchique détermine ainsi un ordre partiel de décomposition sur les problèmes pendants de l'acm.

Exemple:

- . Le tableau 4.1, ci-après met en évidence l'enchaînement des étapes de la stratégie hiérarchique dans le cheminement avant décrit par le tableau 2.1. Cet enchaînement résulte de l'exécution itérative de NIVEAU(TR1,q) avec $q = 0, 1, 2$.
- . La figure 4.2, montre à nouveau l'acm qui est construite lors de ce cheminement avant. Des cadres regroupent les problèmes et les ircm qui ont le même couple de dépendance. L'ordre de décomposition des problèmes formels est indiqué entre parenthèses à côté de chaque sommet-ET.

Commentaires

a/ Chaque fois qu'un sommet-ET est ajouté à l'acm, Tropic range le problème formel correspondant à la fin d'une liste notée Problèmes (c,q) telle que (c q) soit le couple de dépendance de ce problème. Initialement, toutes les listes Problèmes(c,q) sont vides.

Pendant l'exécution de ETAPE(c,q), Tropic décompose les problèmes contenus dans Problèmes(c,q) dans l'ordre où ils y figurent (durant cette exécution, de nouveaux problèmes formels peuvent être ajoutés à la liste). Ce

niveau de détail engendré	ETAPE (c,q) exécutée	ircm appliquée	littéraux engendrés	commentaires
0	ETAPE (TR1,0)	rcm1: NIL	(TYPE TR1 TOUR)	
1	ETAPE (TR1,1)	rcm2: {(\$TR TR1)}	(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1) (SUR COL1 TT1)	$\Omega(\text{TR1},1) = \{\text{COL1 TT1}\}$
	ETAPE (COL1,0)	rcm3: {(\$COL COL1)}	(TYPE COL1 COLONNE)	
	ETAPE (TT1,0)	rcm5: {(\$COL COL1)}	(NOMBRE-DE-BLOCS COL1 2)	
		rcm12: {(\$TT TT1)}	(TYPE TT1 TOIT)	
		rcm13: {(\$TT TT1)}	(FORME TT1 PYRAMIDALE)	
		rcm15: {(\$TT TT1)}	(COULEUR TT1 ROUGE)	
2	ETAPE (COL1,1)	rcm6: {(\$COL COL1) (\$N 2)}	(COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2) (SUR BLC1 BLC2)	$\Omega(\text{COL1},1) = \{\text{BLC1 BLC2}\}$
	ETAPE (BLC1,0)	rcm7: {(\$BLC BLC1)}	(TYPE BLC1 BLOC)	
		rcm9: {(\$BLC BLC1)}	(SECTION BLC1 RONDE)	
		rcm10: {(\$BLC BLC1)}	(COULEUR BLC1 BLANCHE)	
	ETAPE (BLC2,0)	rcm7: {(\$BLC BLC2)}	(TYPE BLC2 BLOC)	
		rcm9: {(\$BLC BLC2)}	(SECTION BLC2 RONDE)	
		rcm10: {(\$BLC BLC2)}	(COULEUR BLC2 BLANCHE)	
	ETAPE (TT1,1)	-	-	$\Omega(\text{TT1},1) = \text{NIL}$, donc la description de TT1 est complète
3	ETAPE (BLC1,1)	-	-	$\Omega(\text{BLC1},1) = \text{NIL}$, donc la description de BLC1 est complète
	ETAPE (BLC2,1)	-	-	$\Omega(\text{BLC2},1) = \text{NIL}$, donc la description de BLC2 est complète de même que celles de COL1 et de TR1

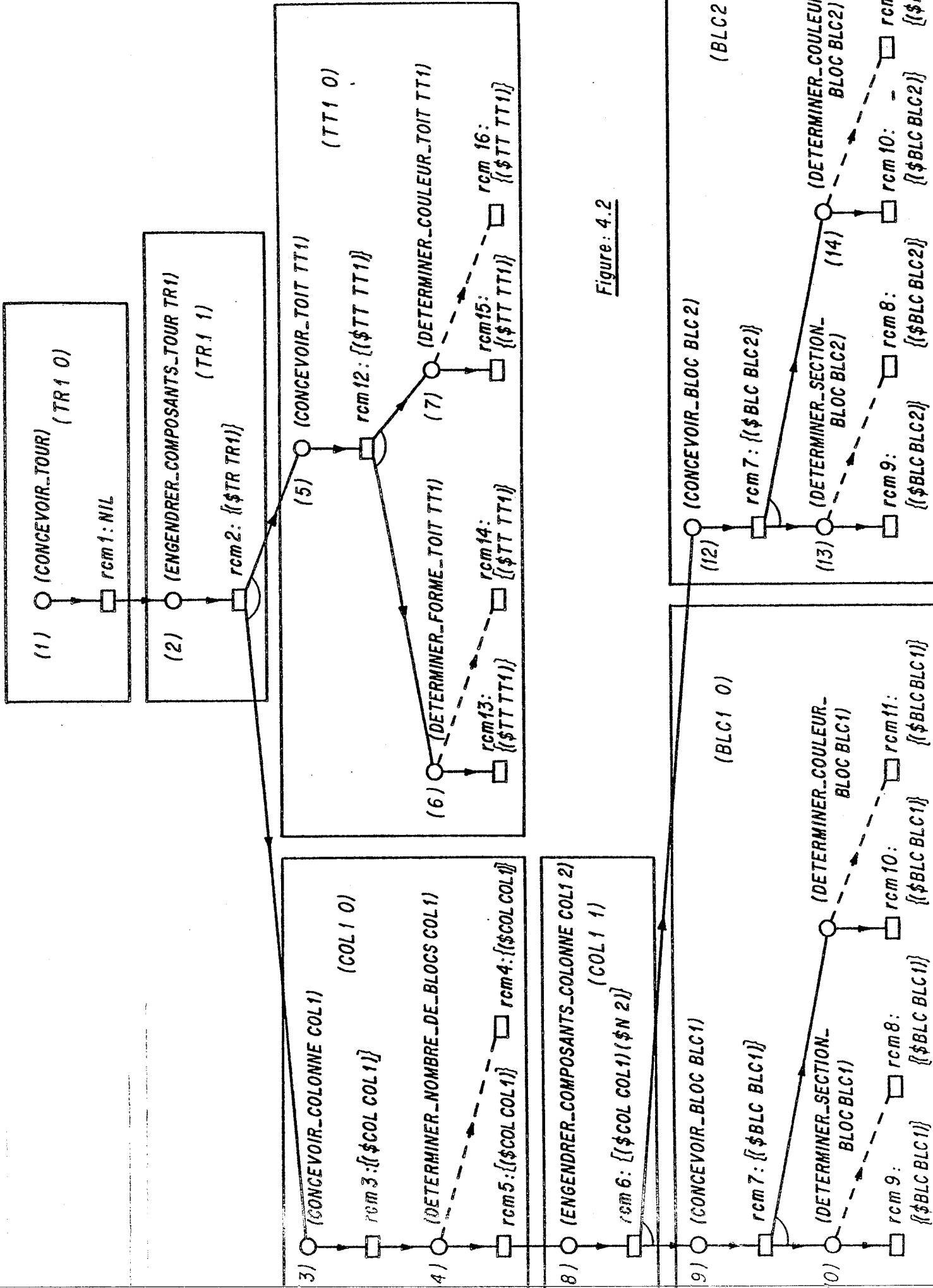


Figure: 4.2

traitement pourrait être amélioré (du point de vue heuristique) en retardant la décomposition d'un problème (dans le cadre de ETAPE(c,q)), quand plusieurs ircm d'égal "mérite" (cf. § 3.2.) lui sont applicables.

b/ La vérification des deux conditions énoncées au début du paragraphe 2.4.2. n'est pas indispensable à l'application de la stratégie hiérarchique. Quand ces conditions ne sont pas vérifiées, Tropic transforme les ircm concernées de façon à pouvoir appliquer le traitement décrit dans ce paragraphe. Ci-dessous nous illustrons le principe de ces transformations sur un exemple ; des transformations similaires sont effectuées pour d'autres exemples.

Soit α et m l'acm et l'état courants à un instant quelconque du cheminement avant. Nous supposons que le problème formel (CONCEVOIR-COLONNE COL1) est associé à un sommet-ET pendant de α et que l'ircm I suivante est une des ircm applicables à ce problème:

I = ((CONCEVOIR-COLONNE COL1)
 → (({ (TYPE COL1 COLONNE) (NOMBRE-DE-BLOCS COL1 2) (COMPOSANT COL1 BLC1)
 (COMPOSANT COL1 BLC2) (SUR BLC1 BLC2) (TYPE BLC1 BLOC) (TYPE BLC2
 BLOC) }
 { (DETERMINER-SECTION-BLOC BLC1) (DETERMINER-COULEUR-BLOC BLC1)
 (DETERMINER-SECTION-BLOC BLC2) (DETERMINER-COULEUR-BLOC BLC2) })))

Soit $L = \{ (TYPE COL1 COLONNE) (NOMBRE-DE-BLOCS COL1 2) (COMPOSANT COL1 BLC1)
 (COMPOSANT COL1 BLC2) (SUR BLC1 BLC2) (TYPE BLC1 BLOC) (TYPE BLC2 BLOC) \}$.

L'ircm I ne vérifie pas la première condition car:

- . (TYPE COL1 COLONNE) et (NOMBRE-DE-BLOCS COL1 2) ont (COL1 0) pour couple de dépendance dans $m \cup L$,
- . (COMPOSANT COL1 BLC1), (COMPOSANT COL1 BLC2) et (SUR BLC1 BLC2) ont (COL1 1) pour couple de dépendance dans $m \cup L$,
- . (TYPE BLC1 BLOC) a (BLC1 0) pour couple de dépendance dans $m \cup L$,
- . (TYPE BLC2 BLOC) a (BLC2 0) pour couple de dépendance dans $m \cup L$.

De par la définition de la stratégie hiérarchique (§ 2.4.1.), Tropic sait que l'étape ETAPE(COL1,0) sera exécutée avant l'étape ETAPE(COL1,1), qui sera elle-même exécutée avant les étapes ETAPE(BLC1,0) et ETAPE(BLC2,0).

Il scinde alors l'ircm I en quatre ircm I_1 , I_2 , I_3 et I_4 énoncées ci-dessous et ayant respectivement (COL1 0), (COL1 1), (BLC1 0) et (BLC2 0) pour couple de dépendance ; cette scission est faite en engendrant trois nouveaux problèmes formels dits problèmes fantômes, (FANTOME1), (FANTOME2) et (FANTOME3).

$$I_1 = ((\text{CONCEVOIR-COLONNE COL1})$$

$$\rightarrow \{ \{ (\text{TYPE COL1 COLONNE})(\text{NOMBRE-DE-BLOCS COL1 2})$$

$$\{ (\text{FANTOME1})(\text{DETERMINER-SECTION-BLOC BLC1})(\text{DETERMINER-COULEUR-BLOC BLC1})$$

$$(\text{DETERMINER-SECTION-BLOC BLC2})(\text{DETERMINER-COULEUR-BLOC BLC2}) \} \})$$

$$I_2 = (\text{FANTOME 1})$$

$$\rightarrow \{ \{ (\text{COMPOSANT COL1 BLC1})(\text{COMPOSANT COL1 BLC2})(\text{SUR BLC1 BLC2})$$

$$\{ (\text{FANTOME2})(\text{FANTOME3}) \} \})$$

$$I_3 = ((\text{FANTOME2}) \rightarrow \{ \{ (\text{TYPE BLC1 BLOC}) \} \text{NIL})$$

$$I_4 = ((\text{FANTOME3}) \rightarrow \{ \{ (\text{TYPE BLC2 BLOC}) \} \text{NIL})$$

Le traitement du paragraphe 2.4.2. peut alors être exécuté. Il permet d'engendrer la partie d'acm montrée à la figure 4.3. Les ircm I_1 , I_2 , I_3 et I_4 sont respectivement appliquées durant les étapes $\text{ETAPE}(\text{COL1},0)$, $\text{ETAPE}(\text{COL1},1)$, $\text{ETAPE}(\text{BLC1},0)$ et $\text{ETAPE}(\text{BLC2},0)$.

Toutefois, il est généralement facile de satisfaire les deux conditions énoncées au début du paragraphe 2.4.2. en ne définissant que des rcm courtes. Aussi, dans la suite de ce document, nous supposons que ces conditions sont vérifiées, ce qui nous permettra de simplifier la description de certains algorithmes, notamment ceux qui concernent le choix de l'ircm à appliquer (§ 3.2. de ce chapitre) et l'activation des contraintes, des conseils et des diagnostics (chapitre 7). L'extension de ces algorithmes au cas où les conditions ne sont pas vérifiées ne présente pas de difficulté majeure (cette extension n'est en fait qu'une amélioration, car les algorithmes que nous décrivons peuvent être utilisés dans le cas où les conditions ne sont pas vérifiées).

c/ Une "couche" de sommets-ET et OU de l'acm correspond à chaque niveau de détail engendré (cf. figure 4.4.). Aux sommets-ET de cette couche sont associés les problèmes formels dont la décomposition a contribué à la génération de ce niveau de détail ; aux sommets-OU sont associées les ircm appliquées à ces problèmes. La décomposition "séquentielle" du problème original par l'application de la stratégie hiérarchique se combine ainsi à la décomposition "parallèle" de ce problème par l'application des ircm.

Dans le cas particulier où la décomposition des problèmes formels situés à la même profondeur dans l'acm contribue à engendrer un même niveau de détail, la stratégie hiérarchique impose un parcours de l'acm selon des couches horizontales d'"épaisseurs" constantes (cas de la figure 4.4.).

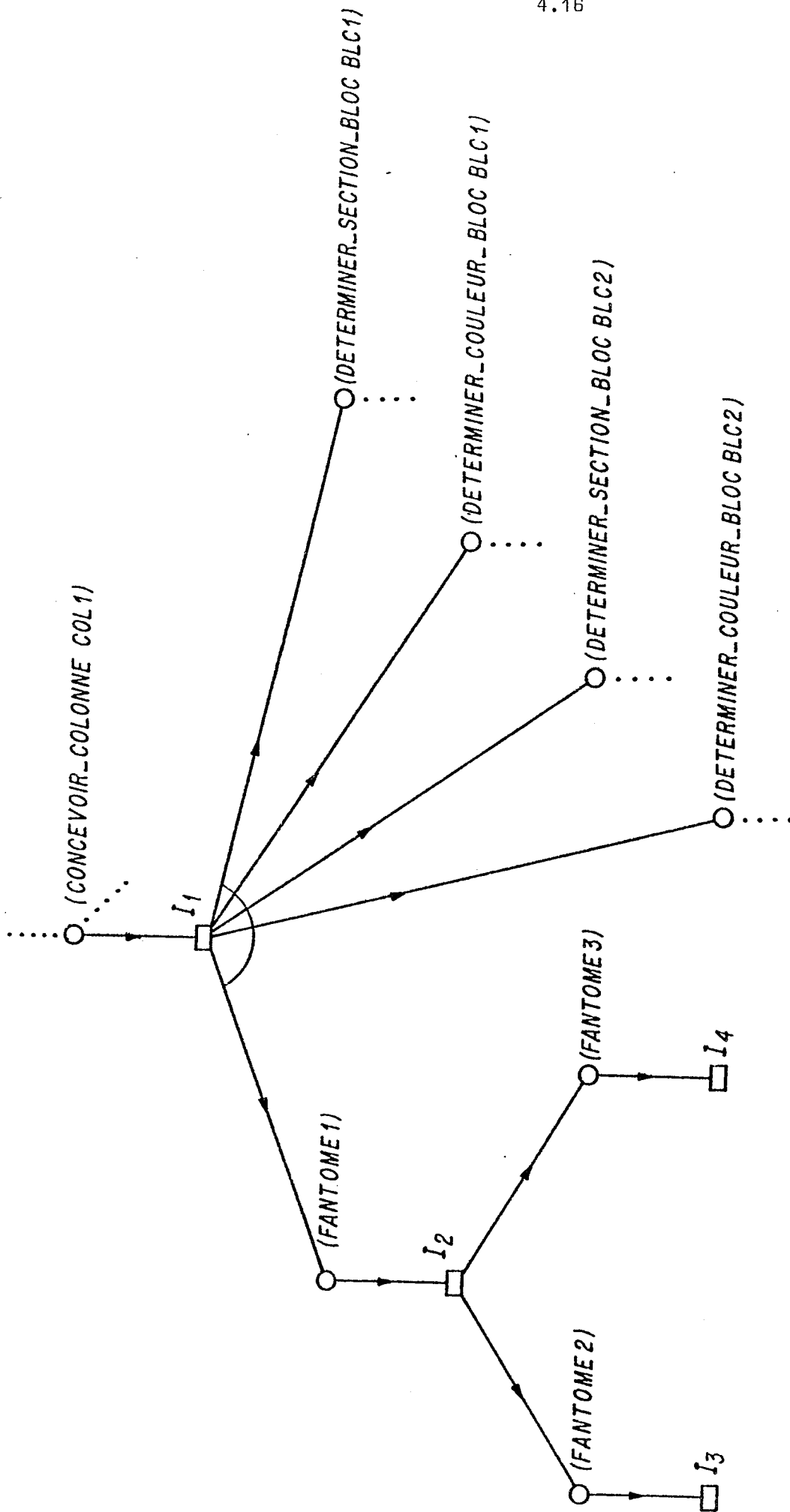


Figure: 4.3

(1) (CONCEVOIR_TOUR)

rcm 1: NIL

(2) (ENGENDRER_COMPOSANTS_TOUR TR1)

rcm 2: {(\$TR TR1)}

(5) (CONCEVOIR_TOIT TT1)

rcm 12: {(\$TT TT1)}

(7) (DETERMINER_COULEUR_TOIT TT1)

rcm 15: {(\$TT TT1)}

rcm 16: {(\$TT TT1)}

(3) (CONCEVOIR_COLONNE COL1)

rcm 3: {(\$COL COL1)}

(4) (DETERMINER_NOMBRE_DE_BLOCS COL1)

rcm 4: {(\$COL COL1)}

(6) (DETERMINER_FORME_TOIT TT1)

rcm 13: {(\$TT TT1)}

(8) (ENGENDRER_COMPOSANTS_COLONNE COL1 2)

rcm 6: {(\$COL COL1) (\$N 2)}

(9) (CONCEVOIR_BLOC BLC1)

rcm 7: {(\$BLC BLC1)}

(10) (DETERMINER_SECTION_BLOC BLC1)

rcm 8: {(\$BLC BLC1)}

(11) (DETERMINER_COULEUR_BLOC BLC1)

rcm 9: {(\$BLC BLC1)}

(12) (CONCEVOIR_BLOC BLC2)

rcm 7: {(\$BLC BLC2)}

(13) (DETERMINER_SECTION_BLOC BLC2)

rcm 8: {(\$BLC BLC2)}

(14) (DETERMINER_COULEUR_BLOC BLC2)

rcm 9: {(\$BLC BLC2)}

rcm 10: {(\$BLC BLC2)}

rcm 11: {(\$BLC BLC2)}

rcm 10: {(\$BLC BLC2)}

rcm 11: {(\$BLC BLC2)}

Figure: 4.4

d/ Si l'application d'une ircm ajoute à l'acm un problème formel dont le couple de dépendance (c q) est tel que ETAPE(c,q) a déjà été exécutée, alors ce problème n'est jamais décomposé, et l'utilisateur en est averti par un message. Ce cas correspond en général à une écriture assez peu naturelle des rcm.

2.4.3. Implications sur la valeur logique des littéraux

Au paragraphe 2 du chapitre 3 et au paragraphe 2.1.3. du présent chapitre nous avons énoncé des règles qui déterminent la valeur logique de certains littéraux constants dans un état donné. La stratégie hiérarchique nous permet de compléter ces règles.

Soient m l'état courant à un instant quelconque du cheminement avant, et l un littéral constant ayant dans m un couple de dépendance (c q) tel que ETAPE(c,q) a déjà été exécuté:

- . si l est positif et si $l \notin m$, alors l a la valeur FAUX dans m ,
- . si l est un littéral négatif (NON Pred $t_1 t_2 \dots t_n$) et si (Pred $t_1 t_2 \dots t_n$) $\notin m$, alors l a la valeur VRAI dans m ,
- . si l est un littéral supérieur (PG Pred $t_1 t_2$) et s'il n'existe pas dans m un littéral (Pred $t_1 t'_2$) tel que $t'_2 > t_2$, alors l a la valeur FAUX dans M
- . si l est un littéral inférieur (PP Pred $t_1 t_2$) et s'il n'existe pas dans m un littéral (Pred $t_1 t'_2$) tel que $t'_2 < t_2$, alors l a la valeur FAUX dans M

Exemple:

Soit le cheminement avant du tableau 4.1. Tant que l'exécution de ETAPE(COL1 1) n'est pas terminée, la valeur du littéral (SUR BLC2 BLC1) dans l'état courant est indéterminée. Ensuite, cette valeur est FAUX.

Commentaires

a/ Soit l un littéral constant ayant (c q) pour couple de dépendance. On vérifie que:

- . si l'exécution de ETAPE(c,q) est terminée, alors la valeur de l dans l'état courant est toujours déterminée ;
- . si l'exécution de ETAPE(c,q) est en cours, alors la valeur de l dans l'état courant peut être indéterminée ;
- . si l'exécution de ETAPE(c,q) n'est pas commencée, alors la valeur de l est toujours indéterminée.

b/ Si les règles énoncées au chapitre 3 et au paragraphe 2.1.3. du présent chapitre déterminent la valeur d'un littéral l dans un état m , alors cette valeur résulte toujours de la présence d'un littéral l' dans m .

Si ces règles ne déterminent pas la valeur d'un littéral l dans un état m , et si les règles énoncées ci-dessus la déterminent, alors la valeur de l résulte toujours de l'absence de certains littéraux dans m .

2.4.4. Implications sur la satisfaction des contraintes

Supposons qu'une instance de contrainte $C:\sigma$ devienne active pendant l'exécution de $ETAPE(c,q)$, et soit r une restriction imposée par $C:\sigma$. Plusieurs cas sont possibles pour r :

a/ Le couple de dépendance de r est (c',q') tel que $ETAPE(c',q')$ a déjà été exécuté. La restriction r est alors soit satisfaite, soit contredite (cf. commentaire a, § 2.4.3.). Dans le deuxième cas, Tropic est en échec

Exemple:

Soit la contrainte C donnée en exemple au paragraphe 2 du chapitre 3. Considérons le cheminement avant du tableau 4.1. L'instance $C:\{(\$TR TR1) (\$COL COL1) (\$TT TT1) (\$BLC BLC1)\}$ devient active après l'application de rcm 9: $\{(\$BLC BLC1)\}$ pendant l'exécution de $ETAPE(BLC1,0)$. La restriction imposée (NON FORME TT1 PYRAMIDALE) a (TT1 0) pour couple de dépendance. $ETAPE(TT1,0)$ a déjà été exécuté et la restriction est contredite.

b/ Le couple de dépendance de r est (c'',q'') tel que $ETAPE(c'',q'')$ n'a pas encore été exécuté. La restriction r est alors ni satisfaite, ni contredite (cf. commentaire a, § 2.4.3.) Tropic la range comme nouvel élément d'un ensemble noté Restrictions(c'',q''). Cet ensemble contient toutes les restrictions ayant (c'',q'') pour couple de dépendance, qui sont imposées par les instances de contraintes actives dans l'état courant.

Avant de ranger r dans Restrictions(c'',q''), Tropic vérifie que r n'est pas contradictoire avec une restriction déjà contenue dans cet ensemble (deux restrictions sont contradictoires si elles ne peuvent être satisfaites simultanément dans un état, par exemple (SECTION BLC1 RONDE) et (NON SECTION BLC1 RONDE)). Si c'était le cas, Tropic serait en échec (bien qu'il n'y ait pas alors contradiction effective d'une contrainte).

c/ Le couple de dépendance de r est (c,q). La restriction r peut alors être satisfaite, ou contredite, ou n'être ni l'un ni l'autre (cf. commentaire a, § 2.4.3.). Si elle est contredite, Tropic est en échec. Si elle n'est ni satisfaite, ni contredite, Tropic applique le traitement du paragraphe b.

Exemple:

Soient le cheminement avant du tableau 4.1. et la contrainte suivante:

*{ (TYPE \$TR TOUR) (COMPOSANT \$TR \$COL) (COMPOSANT \$TR \$TT) (TYPE \$COL COLONNE)
 (TYPE \$TT TOIT) (COULEUR \$TT ROUGE) (COMPOSANT \$COL \$BLC) (TYPE \$BLC BLOC)
 → { (COULEUR \$BLC BLANCHE) } }.*

*L'instance de cette contrainte par { (\$TR TR1) (\$COL COL1) (\$TT TT1)
 (\$BLC BLC1) } devient active pendant l'exécution de ETAPE(BLC1,0)
 après l'application de rem 7: { (\$BLC BLC1) }. Elle applique la restriction
 (COULEUR BLC1 BLANCHE) qui a (BLC1 0) pour couple de dépendance. Cette res-
 triction n'est à cet instant ni satisfaite, ni contredite. Tropic la range
 donc dans Restrictions(BLC1,0), si elle n'est pas contradictoire avec une
 restriction déjà continue dans cet ensemble.*

A la fin de l'exécution de ETAPE(c,q) Tropic vérifie que l'ensemble Restrictions(c,q) ne contient aucun littéral ayant la valeur FAUX dans l'état courant (grâce aux règles énoncées au § 2.4.3., tous ces littéraux ont maintenant une valeur déterminée). Si un littéral de cet ensemble a la valeur FAUX, alors la restriction qu'il représente est contredite, et Tropic est en échec.

3. Le choix des ircm

3.1. Les conseils

Les conseils sont des "contraintes faibles" qu'il n'est pas impératif de satisfaire. L'utilisateur peut s'en servir pour:

- guider le cheminement avant en privilégiant l'application des ircm qui permettent de satisfaire les conseils,
- exprimer des souhaits sur la solution.

Un conseil est une expression

(Activ → Restrict)

où *Activ* et *Restrict* sont des ensembles de littéraux qui représentent respectivement les conditions d'activation du conseil et les restrictions faibles

qu'il applique. Ces ensembles sont aussi appelés partie gauche et partie droite du conseil.

L'instance par σ d'un conseil (Activ \rightarrow Restrict) est dite active dans un état m si et seulement si tous les littéraux de Activ: σ ont la valeur VRAI dans m .

Lorsque l'instance par σ d'un conseil (Activ \rightarrow Restrict) devient active dans l'état courant, chaque littéral de Restrict: σ est rangé dans un ensemble noté F-Restrictions(c,q), tel que (c,q) soit le couple de dépendance de ce littéral. Nous décrivons l'utilisation de cet ensemble par Tropic au paragraphe suivant.

Commentaires

a/ La partie droite d'un conseil peut aussi être une fonction qui s'évalue à un ensemble de littéraux. Il est alors possible d'utiliser la fonction BUT(1) ; l'évaluation de cette fonction consiste à comparer le littéral l aux restrictions appliquées par les instances de contraintes qui sont actives dans l'état courant à l'instant de l'évaluation:

- . BUT(1) s'évalue à VRAI, si la satisfaction de ces restrictions par un état m implique que l ait la valeur VRAI dans m ,
- . BUT(1) s'évalue à FAUX sinon.

Exemples:

- . BUT (PG HAUTEUR TR1 150) s'évalue à VRAI, si la seule restriction concernant la hauteur de TR1 est (HAUTEUR TR1 200).
- . BUT (PG HAUTEUR BLC1 30) s'évalue à VRAI, si les deux seules restrictions concernant la hauteur de BLC1 sont (PG HAUTEUR BLC1 50) et (PP HAUTEUR BLC1 100). Dans les mêmes conditions, BUT(HAUTEUR BLC1 70) s'évalue à FAUX.
- \. Le conseil ci-dessous recommande donc de choisir le nombre de blocs d'une colonne égal à 2, si la hauteur de la tour à concevoir doit être plus grande que 150 m:

```
(↑(TYPE $TR TOUR)(COMPOSANT $TR $COL)(TYPE $COL COLONNE))
```

→ Fonct: Si BUT (PG HAUTEUR \$TR 150)

alors I?{(NOMBRE-DE-BLOCS \$COL 2)}).

. Le conseil ci-dessous se rapporte à l'exemple du transformateur. Il recommande de ne pas choisir des enroulements concentriques, si la tension de court-circuit du transformateur doit être inférieure à 2,5 pour cent de sa tension nominale:

```
{(TYPE $TRSF TRANSFORMATEUR)(COMPOSANT $TRSF $BOB)(TYPE $BOB BOBINAGE)
  (COMPOSANT $BOB $ENR)(TYPE $ENR ENROULEMENT)}
```

→ Fonct: Si BUT(PP TENSION-COURT-CIRCUIT-% \$TRSF 2.5)
alors I?{(NON DISPOSITION \$ENR CONCENTRIQUE)}).

b/ La partie droite d'un conseil peut aussi contenir des questions auxquelles le concepteur répondra pendant la résolution du problème, d'après la solution partielle engendrée.

Exemples:

Les deux conseils ci-dessous se rapportent à l'exemple du transformateur.

. Le conseil suivant est activé lorsque le circuit magnétique du transformateur est ajouté par Tropic à l'état courant. Le système demande alors au concepteur de lui conseiller la qualité de tole (normalisée) à choisir.

```
{(TYPE $CM CIRCUIT-MAGNETIQUE)}
```

→ {(QUALITE-TOLE \$CM(\$DEMANDER(QUALITE DE TOLE CONSEILLEE?)))}

[Commentaires: DEMANDER est une fonction de Tropic qui imprime le message qu'elle prend pour argument sur le terminal du concepteur ; elle s'évalue à la réponse du concepteur. \$ est une autre fonction qui permet de traiter une fonction comme une variable, lors d'une instanciation ; ainsi, l'instanciation de la partie droite d'une instance de ce conseil consiste à remplacer la variable \$CM par une constante et à évaluer la fonction DEMANDER.]

. Le conseil suivant est activé chaque fois qu'un conducteur est ajouté par Tropic à l'état courant. Le système demande alors au concepteur de lui conseiller la forme (ronde ou méplat) et les dimensions (normalisées) de ce conducteur.

```
{(TYPE $COND CONDUCTEUR )}
```

→ Fonct: \$F ← DEMANDER(FORME CONSEILLEE DE \$COND?)

si \$F = RONDE

```

alors I?{(FORME $COND RONDE)
          (DIAMETRE $COND($ (DEMANDER (DIAMETRE CONSEILLE?))) ) }
sinon I?{(FORME $COND MEPLAT)
          (EPAISSEUR $COND ($ (DEMANDER (EPAISSEUR CONSEILLEE?))) )
          (HAUTEUR $COND ($ (DEMANDER (HAUTEUR CONSEILLEE?))) ) }

```

Ce mode d'interaction est aussi permis avec les rcm, les contraintes et les diagnostics. Toutefois, dans les exemples que nous avons traités (notamment le transformateur), c'est son utilisation dans les conseils et les diagnostics qui s'est révélée la plus intéressante.

3.2. Le choix des ircm

Nous nous plaçons à un instant quelconque du cheminement avant. Soient P_b le problème formel à décomposer, I l'ensemble des ircm applicables à P_b et (c, q) le couple de dépendance de P_b .

Tropic restreint son choix aux seules ircm de I dont l'application ne résulterait pas en la contradiction immédiate d'une restriction contenue dans l'ensemble $Restrictions(c, q)$. Il calcule le mérite $\lambda = (\lambda' \lambda'')$ de chacune de ces ircm:

- . λ' est égal au nombre de restrictions de $Restrictions(c, q)$ qui seraient immédiatement satisfaites par l'application de l'ircm,
- . λ'' est égal au nombre de restrictions de $F-Restrictions(c, q)$ qui seraient immédiatement satisfaites par l'application de l'ircm, moins le nombre de restrictions de $F-Restrictions(c, q)$ qui seraient immédiatement contredites.

Nous disons que le mérite $\lambda_1 = (\lambda'_1 \lambda''_1)$ d'une ircm i_1 est plus grand que le mérite $\lambda_2 = (\lambda'_2 \lambda''_2)$ d'une ircm i_2 , si $\lambda'_1 > \lambda'_2$, ou si $\lambda'_1 = \lambda'_2$ et $\lambda''_1 > \lambda''_2$. Tropic choisit d'appliquer l'ircm qui a le plus grand mérite. Si plusieurs ircm satisfont ce critère, il choisit l'une quelconque d'entre elles.

Tropic est en échec si $I = NIL$ ou si l'application de chaque ircm de I résulterait en la contradiction immédiate d'une restriction contenue dans $Restrictions(c, q)$.

Exemple:

Soit $P_b = (\text{DETERMINER-COULEUR-BLOC BLC1})$. $I = \{i_1 i_2\}$, où $i_1 = \text{rcm } 10: \{(\$BLC BLC1)\}$ et $i_2 = \text{rcm } 11: \{(\$BLC BLC1)\}$. Soient λ_1 et λ_2 les mérites respectifs de i_1 et de i_2 .

. Si Restrictions (BLC1,0) = {(SECTION BLC1 RONDE)} et F-Restrictions (BLC1,0) = {(COULEUR BLC1 BLANCHE)}, alors $\lambda_1 = (0 \ 1)$ et $\lambda_2 = (0 \ -1)$.

. Si Restrictions (BLC1,0) = {(COULEUR BLC1 BLANCHE)} et F-Restrictions(BLC1,0) = {(NON COULEUR BLC1 ROUGE)}, alors $\lambda_1 = (1 \ 1)$; λ_2 n'est pas calculée car l'application de i_2 résulterait en la contradiction immédiate de la restriction (COULEUR BLC1 BLANCHE).

Commentaires

a/ A certains sommets-OU peuvent être associés des indicateurs d'échec qui décrivent les conditions dans lesquelles des échecs ont déjà été rencontrés. Avant de considérer l'ircm associée à un de ces sommets, Tropic vérifie que les conditions décrites par l'indicateur correspondant ne sont plus satisfaites. Dans le cas contraire, il ne prend pas l'ircm en considération (cf. chapitre 6).

b/ Supposons qu'une colonne puisse être composée de 1, 2, 3 ou 4 blocs, et qu'une instance de conseil recommande de choisir un nombre de blocs égal à 3. Supposons de plus qu'un échec résulte de ce choix. L'heuristique alors utilisée par Tropic est sensiblement plus compliquée que celle que nous venons de décrire. Elle lui permet de choisir un nombre de blocs égal à 2 ou 4, avant de choisir (éventuellement) un nombre de blocs égal à 1.

c/ En aucun cas la non satisfaction d'une restriction de conseil ne correspond à un échec. Les conseils ne sont en effet utilisés par Tropic que pour déterminer le mérite des ircm. Une instance de conseil qui devient active après que le choix d'ircm qu'elle permet d'aider ait été effectué, n'est donc pas utilisée par le système.

Chapitre 5: LE TRAITEMENT DES DONNES NUMERIQUES

Parmi les données qui constituent un modèle construit par Tropic, nous distinguons les données symboliques et les données numériques:

- . les données symboliques sont choisies dans des ensembles contenant assez peu d'éléments, par exemple: la couleur d'un bloc, la position relative de deux blocs, la forme des joints d'un circuit magnétique ;
- . les données numériques sont choisies dans des continuum de nombres, par exemple; la hauteur d'une colonne, la puissance d'un transformateur, les pertes magnétiques dans un circuit magnétique.

Les méthodes décrites aux chapitres 3 et 4 permettent d'engendrer les données symboliques d'un modèle, mais se prêtent mal à la génération des données numériques. Pour ces dernières, Tropic applique un traitement différent qui est décrit dans ce chapitre. Ce traitement complète, sans remettre en question, les méthodes présentées précédemment.

1. Les D-S-littéraux et les D-N-littéraux

La valeur de la propriété structurante du prédicat d'un D-littéral l est soit (D S), soit (D N):

. Dans le premier cas, nous disons que l est un D-S-littéral (relation descriptive symbolique). Les constantes (éventuellement numériques) pouvant servir à noter son A-terme doivent apparaître dans les rcm du problème original, par exemple: CARREE, RONDE, ROUGE, NOIRE, 1, 2.

. Dans le deuxième cas, nous disons que l est D-N-littéral (relation descriptive numérique). Les constantes pouvant servir à noter son A-terme sont les nombres réels et n'ont donc pas à apparaître dans les rcm.

Les D-S-littéraux et les A-littéraux correspondent aux données qualifiées de symboliques dans l'introduction ; les D-N-littéraux correspondent aux données numériques.

Exemple:

Dans les exemples à venir, nous supposons que $Struct(TYPE) = Struct(NOMBRE-DE-BLOCS) = Struct(FORME) = Struct(SECTION) = Struct(COULEUR) = (D S)$, et que $Struct(HAUTEUR) = (D N)$. Ainsi, $(FORME \text{ TT1 PYRAMIDALE})$ est un D-S-littéral, et $(HAUTEUR \text{ TT1 } 30)$ est un D-N-littéral.

Commentaire:

Les méthodes numériques actuellement utilisées par Tropic imposent que les A-termes constants des D-N-littéraux soient choisis dans l'ensemble des nombres réels (cf. §4.3.2.). On pourrait toutefois imaginer qu'il y ait plusieurs catégories de D-N-littéraux correspondant à des ensembles différents de nombres (\mathbb{R} , \mathbb{N} , \mathbb{C}).

2. Les restrictions symboliques et les restrictions numériques

Soient une contrainte ($Activ \rightarrow Restrict$) et $r \in Restrict$:

- . si r est un D-S-littéral ou un A-littéral, nous disons que r est une restriction symbolique ;
- . si r est un D-N-littéral, nous disons que r est une restriction numérique déclarative ;
- . dans les autres cas, nous disons que r est une restriction numérique procédurale ; elle est alors de la forme $((p \ v) \leftarrow (f(p_1 \ v_1)(p_2 \ v_2) \dots (p_n \ v_n)))$ par exemple $((HAUTEUR \ \$TR) \leftarrow (+ (HAUTEUR \ \$COL)(HAUTEUR \ \$TT)))$; f est le nom d'une fonction connue de Tropic (cas de la fonction $+$) ou programmée par l'utilisateur (dans ce cas, la fonction fait partie de l'énoncé du problème

Les restrictions symboliques et les restrictions numériques déclaratives sont surtout utiles pour exprimer des objectifs. Les restrictions numériques procédurales servent principalement à exprimer les équations d'analyse par exemple: $P = \sqrt{3}UI \cos\phi$, $\Delta\theta = k\left(\frac{W}{S}\right)^{0,8}$.

Exemples:

. La contrainte:

```
{(TYPE $BLC BLOC)(SECTION $BLC CARREE)}
  → {(COULEUR $BLC BLANCHE)
      (PG HAUTEUR $BLC 50)(PP HAUTEUR $BLC 100)}
```

impose une restriction symbolique, (COULEUR \$BLC BLANCHE), et les deux restrictions numériques déclaratives, (PG HAUTEUR \$BLC 50) et (PP HAUTEUR \$BLC 100). Elle impose que la couleur d'un bloc de section carrée soit blanche et que sa hauteur soit comprise entre 50 et 100m.

. La contrainte:

```
{(TYPE $TR TOUR)(COMPOSANT $TR $COL)(COMPOSANT $TR $TT)(TYPE $COL COLONNE)
 (TYPE $TT TOIT)} → {((HAUTEUR $TR) ← (+ (HAUTEUR $COL) (HAUTEUR $TT)))}
```

impose une restriction numérique procédurale. Elle impose que la hauteur d'une tour composée d'une colonne et d'un toit soit égale à la somme des hauteurs de ses deux composants.

3. Définition d'une solution

Dans ce paragraphe, nous complétons la définition d'une solution donnée au chapitre 2 en tenant compte des restrictions numériques à satisfaire. Auparavant nous définissons la compatibilité d'un ensemble de restrictions numériques.

3.1. Compatibilité d'un ensemble de restrictions numériques

Soit $N = \{r_1, r_2, \dots, r_q\}$ un ensemble de restrictions numériques constantes. A chaque restriction r_i ($i=1$ à q), nous faisons correspondre une équation ou une inéquation de la façon suivante:

- . si $r_i = (p \ c \ a)$, l'équation correspondante est $x(p,c)=a$, où $x(p,c)$ est une variable associée de manière unique au couple $(p \ c)$,
- . si $r_i = (PG \ p \ c \ a)$, l'inéquation correspondante est $x(p,c) > a$,
- . si $r_i = (PP \ p \ c \ a)$, l'inéquation correspondante est $x(p,c) < a$,
- . si $r_i = ((p \ c) \leftarrow \{f(p_1 \ c_1)(p_2 \ c_2) \dots (p_n \ c_n)\})$, l'équation correspondante est $x(p,c) = f(x(p_1, c_1) \ x(p_2, c_2) \dots x(p_n, c_n))$.

[Note: Nous imposons qu'une restriction numérique déclarative ne soit pas exprimée par un D-N-littéral négatif.]

Soit S le système d'équations et d'inéquations ainsi formé. S'il admet une solution, nous disons que l'ensemble N est compatible ; sinon, nous disons qu'il est incompatible.

Nous disons qu'un ensemble N de D-N-littéraux positifs constants satisfait N si et seulement si les deux conditions suivantes sont vérifiées :

- . à chaque variable $x(p,c)$ de S , il correspond un littéral $(p \ c \ a)$ unique dans N ,
- . l'ensemble des A-termes des littéraux de N est une solution de S (le A-terme de chaque littéral $(p \ c \ a)$ étant donné comme valeur à $x(p,c)$).

Exemple:

Soit $N = \{(HAUTEUR \ TR1 \ 215)(HAUTEUR \ TT1 \ 30)(PG \ HAUTEUR \ BLC1 \ 50)(PP \ HAUTEUR \ BLC1 \ 100)(PG \ HAUTEUR \ BLC2 \ 50)(PP \ HAUTEUR \ BLC2 \ 100)((HAUTEUR \ TR1) \leftarrow (+ (HAUTEUR \ COL1)(HAUTEUR \ TT1))) \ ((HAUTEUR \ COL1) \leftarrow (+ (HAUTEUR \ BLC1)(HAUTEUR \ BLC2)))\}$.

Le système d'équations et d'inéquations correspondant est:

$$x(HAUTEUR, TR1) = 215$$

$$x(HAUTEUR, TT1) = 30$$

$$50 < x(HAUTEUR, BLC1) < 100$$

$$50 < x(HAUTEUR, BLC2) < 100$$

$$x(HAUTEUR, TR1) = (+ (HAUTEUR \ COL1)(HAUTEUR \ TT1))$$

$$x(HAUTEUR, COL1) = (+ (HAUTEUR \ BLC1)(HAUTEUR \ BLC2))$$

Une solution particulière de ce système est $x(HAUTEUR, TR1) = 215$, $x(HAUTEUR, COL1) = 185$, $x(HAUTEUR, TT1) = 30$, $x(HAUTEUR, BLC1) = 95$, $x(HAUTEUR, BLC2) = 90$. N est donc compatible et l'ensemble $N = \{(HAUTEUR \ TR1 \ 215)(HAUTEUR \ COL1 \ 185)(HAUTEUR \ TT1 \ 30)(HAUTEUR \ BLC1 \ 95)(HAUTEUR \ BLC2 \ 90)\}$ satisfait N .

3.2. Définition d'une solution

Pour cette définition, nous imposons que les deux conditions suivantes soient vérifiées:

- . Les rcm ne doivent contenir aucun D-N-littéral dans leurs parties droites. Elles ne permettent donc d'engendrer que les données symboliques d'une solution.
- . Les conditions d'activation d'une contrainte ne doivent contenir aucun

D-N-littéral. Nous interdisons ainsi des contraintes telles que: $\{(TYPE \$TT TOIT)(PG HAUTEUR \$TT 20)\} \rightarrow \{(FORME \$TT PYRAMIDALE)\}$. Une telle contrainte doit être écrite: $\{(TYPE \$TT TOIT)(NON FORME \$TT PYRAMIDALE)\} \rightarrow \{(PP HAUTEUR \$TT 20)\}$.

Nous disons qu'un ensemble de littéraux positifs et constants $M = M_1 \cup M_2$, où M_1 est un ensemble de A- et de D-S-littéraux et M_2 est un ensemble de D-N-littéraux, est une solution du problème original si et seulement si il vérifie les trois conditions suivantes:

- . $(M_1 \text{ NIL})$ appartient à l'espace des possibilités déterminé par les rcm,
- . M_1 satisfait toutes les restrictions symboliques appliquées par les instances de contraintes actives dans M_1 (i.e.: les littéraux qui expriment ces restrictions ont la valeur VRAI dans M_1).
- . M_2 satisfait l'ensemble des restrictions numériques appliquées par les instances de contraintes actives dans M_1 .

Cette définition ne remet pas en question la résolution d'un problème telle que nous l'avons décrite aux chapitres 3 et 4. Elle impose en plus que Tropic vérifie la compatibilité des restrictions numériques actives.

Commentaire:

Le formalisme des restrictions numériques ne permet pas au concepteur d'imposer à Tropic l'optimisation d'une fonction comme par exemple le coût d'un transformateur. Le concepteur doit définir cette fonction à l'aide d'une restriction numérique procédurale, et imposer par une restriction numérique déclarative que sa valeur soit inférieure (ou supérieure) à une valeur qu'il juge acceptable. Cette dernière valeur peut être ajustée au cours d'une utilisation itérative du système.

4. Vérification de la compatibilité des restrictions numériques

4.1. Fréquence de cette vérification

Soient $(m d)$ la possibilité engendrée à un instant quelconque du cheminement avant et N l'ensemble des restrictions numériques imposées par les

instances de contraintes actives dans m . Si N est incompatible, alors il est impossible d'engendrer une solution en continuant le cheminement avant sans modification. La reconnaissance par Tropic de l'incompatibilité de N correspond donc à un échec.

Tropic éviterait de décomposer inutilement certains problèmes formels de l'acm s'il reconnaissait l'incompatibilité des restrictions numériques actives dès que celle-ci est effective. Toutefois, les calculs nécessaires à cette reconnaissance sont généralement plus longs que la décomposition de quelques problèmes formels. Aussi, dans la version actuelle de Tropic, nous avons choisi (arbitrairement) de ne les effectuer qu'après la génération de chaque niveau de détail.

4.2. Le graphe des restrictions numériques

Au cours du cheminement avant, Tropic construit un graphe aux sommets duquel il associe les restrictions numériques actives (la figure 5.1. en montre un exemple). Nous appelons ce graphe le graphe des restrictions numériques, ou plus brièvement le grn. Tropic s'en sert lors de la vérification de la compatibilité des restrictions numériques (cf. § 4.3.1.).

Parmi les sommets du grn, nous distinguons les D-sommets (notés \circ sur la figure 5.1.) et les P-sommets (notés \square):

- . Chaque D-sommet est étiqueté par un couple $(p \ c)$ tel que $\text{struct}(p) = (D \ N)$ et c soit une constante servant à noter un C-terme de l'état courant. De plus l'ensemble $N\text{-Restrictions}(p,c)$ des restrictions numériques déclaratives actives ayant p pour prédicat et c pour C-terme est associé à ce sommet.

- . A chaque P-sommet est associé une restriction numérique procédurale active. Si $((p \ c) \leftarrow (f(p_1 \ c_1)(p_2 \ c_2) \dots (p_n \ c_n)))$ est cette restriction, alors:

- un arc unique part du P-sommet ; l'extrémité de cet arc est le D-sommet étiqueté par $(p \ c)$;
- n arcs, et n seulement, arrivent au P-sommet ; leurs origines respectives sont les D-sommets étiquetés par $(p_1 \ c_1)$, $(p_2 \ c_2)$, ..., $(p_n \ c_n)$.

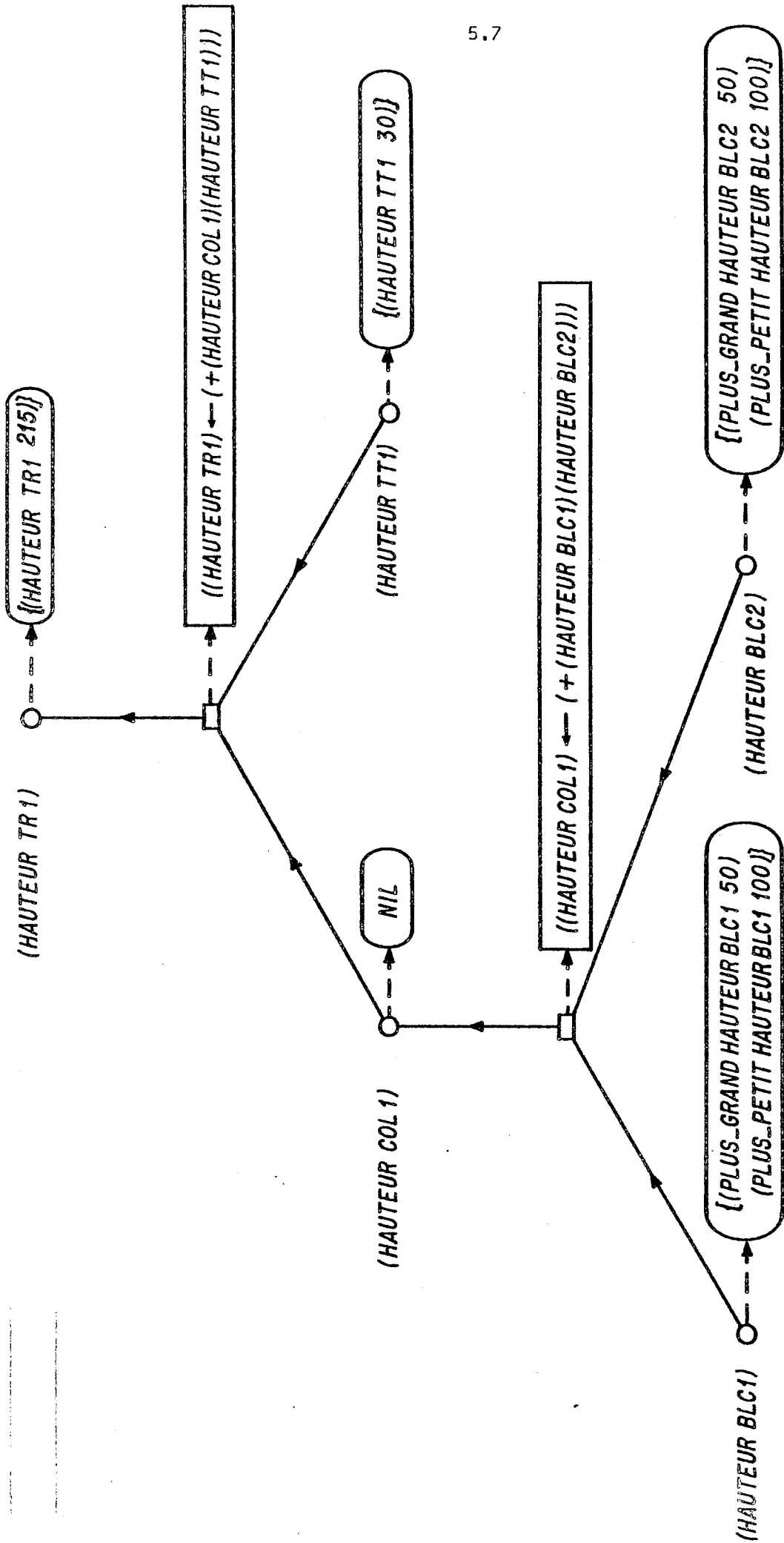


Figure: 5.1

Chaque restriction numérique active est associée à un sommet et à un seul du grn. De plus, le grn ne contient pas de D-sommet isolé auquel est associé l'ensemble vide de restriction numérique déclarative. Le grn peut toutefois ne pas être connexe.

Commentaire

Avant de ranger une restriction numérique déclarative dans un ensemble N-Restrictions(p,c), Tropic vérifie qu'elle n'est pas incompatible avec une restriction déjà contenue dans cet ensemble (par exemple, (HAUTEUR TT1 30) et (PG HAUTEUR TT1 35) sont deux restrictions incompatibles). Tropic reconnaît donc ces cas simples d'incompatibilité sans avoir besoin d'effectuer les calculs décrits au paragraphe suivant, et les traite immédiatement.

4.3. Vérification de la compatibilité des restrictions numériques

La vérification de la compatibilité des restrictions numériques est traitée comme un problème de recherche de minimum d'une fonction. La formulation actuelle de ce problème requiert que le grn soit un graphe sans cycle et qu'un arc au plus arrive à chacun de ses D-sommets. Toutefois, nous verrons au paragraphe 4.3.3.b que ces limitations peuvent être évitées.

Les exemples que nous donnons dans les paragraphes 4.3.1. et 4.3.2. se rapportent au grn de la figure 5.1.

4.3.1. Formulation du problème

Nous définissons le niveau de chaque D-sommet N du grn de façon réursive:

- . si N n'a pas de prédécesseur dans le grn, son niveau est 0,
- . si tous les prédécesseurs de N ont un niveau $\leq n$, et si au moins l'un d'eux a le niveau n, alors le niveau de N est n+1.

Exemple:

Le niveau des D-sommets étiquetés par (HAUTEUR BLC1), (HAUTEUR BLC2) et (HAUTEUR TT1) est 0 ; le niveau du D-sommet étiqueté par (HAUTEUR COL1) est 1 et celui du D-sommet étiqueté par (HAUTEUR TR1) est 2.

A chaque D-sommet du grn, nous associons de manière biunivoque une variable que nous notons X_i^n où n désigne le niveau du D-sommet.

Exemple:

Nous notons $X_1^0, X_2^0, X_3^0, X_1^1, X_1^2$ les variables associées aux D-sommets respectivement étiquetés par (HAUTEUR BLC1), (HAUTEUR BLC2), (HAUTEUR TT1), (HAUTEUR COL1) et (HAUTEUR TR1).

Nous définissons le programme du grn comme une séquence d'instructions construite de la façon suivante:

. Une instruction est engendrée pour chaque P-sommet du grn ; si $((p\ c) \leftarrow (f(p_1\ c_1)(p_2\ c_2)\dots(p_m\ c_m)))$ est la restriction associée à un P-sommet, l'instruction engendrée est $X_i^n \leftarrow f(X_{i_1}^{n_1} X_{i_2}^{n_2} \dots X_{i_m}^{n_m})$, où $X_i^n, X_{i_1}^{n_1}, X_{i_2}^{n_2}, \dots, X_{i_m}^{n_m}$ sont les variables associées aux D-sommets étiquetés par $(p\ c), (p_1\ c_1), (p_2\ c_2), \dots, (p_m\ c_m)$.

[Note: On remarque qu'une variable X_i^0 n'apparaît jamais dans la partie gauche d'une instruction ainsi formée. De plus, chaque variable X_i^q ($q \geq 1$) apparaît dans la partie gauche d'une instruction et d'une seule, car nous avons imposé qu'un arc et un seul arrive à chaque D-sommet du grn.]

Les instructions ainsi formées sont ordonnées en une séquence telle que, pour chaque variable X_i^q ($q \geq 1$), l'instruction où X_i^q figure en partie gauche précède les instructions éventuelles où X_i^q figure en partie droite. Le programme du grn calcule donc les valeurs des variables X_i^q ($q \geq 1$) en fonction des valeurs des variables X_i^0 de façon à satisfaire les restrictions numériques procédurales associées aux P-sommets du grn.

Exemple:

Le programme du grn de la figure 4.1. est:

$$X_1^1 \leftarrow + (X_1^0 X_2^0)$$

$$X_1^2 \leftarrow + (X_1^1 X_3^0)$$

[Note: Dans l'exemple du transformateur, ce programme peut avoir près d'une centaine d'instructions.]

Soient N un D-sommet du grn, X_i^n la variable associée à N et r une restriction contenue dans l'ensemble N -Restrictions(p,c) associée à N . Nous définissons la violation de r comme une fonction de X_i^n :

. si $r = (p \ c \ a)$, la violation de r est $\left[\frac{X_i^n - a}{a} \right]^2$ ($[X_i^n]^2$ dans le cas où $a = 0$),

. si $r = (PG \ p \ c \ a)$, la violation de r est $\left[\frac{X_i^n - a}{a} \right]^2$ ($[X_i^n]^2$ dans le cas où $a = 0$) quand $X_i^n < a$, et 0 quand $X_i^n > a$,

. si $r = (PP \ p \ c \ a)$, la violation de r est $\left[\frac{X_i^n - a}{a} \right]^2$ ($[X_i^n]^2$ dans le cas où $a = 0$) quand $X_i^n > a$, et 0 quand $X_i^n < a$.

Exemples:

. La violation de (HAUTEUR TRI 215) est $\left[\frac{X_1^2 - 215}{215} \right]^2$.

. La violation de (PG HAUTEUR BLC1 50) est $\left[\frac{X_1^0 - 50}{50} \right]^2$ quand $X_1^0 < 50$, et 0 quand $X_1^0 > 50$.

Nous définissons la violation du grn comme étant la somme des violations de toutes les restrictions numériques associées aux D-sommets du grn.

La vérification de la compatibilité des restrictions numériques actives est ainsi formulée comme un problème de recherche de minimum dans \mathbb{R}^d , d étant le nombre de D-sommets de niveau 0. Il s'agit en effet de déterminer un point $(a_1 \ a_2 \ \dots \ a_d)$ de \mathbb{R}^d telle que la violation du grn soit nulle (minimum de cette fonction) lorsque X_i^0 prend la valeur a_i ($\forall i \in [1,d]$), les valeurs des variables X_j^q ($q \geq 1$) étant calculées par le programme du grn. Si un tel point peut être déterminé, les restrictions numériques enregistrées dans le grn sont compatibles. Tropic utilise alors les coordonnées a_1, a_2, \dots, a_d de ce point pour engendrer un ensemble de D-N-littéraux qui satisfait ces restrictions.

4.3.2. Résolution du problème

Dans la version actuelle de Tropic, le problème ainsi formulé est résolu en appliquant la méthode de Hooke et Jeeves (Hooke 1961). Nous avons choisi cette méthode en raison de son utilisation satisfaisante dans le système CADSYS (Milley 1972, Roxlo 1973, Whitney 1975) pour traiter des problèmes analogues et parce qu'elle s'accommode de fonctions non parfaitement continues. De plus, c'est une méthode simple.

Le programme qui met en oeuvre cette méthode travaille sur une liste de d éléments représentant les valeurs courantes des variables $X_1^0, X_2^0, \dots, X_d^0$ (point de \mathbb{R}^d).

En entrée, ce programme reçoit la liste des valeurs initiales des variables $X_1^0, X_2^0, \dots, X_d^0$. Ces valeurs sont soit déterminées par Tropic (à l'aide des restrictions numériques déclaratives les concernant, ou en utilisant les valeurs obtenues lors de calculs précédents), soit demandées à l'utilisateur.

Pendant son exécution, le programme effectue des déplacements dans \mathbb{R}^d ("exploration moves" et "pattern moves"). Après chaque déplacement, le programme du grn est exécuté et la violation du grn est calculée.

En sortie, le programme fournit les valeurs des variables X_i^0 qui minimisent la violation du grn et la valeur correspondante de cette violation. Si cette valeur est nulle, les restrictions numériques sont compatibles, sinon Tropic considère qu'elles sont incompatibles (cf. § 4.3.3.b)

Remarques

a/ Le programme arrête la recherche numérique dès que la violation du grn est nulle.

b/ Si l'ensemble N -Restrictions(p, c) associé à un D -sommet de niveau 0 contient une restriction positive ($p < a$), alors la dimension de l'espace de recherche numérique est diminuée d'une unité. Ceci est réalisé comme dans CADSYS en retirant la variable X_i^0 correspondante de la liste des valeurs sur laquelle travaille le programme de minimisation (cas de la variable X_3^0 dans le grn de la figure 5.1.).

c/ Les déplacements effectués dans l'espace de recherche sont restreints à l'hypervolume connexe dans lequel les restrictions associées aux D-sommets de niveau 0 sont satisfaites. Cette modification nous a été suggérée par l'expérience de CADSYS (Roxlo 1973) où il a été constaté que certains déplacements ("pattern moves" notamment) trop amples effectués par la méthode de Hooke et Jeeves résultaient en des valeurs sans "signification physique" (dimension négative par exemple).

4.3.3. Commentaires

a/ La formulation du problème de minimisation pourrait être modifiée pour éviter les limitations que nous avons imposées sur le grn au début du paragraphe 4.3.

Une formulation consisterait à traiter uniformément les restrictions numériques déclaratives et procédurales en définissant la violation d'une

restriction numérique procédurale par
$$\left[\frac{X_i^n - f(X_{i_1}^{n_1}, X_{i_2}^{n_2}, \dots, X_{i_m}^{n_m})}{f(X_{i_1}^{n_1}, X_{i_2}^{n_2}, \dots, X_{i_m}^{n_m})} \right]^2$$
. Le

problème consisterait alors à trouver un point de l'espace \mathbb{R}^{d_1} (d_1 étant le nombre total de D-sommets dans le grn) où les violations de toutes les restrictions numériques seraient nulles. Le programme du grn n'aurait plus d'existence et, à chaque déplacement dans \mathbb{R}^{d_1} , seule la violation totale serait calculée. Toutefois, cette formulation présenterait deux inconvénients :

- . la dimension de l'espace de recherche numérique serait augmentée du nombre de D-sommets ayant un niveau non nul,
- . à un instant de la recherche numérique, certaines restrictions procédurales représentant des lois physiques ou géométriques pourraient ne pas être satisfaites. Les valeurs des variables X_i^n manqueraient alors de "signification physique" (par exemple, la hauteur d'une tour pourrait être supérieure ou inférieure à la somme des hauteurs de sa colonne et de son toit).

Une autre formulation permettrait de réduire ces inconvénients. Elle consisterait à déterminer un sous-graphe du grn dans lequel les conditions imposées au début du paragraphe 4.3. seraient vérifiées. Le programme du grn serait alors construit à l'aide de ce sous-graphe. La violation du grn serait définie comme la somme de violations des restrictions numériques déclaratives et des restrictions numériques procédurales associées aux P-sommets n'appartenant pas au sous-graphe. La dimension de l'espace de recherche numérique serait alors comprise entre d et d_1 . De plus, le nombre de restrictions procédurales pouvant ne pas être satisfaites serait diminué. La formulation pourrait encore être améliorée en surpondérant les violations de ces restrictions de façon à ce que ces restrictions soient satisfaites en priorité.

b/ Tropic considère que les restrictions numériques sont incompatibles si la minimisation s'arrête avec une violation du grn non nulle. Il y a là un abus de confiance certain dans la méthode de minimisation appliquée. En effet, aucune méthode de minimisation n'est absolument fiable et, en particulier, celle de Hooke et Jeeves peut se bloquer sur un point d'une arête qui ne correspond pas au minimum de la violation du grn.

Chapitre 6: LE TRAITEMENT DES ECHECS

Dans les chapitres 4 et 5, nous avons identifié les différents cas d'échec que Tropic peut subir au cours du cheminement avant. Dans chaque cas, le traitement de l'échec comporte deux parties (cf. chapitre 3, § 3):

- . la détermination d'un point de retour: il s'agit de déterminer le sommet-ET N de l'acm à partir duquel le cheminement avant sera repris,
- . le cheminement arrière au point de retour: il s'agit d'effacer les effets résultant de l'application des ircm qui sont associées aux sommets-OU successeurs de N dans l'acm, avant de reprendre le cheminement avant.

Le traitement appliqué par Tropic diffère nettement du traitement aveugle qui est incorporé dans les systèmes de programmation développés en Intelligence Artificielle, comme PLANNER (Hewitt 1971), QA4 (Rulifson 1972) et QLISP (Sacerdoti 1976) :

. Dans ces systèmes, le point de retour est toujours le point de décision chronologiquement le plus proche de l'échec, où il existe une alternative encore inexplorée. De plus, les décisions sont enregistrées le long d'un chemin d'un graphed'état, de telle sorte que le travail effectué entre ce point de décision et l'instant où l'échec a été reconnu est entièrement effacé.

. Dans Tropic, la détermination d'un point de retour suit une analyse des conditions dans lesquelles l'échec a été subi. Cette analyse évite à Tropic d'effectuer des cheminements arrière inutiles à des sommets-ET où les choix effectués ne peuvent pas être responsables de l'échec. Par ailleurs, la mémorisation des décisions prises pendant le cheminement avant dans une arborescence ET/OU lui permet de n'effacer que la partie du travail qui dérive du point de retour.

Au paragraphe 1, nous introduisons les principaux concepts du traitement des échecs appliqué par Tropic, en les illustrant par des exemples. Au

Nous supposons que Tropic effectue le cheminement avant décrit par le tableau 6.1. (parmi les heuristiques qui guident ce cheminement figurent éventuellement des conseils). L'acm engendrée est montrée à la figure 6.1. ; l'ordre de décomposition des problèmes formels est indiqué entre parenthèses à coté de chaque sommet-ET.

Au cours de ce cheminement avant:

- . C1: $\{(\$TR TR1)\}$ devient active après l'application de rcm1:NIL. Elle impose la restriction numérique (HAUTEUR TR1 200).
- . C4: $\{(\$TR TR1)(\$COL COL1)(\$TT TT1)\}$ devient active après l'application de rcm12: $\{(\$TT TT1)\}$. Elle impose la restriction numérique $\{(HAUTEUR TR1) + (+ (HAUTEUR COL1)(HAUTEUR TT1))\}$.
- . C2: $\{(\$TT TT1)(\$F PRISMATIQUE)\}$ devient active après l'application de rcm14: $\{(\$TT TT1)\}$. Elle impose la restriction numérique (HAUTEUR TT1 15).
- . C3: $\{(\$BLC BLC1)\}$ devient active après l'application de rcm7: $\{(\$BLC BLC1)\}$. Elle impose les restrictions numériques (PG HAUTEUR BLC1 50) et (PP HAUTEUR BLC1 90).
- . C3: $\{(\$BLC BLC2)\}$ devient active après l'application de rcm7: $\{(\$BLC BLC2)\}$. Elle impose les restrictions numériques (PG HAUTEUR 50) et (PP HAUTEUR BLC2 90).
- . C5: $\{(\$COL COL1)(\$BLC \{BLC1 BLC2\})\}$ devient active après l'application de rcm7: $\{(\$BLC BLC2)\}$. Elle impose la restriction numérique $\{(HAUTEUR COL1) + (+ (HAUTEUR BLC1)(HAUTEUR BLC2))\}$.

Après avoir engendré le niveau de détail 2 de la description de TR1, Tropic constate que les restrictions numériques actives sont incompatibles (cf. § 4.1., chapitre 5). Le grn courant est montré à la figure 6.2. L'incompatibilité des restrictions (due à l'inégalité $90+90+15 < 200$) constitue un échec dont nous décrivons le traitement en exemple dans les trois paragraphes suivants.

Etape exécutée	Ircm appliquée	Littéraux engendrés
Etape (TR1,0)	rcm1 : NIL	(TYPE TR1 TOUR)
Etape (TR1,1)	rcm2: {(\$TR TR1)}	(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1 (SUR COL1 TT1)
Etape (COL1,0)	rcm3: {(\$COL COL1)}	(TYPE COL1 COLONNE)
	rcm5: {(\$COL COL1)}	(NOMBRE-DE-BLOCS COL1 2)
Etape (TT1,0)	rcm12: {(\$TT TT1)}	(TYPE TT1 TOIT)
	rcm14: {(\$TT TT1)}	(FORME TT1 PRISMATIQUE)
	rcm15: {(\$TT TT1)}	(COULEUR TT1 ROUGE)
Etape (COL1,1)	rcm6: {(\$COL COL1)(\$N 2)}	(COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2)(SUR BLC1 BLC2)
Etape (BLC1,0)	rcm7: {(\$BLC BLC1)}	(TYPE BLC1 BLOC)
	rcm9: {(\$BLC BLC1)}	(SECTION BLC1 RONDE)
	rcm10: {(\$BLC BLC1)}	(COULEUR BLC1 BLANCHE)
Etape (BLC2,0)	rcm7: {(\$BLC BLC2)}	(TYPE BLC2 BLOC)
	rcm9: {(\$BLC BLC2)}	(SECTION BLC2 RONDE)
	rcm10: {(\$BLC BLC2)}	(COULEUR BLC2 BLANCHE)

Tableau 6.1.

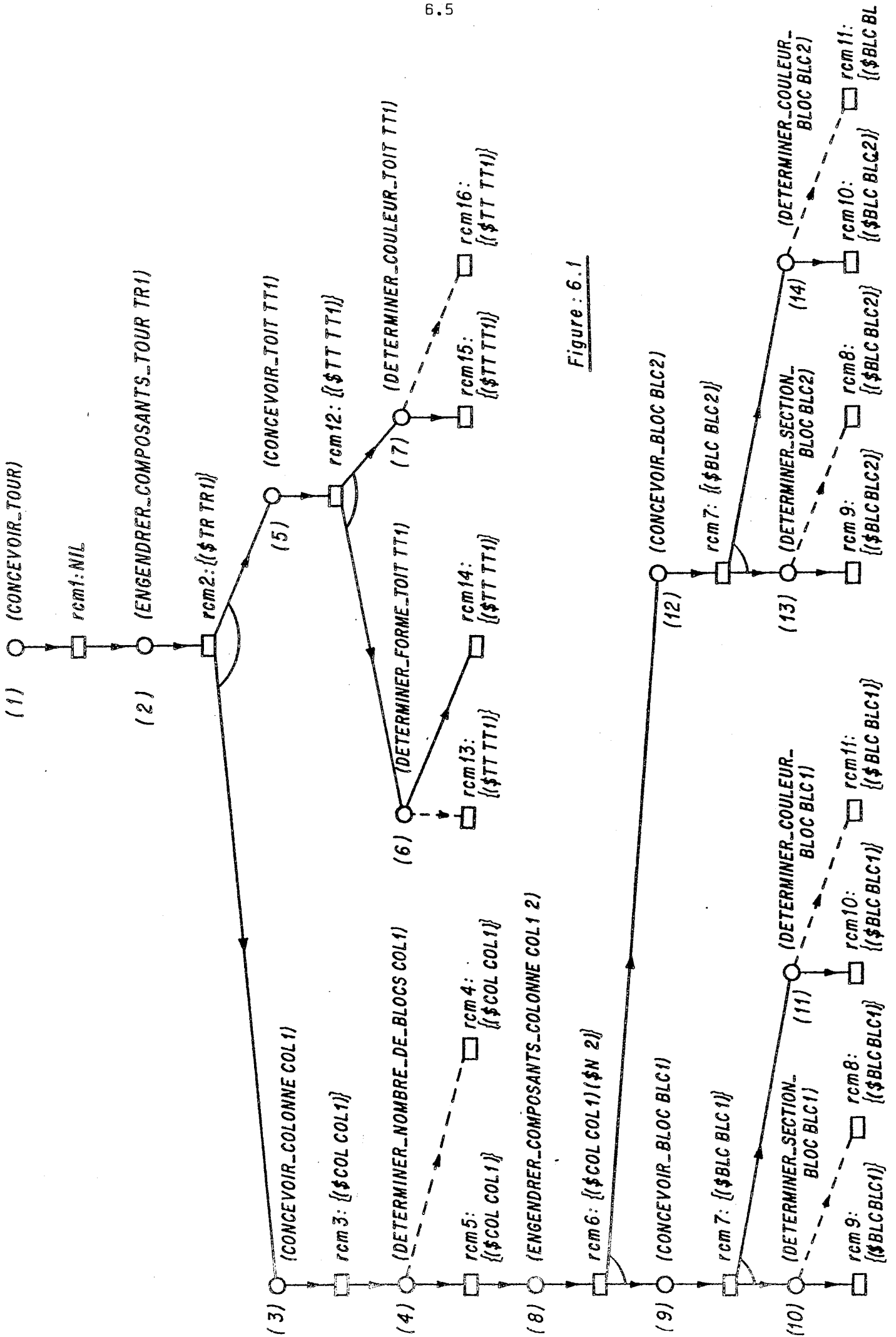


Figure : 6.1

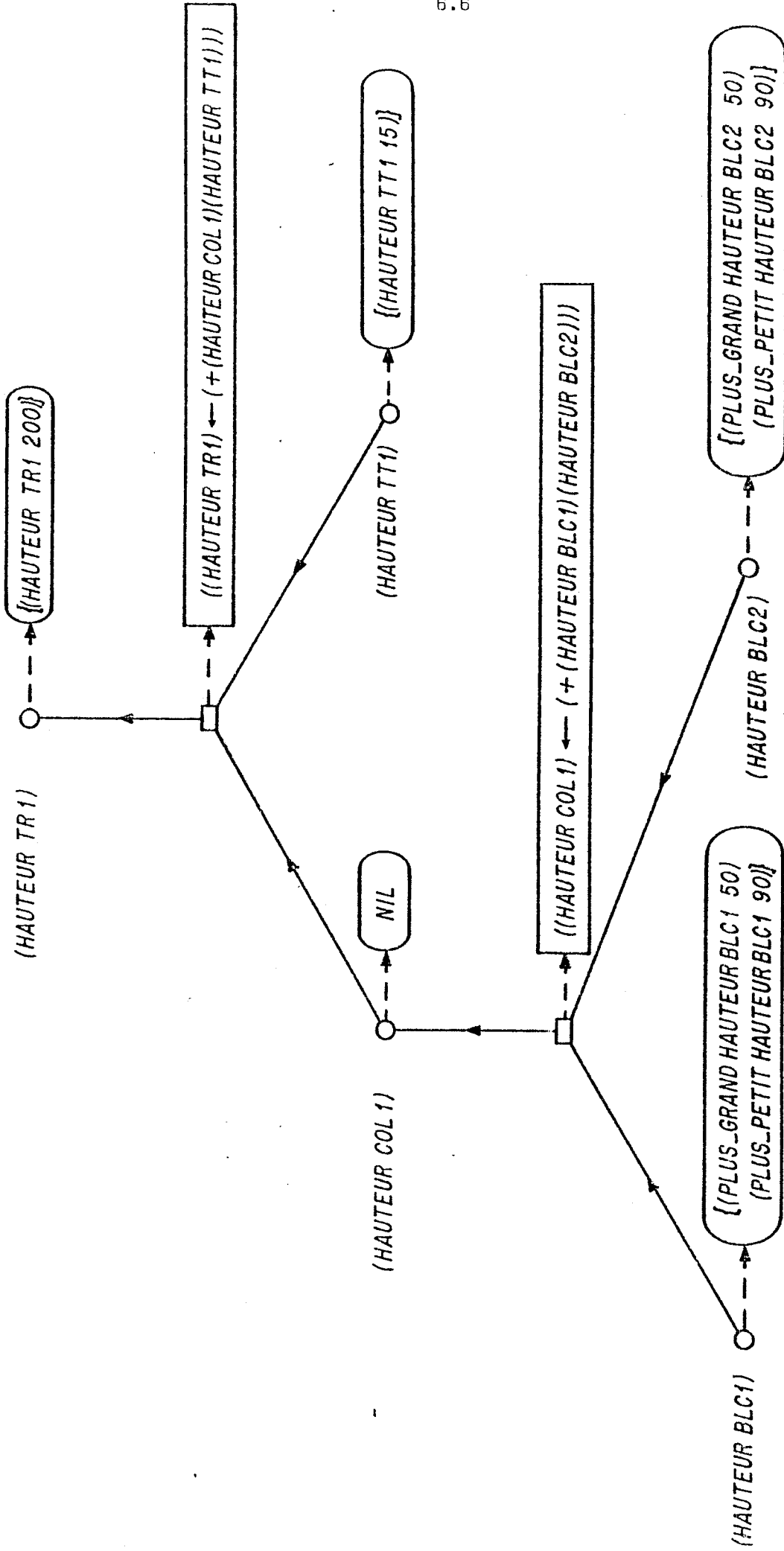


Figure: 6.2

1.1.2. Formation de l'expression causale

Dès qu'il est en échec, Tropic forme une expression, que nous appelons expression causale, à l'aide de conditions qu'il considère comme les causes de l'échec. Nous verrons au paragraphe 3.1.1. que cette expression détermine en fait une condition suffisante d'échec.

L'expression causale est un ensemble $\{c_1 c_2 \dots c_n\}$, où $\forall k \in [1, n]$, c_k est une condition qui est satisfaite à l'instant de l'échec. Cette condition a l'une des formes suivantes:

- 1/ "le littéral l a la valeur VRAI dans l'état courant",
- 2/ "le littéral l a la valeur FAUX dans l'état courant",
- 3/ "le problème formel P_b est associé à un sommet-ET de l'acm courante".

Au paragraphe 3.1.1., nous passerons en revue les cas d'échec que nous avons identifiés dans les chapitres 4 et 5, et nous définirons l'expression causale qui est formée par TROPIC dans chaque cas.

Dans tous les exemples du paragraphe 1, les conditions figurant dans les expressions causales considérées ont la forme 1/; pour simplifier, nous écrirons alors ces expressions $\{l_1 l_2 \dots l_n\}$, où l_i remplace la condition "le littéral l_i a la valeur VRAI dans l'état courant".

Exemple:

Après l'échec du paragraphe 1.1.1., Tropic forme une expression causale avec les littéraux qui représentent les conditions d'activation des instances de contraintes qui imposent les restrictions numériques incompatibles. La valeur VRAI de ces littéraux dans l'état courant est en effet responsable de l'activation de ces restrictions, et donc de l'échec. Les instances de contraintes concernées étant $C1: \{(\$TR TR1)\}$, $C4: \{(\$TR TR1)(\$COL COL1)(\$TT TT1)\}$, $C2: \{(\$TT TT1)(\$F PRISMATIQUE)\}$, $C3: \{(\$BLC BLC1)\}$, $C3: \{(\$BLC BLC2)\}$ et $C5: \{(\$COL COL1)(\$BLC \{BLC1 BLC2\})\}$, l'expression causale formée est:

$\{ (TYPE TR1 TOUR)(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1)(TYPE COL1 COLONNE)$
 $(TYPE TT1 TOIT) \quad (FORME TT1 PRISMATIQUE)(COMPOSANT$
 $COL1 BLC1)(COMPOSANT COL1 BLC2) \quad (TYPE BLC1 BLOC)(TYPE BLC2$
 $BLOC) \}$.

1.1.3. La détermination du point de retour

Tropic utilise l'expression causale pour déterminer le point de retour du cheminement arrière ; son but étant d'effacer une cause de l'échec, le point de retour est donc un sommet-ET N non pendants de l'acm auquel est associé un problème formel Pb tel que :

- . lors du cheminement avant, Tropic a appliqué à Pb une ircm i choisie parmi plusieurs ircm de la forme (Pb →...),
- . au moins une de ces ircm est telle que son application (en remplacement de : permettrait de rendre non satisfaite une condition contenue dans l'expression causale.

Tropic passe en revue tous les sommets-ET non pendants de l'acm dans l'ordre inverse de la décomposition des problèmes formels. Il choisit pour point de retour le premier sommet-ET examiné tel que le problème formel associé vérifie les deux conditions précédentes. Si aucun point de retour ne peut être déterminé, il considère que l'échec est irrémédiable (cf. § 2) et il termine alors la résolution du problème sur un échec.

Exemple:

Grace à l'expression causale du paragraphe 1.1.2, Tropic va donc déterminer pour point de retour un sommet-ET à partir duquel il pourra désactiver au moins une des restrictions numériques qui étaient incompatibles au moment de l'échec. Pour ce faire, il examine tour à tour les sommets-ET auxquels sont associés les problèmes formels suivants (cf. l'acm de la figure 6.1.):

- . (DETERMINER-COULEUR-BLOC BLC2): l'application de $rcm11: \{(\$BLC BLC2)\}$, en remplacement de $rcm10: \{(\$BLC BLC2)\}$, ne permettrait pas de modifier la valeur d'un littéral contenu dans l'expression causale ; Tropic ne choisit donc pas ce sommet pour point de retour.
- . (DETERMINER-SECTION-BLOC BLC2): pour la même raison, Tropic ne choisit pas ce sommet pour point de retour.
- . (CONCEVOIR-BLOC BLC2): il n'y a pas d'alternative à ce sommet ; Tropic ne le choisit donc pas pour point de retour.
- . (DETERMINER-COULEUR-BLOC BLC1), (DETERMINER-SECTION-BLOC BLC1), (CONCEVOIR-BLOC BLC1), (ENGENDRER-COMPOSANTS-COLONNE COL1 2), (DETERMINER-FORME-TOIT TT1): pour l'une ou l'autre des raisons précédentes, Tropic ne choisit aucun de ces sommets pour point de retour.

. (DETERMINER-FORME-TOIT TT1): l'application de $rcm13: \{(\$TT TT1)\}$, en remplacement de $rcm14: \{(\$TT TT1)\}$, permet de modifier la valeur du littéral (FORME TT1 PRISMATIQUE) qui est contenu dans l'expression causale. Tropic choisit donc ce sommet pour point de retour.

[Note: Un traitement aveugle de l'échec aurait consisté à effectuer des cheminement arrière successifs aux sommets correspondant aux problèmes (DETERMINER-COULEUR-BLOC BLC2), (DETERMINER-SECTION-BLOC BLC2), (DETERMINER-COULEUR-BLOC BLC1), (DETERMINER-SECTION-BLOC BLC1) et (DETERMINER-FORME-TOIT TT1), où il existe une alternative. Le point de retour qui est immédiatement choisi par Tropic aurait alors été déterminé par élimination après 63 échecs successifs].

1.1.4 Le cheminement arrière et la reprise du cheminement avant

Soient N le point de retour déterminé, Pb le problème formel associé à N, et i l'ircm qui est associée au sommet-OU successeur immédiat de N dans l'acm courante.

Le cheminement arrière consiste à effacer les conséquences de l'application de i dans l'acm et l'état courants. L'enregistrement des décisions prises pendant le cheminement avant dans une arborescence ET/OU permet simplement à Tropic de conserver des parties de modèle engendrées après la décomposition de Pb par i, mais indépendamment de cette décomposition (cf. § 3 et 5.1. du chapitre 3). Les mises à jour effectuées pendant le cheminement arrière conduisent aussi Tropic à désactiver des instances de contraintes, de conseils et de diagnostics qui étaient actives dans l'état courant à l'instant de l'échec.

Le cheminement avant est ensuite repris à partir du sommet N. Il est effectué comme nous l'avons décrit au paragraphe 1 du chapitre 3 et au paragraphe 2.4.2. du chapitre 4, à la différence principale près suivante: avant de décomposer un problème formel, Tropic vérifie que ce problème n'est pas déjà décomposé dans l'acm ; si c'est le cas, il ne modifie pas cette décomposition, et passe au problème formel suivant (nous verrons cependant aux paragraphes 1.2. et 2.3. que cela n'est pas tout à fait aussi simple).

Exemple:

Lors du cheminement arrière qui suit l'échec du paragraphe 1.1., Tropic ne retire de l'acm que la partie issue du sommet-ET correspondant au problème

(DETERMINER-FORME-TOIT TT1) qui a été choisi pour point de retour, (cf. figure 6.3.). Simultanément, le littéral (FORME TT1 PRISMATIQUE) qui a été engendré par l'application de $rcm14: \{(\$TT TT1)\}$ est retiré de l'état courant, ce qui a pour effet de désactiver l'instance de contrainte $C2: \{(\$TT TT1) (\$F PRISMATIQUE)\}$ qui imposait la restriction numérique (HAUTEUR TR1 15). Dans cet exemple, Tropic conserve donc une partie importante du travail effectué entre l'instant où le problème (DETERMINER-FORME-TOIT TT1) a été décomposé et l'instant où l'échec a été constaté (partie en trait épais sur la figure 6.3.).

Tropic reprend alors le cheminement avant à partir du sommet-ET correspondant au problème (DETERMINER-FORME-TOIT TT1):

. Il applique $rcm13: \{(\$TT TT1)\}$ à ce problème, ce qui a pour effet d'ajouter le littéral (FORME TT1 PYRAMIDALE) à l'état courant. $C2: \{(\$TT TT1) (\$F PYRAMIDALE)\}$ devient alors active ; elle impose la restriction numérique (HAUTEUR TT1 25).

. Tropic constate ensuite que tous les autres problèmes formels sont déjà décomposés dans l'acm, et il n'effectue aucune modification. Les restrictions numériques actives étant alors compatibles, Tropic résout le problème original sans subir de nouvel échec.

1.2. Cas de plusieurs échecs consécutifs

L'objet de ce paragraphe est de décrire le comportement de Tropic face à plusieurs échecs consécutifs en mettant en évidence (§ 1.2.1.), puis en illustrant sur un exemple (§ 1.2.2.), le rôle des indicateurs d'échec.

1.2.1. Principe de l'utilisation des indicateurs d'échec

Un indicateur d'échec est une expression qui a la forme d'une expression causale. Il décrit les conditions dans lesquelles l'application d'une ircm a déjà conduit à un échec, et est associé au sommet-OU qui correspond à cette ircm. Il a trois fonctions dont nous décrivons le principe ci-dessous.

Supposons que lors du cheminement avant Tropic subisse un premier échec E_1 et que l'expression causale formée après cet échec le conduise à

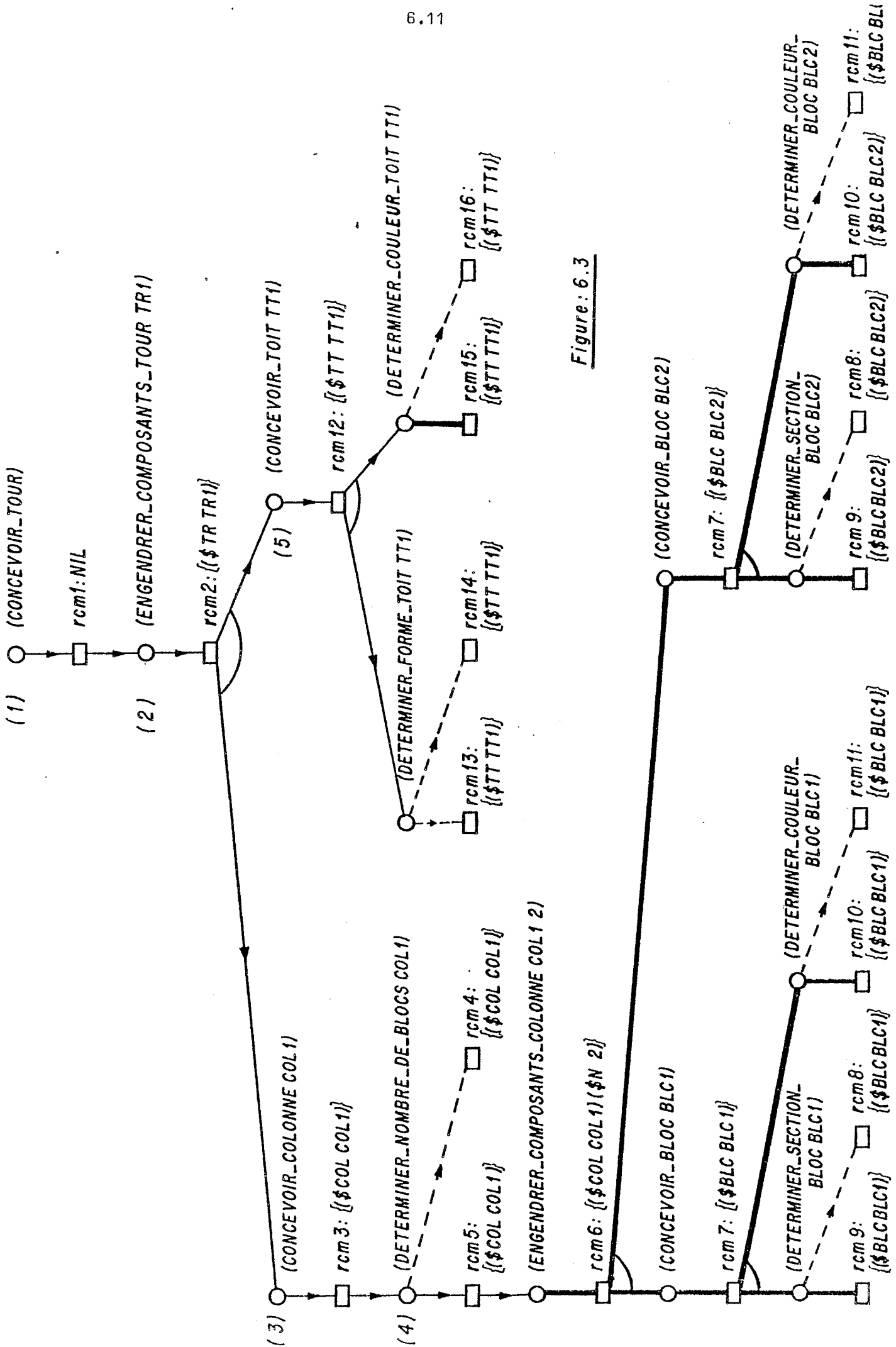


Figure: 6.3

effectuer un cheminement arrière au sommet-ET N_1 . Il associe alors cette expression comme indicateur d'échec au sommet-OU qui correspond à l'ircm i dont l'application au problème formel Pb_1 associé à N_1 est abandonnée. Il reprend ensuite le cheminement avant à partir du sommet N_1 en appliquant une ircm $i' \neq i$ au problème Pb_1 .

Supposons maintenant que Tropic subisse un deuxième échec E_2 qui le conduise à réexaminer le sommet N_1 , et que l'ircm i soit, d'après la nouvelle expression causale, la seule ircm à faire de N_1 un point de retour possible (i.e., l'application de i , en remplacement de i' , permettrait de rendre non satisfaite une condition contenue dans l'expression causale). Tropic ne choisit pas N_1 pour point de retour, car l'indicateur d'échec qui est associé au sommet-OU correspondant à i lui signale que l'application de i le conduirait à subir un échec déjà subi (E_1 dans ce cas). [La première fonction d'un indicateur d'échec est donc d'éviter à Tropic de choisir pour point de retour un sommet-ET à partir duquel le cheminement avant conduirait à un échec déjà subi.] Toutefois, avant d'examiner les sommets-ET antérieurs à N_1 , Tropic effectue les opérations suivantes:

- . L'expression causale courante (qui contient les conditions de l'échec E_2) est associée comme indicateur d'échec au sommet-OU correspondant à i' .
- . Le contenu de l'indicateur d'échec du sommet-OU correspondant à i est ajouté à l'expression causale courante.

La nouvelle expression causale regroupe ainsi les conditions de l'échec E_1 et celles de l'échec E_2 . Le point de retour qui sera éventuellement déterminé sera donc, soit un nouveau point de retour relatif à l'échec E_1 (dans ce cas, le cheminement arrière aura pour effet de "libérer" l'ircm i dont l'application au sommet N_1 peut permettre d'éviter E_2 , car l'échec E_1 sera ainsi corrigé différemment), soit un point de retour relatif à l'échec E_2 .

[La deuxième fonction d'un indicateur d'échec est donc de permettre à Tropic de reprendre en considération les conditions d'un échec déjà traité pour le corriger différemment.]

Supposons que Tropic détermine alors un point de retour N_2 . Il effectue un cheminement arrière à N_2 , puis reprend le cheminement avant à partir de ce sommet. Si le sommet N_1 n'a pas été retiré de l'acm lors du cheminement arrière, Tropic ne se contente pas alors de constater que le problème formel Pb_1 a déjà été décomposé. Il examine auparavant le contenu des indicateurs d'échec associés aux sommets-OU successeurs de N_1 . Si une condition de l'échec E_1 (resp. E_2) enregistrée dans l'indicateur d'échec du sommet-OU correspondant à l'ircm i (resp. i') n'est plus satisfaite, Tropic s'en aperçoit lors de cet examen, et il supprime l'indicateur d'échec. Ainsi, au moins un des deux indicateurs d'échec est supprimé. Deux cas sont alors possibles:

. L'indicateur d'échec du sommet-OU correspondant à i' a été supprimé (i.e., une condition de l'échec E_2 n'est plus satisfaite). Tropic conserve alors la décomposition du problème Pb_1 par l'ircm i' , et il passe au traitement du problème formel suivant.

. L'indicateur d'échec du sommet-OU correspondant à i' n'a pas été supprimé. Dans ce cas, les conditions de l'échec E_2 sont encore satisfaites et Tropic ne peut conserver la décomposition du problème Pb_1 par l'ircm i' , car cela le conduirait à subir à nouveau l'échec E_2 . Il efface donc les effets de l'application de i' sur l'état et l'acm courants, puis décompose Pb_1 en appliquant i (l'indicateur d'échec correspondant à i a alors été supprimé), ce qui rend non satisfaite une condition de l'échec E_2 . [La troisième fonction d'un indicateur d'échec est donc de guider le cheminement avant hors des traces d'un échec déjà subi.] Tropic passe ensuite au traitement du problème formel suivant.

Ce traitement s'appliquerait d'une façon similaire à une succession de plus de deux échecs.

Commentaires

L'exemple du paragraphe suivant nécessite de compléter le principe ci-dessus par les informations suivantes:

a/ Au paragraphe 3.2. du chapitre 4, nous avons décrit comment Tropic choisit l'ircm à appliquer à un problème formel. Nous avons vu que, I étant l'ensemble des ircm applicables à ce problème, Tropic restreint son choix au sous-ensemble I' de I qui contient les ircm dont l'application ne résulterait pas en la contradiction immédiate d'une restriction symbolique déjà active.

Tropic associe alors un indicateur d'échec e à chaque sommet-OU qui correspond à une ircm i de $I-I'$. Si $\{l_1, l_2, \dots, l_n\}$ est la partie gauche de l'instance de contrainte qui impose la restriction qui serait contredite après l'application de i , alors $e = \{c_1, c_2, \dots, c_n\}$, où, $\forall k \in [1, n]$, c_k représente la condition "le littéral l_k a la valeur VRAI dans l'état courant" [la valeur VRAI des littéraux l_1, l_2, \dots, l_n serait en effet responsable de l'activation de la contrainte, et donc de l'échec qui résulterait de l'application de i]. Cet indicateur d'échec joue un rôle identique aux autres indicateurs d'échec.

b/ Soient N un sommet-ET examiné lors de la détermination d'un point de retour, i l'ircm associée au sommet-OU successeur immédiat de N dans l'acm courante, et e l'expression causale courante lors de l'examen de N .

Pendant l'examen de N , Tropic simplifie l'expression causale courante de la façon suivante: il retire de e toutes les conditions dont la satisfaction est une conséquence directe de l'application de i . Par exemple, si c est la condition "le littéral l a la valeur VRAI dans l'état courant", si $c \in e$, et si la valeur VRAI de l dans l'état courant résulte de la présence dans cet état d'un littéral l' engendré par l'application de i , alors c est retiré de e .

c/ Soient N le point de retour choisi à la suite d'un échec et P_b le problème formel associé à N . Lors du cheminement arrière au sommet N , Tropic masque les effets (s'ils n'ont pas déjà été effacés) des ircm dont l'application est postérieure à la décomposition de P_b :

- . les littéraux qui ont été ajoutés à l'état courant par l'application de ces ircm ne sont plus alors considérés comme appartenant à cet état, bien qu'il n'en soit pas retirés effectivement,
- . les problèmes formels décomposés par ces ircm ne sont plus alors considérés comme décomposés, bien que leur décomposition ne soit pas effacée dans l'acm courante.

Ce masquage le conduit à désactiver les instances de contraintes, de conseils et de diagnostics qui contiennent dans leurs parties gauches des littéraux do

la valeur VRAI dans l'état courant avant l'échec résultait de la présence dans cet état littéraux maintenant masqués. Toutefois, nous verrons au paragraphe 2 du chapitre 7 que le travail qui précède leur activation n'est pas détruit, ce qui n'aurait pas été le cas si, au lieu d'être masqués, les littéraux avaient été retirés de l'état courant.

Pourquoi ce masquage est-il nécessaire? Nous avons vu (cf. 3ème fonction d'un indicateur d'échec) que certaines parties de l'acm, qui sont conservées lors d'un cheminement arrière, peuvent être effacées après la reprise du cheminement avant. Certains littéraux sont alors retirés de l'état courant. Si ces littéraux n'avaient pas été masqués, ils auraient pu être la cause de contradictions temporaires ; par exemple, avant d'être retirés de l'état courant, certains littéraux auraient pu rendre temporairement active une instance de contrainte appliquant une restriction contredite. Ces contradictions temporaires auraient remis en question la définition des échecs du chapitre 3 (§ 2).

1.2.2. Illustration sur un exemple

a/ Description du problème utilisé comme exemple.

L'énoncé de ce problème est formé par les rcm données en exemple au paragraphe 3.2. du chapitre 2 et par les contraintes C1, C2, C3, C4, C5, C6, C7 et C8 suivantes:

C1: ({{(TYPE \$TR TOUR)}} → {{(HAUTEUR \$TR 200)}})

C2: ({{(TYPE \$TT TOIT)}(FORME \$TT \$F)})

→ Fonct: si \$F = PRISMATIQUE alors I?{{(HAUTEUR \$TT 25)}}
sinon I?{{(HAUTEUR \$TT 35)}})

C3: ({{(TYPE \$BLC BLOC)}(SECTION \$BLC \$S)})

→ Fonct: si \$S = RONDE alors I?{{(PG HAUTEUR \$BLC 45)}(PP HAUTEUR \$BLC 85)}}
sinon I?{{(PG HAUTEUR \$BLC 55)}(PP HAUTEUR \$BLC 95)}})

C4: ({{(TYPE \$COL COLONNE)}(COMPOSANT \$COL \$BLC1)}(COMPOSANT \$COL \$BLC2)}(TYPE \$BLC1 BLOC)}(TYPE \$BLC2 BLOC)}(SECTION \$BLC1 \$S)})

→ {{(SECTION \$BLC2 \$S)}})

C5: ({{(TYPE \$BLC BLOC)}(SECTION \$BLC CARREE)}} → {{(COULEUR \$BLC VERTE)}})

C6: ({{(TYPE \$COL COLONNE)}(NOMBRE-DE-BLOCS \$COL 2)}(COMPOSANT \$COL \$BLC)}(TYPE \$BLC BLOC)})

→ {{(COULEUR \$BLC BLANCHE)}})

C7: $\{((\text{TYPE } \$\text{TR } \text{TOUR})(\text{COMPOSANT } \$\text{TR } \$\text{COL})(\text{COMPOSANT } \$\text{TR } \$\text{TT})(\text{TYPE } \$\text{COL } \text{COLONNE})$
 $(\text{TYPE } \$\text{TT } \text{TOIT}))\}$
 $\rightarrow \{((\text{HAUTEUR } \$\text{TR}) \leftarrow (+(\text{HAUTEUR } \$\text{COL})(\text{HAUTEUR } \$\text{TT})))\}$

C8: $\{(\$BLC)\{(\text{TYPE } \$\text{COL } \text{COLONNE})(\text{COMPOSANT } \$\text{COL } \$BLC)(\text{TYPE } \$BLC } \text{BLOC})\}$
 $\rightarrow \{((\text{HAUTEUR } \$\text{COL}) \leftarrow (*(\text{HAUTEUR } \$BLC)))\}$

b/ Le premier échec

Nous supposons que Tropic effectue le cheminement avant décrit par le tableau 6.1. L'acm engendré pendant ce cheminement avant est montrée aux figures 6.4. (sans les indicateurs d'échec) et 6.5. (avec les indicateurs d'échec):

. L'indicateur d'échec A est formé avec la partie gauche de C6: $\{(\$COL } \text{COL1})$
 $(\$BLC } \text{BLC1})\}$, car l'application de $\text{rcm11:}\{(\$BLC } \text{BLC1})\}$ aurait résulté en la contradiction immédiate de la restriction (COULEUR BLC1 BLANCHE) imposée par cette instance de contrainte (cf. commentaire a, § 1.2.1.).

[Note: Nous rappelons que dans les expressions causales et les indicateurs d'échec des exemples du paragraphe 1, un littéral l remplace la condition "le littéral l a la valeur VRAI dans l'état courant" (cf. § 1.1.2.)].

. L'indicateur d'échec B est formé avec la partie gauche de C4: $\{(\$COL } \text{COL1})$
 $(\$BLC1 } \text{BLC1})(\$BLC2 } \text{BLC2})(\$S } \text{RONDE})\}$, car l'application de $\text{rcm8:}\{(\$BLC } \text{BLC2})\}$ aurait résulté en la contradiction immédiate de la restriction (SECTION BLC2 RONDE) imposée par cette instance de contrainte.

. L'indicateur d'échec C est formé avec la partie gauche de C6: $\{(\$COL } \text{COL1})$
 $(\$BLC } \text{BLC2})\}$, car l'application de $\text{rcm11:}\{(\$BLC } \text{BLC2})\}$ aurait résulté en la contradiction immédiate de la restriction (COULEUR BLC2 BLANCHE) imposée par cette instance de contrainte.

Après avoir engendré le niveau de détail 2 de la description de TR1, Tropic constate l'incompatibilité des restrictions numériques actives (le grn constant est montré à la figure 6.6.). Il subit ainsi son premier échec E_1 .

c/ Le traitement de E_1

L'expression causale formée après E_1 est l'union des parties gauches de C1: $\{(\$TR } \text{TR1})\}$, C2: $\{(\$TT } \text{TT1})(\$F } \text{PRISMATIQUE})\}$, C7: $\{(\$TR } \text{TR1})(\$COL } \text{COL1})$
 $(\$TT } \text{TT1})\}$, C8: $\{(\$COL } \text{COL1})(\$BLC } \{BLC1 } \text{BLC2})\}$, C3: $\{(\$BLC } \text{BLC1})(\$S } \text{RONDE})\}$ et
 $C3: \{(\$BLC } \text{BLC2})(\$S } \text{RONDE})\}$, soit:

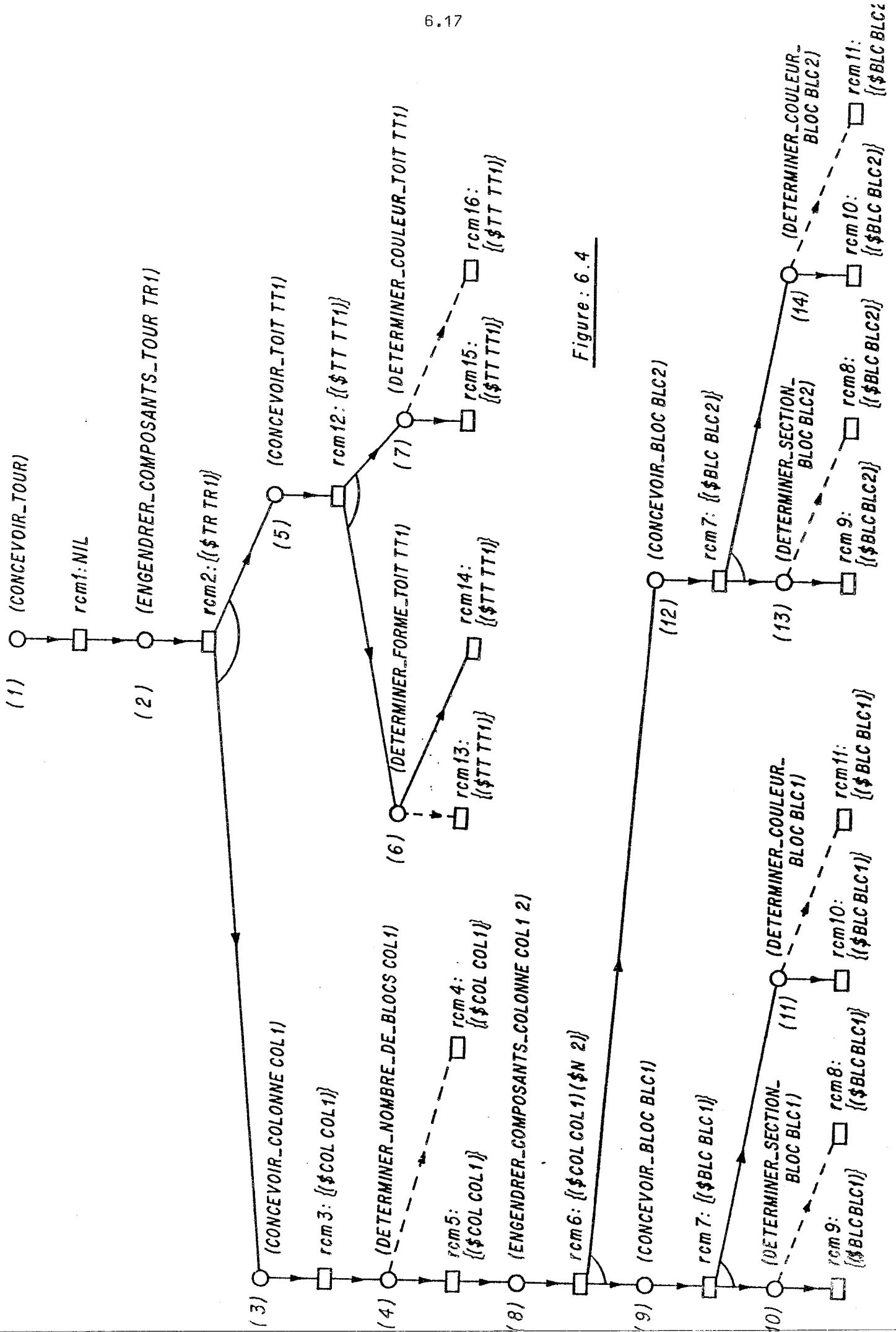


Figure : 6.4

A = {(TYPE COL1 COLONNE)(NOMBRE-DE-BLOCS COL1 2)
 (COMPOSANT COL1 BLC1)(TYPE BLC1 BLOC)}
 B = {(TYPE COL1 COLONNE)(COMPOSANT COL1 BLC1)(COMPOSANT CO
 BLC2)(TYPE BLC1 BLOC)(TYPE BLC2 BLOC)(SECTION BLC1 RO
 C = {(TYPE COL1 COLONNE)(NOMBRE-DE-BLOCS COL1 2)
 (COMPOSANT COL1 BLC2)(TYPE BLC2 BLOC)}

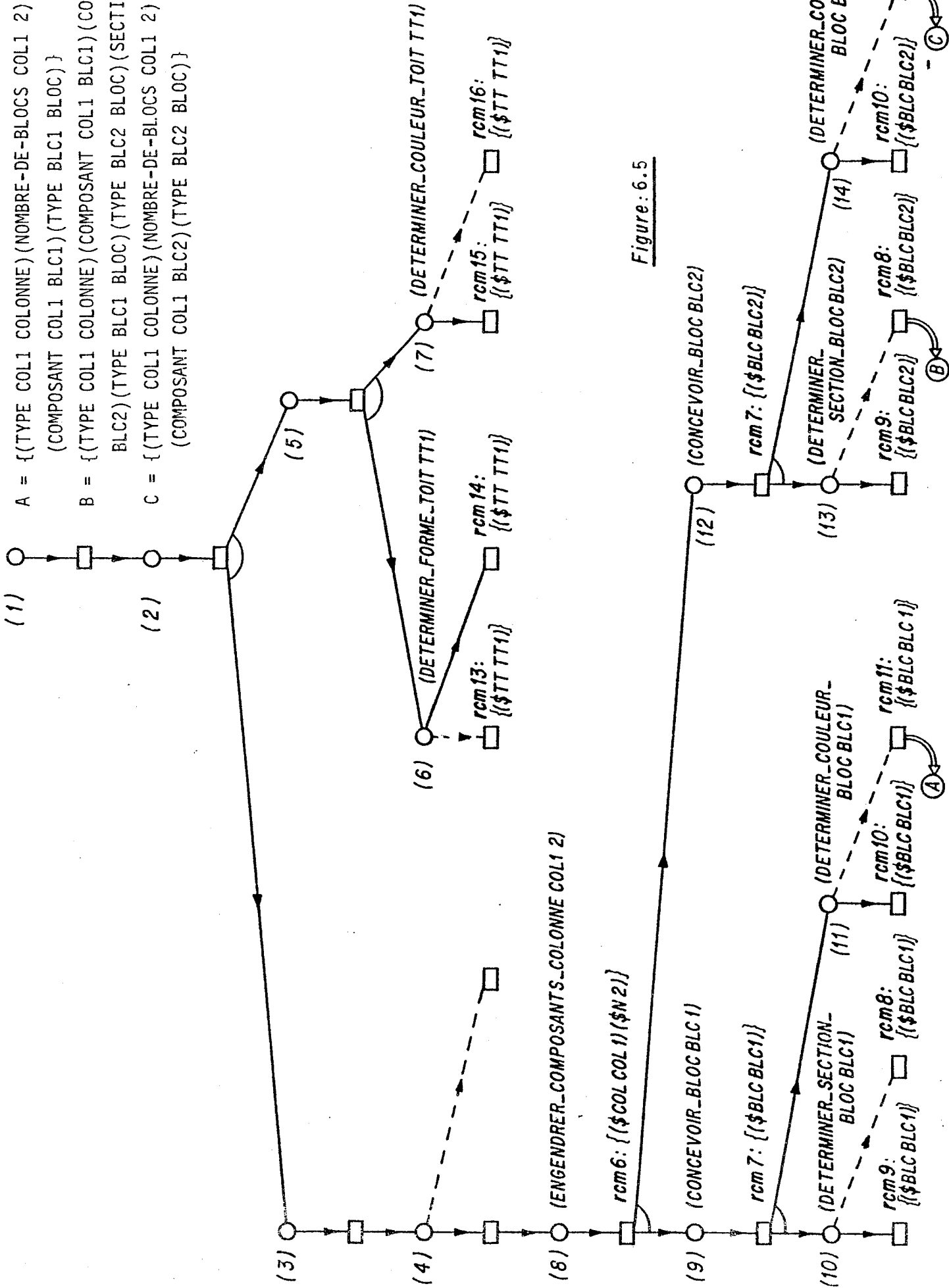


Figure: 6.5

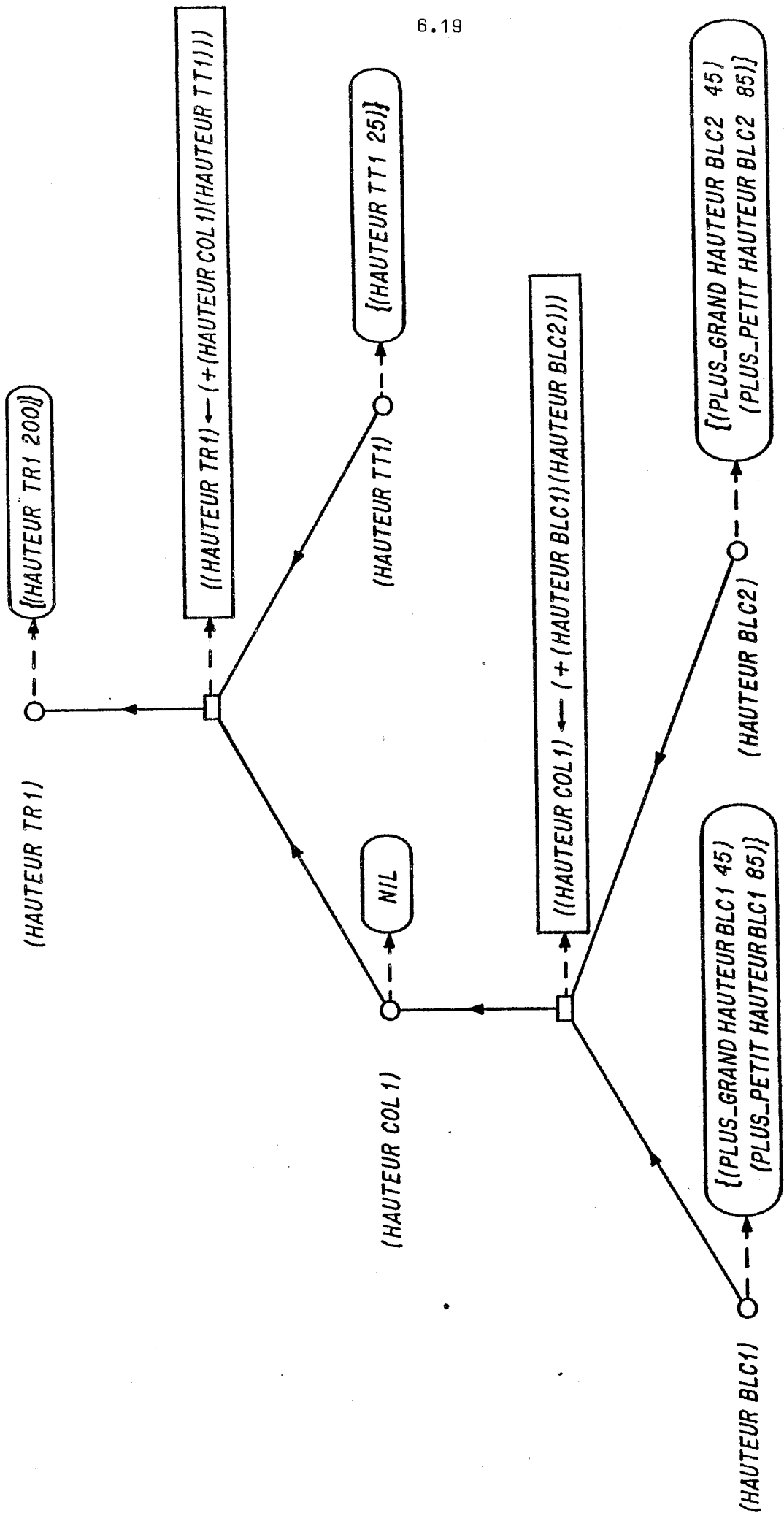


Figure: 6.6

{(TYPE TR1 TOUR)(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1)(TYPE COL1 COLONNE)
 (TYPE TT1 TOIT)(NOMBRE-DE-BLOCS COL1 2)(FORME TT1 PRISMATIQUE)(COMPOSANT
 COL1 BLC1)(COMPOSANT COL1 BLC2) (TYPE BLC1 BLOC)(TYPE BLC2 BLOC)
 (SECTION BLC1 RONDE)(SECTION BLC2 RONDE)}.

Tropic examine tour à tour les sommets-ET de l'acm (cf. figure 6.4 et 6.5) auxquels sont associés les problèmes formels suivants :

- . (DETERMINER-COULEUR-BLOC BLC2) : l'application de rcm11 : {(\$BLC BLC2)}, en remplacement de rcm10 : {(\$BLC BLC2)}, ne permettrait pas de modifier la valeur d'un littéral contenu dans l'expression causale ; Tropic ne choisit donc pas ce sommet pour point de retour.
- . (DETERMINER-SECTION-BLOC BLC2) : l'application de rcm8 : {(\$BLC BLC2)}, en remplacement de rcm9 : {(\$BLC BLC2)}, permettrait de modifier la valeur du littéral (SECTION BLC2 RONDE) contenu dans l'expression causale ; un indicateur d'échec étant associé au sommet-OU correspondant à rcm8 : {(\$BLC BLC2)}, Tropic ne choisit pas le sommet-ET examiné pour point de retour, et effectue les trois opérations suivantes :
 - il retire le littéral (SECTION BLC2 RONDE) de l'expression causale (cf. commentaire b, § 1.2.1) ;
 - il associe l'expression causale comme indicateur d'échec au sommet-OU correspondant à rcm9 : {(\$BLC BLC2)} (cf. l'indicateur d'échec D, figure 6.7) ;
 - il ajoute le contenu de l'indicateur d'échec B à l'expression causale (dans ce cas particulier, l'expression causale n'est pas modifiée, car elle contenait déjà tous les littéraux contenus dans B).
- . (CONCEVOIR-BLOC BLC2) : il n'y a pas d'alternative à ce sommet ; Tropic ne le choisit donc pas pour point de retour, et retire le littéral (TYPE BLC2 BLOC) de l'expression causale.
- . (DETERMINER-COULEUR-BLOC BLC1) : l'application de rcm11 : {(\$BLC BLC1)}, en remplacement de rcm10 : {(\$BLC BLC1)}, ne permettrait pas de modifier la valeur d'un littéral de l'expression causale ; Tropic ne choisit donc pas ce sommet pour point de retour.

D = {(TYPE TR1 TOUR)(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1)
 (TYPE COL1 COLONNE)(TYPE TT1 TOIT)(NOMBRE-DE-BLOCS COL
 (FORME TT1 PRISMATIQUE)(COMPOSANT COL1 BLC1)(COMPOSANT
 COL1 BLC2)(SUR BLC1 BLC2)(TYPE BLC1 BLOC)(TYPE BLC2 EL
 (SECTION BLC1 RONDE)}
 E = {(TYPE TR1 TOUR)(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1)
 (TYPE COL1 COLONNE)(TYPE TT1 TOIT)(NOMBRE-DE-BLOCS COL
 (FORME TT1 PRISMATIQUE)(COMPOSANT COL1 BLC1)
 (COMPOSANT COL1 BLC2)(SUR BLC1 BLC2)(TYPE BLC1 BLOC)}

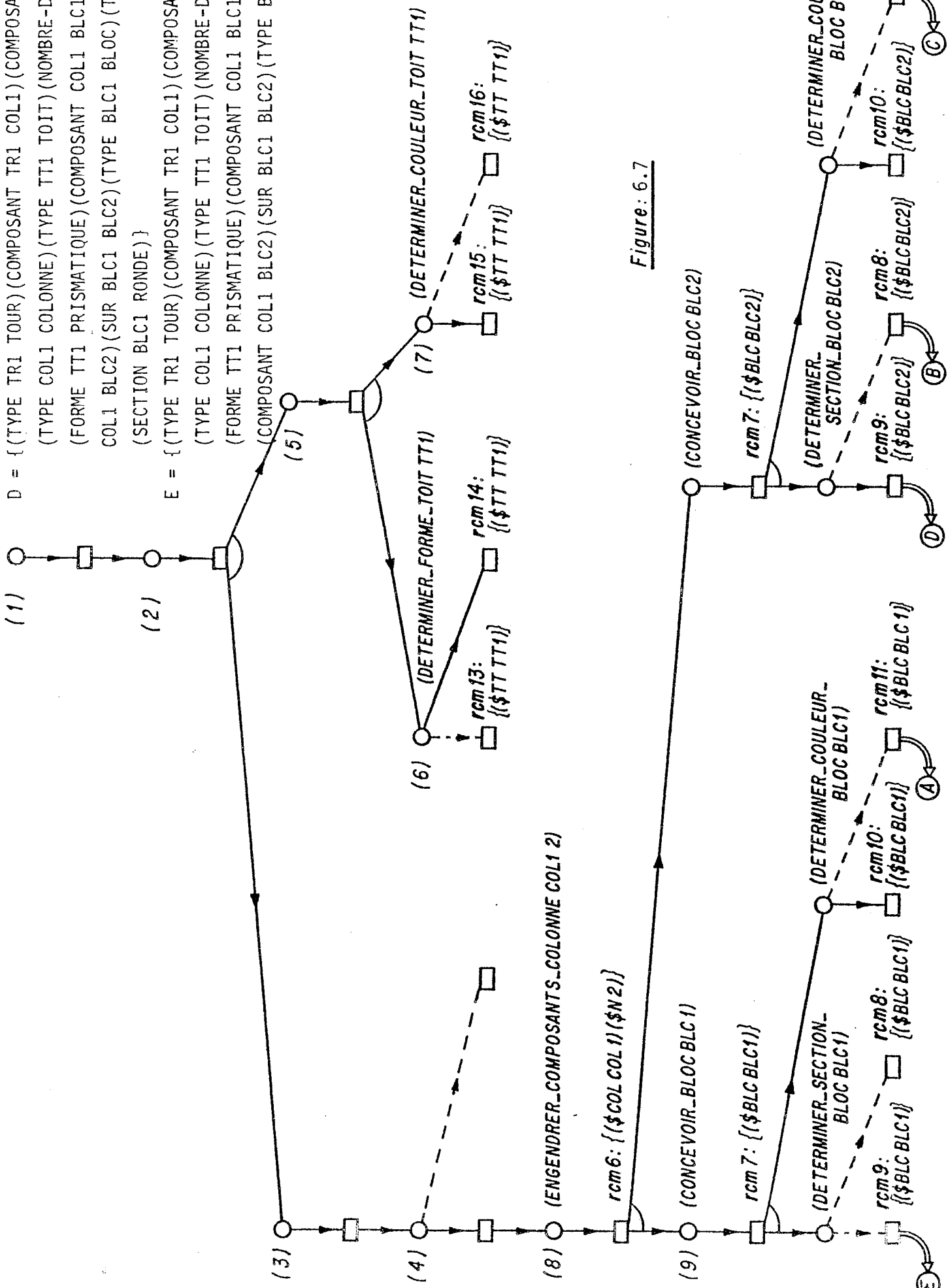


Figure: 6.7

- . (DETERMINER-SECTION-BLOC BLC1) : l'application de rcm8 : {(\$BLC BLC1)}, en remplacement de rcm9 : {(\$BLC BLC1)}, permet de modifier la valeur du littéral (SECTION BLC1 RONDE) contenu dans l'expression causale ; comme il n'y a pas d'indicateur d'échec au sommet-OU correspondant à rcm8 : {(\$BLC BLC1)}, Tropic choisit ce sommet-ET pour point de retour ; il retire le littéral (SECTION BLC1 RONDE) de l'expression causale, puis il associe celle-ci comme indicateur d'échec au sommet-OU correspondant à rcm9 : {(\$BLC BLC1)} (cf. l'indicateur d'échec E , figure 6.7).

Tropic effectue alors un cheminement arrière au sommet-ET auquel est associé le problème (DETERMINER-SECTION-BLOC BLC1) :

- . Il efface les effets de l'ircm rcm9 : {(\$BLC BLC1)}:
 - il retire le l'acm la partie qui est issue de ce sommet ; le sommet-OU correspondant à rcm9 : {(\$BLC BLC1)}, avec son indicateur d'échec E, est conservé, mais n'appartient plus à l'acm courante (cf. figure 6.7) ;
 - il retire de l'état courant le littéral (SECTION BLC1 RONDE), ce qui le conduit à désactiver C3 : {(\$BLC BLC1)(\$S RONDE)}, qui imposait deux des restrictions numériques incompatibles au moment de l'échec, et C4 : {(\$COL COL1)(\$BLC1 BLC1)(\$BLC2 BLC2)(\$S RONDE)}.
- . Il masque les effets des ircm rcm10 : {(\$BLC BLC1)}, rcm7 : {(\$BLC BLC2)}, rcm9 : {(\$BLC BLC2)} et rcm10 : {(\$BLC BLC2)} (cf. commentaire c, § 1.2.1) ; les littéraux (COULEUR BLC1 BLANCHE), (TYPE BLC2 BLOC), (SECTION BLC2 RONDE) et (COULEUR BLC2 BLANCHE) sont donc masqués, ce qui conduit Tropic à désactiver C6 : {(\$COL COL1)(\$BLC BLC2)}, C8 : {(\$COL COL1)(\$BLC{BLC1 BLC2})}, C3 : {(\$BLC BLC2)(\$S RONDE)} et C5 : {(\$COL COL1)(\$BLC1 BLC2)(\$BLC2 BLC1)(\$S RONDE)}.

[Note : Nous indiquons sur les figures que la décomposition d'un problème formel est masquée en retirant la numérotation du sommet-ET correspondant (cf. figure 6.7)].

d/ Le deuxième échec

Tropic reprend le cheminement avant à partir du sommet-ET correspondant au problème (DETERMINER-SECTION-BLOC BLC1). Il applique rcm8 :

$\{(\$BLC\ BLC1)\}$ à ce problème (cf. figure 6.8), ce qui a pour effet d'ajouter le littéral (SECTION BLC1 CARREE) à l'état courant. On note que C4 : $\{(\$COL\ COL1)(\$BLC1\ BLC1)(\$BLC2\ BLC2)(\$S\ ^\circ\text{CARREE})\}$ et C4 : $\{(\$COL\ COL1)(\$BLC1\ BLC2)(\$BLC2\ BLC1)(\$S\ \text{RONDE})\}$ ne sont pas actives (et donc pas contredites), car les littéraux (TYPE BLC2 BLOC) et (SECTION BLC2 RONDE) sont masqués.

L'ajout du littéral (SECTION BLC1 CARREE) à l'état courant rend active C3 : $\{(\$BLC\ BLC1)(\$S\ \text{CARREE})\}$ et C5 : $\{(\$BLC\ BLC1)\}$ qui imposent les restrictions numériques (PG HAUTEUR BLC1 55) et (PP HAUTEUR BLC1 95), et la restriction symbolique (COULEUR BLC1 VERTE). Cette dernière est contradictoire avec la restriction (COULEUR BLC1 BLANCHE) déjà imposée par C6 : $\{(\$COL\ COL1)(\$BLC\ BLC1)\}$. Tropic subit ainsi un deuxième échec E_2 (cf. § 2.4.4 b, chapitre 4).

e/ Le traitement de E_2

L'expression causale formée après E_2 est l'union des parties gauches de C5 : $\{(\$BLC\ BLC1)\}$ et de C6 : $\{(\$COL\ COL1)(\$BLC\ BLC1)\}$, soit : $\{(\text{TYPE COL1 COLONNE})(\text{NOMBRE-DE-BLOCS COL1 2})(\text{COMPOSANT COL1 BLC1})(\text{TYPE BLC1 BLOC})(\text{SECTION BLC1 CARREE})\}$.

Tropic passe en revue les sommets-ET de l'acm (cf. figure 6.8) dans l'ordre inverse de la décomposition des problèmes qui leur sont associés, en commençant par le sommet qui correspond au dernier problème décomposé avant E_2 . Il examine ainsi tour à tour les sommets-ET auxquels sont associés les problèmes suivants :

- . (DETERMINER-SECTION-BLOC BLC1) : l'application de rcm9 : $\{(\$BLC\ BLC1)\}$, en remplacement de rcm8 : $\{(\$BLC\ BLC1)\}$, permettrait de modifier la valeur du littéral (SECTION BLC1 CARREE) contenu dans l'expression causale ; un indicateur d'échec étant associé au sommet-OU correspondant à rcm9 : $\{(\$BLC\ BLC1)\}$, Tropic ne choisit pas le sommet-ET examiné pour point de retour, et effectue les trois opérations suivantes :
 - il retire le littéral (SECTION BLC1 CARREE) de l'expression causale,
 - il associe l'expression causale comme indicateur d'échec au sommet-OU correspondant à rcm8 : $\{(\$BLC\ BLC1)\}$ (cf. l'indicateur d'échec F, figure 6.9)

F = {(TYPE COL1 COLONNE)(NOMBRE-DE-BLOCS COL1 2)
 (COMPOSANT COL1 BLC1)(TYPE BLC1 BLOC)}
 G = {(TYPE TR1 TOUR)(COMPOSANT TR1 COL1)(COMPOSANT TR1 TT1)
 (TYPE COL1 COLONNE)(TYPE TT1 TOIT)(NOMBRE-DE-BLOCS COL1

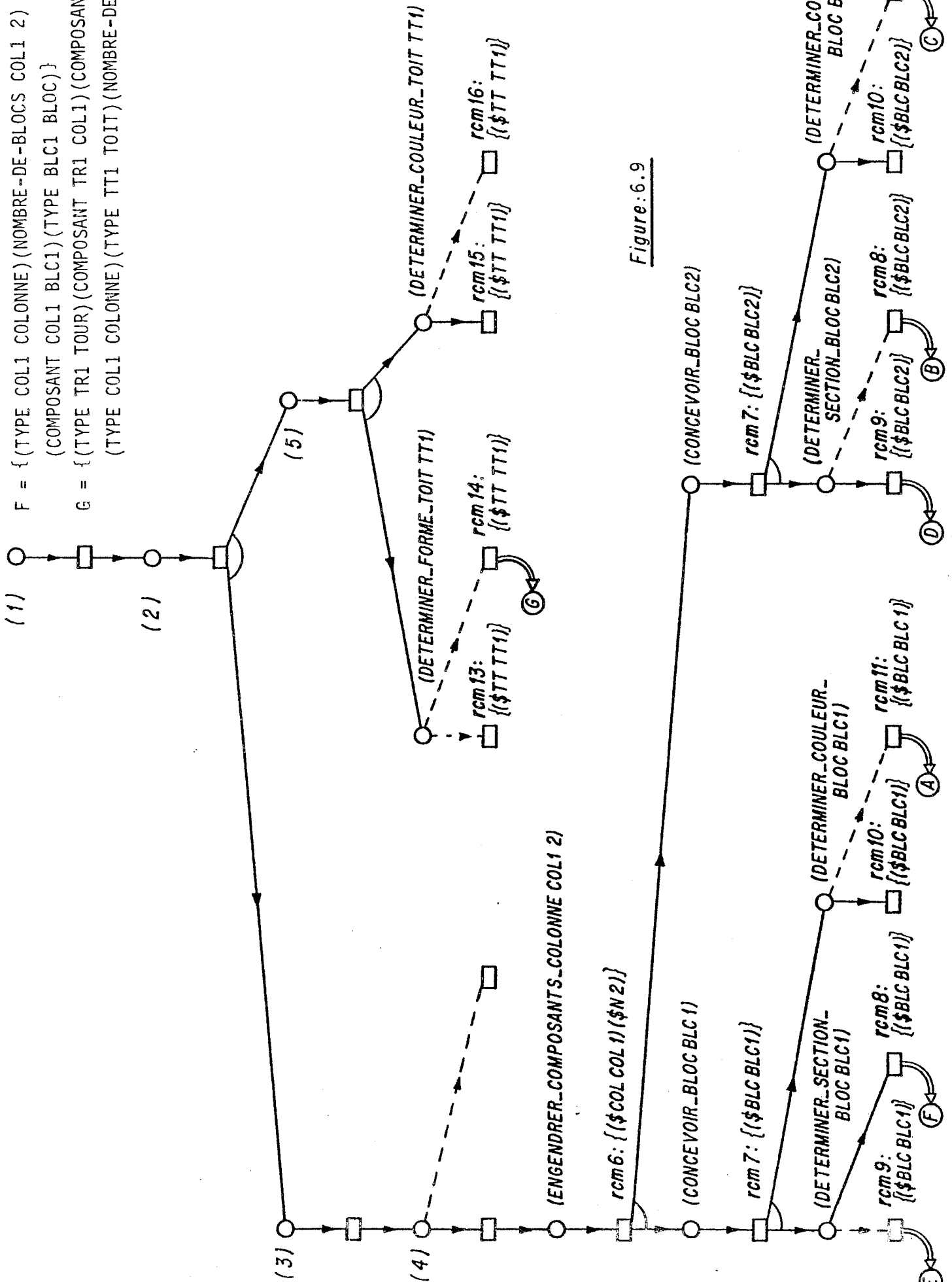


Figure: 6.9

- il ajoute le contenu de l'indicateur d'échec E à l'expression causale qui devient : $\{(TYPE\ TR1\ TOUR)(COMPOSANT\ TR1\ COL1)(COMPOSANT\ TR1\ TT1)\ (TYPE\ COL1\ COLONNE)(TYPE\ TT1\ TOIT)(NOMBRE-DE-BLOCS\ COL1\ 2)(FORME\ TT1\ PRISMATIQUE)(COMPOSANT\ COL1\ BLC1)(COMPOSANT\ COL1\ BLC2)(SUR\ BLC1\ BLC2)\ (TYPE\ BLC1\ BLOC)\}$.

[L'expression causale réunit maintenant les conditions de l'échec E_1 qui n'ont pas été utilisées au paragraphe 1.2.2 c, et celles de l'échec E_2].

- . (CONCEVOIR-BLOC BLC1), (ENGENDRER-COMPOSANTS-COLONNE COL1 2), (DETERMINER-COULEUR-TOIT TT1) : Tropic ne choisit aucun de ces sommets-ET pour point de retour ; il retire successivement les littéraux (TYPE BLC1 BLOC), (COMPOSANT COL1 BLC1), (COMPOSANT COL1 BLC2) et (SUR BLC1 BLC2) de l'expression causale.
- . (DETERMINER-FORME-TOIT TT1) : l'application de rcm13 : $\{(\$TT\ TT1)\}$, en remplacement de rcm14 : $\{(\$TT\ TT1)\}$, permet de modifier la valeur du littéral (FORME TT1 PRISMATIQUE) contenu dans l'expression causale ; comme il n'y a pas d'indicateur d'échec au sommet-OU correspondant à rcm13 : $\{(\$TT\ TT1)\}$, Tropic choisit ce sommet-ET pour point de retour ; il retire le littéral (FORME TT1 PRISMATIQUE) de l'expression causale, puis il associe celle-ci comme indicateur d'échec au sommet-OU correspondant à rcm14 : $\{(\$TT\ TT1)\}$ (cf. l'indicateur d'échec G , figure 6.9).

Tropic effectue alors un cheminement arrière au sommet-ET auquel est associé le problème (DETERMINER-FORME-TOIT TT1) :

- . Il efface les effets de l'ircm rcm14 : $\{(\$TT\ TT1)\}$:
 - il retire de l'acm la partie qui est issue de ce sommet (cf. figure 6.9),
 - il retire de l'état courant le littéral (FORME TT1 PRISMATIQUE), ce qui le conduit à désactiver C2 : $\{(\$TT\ TT1)\ (\$F\ PRISMATIQUE)\}$ qui imposait une des restrictions numériques incompatibles lors de l'échec E_1 .
- . Il masque les effets des ircm rcm15 : $\{(\$TT\ TT1)\}$, rcm6 : $\{(\$COL\ COL1)\ (\$N\ 2)\}$,

rcm7 : {(\$BLC BLC1)} et rcm8 : {(\$BLC BLC1)} ; les littéraux (COULEUR TT1 ROUGE), (TYPE BLC1 BLOC) et (SECTION BLC1 CARREE) sont donc masqués, ce qui conduit Tropic à désactiver C3 : {(\$BLC BLC1)(\$S CARREE)}, C5 : {(\$BLC BLC1)} et C6 : {(\$COL COL1)(\$BLC BLC1)}.

f/ L'obtention d'une solution

Tropic reprend le cheminement avant de partir du sommet-ET correspondant au problème (DETERMINER-FORME-TOIT TT1). Il traite successivement les problèmes suivants (les figures 6.9 et 6.10 montrent l'acm à la reprise et à la fin du cheminement avant) :

- . (DETERMINER-FORME-TOIT TT1) : rcm13 : {(\$TT TT1)} est appliquée à ce problème. Le littéral (FORME TT1 PYRAMIDALE) est ajouté à l'état courant.
- . (DETERMINER-COULEUR-TOIT TT1), (ENGENDRER-COMPOSANTS-COLONNE COL1 2), (CONCEVOIR-BLOC BLC1) : la décomposition de ces problèmes par rcm15 : {(\$TT TT1)}, rcm6 : {(\$COL COL1)(\$N 2)} et rcm7 : {(\$BLC BLC1)} est démasquée.
- . (DETERMINER-SECTION-BLOC BLC1) : Tropic supprime l'indicateur d'échec E, car le littéral (FORME TT1 PRISMATIQUE) a la valeur FAUX dans l'état courant, mais garde l'indicateur d'échec F. Il ne peut donc conserver la décomposition du problème (DETERMINER-SECTION-BLOC BLC1) par rcm8 : {(\$BLC BLC1)} (cf. la troisième fonction d'un indicateur d'échec, § 1.2.1), et il efface les effets de cette ircm. Il applique ensuite rcm9 : {(\$BLC BLC1)}, ce qui a pour effet d'ajouter le littéral (SECTION BLC1 RONDE) à l'état courant.
- . (DETERMINER-COULEUR-BLOC BLC1) : Tropic garde l'indicateur d'échec A. Il démasque la décomposition de (DETERMINER-COULEUR-BLOC BLC1) par rcm10 : {(\$BLC BLC1)}.
- . (CONCEVOIR-BLOC BLC2) : la décomposition de ce problème par rcm7 : {(\$BLC BLC1)} est démasquée.
- . (DETERMINER-SECTION-BLOC BLC2) : Tropic supprime l'indicateur d'échec D et garde l'indicateur B. Il démasque la décomposition de (DETERMINER-SECTION-BLOC BLC2) par rcm9 : {(\$BLC BLC2)}.
- . (DETERMINER-COULEUR-BLOC BLC2) : Tropic garde l'indicateur C. Il démasque la décomposition de (DETERMINER-COULEUR-BLOC BLC2) par rcm10 : {(\$BLC BLC2)}.

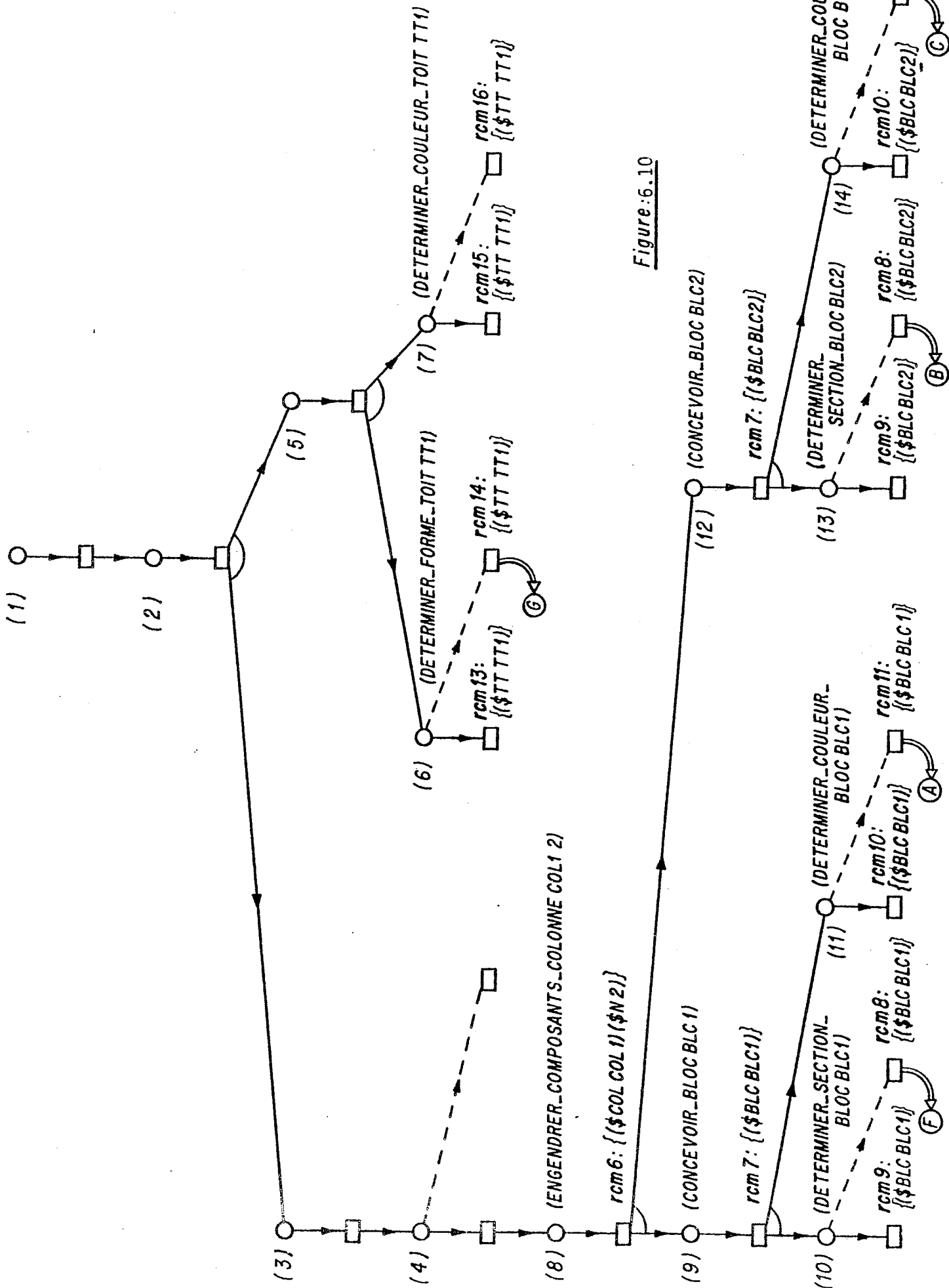


Figure: 6.10

Après avoir engendré les niveaux de détail 1 et 2 de la description de TR1, Tropic vérifie que les restrictions numériques actives sont compatibles. Les restrictions symboliques actives étant satisfaites, il résoud ainsi le problème original sans rencontrer de nouvel échec.

1.3. Les diagnostics et leur utilisation

1.3.1. Principe

Supposons que Tropic subisse un échec du à l'incompatibilité de restrictions numériques (cas des échecs des § 1.1.1 et 1.2.3). L'expression causale alors formée évite de choisir pour point de retour un sommet-ET tel que la reprise du cheminement avant à partir de ce sommet résulterait à nouveau en l'activation de ces mêmes restrictions. Mais, parce qu'elle ne décrit pas les interdépendances entre les conditions qu'elle contient et les restrictions numériques à satisfaire, elle peut conduire Tropic à choisir pour point de retour un sommet-ET à partir duquel il ne pourra faire autrement que subir un nouvel échec comparable au précédent. Par exemple, la présence de la condition "le littéral (FORME TT1 PRISMATIQUE) a la valeur VRAI dans l'état courant" dans une expression causale indique seulement qu'un changement de la forme de TT1 modifierait les conditions dans lesquelles l'échec a été subi ; elle n'indique pas si le choix d'un toit pyramidal résulterait en un accroissement ou en une réduction de la contradiction des restrictions numériques, ni à plus forte raison, si dans le deuxième cas le nouveau choix apporterait une modification suffisante pour éviter l'échec.

Les diagnostics sont des heuristiques spécialisées que le concepteur peut fournir à Tropic pour combler ces lacunes. Chaque diagnostics est une expression de la forme (Activ → Indic), où Indic est en général une fonction qui s'évalue à un ensemble de conditions. Après avoir constaté un échec dû à l'incompatibilité de restrictions numériques actives, Tropic utilise les diagnostics de la façon suivante :

- . si des instances de diagnostics s'appliquent à l'échec, alors Tropic forme une expression causale avec les conditions contenues dans les indications fournies par ces instances,

- . si aucune instance de diagnostic ne s'applique à l'échec, alors Tropic forme une expression causale comme nous l'avons déjà montré sur des exemples aux paragraphes 1.1.2 et 1.2.2.c.

1.3.2. Exemple

Soit le problème original donné en exemple au paragraphe 1.2.2.a, auquel nous ajoutons la contrainte C9 suivante [on notera que le problème n'a plus alors de solution] :

C9 : $\{(TYPE \$COL COLONNE)\} \rightarrow \{(NOMBRE-DE-BLOCS \$COL 1)\}$.

Cette contrainte impose de choisir une colonne COL1 composée d'un seul bloc BLC1. Ainsi, quelques soient les décisions prises concernant la forme du toit TT1 et la section du bloc BLC1, les restrictions numériques seront fortement incompatibles. En l'absence de diagnostic, Tropic subit une succession de plusieurs échecs, avant de constater l'incompatibilité de résoudre le problème original. En effet, les expressions formées après ces échecs le conduisent à effectuer des modifications (changement de la section de BLC1, puis de la forme de TT1) qui sont insuffisantes pour corriger l'échec.

Cette succession d'échecs est évitée si Tropic dispose du diagnostic suivant :

```
{(TYPE $TR TOUR)(COMPOSANT $TR $COL)(TYPE $COL COLONNE)(NOMBRE-DE-BLOCS
  $COL $N)}
→Fonct : si %ECHEC(HAUTEUR $TR) < -0.3 alors I? {(PG NOMBRE-DE-BLOCS $COL $N)}
          sinon NIL)
```

[Note : La fonction %ECHEC qui figure dans ce diagnostic sera définie au paragraphe 3.1.2 b. Lors des échecs qui nous intéressent ici, %ECHEC(HAUTEUR TR1) s'évalue à $\frac{\mu-h}{h}$, où :

- . h est la hauteur de la tour TR1 qui est imposée par C1 : $\{(\$TR TR1)\}$; donc, $h = 200$.
- . μ est la hauteur de TR1 la plus proche de h qui permet de satisfaire les autres restrictions numériques actives au moment de l'échec ; dans le meilleur des cas on a donc $\mu = 130$ (i.e. $95+35$).

On a donc ici : $\%ECHEC(HAUTEUR TR1) \leq -0.35$.]

Dès le premier échec, Tropic évalue la partie droite de l'instance par $\{(\$TR TR1)(\$COL COL1)(\$N 1)\}$ du diagnostic, car les conditions d'activation de cette instance sont satisfaites par l'état courant. Il utilise le résultat de cette évaluation, $\{(PG NOMBRE-DE-BLOCS COL1 1)\}$, pour former une expression causale contenant "le littéral (PG NOMBRE-DE-BLOCS COL1 1) a la valeur FAUX dans l'état courant "pour unique condition (cf. § 3.1.2 c). Cette expression causale lui impose de choisir pour point de retour un sommet-ET de l'acm à partir duquel il pourra augmenter le nombre de blocs de COL1. Mais un indicateur d'échec a précédemment été associé au sommet-OU qui correspond à $rcm5:\{(\$COL COL1)\}$, car lors du cheminement avant l'application de cette ircm aurait résulté en la contradiction immédiate de la restriction (NOMBRE-DE-BLOCS COL1 1) qui est imposée par $C9:\{(\$COL COL1)\}$, devenue active après l'application de $rcm3:\{(\$COL COL1)\}$. Cet indicateur d'échec empêche de choisir le sommet-ET correspondant au problème (DETERMINER-NOMBRE-DE-BLOCS COL1) pour point de retour. L'expression causale ne déterminant aucun autre point de retour, Tropic constate ainsi l'impossibilité de résoudre le problème original sans rencontrer de nouvel échec.

2. Description formelle simplifiée du traitement des échecs.

Dans ce paragraphe, nous décrivons le traitement des échecs en utilisant un formalisme simplifié. Notre principal objectif est de montrer que, sous certaines hypothèses (discutées au § 2.4), ce traitement est complet, c'est-à-dire qu'il produit une solution chaque fois qu'il en existe au moins une. Notre description procède par étapes successives : nous commençons par décrire un traitement simplifié exécuté dans une représentation de transformation d'état (§ 2.1), puis nous compliquons progressivement ce traitement (§ 2.2 et 2.3) dans la représentation utilisée par Tropic. Nous rappelons qu'une description détaillée des algorithmes exécutés par Tropic sera faite au paragraphe 3.

2.1. Première étape.

Le traitement qui est décrit dans cette première étape est une transposition, dans une représentation de transformation d'état, du traitement qui est exécuté par Tropic.

2.1.1. La recherche d'une solution

Le cheminement avant qui correspond, dans cette représentation, au cheminement avant effectué par Tropic (cf. chapitres 3 et 4) consiste à engendrer un chemin d'une arborescence A . A est définie en associant une possibilité, et une seule, de l'espace Π déterminé par les rcm (cf. chapitre 2, § 3.1) à chacun de ses sommets, et une ircm, et une seule, à chacun de ses arcs. Ainsi :

- . La possibilité $(NIL\{po\})$, où po est le problème initial, est associée à la racine de A .
- . N et N' étant deux sommets quelconques de A tels qu'il existe un arc a de A qui relie N à N' , les possibilités $(m\ d)$ et $(m'\ d')$ qui sont associées respectivement à N et à N' sont telles que $(m'\ d')$ dérive directement de $(m\ d)$ par l'application de l'ircm qui est associée à l'arc a .
- . Les ircm qui sont associées aux arcs issus d'un même sommet N de A décomposent un même problème formel $P_b \in d$, tel que la possibilité associée à N soit $(m\ d)$; on choisit P_b de la même façon que Tropic choisit le problème pendant à décomposer en appliquant la stratégie hiérarchique (§ 2.4.2, chapitre 4).

La recherche d'une solution qui correspond à l'algorithme du paragraphe 4 du chapitre 3 consiste à engendrer un chemin de A , dont l'extrémité initiale est la racine de A , jusqu'à ce qu'on ait produit une solution ou subi un échec. Dans le premier cas, le problème original est résolu ; dans le second cas, il est nécessaire de reprendre le cheminement avant à partir d'un point de retour (un sommet du chemin engendré, autre que l'extrémité terminale de ce chemin), suivant un autre chemin de A . Les ircm appliquées pendant le cheminement avant sont choisies à l'aide des heuristiques décrites au paragraphe 3.2 du chapitre 4.

Commentaires

a/ Le commentaire d du paragraphe 1 du chapitre 3 implique que tous les chemins de A ont une longueur finie. Le nombre d'arcs issus d'un sommet quelconque de A étant aussi fini, A est finie.

b/ $\forall M \in \mu$ (ensemble des modèles implicitement déterminé par les rom, cf. § 3.1, chapitre 2), il existe au moins un sommet pendant de A auquel est associée la possibilité (M NIL). En effet, les possibilités de la forme (M NIL) que l'on peut dériver de la possibilité (NIL {p₀}) ne dépendent pas de l'ordre de décomposition des problèmes formels (cf. commentaires b, § 3.1, chapitre 2). S'il existe une solution, on est donc certain qu'un parcours exhaustif de A permettra de la produire.

c/ Les heuristiques décrites au paragraphe 3.2 du chapitre 4 n'éliminent aucun chemin de A pouvant aboutir à une solution.

2.1.2. La détermination du point de retour

Nous décrivons ci-dessous l'algorithme qui correspond à celui qui est exécuté par Tropic pour déterminer le point de retour à la suite d'un échec E.

Soit Φ le chemin de A qui a conduit à l'échec E. Nous notons :

- N_1, N_2, \dots, N_T les sommets qui se succèdent sur Φ (N_1 est donc la racine de A),
- $a_{p,1}, a_{p,2}, \dots, a_{p,Q_p}, \forall p \in [1, T-1]$, les arcs qui sont issus de N_p dans A,
- $a_{p,q_p}, \forall p \in [1, T-1]$, l'arc de Φ qui relie N_p à N_{p+1} ($q_p \in [1, Q_p]$),
- $i_{p,q_p}, \forall p \in [1, T-1], \forall q_p \in [1, Q_p]$, l'ircm qui est associée à l'arc a_{p,q_p} ,
- $Pb_p, \forall p \in [1, T-1]$, le problème formel qui est décomposé par l'ircm i_{p,q_p} ,
- $E = \{c_1, c_2, \dots, c_n\}$ l'expression causale qui est formée à la suite de E.

L'hypothèse 1 énoncée au paragraphe 2.1.3 suivant étant supposée satisfaite (cette hypothèse peut être lue plus tard), on exécute les opérations suivantes pour $p=T-1, T-2, T-3, \dots$, jusqu'à ce qu'on ait déterminé un point de retour ou abandonné la résolution du problème original : [Note : les identificateurs e^p , $ALTERN^p$ et $ALTERN_1^p$ contenus dans cet algorithme pourraient être notés plus simplement e , $ALTERN$ et $ALTERN_1$; l'exposant p nous permettra par la suite de distinguer les valeurs associées à ces identificateurs à chaque itération].

1.
 - 1.1. $\text{ALTERN}^P \leftarrow \text{NIL}$.
 - 1.2. $\epsilon \leftarrow e^P$ [e^P désigne l'expression causale courante ; initialement, $e^{T-1} = E$].

2. Pour tout $c \in e^P$, tel que la satisfaction de c soit la conséquence directe de l'application de i_{p,q_p} , on effectue :
 - 2.1. $\epsilon \leftarrow \epsilon - \{c\}$,
 - 2.2. pour tout $q \in [1, Q_p]$, tel que :
 - $q \neq q_p$,
 - l'application de $i_{p,q}$ n'aurait pas la satisfaction de c pour conséquence directe, on effectue : $\text{ALTERN}^P \leftarrow \text{ALTERN}^P \cup \{a_{p,q}\}$.

3. Soit ALTERN_1^P l'ensemble des arcs de ALTERN^P tels qu'aucun indicateur d'échec ne leur soit associé [dans ce paragraphe, les indicateurs d'échecs sont associés à des arcs de A (ces arcs correspondent en effet aux sommets-OU de l'acm)].

Si $\text{ALTERN}_1^P \neq \text{NIL}$, alors :

- 3.1. ϵ est associé à a_{p,q_p} comme indicateur d'échec,
- 3.2. N_p est choisi pour point de retour,

sinon :

3.3. Si $p \neq 1$, alors :

- 3.3.1. pour tout $a \in \text{ALTERN}^P$, on effectue : $\epsilon \leftarrow \epsilon \cup e'$, où e' est l'indicateur d'échec associé à a ,
- 3.3.2. Si $\epsilon = e^P$, alors : $e^{p-1} \leftarrow e^P$, sinon : $e^{p-1} \leftarrow \epsilon \cup \{c\}$, où c est la condition "le problème P_{b_p} figure dans la décomposition courante du problème initial",

sinon :

3.3.4. on abandonne la résolution du problème original.

Si N_p ($p \in [1, T-1]$) est choisi pour point de retour (on a alors : $ALTERN_1^D \neq NIL$), le cheminement avant est repris à partir de ce sommet en appliquant à Pb_p une ircm $i_{p,q}$ telle que $a_{p,q} \in ALTERN_1^D$.

2.1.3. Complétude du traitement

Pour montrer que le traitement des paragraphes 2.1.1 et 2.1.2 produit une solution chaque fois qu'il en existe au moins une, il nous suffit de démontrer les trois propositions suivantes :

- (1) Si l'algorithme du paragraphe 2.1.2 choisit le sommet N_p ($p \in [1, T-1]$) pour point de retour, alors, $\forall p' \in [p+1, T-1]$, aucun chemin de A passant par $N_{p'}$, n'aboutit à une solution.
- (2) Si l'algorithme du paragraphe 2.1.2 ne détermine aucun point de retour, alors le problème original n'a pas de solution.
- (3) Si l'algorithme du paragraphe 2.1.2 choisit le sommet N_p ($p \in [1, T-1]$) pour point de retour, alors aucun des arcs de $ALTERN_1^D$ n'a déjà été engendré.

En effet, la vérification de ces trois propositions et le commentaire c du paragraphe 2.1.1 assurent que l'algorithme de recherche du paragraphe 2.1.1 effectue un parcours systématique de tous les chemins de A dont on n'est pas certain qu'ils n'aboutissent pas à une solution [la proposition (3) assure qu'on ne parcourt pas deux fois le même chemin]. A étant finie (cf. commentaire a, § 2.1.1), on est alors certain de produire une solution chaque fois qu'il en existe au moins une (cf. commentaire b, § 2.1.1).

Démonstration des propositions (1) et (2)

Pour démontrer les propositions (1) et (2), nous supposons que les trois hypothèses suivantes sont satisfaites (nous discuterons ces hypothèses au § 2.4) :

- . Hypothèse 1 : Soit c une condition quelconque figurant dans une expression causale ou un indicateur d'échec ; si, à un instant quelconque du cheminement

avant, c est satisfaite, alors la satisfaction de c est la conséquence directe de l'application d'une ircm i associée à un arc a de ϕ (ϕ désigne le chemin de A qui a été engendré), et d'une seule, et il est possible de déterminer i en examinant uniquement les effets immédiats (les littéraux et les problèmes formels engendrés) de l'application des ircm associées aux arcs de ϕ .

[Note : La satisfaction de cette hypothèse est essentielle pour exécuter l'algorithme du paragraphe 2.1.2 et pour énoncer les hypothèses suivantes].

- . Hypothèse 2 : L'expression causale $e = \{c_1 \ c_2 \ \dots \ c_n\}$ formée à la suite d'un échec quelconque est telle que :
 - $\forall c \in e, \exists a \in \phi$ (ϕ désigne le chemin de A qui a conduit à l'échec) tel que l'application de l'ircm i associée à l'arc a a la satisfaction de c pour conséquence directe,
 - la conjonction logique $c_1 \wedge c_2 \wedge \dots \wedge c_n$ est une condition suffisante d'échec i.e. la satisfaction simultanée des conditions c_1, c_2, \dots, c_n implique la rencontre d'un échec indépendamment du chemin de A engendré.
- . Hypothèse 3 : Soient $e = \{c_1 \ c_2 \ \dots \ c_n\}$ un indicateur d'échec quelconque associé à un arc a de A , i l'ircm associée à a , et ϕ le chemin ayant la racine de A pour extrémité initiale et l'origine de a pour extrémité finale :
 - $\forall c \in e, \exists a' \in \phi$ tel que l'application de l'ircm i' associée à l'arc a' ait la satisfaction de c' pour conséquence directe,
 - c représentant la condition "l'ircm i est appliquée à un problème formel qui figure dans la décomposition courante du problème initial", la conjonction $c_1 \wedge c_2 \wedge \dots \wedge c_n \wedge c$ est une condition suffisante d'échec.

Notre démonstration comporte les deux parties a et b suivantes :

a/ Première partie

Les opérations 1, 2 et 3 énoncées au paragraphe 2.1.2 sont exécutées itérativement avec p successivement égal à $T-1, T-2, T-3, \dots$, jusqu'à ce qu'on ait déterminé un point de retour ou abandonné la résolution du problème original. Soit $e^p = \{c_1^p \ c_2^p \ \dots \ c_n^p\}$ l'expression causale courante lors de l'opération 1 à chaque itération. ^p Le sommet N_p est choisi pour point de

retour (opération 3.2) si et seulement si $\text{ALTERN}_1^p \neq \text{NIL}$, c'est-à-dire s'il existe au moins un arc $a_{p,q}$ de A ($q \in [1, Q_p]$, $q \neq q_p$) qui satisfait les deux conditions suivantes :

- (1') $\exists c \in e^p$ tel que :
- . l'application de i_{p,q_p} a la satisfaction de c pour conséquence directe,
 - . l'application de $i_{p,q}$ n'aurait pas la satisfaction de c pour conséquence directe,
- (2') aucun indicateur d'échec n'est associé à $a_{p,q}$.

Dans cette première partie, nous supposons que e^p vérifie les deux propositions suivantes (ce que nous montrerons dans la deuxième partie) :

(1'') $\forall c \in e^p, \exists p' \in [1, p]$ tel que l'application de $i_{p',q_{p'}}$ a la satisfaction de c pour conséquence directe.

(2'') $c_1^p \wedge c_2^p \wedge \dots \wedge c_{n_p}^p$ est une condition suffisante d'échec.

Soit $\phi_{p,q}$ un chemin quelconque de A tel que l'arc $a_{p,q} \in \phi_{p,q}$. On déduit des propositions (1'') et (2''), et de l'hypothèse 1 que, si N_p n'est pas choisi pour point de retour, alors $\phi_{p,q}$ ne peut pas aboutir à une solution. En effet, $\phi_{p,q}$ est alors tel que l'arc $a_{p,q}$ contredit la condition (1') et/ou la condition (2') :

- . Si $a_{p,q}$ contredit (1') [ce cas inclus le cas où $q = q_p$], alors on est certain (cf. proposition (1'')) que les conditions $c_1^p, c_2^p, \dots, c_{n_p}^p$ seront toutes satisfaites après l'application de $i_{p,q}$. $c_1^p \wedge c_2^p \wedge \dots \wedge c_{n_p}^p$ étant une condition suffisante d'échec (cf. proposition (2'')), on est certain de subir un échec en suivant $\phi_{p,q}$.
- . Si $a_{p,q}$ contredit (2'), alors soit $\{c_1', c_2', \dots, c_n'\}$ l'indicateur d'échec qui est associé à $a_{p,q}$. Les conditions c_1', c_2', \dots, c_n' sont toutes satisfaites (cf. hypothèse 3). $c_1' \wedge c_2' \wedge \dots \wedge c_n'$, $\wedge c$, où c représente la condition "l'ircm $i_{p,q}$ est appliquée à un problème formel (P_b) figurant dans la décomposition courante du problème initial", étant une condition suffisante d'échec (cf. hypothèse 3), on est certain de subir un échec en suivant $\phi_{p,q}$.

La vérification des propositions (1) et (2) est alors évidente.

b/ Deuxième partie

Dans cette deuxième partie, nous montrons que e^P vérifie les propositions (1'') et (2''). Pour ce faire, nous raisonnons par induction :

- e^{T-1} vérifie (1'') et (2''). Cela découle directement de l'hypothèse 2. En effet, e^{T-1} est l'expression causale formée à la suite de l'échec E ($e^{T-1} = E$).

- Si e^{P+1} , $p \in [T-2, 1]$, vérifie (1'') et (2'') et si N_{p+1} n'est pas choisi pour point de retour (on a alors : $ALTERN_1^{P+1} = NIL$), alors e^P vérifie (1'') et (2''). En effet :

. e^P est obtenu en effectuant trois transformations sur e^{P+1} :

- L'opération 2.1 retire de e^{P+1} toutes les conditions c telles que c soit une conséquence directe de l'application de $i_{p+1, q+1}$. Soit e_1^{P+1} l'expression obtenue ; e^{P+1} vérifiant (1''), on a : $\forall c \in e_1^{P+1}, \exists p' \in [1, p]$ tel que l'application de $i_{p', q_{p'}}$ a la satisfaction de c pour conséquence directe.
- L'opération 3.3.1 fait l'union de e_1^{P+1} avec les indicateurs d'échec associés à des arcs issus de N_{p+1} . Soit e_2^{P+1} l'expression obtenue ; par suite de l'hypothèse 3, on a : $\forall c \in e_2^{P+1}, \exists p' \in [1, p]$ tel que l'application de $i_{p', q_{p'}}$ a la satisfaction de c pour conséquence directe.
- L'opération 3.3.2 ajoute éventuellement la condition "le problème formel Pb_{p+1} figure dans la décomposition courante du problème initial" à e_2^{P+1} , et on obtient e^P . Cette condition étant une conséquence directe de l'application de i_{p, q_p} , e^P vérifie donc (1'').
- La conjonction des conditions contenues dans e_1^{P+1} et de la condition "une ircm $i_{p+1, q}$ telle que $a_{p+1, q} \notin ALTERN^{P+1}$, est appliquée à Pb_{p+1} " est une condition suffisante d'échec. [Nous déduisons cette proposition de ce que la conjonction des conditions contenues dans e^{P+1} est une condition suffisante d'échec, en remarquant que $\forall a_{p+1, q} \notin ALTERN^{P+1}$,

- on a : $\forall c \in e_1^{p+1} - e_1^{p+1}$, l'application de $i_{p+1,q}$ a la satisfaction de c pour conséquence directe (cf. la définition de $ALTERN^{p+1}$ dans l'opération 2).
- $\forall a_{p+1,q} \in ALTERN^{p+1}$, la conjonction des conditions contenues dans l'indicateur d'échec associé à $a_{p+1,q}$ et de la condition "l'ircm $i_{p+1,q}$ est appliquée à un problème (Pb_{p+1}) figurant dans la décomposition courante du problème initial", est une condition suffisante d'échec (hypothèse 3).
 - La présence de Pb_{p+1} dans la décomposition courante du problème initial implique (sauf si un cheminement arrière retire Pb_{p+1} de cette décomposition) qu'une ircm $i_{p+1,q}$ ($q \in [1, Q_{p+1}]$) sera appliquée à Pb_{p+1} . Des deux conditions suffisantes et de l'implication ci-dessus nous déduisons que la conjonction des conditions contenues dans e_1^{p+1} , des conditions contenues dans les indicateurs d'échec associés aux arcs $a_{p+1,q}$ tels que $a_{p+1,q} \in ALTERN^{p+1}$, et de la condition " Pb_{p+1} figure dans la décomposition courante du problème initial" est une condition suffisante d'échec. Les conditions qui composent cette conjonction étant aussi celles qui composent e^p , e^p vérifie (2").

Démonstration de la proposition (3)

Supposons que l'algorithme du paragraphe 2.1.2 choisisse N_p ($p \in [1, T-1]$) pour point de retour. Supposons de plus que le chemin Φ qui a la racine de A pour extrémité initiale et N_p pour extrémité terminale soit tel que, $\forall a \in \Phi$, il n'y a pas d'indicateur d'échec associé à a . Si $ALTERN_1^p$ contenait un arc $a_{p,q}$ déjà engendré, alors, $a_{p,q}$ étant différent de a_{p,q_p} (de par la formation de $ALTERN^p$), un cheminement arrière à un sommet N_p , tel que $p' \in [1, p]$ aurait nécessairement été effectué entre l'instant où $a_{p,q}$ a été engendré et celui où l'échec E a été constaté.

- . Si $p' = p$, alors un indicateur d'échec aurait été associé à $a_{p,q}$ (cf. opération 3.1), ce qui serait en contradiction avec l'appartenance de $a_{p,q}$ à $ALTERN_1^p$,
- . Si $p' \in [1, p-1]$, alors un indicateur d'échec aurait été associé à $a_{p',q_{p'}}$, ce qui serait en contradiction avec l'hypothèse du raisonnement concernant Φ .

Donc, si N_p est choisi pour point de retour et si, $\forall a \in \Phi$, il n'y a pas d'indicateur d'échec associé à a , alors aucun arc de $ALTERN_1^p$ n'a déjà été engendré. De plus, le chemin Φ' qui est engendré après le cheminement arrière à N_p est aussi tel que, $\forall a \in \Phi'$, il n'y a pas d'indicateur d'échec associé à a . En effet, le cheminement avant n'associe aucun indicateur d'échec aux arcs qu'il engendre. Comme, pour cette même raison, le chemin Φ_0 engendré avant la rencontre du premier échec est tel que, $\forall a \in \Phi_0$, il n'y a pas d'indicateur d'échec associé à a , la proposition (3) est démontrée.

Remarque

Si N_p est choisi pour point de retour, un indicateur d'échec est associé à l'arc a_{p,q_p} . De la partie b de la démonstration des propositions (1) et (2), on déduit facilement que cet indicateur d'échec satisfait l'hypothèse 3.

2.2. Deuxième étape

Le traitement qui est décrit dans cette deuxième étape est, dans un formalisme simplifié, celui qui est exécuté par Tropic, à la différence près suivante (cf. § 3, chapitre 3) : le cheminement arrière à un sommet-ET N consiste à effacer les effets de l'ircm i qui a été appliquée au problème formel associé à N et les effets de toutes les ircm dont l'application a été postérieure à celle de i .

Nous décrivons ci-dessous l'algorithme qui est exécuté pour déterminer le point de retour à la suite d'un échec \bar{E} . α désignant l'acm courante à l'instant où l'échec \bar{E} est constaté, nous notons :

- $Pb_1, Pb_2, \dots, Pb_{T-1}$ les problèmes formels qui sont associés aux sommets-ET non pendants de α (les indices sont choisis de telle sorte que, $\forall p$ et $q \in [1, T-1]$, si $p < q$ alors la décomposition de Pb_p est antérieure à celle de Pb_q),
- $N_p, \forall p \in [1, T-1]$, le sommet-ET de α auquel est associé Pb_p ,
- $N'_{p,1}, N'_{p,2}, \dots, N'_{p,Q_p}, \forall p \in [1, T-1]$, les sommets-OU successeurs immédiats de N_p [Note : un seul de ces sommets appartient à α (cf. commentaires b, § 1, chapitre 3)],

- . N'_{p,q_p} , $\forall p \in [1, T-1]$, le sommet-OU successeur immédiat de N_p dans α
($q_p \in [1, Q_p]$),
- . $i_{p,q}$, $\forall p \in [1, T-1]$, $\forall q \in [1, Q_p]$, l'iron qui est associée à $N'_{p,q}$,
- . $E = \{c_1 c_2 \dots c_n\}$ l'expression causale qui est formée à la suite de E .

L'hypothèse 1' énoncée plus loin dans ce paragraphe étant supposée satisfaite, on exécute les opérations suivantes pour $p = T-1, T-2, T-3, \dots$, jusqu'à ce qu'on ait déterminé un point de retour ou abandonné la résolution du problème original :

1. 1.1. $ALTERN^p \leftarrow NIL$.
- 1.2. $\epsilon \leftarrow e^p [e^{T-1} = E]$.

2. Pour tout $c \in e^p$, tel que la satisfaction de c soit la conséquence directe de l'application de i_{p,q_p} , on effectue :
 - 2.1. $\epsilon \leftarrow \epsilon - \{c\}$,
 - 2.2. pour tout $q \in [1, Q_p]$, tel que :
 - . $q \neq q_p$,
 - . l'application de $i_{p,q}$ n'aurait pas la satisfaction de c pour conséquence directe, on effectue : $ALTERN^p \leftarrow ALTERN^p \cup \{N'_{p,q}\}$.

3. Soit $ALTERN^p_1$ l'ensemble des sommets de $ALTERN^p$ tels qu'aucun indicateur d'échec ne leur soit associé.

Si $ALTERN^p_1 \neq NIL$, alors :

- 3.1. ϵ est associé à N'_{p,q_p} comme indicateur d'échec,
- 3.2. N_p est choisi pour point de retour,

sinon :

- 3.3. si $p \neq 1$, alors :

- 3.3.1. pour tout $N' \in ALTERN^p$, on effectue : $\epsilon \leftarrow \epsilon \cup e'$, où e' est l'indicateur d'échec associé à N' ,

3.3.2. si $\epsilon = e^p$, alors : $e^{p-1} \leftarrow e^p$, sinon : $e^{p-1} \leftarrow \epsilon u\{c\}$, où c est la condition "le problème Pb_p est associé à un sommet-ET de l'acm courant",

sinon :

3.3.4. on abandonne la résolution du problème original.

Si N_p ($p \in [1, T-1]$) est choisi pour point de retour (on a alors : $ALTERN_1^p \neq NIL$), le cheminement avant est repris à partir de ce sommet en appliquant à Pb_p une ircm $i_{p,q}$ telle que $N'_{p,q} \in ALTERN_1^p$.

Bien qu'utilisant des représentations différentes pour mémoriser leur décision, les traitements décrits dans ce paragraphe et dans le paragraphe 2.1 explorent l'espace Π déterminé par les rcm en engendrant les mêmes possibilités dans le même ordre. Une démonstration semblable à celle du paragraphe 2.1.3 permettrait donc de montrer que le traitement ci-dessus produit une solution chaque fois qu'il en existe au moins une, si les hypothèses 1', 2' et 3' suivantes sont satisfaites (ce sont les hypothèses 1, 2 et 3 reformulées dans la représentation utilisée par ce traitement) :

- . Hypothèse 1' : Soit c une condition quelconque figurant dans une expression causale ou dans un indicateur d'échec ; si, à un instant quelconque du cheminement avant, c est satisfaite, alors la satisfaction de c est la conséquence directe de l'application d'une ircm i associée à un sommet-OU de l'acm courante α , et d'une seule, et il est possible de déterminer i en examinant uniquement les effets immédiats (les littéraux et les problèmes formels engendrés) de l'application des ircm associées aux sommets-OU de α .
- . Hypothèse 2' : L'expression causale $e = \{c_1 c_2 \dots c_n\}$ formée à la suite d'un échec quelconque est telle que :
 - $\forall c \in e$, il existe un sommet-OU N' de l'acm courante à l'instant de l'échec tel que l'application de l'ircm i associée à N' a la satisfaction de c pour conséquence directe,

- la conjonction logique $c_1 \wedge c_2 \wedge \dots \wedge c_n$ est une condition suffisante d'échec, i.e. la satisfaction simultanée des conditions c_1, c_2, \dots, c_n implique la rencontre d'un échec quelque soit l'acm courante.
- . Hypothèse 3' : Soient $e = \{c_1, c_2, \dots, c_n\}$ un indicateur d'échec quelconque associé à un sommet-OU N'_1 d'une acm α et i_1 l'ircm associée à N'_1 :
 - $\forall c \in e$, il existe un sommet-OU N'_2 de α tel que l'application de l'ircm i_2 associée à N'_2 ait la satisfaction de c pour conséquence directe, et tel que l'application de i_2 a précédé celle de i_1 lors de la génération de α ,
 - c représentant la condition "l'ircm i_1 est associée à un sommet-OU de l'acm courante", la conjonction $c_1 \wedge c_2 \wedge \dots \wedge c_n \wedge c$ est une condition suffisante d'échec.

2.3. Troisième étape

Le traitement qui est décrit dans cette troisième étape est, dans un formalisme simplifié, celui qui est exécuté par Tropic. Nous supposons que les hypothèses 1', 2' et 3' sont satisfaites.

A la suite d'un échec E , le point de retour N_p est déterminé en exécutant l'algorithme énoncé au paragraphe 2.2, auquel on ajoute l'opération 3.3.0 suivante (cette opération précède l'opération 3.3.1) :

3.3.0. Si $ALTERN^P \neq NIL$, alors ϵ est associé, comme indicateur d'échec, à tous les sommets $N'_{p,q}$ ($q = 1$ à Q_p) qui n'appartiennent pas à $ALTERN^P$.

[Remarque : en s'inspirant de la remarque énoncée à la fin du paragraphe 2.1.3, on montrerait facilement que les indicateurs d'échec mémorisés par cette opération satisfont l'hypothèse 3'].

Le cheminement arrière au sommet-ET N_p choisi pour point de retour consiste à effectuer les opérations suivantes :

- . les effets des ircm associées aux sommets-OU successeurs de N_p dans l'acm courante sont effacés (cf. §3, chapitre 3),
- . les effets non déjà effacés des ircm dont l'application est postérieure à la décomposition du problème formel Pb_p associé à N_p sont masqués (cf. commentaire c, § 1.2.1).

Le cheminement avant est repris à partir de N_p en appliquant à Pb_p une ircm $i_{p,q}$ telle que $N'_{p,q} \in \text{ALTERN}_1^p$. Ensuite, la décomposition d'un problème Pb consiste à exécuter les opérations suivantes :

1. Si une ircm i est déjà appliquée à Pb dans la partie masquée de l'acm courante, alors on exécute 2, sinon on applique à Pb une ircm choisie à l'aide des heuristiques du paragraphe 3.2 du chapitre 4.
2. 2.1. Pour chaque sommet-OU N' successeur immédiat du sommet-ET N auquel est associé Pb , soit e' l'indicateur d'échec éventuellement associé à N' ; s'il existe $c \in e'$ tel que c ne soit pas satisfaite, alors on supprime e' . [On supprime ainsi tous les identificateurs d'échec qui ne satisfont pas la première partie de l'hypothèse 3].
- 2.2. Si aucun indicateur d'échec n'est associé au sommet-OU correspondant à i , alors :
 - 2.2.1. on démasque les effets de la décomposition de Pb par i , sinon :
 - 2.2.2. on efface les effets (déjà masqués) des ircm associés aux sommets-OU successeurs de N dans la partie masquée de l'acm,
 - 2.2.3. on applique à Pb une ircm i' choisie à l'aide des heuristiques du paragraphe 3.2 du chapitre 4, et telle qu'il n'y ait pas d'indicateur d'échec associé au sommet-OU correspondant à i' .

Le traitement ci-dessus est identique au traitement décrit dans le paragraphe 2.2, au choix des ircm prés : en masquant, puis en démasquant

les effets de certaines ircm, il évite autant que possible de refaire un travail déjà effectué. La démonstration des propositions (1), (2) et (3) faite au paragraphe 2.4.3 ne dépendant pas de la façon dont on choisit les ircm, une démonstration semblable pourrait être faite avec le traitement ci-dessus. Si les hypothèses 1', 2' et 3' sont satisfaites, le traitement ci-dessus produit donc une solution chaque fois qu'il en existe au moins une, car les opérations 1 et 2 précédentes n'éliminent alors aucune ircm dont on n'est pas certain que l'application conduirait à un échec.

2.4. Discussion des hypothèses 1', 2' et 3'.

2.4.1. Hypothèse 1'

Soit e une expression causale ou un indicateur d'échec quelconque. Soit c a l'une des formes suivantes (cf. § 1.1.2):

- 1/ "le littéral l a la valeur VRAI dans l'état courant",
- 2/ "le littéral l a la valeur FAUX dans l'état courant",
- 3/ "le problème formel P_b est associé à un sommet-ET de l'acm courante"

a/ c a la forme 1/ ou 2/

Si, à un instant quelconque du cheminement avant, c est satisfaite, alors, si la valeur VRAI ou FAUX du littéral l résulte de la présence d'un littéral l' dans l'état courant (cf. commentaire b, § 2.4.3, chapitre 4), c satisfait l'hypothèse 1'. En effet, il est alors possible de déterminer l'ircm i dont l'application a engendré le littéral l' , en examinant uniquement les effets immédiats de l'application des ircm associées aux sommets-OU de l'acm courante (d'après le commentaire c du § 1 du chapitre 3, i est unique).

Si, à un instant quelconque du cheminement avant, c est satisfaite, alors, si la valeur VRAI ou FAUX du littéral l résulte de l'absence de certains littéraux dans l'état courant (cf. commentaire b, § 2.4.3, chapitre 4), c ne satisfait pas l'hypothèse 1'. Nous verrons au paragraphe 3.2 (commentaire) comment Tropic traite ce cas.

b/ c a la forme 3/

c satisfait alors l'hypothèse 3'. En effet, si c est satisfaite, il est possible de déterminer l'ircm i dont l'application a engendré le problème formel Pb, en examinant uniquement les effets immédiats de l'application des ircm associées aux sommets-OU de l'acm courante (d'après le commentaire c du § 1 du chapitre 3, i est unique).

2.4.2. Hypothèse 2'

L'expression causale $e = \{c_1 c_2 \dots c_n\}$ qui est formée par Tropic à la suite d'un échec est un ensemble de conditions qui sont satisfaites à l'instant de l'échec. e satisfait donc la première partie de l'hypothèse 2'.

Si e est formée sans l'aide de diagnostics fournis par le concepteur, alors e satisfait la deuxième partie de l'hypothèse 2', i.e. $c_1 \wedge c_2 \wedge \dots \wedge c_n$ est une condition suffisante d'échec (nous le vérifierons au § 3.1.1).

Si e est formée avec l'aide de diagnostics fournis par le concepteur, alors e peut ne pas satisfaire la deuxième partie de l'hypothèse 2' (cf. § 3.1.2 c).

2.4.3. Hypothèse 3'

Un indicateur d'échec e peut avoir été associé à un sommet-OU N' de l'acm :

- . lors du cheminement avant (cf. commentaire a, § 1.2.1) : il est alors facile de vérifier que e satisfait l'hypothèse 3' ;
- . lors de la détermination d'un point de retour (cf. les opérations 3.1 et 3.3.0 de l'algorithme du § 2.2, complété au § 2.3) : on vérifie alors que e satisfait l'hypothèse 3', si l'expression causale formée à la suite de l'échec satisfait l'hypothèse 2' et si les indicateurs d'échec associés aux sommets-OU de l'acm à l'instant de l'échec satisfont l'hypothèse 3' (cf. les remarques à la fin du § 2.1 et à la suite de l'opération 3.3.0 au § 2.3).

Les indicateurs d'échec satisfaisaient donc l'hypothèse 3' si toutes les expressions causales formées par Tropic satisfaisaient l'hypothèse 2'. Comme cela n'est pas toujours le cas, certains indicateurs d'échec peuvent ne pas satisfaire l'hypothèse 3'.

2.4.4. Conclusion

Tropic peut terminer la résolution d'un problème sur un échec, alors qu'il existe au moins une solution, pour trois raisons :

- 1/ Certaines conditions des expressions causales et des indicateurs d'échec utilisées par Tropic peuvent ne pas satisfaire l'hypothèse 1'.
- 2/ Lorsque des diagnostics sont utilisés pour former l'expression causale, celle-ci peut ne pas satisfaire l'hypothèse 2' ; par la suite, certains indicateurs d'échec peuvent ne pas satisfaire 3'.
[On note que cette raison correspond à un choix que nous avons effectué volontairement pour permettre à l'utilisateur de participer au traitement de certains échecs].
- 3/ La non vérification de la compatibilité des restrictions numériques actives à un instant donné n'implique pas l'incompatibilité de ces restrictions (cf. commentaire b, § 4.3.3, chapitre 5) ; Tropic traite alors un échec qui n'est pas un.
[Cette raison est indépendante du traitement des échecs.]

Nous avons soumis à Tropic une trentaine de problèmes. La résolution d'un seul de ces problèmes s'est terminée sur un échec, alors qu'il existait une solution. La raison 3 en a été responsable (cf. § 2.2.3, chapitre 8).

3. Description des algorithmes

Nous décrivons successivement la formation de l'expression causale (§ 3.1), la détermination du point de retour (§ 3.2), le cheminement arrière (§ 3.3) et la reprise du cheminement avant (§ 3.4).

Notation

Dans les paragraphes 1 et 2 nous avons exprimé les expressions causales et les indicateurs d'échec par des ensembles $\{c_1 c_2 \dots c_n\}$, où $c_i, \forall i \in [1, n]$, représentait une condition ayant l'une des trois formes suivantes (cf. § 1.1.2) :

- . "le littéral l a la valeur VRAI dans l'état courant",
- . "le littéral l a la valeur FAUX dans l'état courant",
- . "le problème formel P_b est associé à un sommet-ET de l'acm courante",

Dans le paragraphe 3, nous regroupons les conditions ayant une même forme, et nous exprimons chaque expression causale et indicateur d'échec par une liste $(L_V L_F P)$, où L_V et L_F sont des ensembles de littéraux et P est un ensemble de problèmes formels :

- . chaque littéral l de L_V (resp. L_F) représente la condition "le littéral l a la valeur VRAI (resp. FAUX) dans l'état courant",
- . chaque problème formel P_b de P représente la condition "le problème formel P_b est associé à un sommet-ET de l'acm courante".

3.1. Formation de l'expression causale initiale

Nous appelons expression causale initiale l'expression causale que Tropic forme immédiatement après avoir subi un échec. Nous la définissons d'abord en l'absence de diagnostic (§ 3.1.1), puis en présence de diagnostics (§ 3.1.2).

3.1.1. L'expression causale initiale en l'absence de diagnostic

Nous passons en revue tous les cas d'échec que Tropic peut rencontrer. Pour chaque cas, nous définissons l'expression causale initiale qui est formée en l'absence de diagnostic. $(L_V L_F P)$, où $L_V = \{l_1 l_2 \dots l_m\}$, $L_F = \{l'_1 l'_2 \dots l'_m\}$ et $P = \{P_{b_1} P_{b_2} \dots P_{b_n}\}$, désignant cette expression, il

est facile de vérifier que la conjonction logique $c_1 \wedge c_2 \wedge \dots \wedge c_m \wedge c'_1 \wedge c'_2 \wedge \dots \wedge c'_m \wedge c''_1 \wedge c''_2 \wedge \dots \wedge c''_n$, où c_i ($\forall i \in [1, m]$) est la condition "le littéral l_i a la valeur VRAI dans l'état courant", c'_j ($\forall j \in [1, m']$) est la condition "le littéral l'_j a la valeur FAUX dans l'état courant" et c''_k ($\forall k \in [1, n]$) est la condition "le problème formel Pb_k est associé à un sommet-ET de l'acm courante", est une condition suffisante d'échec.

a/ Cas d'échec 1 (cf. § 2.4.4.a et c, chapitre 4)

L'étape ETAPE(c,q) de la stratégie hiérarchique est en cours d'exécution. A la suite de l'application d'une ircm, l'instance de contrainte (Activ: σ → Restrict: σ) devient active. Une restriction symbolique r de Restrict: σ est contredite par l'état courant (i.e. le littéral r a la valeur FAUX dans cet état).

L'expression causale initiale est alors $(\{l_1 \ l_2 \ \dots \ l_m\} \{r\} \text{NIL})$ où $\{l_1 \ l_2 \ \dots \ l_m\} = \text{Activ}:\sigma$.

b/ Cas d'échec 2 (cf. fin du § 2.4.4, chapitre 4)

A la fin de l'exécution de ETAPE(c,q), une restriction symbolique r contenue dans l'ensemble Restrictions(c,q) est contredite par l'état courant.

L'expression causale initiale est alors $(\{l_1 \ l_2 \ \dots \ l_m\} \{r\} \text{NIL})$ où $\{l_1 \ l_2 \ \dots \ l_m\}$ est la partie gauche de l'instance de contrainte qui impose r.

c/ Cas d'échec 3 (cf. § 2.4.4.b et c, chapitre 4)

ETAPE(c,q) est en cours d'exécution. A la suite de l'application d'une ircm, l'instance de contrainte (Activ: σ → Restrict: σ) devient active. Une restriction symbolique r de Restrict: σ , ayant (c' q') pour couple de dépendance, n'est ni satisfaite, ni contredite par l'état courant, et est contradictoire avec une restriction r' contenue dans l'ensemble Restrictions(c',q').

L'expression causale initiale est alors $(\{l_1 \ l_2 \ \dots \ l_m\} \cup \{l'_1 \ l'_2 \ \dots \ l'_m\} \text{NIL NIL})$ où $\{l_1 \ l_2 \ \dots \ l_m\}$ et $\{l'_1 \ l'_2 \ \dots \ l'_m\}$ sont les parties gauches des instances de contraintes qui imposent r et r'.

d/ Cas d'achec 4 (cf. commentaire du § 4.2, chapitre 5)

L'instance de contrainte (Activ: $\sigma \rightarrow$ Restrict: σ) devient active. Une restriction numérique déclarative r de Restrict: σ , ayant p pour prédicat et c pour C-terme, est contradictoire avec une restriction numérique déclarative r' contenue dans N-Restrictions(p, c).

L'expression causale initiale est alors $(\{l_1 \ l_2 \dots \ l_m\} \cup \{l'_1 \ l'_2 \dots \ l'_m\})$ NIL NIL) où $\{l_1 \ l_2 \dots \ l_m\}$ et $\{l'_1 \ l'_2 \dots \ l'_m\}$ sont les parties gauches des instances qui imposent r et r' .

e/ Cas d'échec 5 (cf. § 3.2, chapitre 4)

ETAPE(c, q) est en cours d'exécution. Tropic tente de décomposer le problème formel Pb. L'application de chacune des ircm applicables à ce problème résulterait en la contradiction immédiate d'une restriction contenue dans Restrictions(c, q).

Soient i_1, i_2, \dots, i_p les ircm qui sont applicables au problème Pb. Soient r_j ($j=1$ à p) la restriction de Restrictions(c, q) qui serait immédiatement contredite par l'application de i_j , et (Activ: $\sigma_j \rightarrow$ Restrict: σ_j) l'instance de contrainte qui impose r_j .

L'expression causale initiale est alors :

$$(\{l_{1,1} \ l_{1,2} \dots \ l_{1,m_1}\} \cup \{l_{2,1} \ l_{2,2} \dots \ l_{2,m_2}\} \cup \dots \cup \{l_{p,1} \ l_{p,2} \dots \ l_{p,m_p}\}) \text{ NIL (Pb)}$$

où, $\forall j \in [1, p], \{l_{j,1} \ l_{j,2} \dots \ l_{j,m_j}\} = \text{Activ}_j : \sigma_j$.

f/ Cas d'échec 6 (cf. § 3.2, chapitre 4)

Tropic tente de décomposer un problème formel Pb auquel aucune ircm n'est applicable.

L'expression causale initiale est alors (NIL NIL {Pb}).

g/ Cas d'échec 7 (cf. § 4.1, chapitre 5)

La vérification de la compatibilité des restrictions numériques actives échoue. Tropic considère alors que ces restrictions sont incompatibles (cf. § 4.3.3.b, chapitre 5).

Soient r_1, r_2, \dots, r_p les restrictions numériques déclaratives et procédurales qui sont actives lors de l'échec. Soit $(\text{Activ}_j : \sigma_j \rightarrow \text{Restrict}_j : \sigma_j)$ l'instance de contrainte qui impose $r_j (\forall j \in [1, p])$.

En première approximation, l'expression causale initiale pourrait alors être :

$(\{l_{1,1} \ l_{1,2} \dots \ l_{1,m_1}\} \cup \{l_{2,1} \ l_{2,2} \dots \ l_{2,m_2}\} \cup \dots \cup \{l_{p,1} \ l_{p,2} \dots \ l_{p,m_p}\} \text{ NIL NIL})$
 où, $\forall j \in [1, p], \{l_{j,1} \ l_{j,2} \dots \ l_{j,m_j}\} = \text{Activ}_j : \sigma_j$. L'expression qui est en fait formée par Tropic résulte d'une double simplification de cette expression :

- Soit le grn de la figure 6.11. Les restrictions numériques qui sont associées aux sommets de ce grn sont incompatibles, mais les restrictions (HAUTEUR TR1 200), (PG HAUTEUR TT1 5), (PP HAUTEUR TT1 25) et ((HAUTEUR TR1)+(+HAUTEUR COL1)(HAUTEUR TT1))) ne sont pas responsables de cette incompatibilité. Une expression causale initiale (L_V NIL NIL), telle que les parties gauches des instances de contraintes qui appliquent ces quatre restrictions aient été utilisées pour former L_V , pourrait conduire Tropic à désactiver seulement une de ces restrictions, puis à subir un nouvel échec qui, comme le précédent, serait dû à l'incompatibilité des restrictions (HAUTEUR BLC1 90), (HAUTEUR BLC2 90), (PG HAUTEUR COL1 185) et ((HAUTEUR COL1)+(+HAUTEUR BLC1)(HAUTEUR BLC2))).

Pour éviter cela, Tropic divise le grn en "fragments" qu'il traite successivement. Auparavant, il construit une liste $\chi = (X_{i_1}^{m_1} \ X_{i_2}^{m_2} \ \dots \ X_{i_q}^{m_q})$ de variables associées aux D-sommets du grn, telle que :

- Chaque variable qui figure dans χ est associée à un D-sommet auquel correspond un ensemble N-Restrictions(p,c) non vide ; de plus, chaque variable X_i^m associée à un D-sommet du grn auquel correspond un ensemble de N-Restrictions(p,c) non vide figure dans χ , si $m > 0$.
- $X_{i_a}^{m_a}$ et $X_{i_b}^{m_b}$ étant deux variables quelconques de χ , si $b > a$ alors $m_b \geq m_a$.
- $m_1 > 0$.

Dans le cas du grn de la figure 6.11, on a donc $\chi = (X_1^1 \ X_1^2)$ [la variable X_i^m associée à chaque D-sommet du grn est indiquée sur la figure].

A chaque variable $X_{i_k}^{m_k}$ de χ , Tropic fait correspondre le fragment du grn qui est constitué par le sous-graphe du grn ayant le D-sommet auquel est associée $X_{i_k}^{m_k}$ pour unique sommet pendant. Par exemple, le fragment du grn

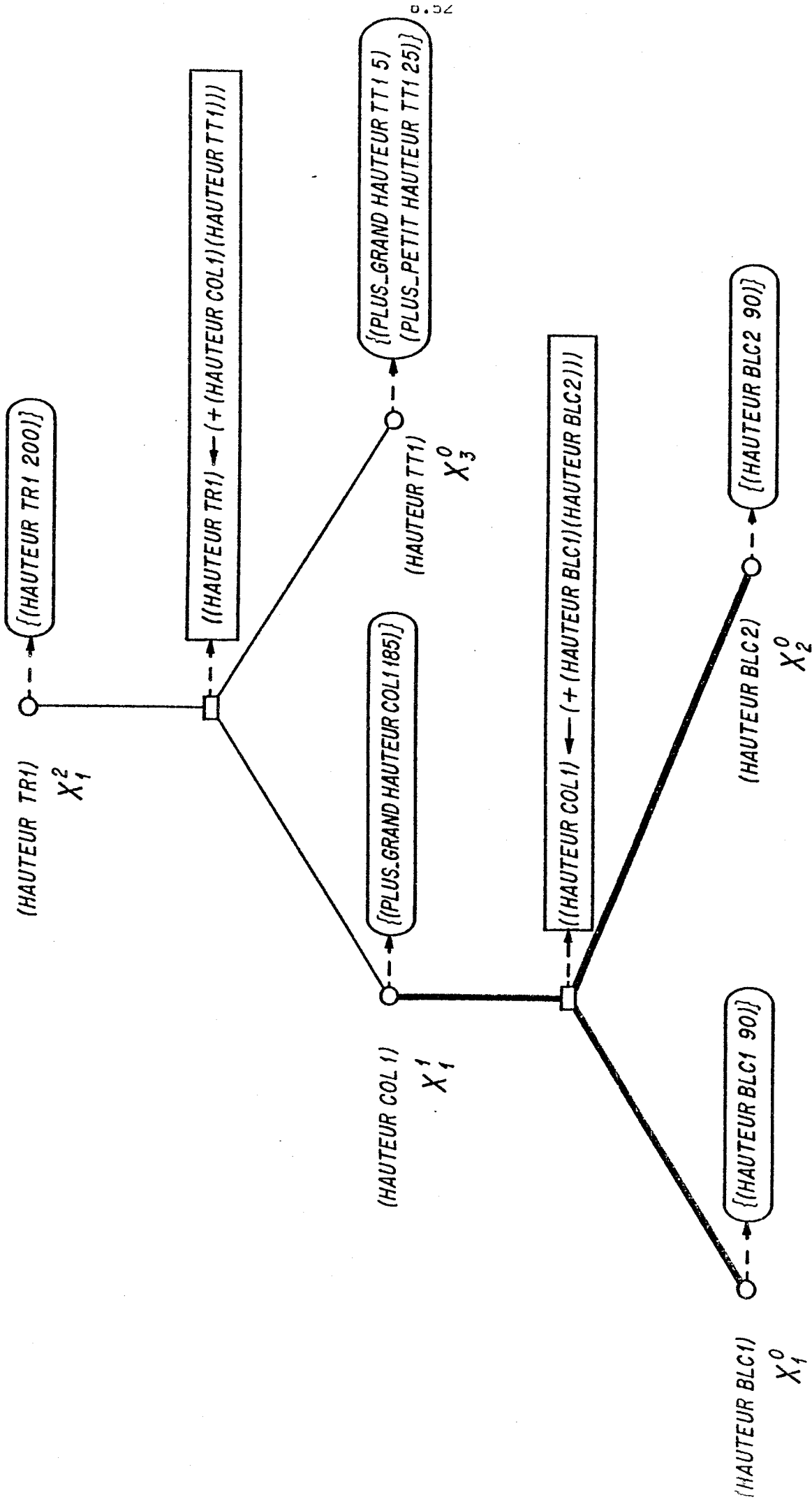


Figure : 6.11

de la figure 6.11 correspondant à χ_1^1 est montré en trait épais sur cette figure.

Tropic applique alors le traitement du paragraphe 4.3 du chapitre 5 aux fragments du grn qui correspondent aux variables contenues dans χ , dans l'ordre défini par cette liste, avant de l'appliquer à l'ensemble du grn. Si la vérification de la compatibilité des restrictions numériques associées aux sommets d'un fragment échoue, alors l'expression causale initiale est $(L_V \text{ NIL NIL})$, où L_V est l'union des parties gauches des instances de contraintes qui imposent ces restrictions. Ainsi, dans le cas du grn de la figure 6.11, Tropic traite d'abord le fragment correspondant à χ_1^1 ; il constate alors l'incompatibilité des restrictions $(\text{HAUTEUR BLC1 90}), (\text{HAUTEUR BLC2 90}), (\text{PG HAUTEUR COL1 185})$ et $((\text{HAUTEUR COL1}) \leftarrow (\text{HAUTEUR BLC1})(\text{HAUTEUR BLC2}))$, et forme l'expression causale initiale $(L_V \text{ NIL NIL})$ uniquement avec les parties gauches des instances de contraintes qui imposent ces restrictions.

- Si parmi les restrictions numériques incompatibles, il existe deux restrictions déclaratives r' et r'' ayant le même prédicat et le même C-terme, et telles que r' soit plus restrictive que r'' (par exemple, $(\text{PP HAUTEUR BLC1 90})$ est plus restrictive que $(\text{PP HAUTEUR BLC1 100})$), alors Tropic ne tient pas compte de r'' pour former l'expression causale initiale.

3.1.2. L'expression causale initiale en présence des diagnostics

Le présence de diagnostics fournis par l'utilisateur affecte la formation de l'expression causale initiale uniquement dans le cas 7 d'échec. Dans toutes les autres cas, l'expression causale initiale est celle que nous avons définie dans le paragraphe 3.1.1.

a/ Définition d'un diagnostic

Un diagnostic est une expression de la forme

$(\text{Activ} \rightarrow \text{Indic})$

où :

- . *Activ* est un ensemble de littéraux qui représentent les conditions d'activation du diagnostic,
- . *Indic* est soit un ensemble de littéraux, soit une fonction qui s'évalue à un ensemble de littéraux.

Les expressions Activ et Indic sont respectivement appelées partie gauche et partie droite du diagnostic.

b/ La fonction %ECHEC

La fonction %ECHEC que nous définissons ci-dessous sert à caractériser un échec du cas 7. Elle peut être utile pour exprimer la partie droite de certains diagnostics (cf. l'exemple de diagnostic au § 1.3.2).

Soit $\chi = (X_{i_1}^{m_1} X_{i_2}^{m_2} \dots X_{i_q}^{m_q})$ la liste de variables construite pour fragmenter le grn (cf. § 3.1.1.g). Tropic contrôle successivement la compatibilité des restrictions numériques associées aux fragments F_1, F_2, \dots, F_q du grn qui correspondent aux variables $X_{i_1}^{m_1}, X_{i_2}^{m_2}, \dots, X_{i_q}^{m_q}$ de χ . Lors du contrôle de la compatibilité des restrictions associées aux sommets de F_k ($k=2,3,\dots,q$), on est certain (de par la définition des fragments) de la compatibilité des restrictions associées aux sommets de F_k autres que le D-sommet N_k correspondant à $X_{i_k}^{m_k}$.

Pour calculer la violation du fragment F_k , Tropic sous-pondère la violation des restrictions les plus restrictives associées au D-sommet N_k (relativement à la violation des restrictions associées aux D-sommets de F_k autres que N_k) et annule les violations des autres restrictions associées à N_k . Ainsi, si les restrictions associées aux sommets de F_k sont incompatibles seule une restriction associée à N_k est nettement contredite à la fin du traitement de F_k . Soient r cette restriction éventuelle, p son prédicat, c son C-terme, a son A-terme et μ la valeur de $X_{i_k}^{m_k}$ à la fin du traitement de F_k . La fonction %ECHEC($p' c'$) s'évalue alors de la façon suivante :

- . si $p' \neq p$ et/ou $c' \neq c$, alors %ECHEC($p' c'$)=0,
- . si $p' = p$ et $c' = c$, alors %ECHEC($p' c'$)= $\frac{\mu-a}{a}$.

c/ Utilisation des diagnostics

Dans le cas 7 d'échec (et uniquement dans ce cas), Tropic forme l'expression causale initiale de la façon suivante :

Soit INDIC l'union des ensembles de littéraux obtenus en évaluant les parties droites des instances de diagnostics dont les conditions d'activation sont satisfaites par l'état courant à l'instant de l'échec :

- . si INDIC \neq NIL, alors l'expression causale initiale est $(L_V L_F \text{ NIL})$, où L_V (resp. L_F) est l'ensemble des littéraux de INDIC qui ont la valeur VRAI (resp. FAUX) dans l'état courant à l'instant de l'échec,
- . si INDIC = NIL, alors l'expression causale initiale est l'expression $(L_V \text{ NIL NIL})$ définie au § 3.1.1.g.

Commentaire

Soit $(\{l_1 l_2 \dots l_m\} \{l'_1 l'_2 \dots l'_m\} \text{ NIL})$ l'expression causale initiale qui est formée si $\text{INDIC} \neq \text{NIL}$. La conjonction $c_1 \wedge c_2 \wedge \dots \wedge c_m \wedge c'_1 \wedge c'_2 \wedge \dots \wedge c'_m$, où c_i ($\forall i \in [1, m]$) représente la condition "le littéral l_i a la valeur VRAI dans l'état courant" et c'_j ($\forall j \in [1, m']$) représente la condition "le littéral l'_j a la valeur FAUX dans l'état courant", peut ne pas être une condition suffisante d'échec.

3.2. La détermination du point de retour

A la suite d'un échec, Tropic détermine un point de retour en passant en revue les sommets-ET de l'acm, dans l'ordre inverse de la décomposition des problèmes formels qui leur sont associés et en commençant par le sommet-ET auquel est associé le problème formel le plus récemment décomposé. Pour chaque sommet-ET N examiné, il exécute l'algorithme ci-dessous, dans lequel nous notons :

- . P_b le problème formel associé à N ,
- . i l'ircm qui est appliquée à P_b dans l'acm courante,
- . $(L_V L_F P)$ l'expression causale courante (l'expression causale initiale, si N est le premier sommet-ET examiné).

[Note : Dans cet algorithme, nous supposons que la valeur VRAI (resp. FAUX) de chaque littéral l de L_V (resp. L_F) résulte de la présence d'un littéral l' dans l'état courant. Nous verrons en commentaire le cas où il existe dans L_V et L_F des littéraux dont la valeur résulte de l'absence de certains littéraux dans l'état courant].

1. 1.1. $\text{ALTERN} \leftarrow \text{NIL}$,
- 1.2. $\varepsilon \leftarrow (L_V L_F P)$.

2. 2.1. Pour tout $l \in L_V$ (resp. L_F) tel que la valeur VRAI (resp. FAUX) de l dans l'état courant résulte de la présence dans cet état

d'un littéral engendré par l'application de i , on effectue :

2.1.1. $L_V \leftarrow L_V - \{1\}$ (resp. $L_F \leftarrow L_F - \{1\}$),

2.1.2. pour toute ircm i' de la forme $(Pb \rightarrow \dots)$ dont l'application n'engendrerait aucun littéral donnant à 1 la valeur VRAI (resp. FAUX), on effectue : $ALTERN \leftarrow ALTERN \cup \{N'\}$, où N' note le sommet-OU auquel est associée i' .

2.2. Pour tout $pb \in P$ tel que pb a été engendré par l'application de i à Pb , on effectue

2.2.1. $P \leftarrow P - \{pb\}$

2.2.2. pour toute ircm i' de la forme $(Pb \rightarrow \dots)$ dont l'application n'engendrerait pas pb , on effectue : $ALTERN \leftarrow ALTERN \cup \{N'\}$, où N' note le sommet-OU auquel est associée i' .

3. Soit $ALTERN_1$ l'ensemble des sommets de $ALTERN$ tels qu'aucun indicateur d'échec ne leur soit associé.

Si $ALTERN_1 \neq NIL$, alors :

3.1. $(L_V L_F P)$ est associé comme indicateur d'échec au sommet-OU qui correspond à i ,

3.2. N est choisi pour point de retour et l'ensemble $ALTERN_1$ lui est associé [à la reprise du cheminement avant, Tropic appliquera à Pb une ircm associée à un sommet-OU de cet ensemble (cf. § 3.4)],

sinon :

3.3. Si N n'est pas la racine de l'acm, alors :

3.3.1. si $ALTERN \neq NIL$, alors $ALTERN$ est associé à N [nous verrons l'utilisation de cet ensemble au paragraphe 3.4.],

3.3.2. pour tout $N' \in ALTERN$, $(L'_V L'_F P')$ étant l'indicateur d'échec associé à N' , on effectue :

. $L_V \leftarrow L_V \cup L'_V$,

. $L_F \leftarrow L_F \cup L'_F$,

. $P \leftarrow P \cup P'$,

3.3.3. si $\varepsilon \neq (L_V L_F P)$, alors $P \leftarrow P \cup \{Pb\}$,

sinon :

3.3.4. on abandonne la résolution du problème original.

Commentaire

Pour tenir compte du cas où il existe dans L_V et L_F des littéraux ayant dans l'état courant une valeur VRAI ou FAUX qui résulte de l'absence de certains littéraux dans cet état, nous ajoutons à l'algorithme ci-dessus l'opération 2.3. suivante :

2.3. pour toute ircm i' de la forme $(Pb \rightarrow \dots)$ dont l'application engendrerait un littéral donnant la valeur FAUX (resp. VRAI) à un littéral l de L_V (resp. L_F), on effectue :

2.3.1. $L_V \leftarrow L_V - \{l\}$ (resp. $L_F \leftarrow L_F - \{l\}$),

2.3.2. $ALTERN \leftarrow ALTERN \cup \{N'\}$, où N' est le sommet-OU auquel est associée i' .

De plus, lorsque tous les problèmes formels ayant un même couple de dépendance (c, q) ont été examinés, Tropic modifie l'expression causale courante en retirant de L_V et de L_F tous les littéraux ayant (c, q) pour couple de dépendance.

Toutefois, ces modifications n'assurent pas que Tropic engendre une solution du problème original, chaque fois qu'il en existe au moins une (cf. § 2.4).

3.3. Le cheminement arrière

Soient N_R le sommet-ET choisi pour point de retour, Pb_R le problème formel qui lui est associé et i_R l'ircm qui est appliquée à Pb_R dans l'acm courante.

Le cheminement arrière au sommet-ET N_R consiste à effectuer les opérations suivantes :

1. Les effets des ircm associées au sommets-OU successeurs de N_R dans l'acm courante (y compris la partie masquée) sont effacés (cf. § 3, chapitre 3).
2. Les effets non déjà effacés des ircm dont l'application est postérieure à celle de i_R sont masqués :

- . les littéraux qui ont été ajoutés à l'état courant par l'application de ces ircm ne sont plus considérés comme appartenant à cet état, bien qu'il n'en soit pas retirés effectivement,
- . les problèmes formels associés aux sommets-ET non pendants de la partie de l'acm engendrée par l'application de ces ircm ne sont plus considérés comme décomposés, bien que leur décomposition ne soit pas effacée .

[Commentaire : Soient $ETAPE(c_1, q_1), ETAPE(c_2, q_2), \dots, ETAPE(c_N, q_N)$ les étapes de la stratégie hiérarchique qui étaient exécutées à l'instant de l'échec. Soit $(c_i, q_i), i \in [1, N+1]$, le couple de dépendance de Pb_R . Après l'opération 2, seule les étapes $ETAPE(c_1, q_1), ETAPE(c_2, q_2), \dots, ETAPE(c_{i-1}, q_{i-1})$ sont considérées comme exécutées par Tropic. Ce commentaire est important car, dans certains cas, la valeur logique d'un littéral ayant (c, q) pour couple de dépendance peut être déterminée ou non suivant que $ETAPE(c, q)$ est considéré comme exécuté ou non].

3. Le travail d'activation des contraintes, des conseils et des diagnostics est mis à jour (cf. § 2, chapitre 7).

3.4. La reprise du cheminement avant

Soient N_R le point de retour du cheminement arrière et Pb_R le sommet-ET associé à N_R . Soient $ALTERN_1$ l'ensemble de sommets-OU associé à N_R (cf. opération 3.2., § 3.2.) et I l'ensemble des ircm correspondant à ces sommets.

Tropic reprend le cheminement avant à partir de N en appliquant à Pb une ircm qu'il choisit dans I en utilisant les heuristiques décrites au paragraphe 3.2 du chapitre 4 (auparavant, il supprime l'ensemble $ALTERN_1$ associé à N_R). Il décompose ensuite les problèmes formels associés aux sommets-ET pendants de la partie non masquée de l'acm dans l'ordre déterminé par l'application de la stratégie hiérarchique (cf. § 2.4.2, chapitre 4). Pour chaque problème, il exécute l'algorithme ci-dessous, dans lequel nous notons :

- . Pb le problème formel à décomposer,
- . N le sommet-ET auquel est associé Pb .

1. Si une ircm i est déjà appliquée à P_b dans la partie masquée de l'acm courante, alors on exécute 2, sinon on applique à P_b une ircm choisie parmi les ircm de la forme $(P_b \rightarrow \dots)$ en utilisant les heuristiques décrites au paragraphe 3.2 du chapitre 4.

2. 2.1. Pour chaque sommet-OU N' successeur immédiat de N , tel qu'un indicateur d'échec $(L_V L_F P)$ soit associé à N' , on effectue :
 - si L_V (resp. L_F) contient un littéral dont la valeur dans l'état courant est FAUX (resp. VRAI) ou indéterminée, et/ou si P contient un problème qui n'est associé à aucun sommet-ET de l'acm, alors l'indicateur d'échec est supprimé.

 - 2.2. Si aucun indicateur d'échec n'est associé au sommet-OU correspondant à i , alors :
 - 2.2.1. on supprime l'ensemble ALTERN de sommets-OU qui a éventuellement été associé à N (cf. opération 3.3.1, § 3.2),
 - 2.2.2. on démasque les effets de la décomposition de P_b par i ,

 - sinon :
 - 2.2.3. on effectue les opérations 1 et 3 de l'algorithme du paragraphe 3.3.,
 - 2.2.4. ALTERN étant l'ensemble de sommets-OU associé à N (cf. opération 3.3.1, § 3.2), $ALTERN_1$ étant l'ensemble des sommets-OU de ALTERN auxquels aucun indicateur d'échec n'est associé, et I étant l'ensemble des ircm associées aux sommets-OU de $ALTERN_1$, on applique à P_b une ircm choisie dans I en utilisant les heuristiques décrites au paragraphe 3.2 du chapitre 4.
 [Commentaire : Si chaque ircm de I est telle que son application résulterait en la contradiction immédiate d'une restriction contenue dans un ensemble Restrictions(c,q), on est en échec. L'expression causale initiale est alors celle du paragraphe 3.1.1.e (les ircm i_1, i_2, \dots, i_p étant les ircm contenues dans I), à laquelle on ajoute le contenu des indicateurs d'échecs qui sont associés aux sommets-OU de $ALTERN-ALTERN_1$].

Chapitre 7: ACTIVATION DES CONTRAINTES, DES CONSEILS ET DES DIAGNOSTICS

Chaque contrainte, conseil et diagnostic est une règle de production (Activ \rightarrow ...) dans laquelle Activ est un ensemble de littéraux qui représentent les conditions d'activation de la règle. L'instance par σ de cette règle est active dans un état m si et seulement si ces conditions sont satisfaites par m .

Le paragraphe 1 de ce chapitre décrit l'algorithme d'activation qui détermine les instances de contraintes, de conseils et de diagnostics qui deviennent actives pendant le cheminement avant. Le paragraphe 2 décrit l'algorithme qui met à jour le travail d'activation lors des cheminement arrière.

1. Activation des contraintes, des conseils et des diagnostics pendant le cheminement avant.

1.1. Hypothèses de travail

La description des connaissances à l'aide de règles de production a l'avantage d'être modulaire (cf. § 2.7., chapitre 1). Toutefois, elle nécessite de maintenir une partie de Tropic constamment en éveil, car de nouvelles règles de production peuvent devenir actives dans l'état courant chaque fois qu'une ircm est appliquée. De plus, la détermination des instances de règles qui sont éventuellement actives peut être parfois combinatoire. Pour éviter ces inconvénients, nous avons conçu l'algorithme d'activation en imposant que la partie gauche Activ d'une contrainte, d'un conseil ou d'un diagnostic quelconque possède les trois propriétés a, b et c énoncées ci-dessous:

a/ Soit Ω l'ensemble des atomes qui notent les C-termes de Activ (ces atomes sont nécessairement des variables). Nous définissons dans Activ la relation \succ par: c_1 et c_2 étant deux éléments quelconques de Ω , $c_1 \succ c_2$ si et seulement si il existe dans Activ un A-H-littéral $(p \ c_1 \ c_2)$. Le graphe de cette relation doit être une arborescence. L'atome qui est associé à la racine de cette arborescence est appelé la racine de Activ.

Cette propriété est utilisée pour coordonner l'exécution de l'algorithme d'activation avec l'application de la stratégie hiérarchique. Elle permet de guider le travail d'activation en évitant d'examiner chaque contrainte, conseil et diagnostic du problème original chaque fois que l'état courant est modifié (cf. le rôle des ensembles $Activ1(c,q)$ et $Activ2(c,q)$ dans les pas 3 et 4 de l'algorithme d'activation, § 1.2.c et d).

[Commentaire: Au paragraphe 2.3.2. du chapitre 4, nous avons défini le couple de dépendance d'un littéral d'un état m dans m . Grâce à la propriété ci-dessus, nous définissons de la même façon le couple de dépendance d'un littéral de $Activ$ dans $Activ$. Par exemple, soit $Activ = \{(TYPE \$TR TOUR)(COMPOSANT \$TR \$COL)(COMPOSANT \$TR \$TT)(TYPE \$COL COLONNE)(TYPE \$TT TOIT)(COULEUR \$TT ROUGE)(COMPOSANT \$COL \$BLC)(TYPE \$BLC BLOC)(SECTION \$BLC \$S)\}$. Les littéraux $(TYPE \$TR TOUR)$, $(COMPOSANT \$TR \$COL)$ et $(COULEUR \$TT ROUGE)$ ont respectivement $(\$TR 0)$, $(\$TR 1)$ et $(\$TT 0)$ pour couple de dépendance.]

b/ $Activ$ doit contenir un D-S-littéral et un seul de la forme $(TYPE r a)$ où r est la racine de $Activ$ et a est une constante. [Note : On a obligatoirement $Struct(TYPE) = (D S)$].

Cette propriété permet à l'algorithme d'activation de reconnaître une nouvelle occasion d'activer la règle de production ayant $Activ$ pour partie gauche, chaque fois qu'un littéral de la forme $(TYPE c a)$ est ajouté à l'état courant (cf. le rôle de l'ensemble $Type(a)$ dans le pas 2 de l'algorithme d'activation, § 1.2.b).

c/ $Activ$ ne doit contenir aucun D-N-littéral

Cette propriété permet de ne pas remettre en question le travail de l'algorithme d'activation chaque fois que la compatibilité des restrictions numériques actives est vérifiée.

1.2. L'algorithme d'activation

L'algorithme d'activation comporte quatre pas exécutés à des instants différents du cheminement avant. Nous décrivons ces pas ci-dessous en les illustrant sur un exemple.

a/ Pas 1

Il s'agit d'un pas de structuration des informations qui est exécuté une seule fois au début de la résolution du problème original. Il consiste à effectuer les opérations 1.1 et 1.2 ci-dessous pour chaque contrainte, conseil et diagnostic de ce problème (les hypothèses du § 1.1. sont simultanément vérifiées).

1.1. Un nom interne δ est engendré, et les propriétés suivantes lui sont associées:

- 1.1.1. $Catg(\delta) \leftarrow CNTR$ (pour une contrainte), $CNSL$ (pour un conseil) ou $DIAG$ (pour un diagnostic),
- 1.1.2. $Pgauche(\delta) \leftarrow$ partie gauche de la règle de production,
- 1.1.3. $Pdroite(\delta) \leftarrow$ partie droite de la règle de production,
- 1.1.4. $Nbdep(\delta) \leftarrow$ cardinal de la partition P de $Pgauche(\delta)$ telle que chaque classe de P soit composée des littéraux de $Pgauche(\delta)$ ayant un même couple de dépendance dans $Pgauche(\delta)$ (cf. commentaire, § 1.1.a).

1.2. Soit $(TYPE\ r\ a)$ le D-S-littéral de $Pgauche(\delta)$ tel que $r = Racine(\delta)$.
On range δ comme nouvel élément d'un ensemble noté $Type(a)$ initialement vide.

Exemple:

Supposons que la contrainte suivante figure dans l'énoncé du problème original:

$\{(TYPE\ \$TR\ TOUR)(COMPOSANT\ \$TR\ \$COL)(COMPOSANT\ \$TR\ \$TT)(TYPE\ \$COL\ COLONNE)$
 $(TYPE\ \$TT\ TOIT)(COULEUR\ \$TT\ ROUGE)(COMPOSANT\ \$COL\ \$BLC)(TYPE\ \$BLC\ BLOC)$
 $(SECTION\ \$BLC\ \$S)\}$

\rightarrow *Fonct:* si $\$S = CARREE$ alors $I?\{(COULEUR\ \$BLC\ BLANCHE)\}$
 sinon $I?\{(COULEUR\ \$BLC\ VERTE)\}$

Lors du pas 1, Tropic engendre un nom interne pour cette contrainte (soit Δ ce nom) et lui associe les propriétés suivantes:

- $Catg(\Delta) \leftarrow CNTR,$
- $Pgauche(\Delta) \leftarrow \{(TYPE\ \$TR\ TOUR)(COMPOSANT\ \$TR\ \$COL)(COMPOSANT\ \$TR\ \$TT)$
 $(TYPE\ \$COL\ COLONNE)(TYPE\ \$TT\ TOIT)(COULEUR\ \$TT\ ROUGE)(COMPOSANT\ \$COL\ \$BLC)$
 $(TYPE\ \$BLC\ BLOC)(SECTION\ \$BLC\ \$S)\},$

. $Pdroite(\Delta) \leftarrow Fonct$: si $S = CARREE$ alors $I? \{(COULEUR \$BLC BLANCHE)\}$
 sinon $I? \{(COULEUR \$BLC VERTE)\}$

. $Racine(\Delta) = \$TR$,

. $Nbdep(\Delta) \leftarrow 6$.

Ensuite, Δ est rangé dans l'ensemble $Type(TOUR)$.

[Au cours des pas 2, 3 et 4 décrits ci-après, Tropic construit des arborescences, dites arborescences d'activation, aux sommets desquelles sont associées plusieurs propriétés. La figure 7.1. montre l'arborescence d'activation qui est construite pour la contrainte que nous venons de prendre pour exemple, pendant le cheminement avant du tableau 4.1. (une seule propriété associée à chaque sommet est indiquée sur la figure). La construction de cette arborescence est décrite dans les exemples qui illustrent les pas 2, 3 et 4].

b/ Pas 2

Ce pas est exécuté chaque fois qu'un littéral (TYPE c a) est ajouté à l'état courant. Il consiste à effectuer les opérations 2.1 et 2.2 ci-dessous, pour tout $\delta \in Type(a)$.

2.1. La racine ϵ d'une nouvelle arborescence d'activation est créée, et les propriétés suivantes lui sont associées:

2.1.1. $Subst(\epsilon) \leftarrow \{(r c)\}$, où $r = Racine(\delta)$,

2.1.2. $Nbdep(\epsilon) \leftarrow Nbdep(\delta)$,

2.1.3. $Orig(\epsilon) \leftarrow \delta$.

2.2. 2.2.1. Si $Nbdep(\epsilon) > 1$, alors ϵ est rangé comme nouvel élément d'un ensemble noté $Activ1(c,0)$ [l'activation de la règle de production ne se fera pas pendant l'étape $ETAPE(c,0)$ en cours d'exécution].

2.2.2. Si $Nbdep(\epsilon) = 1$, alors ϵ est rangé comme nouvel élément d'un ensemble noté $Activ2(c,0)$ [l'activation de la règle de production se fera éventuellement pendant l'étape $ETAPE(c,0)$ en cours d'exécution].

[Note: Tous les ensembles $Activ1(c,q)$ et $Activ2(c,q)$ sont vides au début du cheminement avant.]

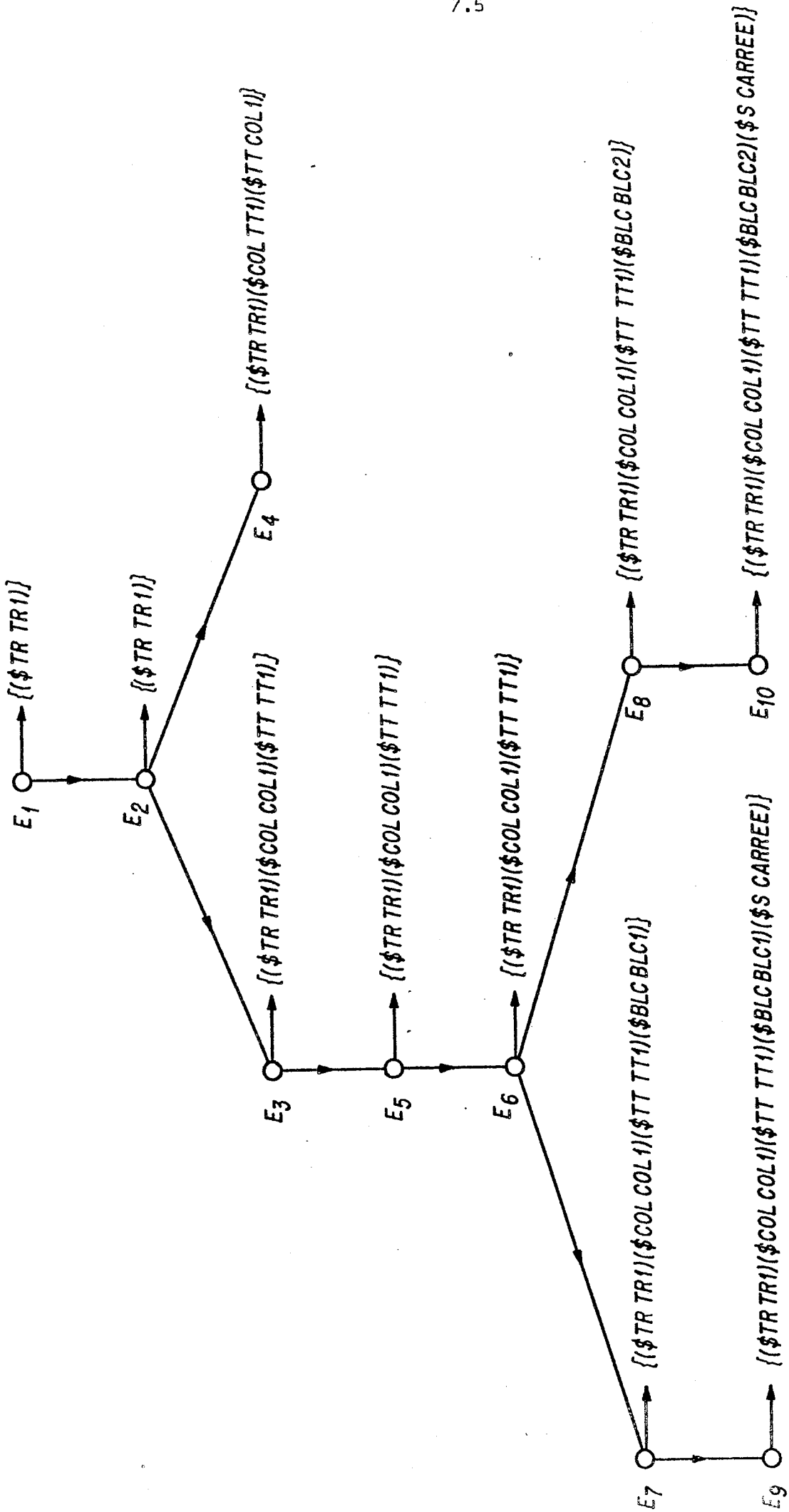


Figure: 7.1

Exemple:

Nous nous plaçons dans le cadre du cheminement avant décrit par le tableau 4.1. Lorsque le littéral (TYPE TR1 TOUR) est ajouté à l'état courant (application de $rcml:NIL$), Tropic crée la racine d'une nouvelle arborescence d'activation pour l'élément Δ de Type(TOUR). Soit E1 cette racine. Les propriétés suivantes lui sont associées.

- . $Subst(E1) \leftarrow \{(\$TR TR1)\}$ [Note: c'est la seule propriété qui soit indiquée sur la figure 7.1.],
- . $Nbdep(E1) \leftarrow 6$,
- . $Orig(E1) \leftarrow \Delta$.

E1 est rangé comme un nouvel élément de $Activ1(TR1,0)$.

c/ Pas 3

Ce pas est exécuté à la fin de chaque ETAPE(c,q) de la stratégie hiérarchique. Il consiste à effectuer les opérations 3.1, 3.2 et 3.3 ci-dessous pour tout $\epsilon \in Activ1(c,q)$.

- 3.1. Soit L l'ensemble de tous les littéraux de $Pgauche(Orig(\epsilon)):Subst(\epsilon)$ qui ont (c q) pour couple de dépendance dans $Pgauche(Orig(\epsilon)):Subst(\epsilon)$. La fonction INSTANCES (qui sera définie au § 1.3) est appliquée à L . Elle s'évalue à $\Sigma = \{(\sigma_1 \dots)(\sigma_2 \dots) \dots (\sigma_n \dots)\}$, où $\sigma_1, \sigma_2, \dots, \sigma_n$ sont toutes les substitutions distinctes telles que:
- . $\forall i \in [1,n]$, tous les littéraux de $L:\sigma_i$ ont la valeur VRAI dans l'état courant,
 - . $\forall i \in [1,n]$, toutes les variables qui figurent dans σ_i apparaissent aussi dans L ,
 - . $\forall i \in [1,n]$, il n'existe pas deux variables différentes utilisées pour noter des C-termes dans L , qui ont la même constante pour substitut dans σ_i . [Note: Cette condition traduit un choix arbitraire que nous avons fait dans la version actuelle de Tropic. Elle permet de conserver dans $Pgauche(Orig(\epsilon)):Subst(\epsilon)$ la structure hiérarchique de $Pgauche(Orig(\epsilon))$, ϵ étant un sommet quelconque d'une arborescence d'activation.]

Σ peut être l'ensemble vide.

3.2. Les opérations 3.2.1 et 3.2.2 ci-dessous sont effectuées pour tout élément (σ_1, \dots) de Σ . Si $\Sigma = \text{NIL}$, alors on va directement effectuer l'opération 3.3.

3.2.1. Un sommet ϵ' est ajouté à l'arborescence d'activation qui contient ϵ , comme successeur immédiat de ϵ . Les propriétés suivantes lui sont associées:

3.2.1.1. $\text{Subst}(\epsilon') \leftarrow \text{Subst}(\epsilon) \cup \sigma_1$

3.2.1.2. $\text{Nbdep}(\epsilon') \leftarrow \text{Nbdep}(\epsilon) - 1$,

3.2.1.3. $\text{Orig}(\epsilon') \leftarrow \text{Orig}(\epsilon)$.

3.2.2. Pour tout couple (c', q') tel que:

. au moins un littéral de $\text{Pgauche}(\text{Orig}(\epsilon')) : \text{Subst}(\epsilon')$ a (c', q') pour couple de dépendance dans l'état courant,

. $\text{ETAPE}(c', q')$ n'a pas été exécutée,

ϵ' est rangé comme nouvel élément de l'ensemble $\text{Activ1}(c', q')$

3.3. ϵ est retiré de tous les ensembles $\text{Activ1}(c, q)$ où il figure.

Si $\Sigma = \text{NIL}$, alors ϵ est déclaré abandonné.

[Au début de chaque étape $\text{ETAPE}(c, q)$, les éléments ϵ de $\text{Activ1}(c, q)$ tels que $\text{Nbdep}(\epsilon) = 1$ sont rangés comme nouveaux éléments d'un ensemble noté $\text{Activ2}(c, q)$. Simultanément, ces éléments sont retirés de $\text{Activ1}(c, q)$. Aucune instance de contrainte, conseil ou diagnostic ne peut donc devenir active pendant l'exécution du pas 3].

Exemple:

. A la fin de $\text{ETAPE}(TR1, 0)$ (cf. Tableau 4.1.), Tropic effectue les opérations du pas 3 pour l'élément $E1$ de $\text{Activ1}(TR1, 0)$. On a alors $L = \{(\text{TYPE } TR1 \text{ TOUR})\}$ et $\text{INSTANCES}(L)$ s'évalue à $\Sigma = \{(\text{NIL} \dots)\}$. Un sommet $E2$ est ajouté à l'arborescence d'activation qui contient $E1$ (cf. figure 7.1.), comme successeur immédiat de $E1$, et les propriétés suivantes lui sont associées: $\text{Subst}(E2) \leftarrow \{(\$TR \text{ } TR1)\}$, $\text{Nbdep}(E2) \leftarrow 5$ et $\text{orig}(E2) \leftarrow \Delta$. $E2$ est rangé dans $\text{Activ1}(TR1, 1)$. $E1$ est retiré de $\text{Activ1}(TR1, 0)$.

. A la fin de $\text{ETAPE}(TR1, 1)$, Tropic effectue les opérations du pas 3 pour l'élément $E2$ de l'ensemble $\text{Activ1}(TR1, 1)$. On a alors $L = \{(\text{COMPOSANT } TR1 \text{ } \$COL)(\text{COMPOSANT } TR1 \text{ } \$TT)\}$ et $\text{INSTANCES}(L)$ s'évalue à $\Sigma = \{(\{(\$COL \text{ } COL1)$

$(\$TT\ TT1)\dots)(\{(\$COL\ TT1)(\$TT\ COL1)\ \dots\})$. Deux sommets $E3$ et $E4$ sont ajoutés à l'arborescence d'activation qui contient $E2$, comme successeurs immédiats de $E2$, et les propriétés suivantes leur sont associées:

- $Subst(E3) \leftarrow \{(\$TR\ TR1)(\$COL\ COL1)(\$TT\ TT1)\}$, $Nbdep(E3) \leftarrow 4$ et $Orig(E3) \leftarrow \Delta$,
 - $Subst(E4) \leftarrow \{(\$TR\ TR1)(\$COL\ TT1)(\$TT\ COL1)\}$, $Nbdep(E4) \leftarrow 4$ et $Orig(E4) \leftarrow \Delta$.
- $E3$ est rangé dans $Activ1(COL1,0)$, $Activ1(COL1,1)$ et $Activ1(TT1,0)$. $E4$ est rangée dans $Activ1(TT1,0)$, $Activ1(TT1,1)$ et $Activ1(COL1,0)$. $E2$ est retiré de $Activ1(TR1,1)$.

. A la fin de ETAPE $(COL1,0)$, Tropic effectue les opérations du pas 3 pour les éléments $E3$ et $E4$ de $Activ1(COL1,0)$:

- Pour $E3$, on a $L = \{(TYPE\ COL1\ COLONNE)\}$ et $INSTANCES(L)$ s'évalue à $\Sigma = \{(NIL, \dots)\}$. Un sommet $E5$ est ajouté à l'arborescence d'activation qui contient $E3$, comme successeur immédiat de $E3$, et les propriétés suivantes lui sont associées: $Subst(E5) \leftarrow \{(\$TR\ TR1)(\$COL\ COL1)(\$TT\ TT1)\}$, $Nbdep(E5) \leftarrow 3$ et $Orig(E5) \leftarrow \Delta$. $E5$ est rangé dans $Activ1(COL1,1)$ et $Activ1(TT1,0)$. $E3$ est retiré de $Activ1(COL1,0)$, $Activ1(COL1,1)$ et $Activ1(TT1,0)$.
- Pour $E4$, on a $L = \{(TYPE\ COL1\ TOIT)(COULEUR\ COL1\ ROUGE)\}$ et $INSTANCES(L)$ s'évalue à $\Sigma = NIL$. $E4$ est retiré des ensembles $Activ1(TT1,0)$, $Activ1(TT1,1)$ et $Activ1(COL1,0)$, puis est déclaré abandonné.

. A la fin de ETAPE $(TT1,0)$, Tropic effectue les opérations du pas 3 pour l'élément $E5$ de $Activ1(TT1,0)$. On a alors $L = \{(TYPE\ TT1\ TOIT)(COULEUR\ TT1\ ROUGE)\}$ et $INSTANCES(L)$ s'évalue à $\Sigma = \{(NIL, \dots)\}$. Un sommet $E6$ est ajouté à l'arborescence d'activation qui contient $E5$, comme successeur immédiat de $E5$, et les propriétés suivantes lui sont associées: $Subst(E6) \leftarrow \{(\$TR\ TR1)(\$COL\ COL1)(\$TT\ TT1)\}$, $Nbdep(E6) \leftarrow 2$ et $Orig(E6) \leftarrow \Delta$. $E6$ est rangé dans $Activ1(COL1,1)$. $E5$ est retiré de $Activ1(TT1,0)$ et $Activ1(COL1,1)$.

. A la fin de ETAPE $(COL1,1)$, Tropic effectue les opérations du pas 3 pour l'élément $E6$ de $Activ1(COL1,1)$. On a alors $L = \{(COMPOSANT\ COL1\ \$BLC)\}$ et $INSTANCES(L)$ s'évalue à $\Sigma = \{(\{(\$BLC\ BLC1)\}\dots)(\{(\$BLC\ BLC2)\}\dots)\}$. Deux sommets $E7$ et $E8$ sont ajoutés à l'arborescence d'activation qui contient $E6$,

comme successeurs immédiats de E6, et les propriétés suivantes leur sont associées:

- $Subst(E7) \leftarrow \{(\$TR TR1)(\$COL COL1)(\$TT TT1)(\$BLC BLC1)\}$, $Nbdep(E7) \leftarrow 1$ et $Orig(E7) \leftarrow \Delta$,
- $Subst(E8) \leftarrow \{(\$TR TR1)(\$COL COL1)(\$TT TT1)(\$BLC BLC2)\}$, $Nbdep(E8) \leftarrow 1$ et $Orig(E8) \leftarrow \Delta$.

E7 est rangé dans $Activ1(BLC1,0)$. E8 est rangé dans $Activ1(BLC2,0)$. E6 est retiré de $Activ1(COL1,1)$.

L'activation de la contrainte se poursuit au pas 4 de l'algorithme.

d/ Pas 4

Nous rappelons qu'au début de chaque étape $ETAPE(c,q)$, tous les éléments ϵ de $Activ1(c,q)$ tels que $Nbdep(\epsilon) = 1$ sont rangés comme nouveaux éléments de l'ensemble $Activ2(c,q)$. Simultanément, ces éléments sont retirés de $Activ1(c,q)$.

Le pas 4 est exécuté au cours de chaque $ETAPE(c,q)$. Il consiste à effectuer les opérations 4.1, 4.2, 4.3 et 4.4 ci-dessous pour tout $\epsilon \in Activ2(c,q)$, après chaque application d'ircm.

- 4.1. Soit L l'ensemble de tous les littéraux de $Pgauche(Orig(\epsilon)):Subst(\epsilon)$ qui ont (c,q) pour couple de dépendance dans $Pgauche(Orig(\epsilon)):Subst(\epsilon)$. La fonction INSTANCES est appliquée à L et s'évalue alors à (cf. § 1.3.
- . $\Sigma = \{(\sigma_1 \dots)(\sigma_2 \dots) \dots (\sigma_n \dots)\}$, où $\sigma_1, \sigma_2, \dots, \sigma_n$ sont toutes les substitutions distinctes qui vérifient les trois conditions énoncées pour l'opération 3.1., si une telle substitution au moins existe,
 - . NIL, si un littéral de $L:\sigma$ a la valeur FAUX dans l'état courant, $\forall \sigma$.
 - . ?, dans les autres cas.

- 4.2. Si $INSTANCES(L)$ s'évalue à $\Sigma = \{(\sigma_1 \dots)(\sigma_2 \dots) \dots (\sigma_n \dots)\}$, $n \geq 1$, alors les opérations 4.2.1. et 4.2.2. ci-dessous sont effectuées pour toute substitution σ_i qui figure dans Σ et telle que ϵ n'ait pas déjà un successeur immédiat ϵ_i tel que $Subst(\epsilon_i) = Subst(\epsilon) \cup \sigma_i$.

- 4.2.1. Un sommet ϵ' est ajouté à l'arborescence d'activation qui contient ϵ , comme successeur immédiat de ϵ , et:

4.2.1.1. $\text{Subst}(\epsilon') \leftarrow \text{Subst}(\epsilon) \cup \sigma_i$,

4.2.1.2. ϵ' est déclaré terminal.

4.2.2. L'instance par $\text{Subst}(\epsilon')$ de la contrainte, du conseil ou du diagnostic qui correspond à $\delta = \text{Orig}(\epsilon)$ est activée.

4.3. Si $\text{INSTANCES}(L)$ s'évalue à NIL, alors ϵ est retiré de $\text{Activ2}(c,q)$ et est déclaré abandonné.

4.4. Si $\text{INSTANCES}(L)$ s'évalue à ?, alors aucune opération n'est effectuée.

Exemple:

. Au début de $\text{ETAPE}(\text{BLC1},0)$ (cf. tableau 4.1.), Tropic range E7 dans $\text{Activ2}(\text{BLC1},0)$ et le retire de $\text{Activ1}(\text{BLC1},0)$. On a alors $L = \{(\text{TYPE BLC1 BLOC})(\text{SECTION BLC1 } \$S)\}$. Après l'application de $\text{rem7}: \{(\$BLC BLC1)\}$, $\text{INSTANCES}(L)$ s'évalue à ?. Après l'application de $\text{rem8}: \{(\$BLC BLC1)\}$, $\text{INSTANCES}(L)$ s'évalue à $\{(\{(\$S \text{CARREE}) \dots\})\}$. Un sommet E9 est ajouté à l'arborescence d'activation qui contient E7, comme successeur de E7. La propriété suivante lui est associée: $\text{Subst}(E9) \leftarrow \{(\$TR \text{TR1})(\$COL \text{COL1})(\$TT \text{TT1})(\$BLC \text{BLC1})(\$S \text{CARREE})\}$, et E9 est déclaré terminal. L'instance par $\text{Subst}(E9)$ de la contrainte prise comme exemple est alors activée.

. L'instance par $\{(\$TR \text{TR1})(\$COL \text{COL1})(\$TT \text{TT1})(\$BLC \text{BLC2})(\$S \text{CARREE})\}$ de cette contrainte est activée de la même façon au cours de l'étape $\text{ETAPE}(\text{BLC2},0)$ immédiatement après l'application de $\text{rem8}: \{(\$BLC \text{BLC2})\}$.

1.3. La fonction INSTANCES

La fonction INSTANCES est utilisée par les opérations 3.1 et 4.2. de l'algorithme d'activation. Elle est appliquée à un ensemble $L = \{l_1, l_2, \dots, l_n\}$ de littéraux.

$\text{INSTANCES}(L)$ s'évalue de la façon suivante:

- . S'il existe au moins une substitution σ telle que :
 - a/ tous les littéraux de L : σ aient la valeur VRAI dans l'état courant,
 - b/ toutes les variables qui figurent dans σ apparaissent dans L ,
 - c/ il n'existe pas deux variables différentes utilisées pour noter des C-termes dans L qui ont la même constante pour substitut dans σ ,

alors INSTANCES(L) s'évalue à un ensemble non vide $\Sigma = \{(\sigma_1 \{l_{1,1} l_{1,2} \dots l_{1,m_1}\} e_1) (\sigma_2 \{l_{2,1} l_{2,2} \dots l_{2,m_2}\} e_2) \dots (\sigma_q \{l_{q,1} l_{q,2} \dots l_{q,m_q}\} e_q)\}$, où :

- $\sigma_1, \sigma_2, \dots, \sigma_q$ sont toutes les substitutions distinctes qui vérifient les trois conditions énoncées ci-dessus pour σ ,
- $\forall i \in [1, q], \forall j \in [1, m_i]$, il existe un littéral de $L:\sigma_i$ dont la valeur VRAI dans l'état courant m résulte de la présence de $l_{i,j}$ dans m (cf. commentaire b, § 2.4.3., chapitre 4),
- $\forall i \in [1, q]$, si la valeur VRAI dans m d'un littéral de $L:\sigma_i$ résulte de la présence dans m d'un littéral l , alors il existe $j \in [1, m_i]$ tel que $l_{i,j}=1$,
- $\forall i \in [1, q]$, si la valeur VRAI dans m d'au moins un littéral de $L:\sigma_i$ résulte de l'absence dans m de certains littéraux (cf. commentaire b, § 2.4.3., chapitre 4), alors $e_i=1$, sinon $e_i=0$.

[Note : Nous verrons l'utilisation des différents composants de Σ au paragraphe 2].

. S'il n'existe pas de substitution σ qui vérifie les conditions a/, b/ et c/ énoncées ci-dessus, alors:

- si toute substitution σ vérifiant b/ et c/ est telle qu'au moins un littéral de $L:\sigma$ ait la valeur FAUX dans m , alors INSTANCES(L) s'évalue à NIL,
- s'il existe au moins une substitution σ vérifiant b/ et c/ telle qu'aucun des littéraux de $L:\sigma$ n'ait la valeur FAUX dans m , la valeur d'au moins l'un d'eux étant indéterminée, alors INSTANCES(L) s'évalue à ?.

Exemples:

Considérons le cheminement avant décrit par le tableau 4.1., et plaçons nous immédiatement après l'application de rem6: {(\$COL COL1)(\$N 2)}

(l'exécution de ETAPE(COL1,1) est alors terminée).

. INSTANCES({(COMPOSANT COL1 \$BLC1)(COMPOSANT COL1 \$BLC2)}) s'évalue à:

$$\Sigma = \{(\{(\$BLC1 BLC1) (\$BLC2 BLC2)\} \{(\text{COMPOSANT COL1 BLC1})(\text{COMPOSANT COL1 BLC2})\} 0) (\{(\$BLC1 BLC2) (\$BLC2 BLC1)\} \{(\text{COMPOSANT COL1 BLC1})(\text{COMPOSANT COL1 BLC2})\} 0)\}.$$

. INSTANCES({(COMPOSANT COL1 \$BLC1)(COMPOSANT COL1 \$BLC2)(SUR \$BLC1 \$BLC2)}) s'évalue à:

$$\Sigma = \{(\{(\$BLC1 BLC1) (\$BLC2 BLC2)\} \{(\text{COMPOSANT COL1 BLC1})(\text{COMPOSANT COL1 BLC2})(\text{SUR BLC1 BLC2})\} 0)\}.$$

- . INSTANCES($\{(COMPOSANT COL1 \$BLC1)(COMPOSANT COL1 \$BLC2)(NON SUR \$BLC2 \$BLC1$
s'évalue à :
- $$\Sigma = \{ \{ (\$BLC1 BLC1)(\$BLC2 BLC2) \} \{ (COMPOSANT COL1 BLC1)(COMPOSANT COL1 BLC2) \} 1 \}.$$

2. Mises à jour lors d'un cheminement arrière

2.1. Compléments sur l'algorithme d'activation

Au cours de la génération des arborescences d'activation, l'algorithme d'activation mémorise des informations qui permettent la mise à jour de ces arborescences lors des cheminements arrière. Ces informations, que nous avons ignorées au paragraphe 1, consistent en plusieurs ensembles initialement vides notés Arbor1(l), Arbor2(l), Arbor3(c,q) et Arbor4(c,q). Ces ensembles sont engendrés lors des pas 2, 3 et 4 de l'algorithme d'activation que nous complétons ci-dessous:

- . Nous ajoutons l'opération 2.1.4. suivante au pas 2 de l'algorithme d'activation:

2.1.4. Arbor1(l) \leftarrow Arbor1(l) \cup { ϵ }, où l = (TYPE c a).

- . L'opération 3.2.1. du pas 3 de l'algorithme d'activation est exécutée pour tout élément $(\sigma_i \{l_{i,1} l_{i,2} \dots l_{i,p}\} e_i)$ de l'ensemble Σ obtenu en évaluant INSTANCES(L). Nous lui ajoutons les deux opérations 3.2.1.4. et 3.2.1.5. suivantes:

3.2.1.4. Pour tout $l_{i,j} \in \{l_{i,1} l_{i,2} \dots l_{i,p}\}$, Arbor2($l_{i,j}$) \leftarrow Arbor2($l_{i,j}$) \cup { ϵ' }.

3.2.1.5. Si $e_i = 1$, alors Arbor3(c,q) \leftarrow Arbor3(c,q) \cup { ϵ' }.

- . Nous ajoutons l'opération 4.2.1.3. au pas 4 de l'algorithme d'activation:

4.2.1.3. Arbor4(c,q) \leftarrow Arbor4(c,q) \cup { ϵ' }.

2.2. L'algorithme des mises à jour lors d'un cheminement arrière

Supposons que Tropic subisse un échec pendant l'exécution de l'étape $e_p = \text{ETAPE}(c_p, q_p)$ de la stratégie hiérarchique (cas 1, 3, 4, 5 et 6 d'échec), ou à la fin de l'exécution de e_p (cas 2 et 7 d'échec):

- . Soient $e_1, e_2, \dots, e_N, \dots, e_{P-1}$ les étapes successivement exécutées avant e_P ; $\forall i \in [1, P-1], e_i = \text{ETAPE}(c_i, q_i)$.
- . Soient (c_N, q_N) le couple de dépendance du problème formel qui est associé au sommet-ET choisi pour point de retour ; $N \in [1, P]$.
- . Soit $L = \{l_1, l_2, \dots, l_n\}$ l'ensemble des littéraux qui sont retirés de l'état courant lors du cheminement arrière.

Tropic met à jour les arborescences d'activation en exécutant l'algorithme suivant:

1. Pour tout ϵ tel que $\epsilon \in \text{Arbor1}(l)$, où $l \in L$ et $l = (\text{TYPE} \dots)$,
 - 1.1. L'arborescence d'activation qui a ϵ pour racine est supprimée,
 - 1.2. Chaque sommet ϵ' supprimé est aussitôt retiré des ensembles $\text{Activ1}, \text{Activ2}, \text{Arbor1}, \text{Arbor2}, \text{Arbor3}$ et Arbor4 où il figure,
 - 1.3. Pour chaque sommet terminal ϵ' supprimé, l'instance par $\text{Subst}(\epsilon')$ de la règle de production correspondant à $\text{Orig}(\epsilon')$ est aussitôt désactivée.

[Commentaire: On supprime ainsi toutes les arborescences d'activation dont les racines ont été engendrées à la suite de l'ajout à l'état courant des littéraux de la forme $(\text{TYPE} \dots)$ qui figurent dans L].

2. Pour tout ϵ' vérifiant l'une des trois conditions suivantes:

- (a) $\epsilon' \in \text{Arbor2}(l)$, où $l \in L$,
- (b) $\epsilon' \in \text{Arbor3}(c, q)$, tel qu'il existe $i \in [N, P]$ pour lequel $(c_i, q_i) = (c, q)$,
- (c) $\epsilon' \in \text{Arbor4}(c, q)$, tel qu'il existe $i \in [N, P]$ pour lequel $(c_i, q_i) = (c, q)$.

- 2.1. Le prédécesseur immédiat ϵ de ϵ' dans l'arborescence d'activation qui contient ϵ' est rangé comme nouvel élément d'un ensemble noté Réactiv qui est vide au début de l'exécution de :
- 2.2. L'arc qui part de ϵ et qui arrive à ϵ' , le sommet ϵ' et la partie de l'arborescence d'activation qui est issue de ϵ' sont supprimés.

2.3. Chaque sommet ϵ'' supprimé est aussitôt retiré des ensembles Activ1, Activ2, Arbor1, Arbor2, Arbor3, Arbor4 et Réactiv où il figure.

2.4. Pour chaque sommet terminal ϵ'' supprimé, l'instance par Subst(ϵ'') de la règle de production correspondant à Orig(ϵ'') est aussitôt désactivée.

[Commentaire :

- . Les opérations 2.1, 2.2, 2.3 et 2.4 pour tous les ϵ' qui vérifient la condition (a) ont pour effet d'effacer le travail d'activation qui dérivait de la présence des littéraux de L dans l'état courant avant le cheminement arrière.
- . Les opérations 2.1, 2.2, 2.3 et 2.4 pour tous les ϵ' qui vérifient la condition (b) ont pour effet d'effacer le travail d'activation qui dérivait de l'absence de certains littéraux dans l'état courant à la fin de l'exécution des étapes e_N, e_{N+1}, \dots, e_P (ces étapes ne sont plus considérées comme étant exécutées, (cf. § 3.3 du chapitre 6).
- . Les opérations 2.1, 2.2, 2.3 et 2.4 pour tous les ϵ' qui vérifient la condition (c) ont pour effet de désactiver les instances de règles de production qui avaient été activées au cours des étapes e_N, e_{N+1}, \dots, e_P . Toutefois le travail d'activation de ces instances n'est pas perdu].

3. Pour tout $\epsilon \in \text{Réactiv}$:

3.1. Si $N_{\text{dbep}}(\epsilon) > 1$, alors pour tout couple (c q) tel que :

- . il existedans Pgauche (Orig(ϵ)):Subst(ϵ) au moins un littéral ayant (c q) pour couple de dépendance dans la partie non masquée de l'état courant,
- . (c q) $\notin \{(c_1 q_1)(c_2 q_2) \dots (c_{N-1} q_{N-1})\}$,
 ϵ est rangé comme nouvel élément de l'ensemble Actic1(c,q).

3.2. Si $N_{\text{dbep}}(\epsilon) = 1$, alors soit (c q) le couple de dépendance qui vérifie les deux conditions énoncées dans l'opération 3.1.

3.2.1. Si (c q) = $(c_N q_N)$, alors ϵ est rangé comme nouvel élément de Activ2(c_N, q_N).

3.2.2. Si (c q) $\neq (c_N q_N)$, alors ϵ est rangé comme nouvel élément de Activ1(c,q).

4. Réactiv \leftarrow NIL.

Exemple :

Reprenons l'exemple utilisé pour illustrer l'algorithme d'activation. A la fin de $ETAPE(BLC2,0)$ on a :

- . $Arbor1((TYPE\ TR1\ TOUR)) = \{\dots E_1 \dots\}$,
- . $Arbor2((COULEUR\ TT1\ ROUGE)) = \{\dots E_6 \dots\}$,
- . $Arbor4(BLC1,0) = \{\dots E_9 \dots\}$,
- . $Arbor4(BLC2,0) = \{\dots E_{10} \dots\}$.

- Si Tropic effectue alors un cheminement arrière au sommet-ET auquel est associé le problème (DETERMINER-FORME-TOIT TT1), seuls les sommets E_9 et E_{10} de l'arborescence d'activation de la figure 7.1 sont effacés.
- Si Tropic effectue un cheminement arrière au sommet-ET auquel est associé le problème (DETERMINER-COULEUR-TOIT TT2), le sommet E_6 et tous ses successeurs sont effacés.

2.3. Compléments sur l'algorithme de mises à jour.

a/ Pendant l'exécution de l'algorithme des mises à jour, certains successeurs immédiats d'un sommet ϵ d'une arborescence d'activation peuvent être effacés, et d'autres non. Pour assurer une reprise correcte de l'algorithme d'activation (§ 1.2), nous modifions le début de l'opération 3.2 du pas 3 de cet algorithme qui devient :

3.2. Les opérations 3.2.1. et 3.2.2. ci-dessous sont effectuées pour tout élément $(\sigma_i \dots)$ de Σ , tel que ϵ n'ait pas déjà un successeur immédiat σ_i pour lequel $Subst(\epsilon_i) = Subst(\epsilon) \cup \sigma_i \dots$

b/ A la suite d'un cheminement arrière, on doit reprendre en considération des sommets des arborescences d'activation qui ont été déclarés abandonnés. Pour ce faire, nous apportons les modifications suivantes aux algorithmes précédents :

- . Lorsqu'un sommet est déclaré abandonné (opérations 3.3 et 4.3 de l'algorithme d'activation), ce sommet est rangé comme nouvel élément d'un ensemble noté $Abandon(c,q)$.

- . Le début de l'opération 3 de l'algorithme du paragraphe 2.2 devient :
 - . Réactiv \leftarrow Réactiv \cup Abandon(c_N, q_N) \cup Abandon(c_{N+1}, q_{N+1}) $\cup \dots$
 \cup Abandon(c_P, q_P).
 - . Pour $i = N$ à P , Abandon(c_i, q_i) \leftarrow NIL.
 - . Pour tout $\varepsilon \in$ Réactiv:
 - 3.1....

c/ A cause de la condition (c), l'opération 2 de l'algorithme du paragraphe 2.2 peut désactiver certaines instances de règles de production dont les conditions d'activation restent satisfaites par les littéraux non masqués de l'état courant après le cheminement arrière. Aussi, avant de reprendre le cheminement avant Tropic exécute le pas 4 de l'algorithme d'activation, avec $c=c_N$ et $q=q_N$. Cette exécution lui permet de réactiver ces instances.

d/ Supposons que pendant le cheminement avant (après un échec), Tropic soit conduit à effacer une partie de l'acm (cf. opération 2.2.3 de l'algorithme du § 3.4, chapitre 6). Nous avons dit qu'il exécute les opérations 1 et 3 de l'algorithme du cheminement arrière (§ 3.3, chapitre 6). L'opération 3 consiste alors à exécuter l'algorithme du paragraphe 2.2 précédent avec les précisions suivantes :

- . L est l'ensemble des littéraux (masqués) qui viennent d'être retirés de l'état courant ;
- . l'opération 2 est exécutée seulement pour les ε' qui vérifient la condition (a) ;
- . c_N et q_N sont tels que $ETAPE(c_N, q_N)$ soit l'étape de la stratégie hiérarchique en cours d'exécution.

Chapitre 8 : IMPLANTATION ET EXPERIMENTATION DE TROPIC ; QUELQUES NOUVELLES DIRECTIONS DE RECHERCHE

Nous avons réalisé une implantation de Tropic qui inclut à peu de détails près tous les aspects décrits dans les chapitres précédents. Cette implantation est opérationnelle sur l'ordinateur IBM 360/67 du CIGC sous le système conversationnel CP/CMS et est écrite en LISP/CMS, une version du langage LISP développée par Lux (1975). Nous l'avons expérimentée sur différents problèmes.

Dans ce chapitre, nous décrivons cette implantation (§ 1) et son expérimentation (§ 2). Nous présentons ensuite quelques nouvelles directions de recherche (§ 3) suggérées par l'expérience obtenue sur cette implantation.

1. Implantation de Tropic

1.1. Historique

L'implantation de Tropic sur IBM 360/67 est dérivée de deux implantations que nous avons réalisées sur l'ordinateur DEC PDP-10 du Centre d'Intelligence Artificielle du SRI (Stanford Research Institute) sous le système conversationnel TENEX.

1.1.1. L'écriture de Tropic dans le langage QLISP

La première implantation de Tropic a été écrite dans le langage QLISP (Wilber 1976, Sacerdoti 1976). Ce langage, développé au SRI sur DEC PDP-10, est le dernier né d'une génération de langages de très haut niveau initialisée par PLANNER (Hewitt 1972). Il offre plusieurs facilités de programmation qui sont particulièrement utiles pour implanter des applications d'intelligence artificielle, et parmi lesquelles figurent notamment des types de données variés, un appel par "pattern" des fonctions ("pattern-directed function invocation"), un mécanisme de cheminement arrière automatique et une base de données associative décomposable en contextes. Ces facilités nous ont permis de mettre en oeuvre un "micro-Tropic" dans un délais relativement court (environ un mois).

Bien que n'incluant pas de nombreuses composantes de l'implantation actuelle (traitement de données numériques, traitement non aveugle des échecs

heuristiques spécialisées), cette première implantation a été un outil déterminant pour valider ou élaborer plusieurs des concepts décrits dans les chapitres précédents (traitement des échecs notamment).

1.1.2. Le passage de QLISP à INTERLISP

Deux inconvénients de QLISP nous ont ensuite conduits à abandonner ce langage :

- . L'exécution des programmes écrits en QLISP est assez lente. Au début, cet inconvénient a été mineur en regard des facilités de programmation offertes par le langage. Mais, par la suite, les temps de réponse sont devenus trop longs pour pouvoir poursuivre un travail efficace.
- . QLISP offre en principe les moyens de contrôler le mécanisme de cheminement arrière qu'il incorpore. Toutefois, nous avons trouvé difficile, sinon impossible, d'utiliser ces moyens pour implanter le traitement des échecs décrit au chapitre 6.

QLISP étant une extension d'INTERLISP (Teitelman 1974), une version de LISP développée sur DEC PDP-10 et abondamment utilisée en Intelligence Artificielle, nous avons pu passer de l'implantation de Tropic en QLISP à une nouvelle implantation en INTERLISP, sans avoir à réécrire complètement les programmes de la première implantation. Pour ce faire, nous avons écrit en INTERLISP certaines fonctions fournies pas QLISP après les avoir simplifiées en restreignant leurs domaines d'application aux seuls cas utiles dans Tropic. Nous avons aussi ajouté à la nouvelle implantation la possibilité de fournir des conseils (cf. § 3.1, chapitre 4), le traitement des données numériques décrit au chapitre 5 et un traitement des échecs semblable à celui du chapitre 6, qui n'existaient pas dans l'implantation en QLISP.

Cette deuxième implantation de Tropic a été expérimentée sur des problèmes simples (conception de tours), et a fait l'objet de la première publication sur Tropic (Latombe 1976a).

1.1.3. Le transport des programmes sur IBM 360/67

L'implantation de Tropic sur IBM 360/67 a été réalisée en transportant sur cet ordinateur les programmes en INTERLISP de la deuxième implantation. Grâce à l'aide d'A. Lux, un travail d'environ 15 jours a suffi pour effectuer l'adaptation de ces programmes (qui utilisaient peu de fonctions spécifiques à INTERLISP) à LISP/CMS.

Ensuite, nous avons complété cette implantation en ajoutant la possibilité de fournir des diagnostics, en améliorant le traitement des échecs et en simplifiant la syntaxe des rcm, des contraintes et des conseils.

1.2. Description générale des programmes

L'implantation de Tropic sur IBM 360/67 comporte environ 4500 lignes de LISP réparties en approximativement 250 fonctions. Ces fonctions sont regroupées en 5 sous-ensembles fonctionnels montrés à la figure 1.2 :

- . le coordonnateur (application de la stratégie hiérarchique) représente environ 10 % des lignes de LISP écrites,
- . le prp-symbolique (choix et application des ircm, construction de l'acm) en représente 20 %,
- . le prp-numérique (construction du grn, vérification de la compatibilité des restrictions numériques) en représente 20 %,
- . le programme de traitement des échecs (formation des expressions causales, choix des points de retour, cheminement arrière) en représente 30 %,
- . le programme d'activation des connaissances (lecture et activation des rcm, des contraintes, des conseils et des diagnostics) en représente 20 %.

Compilée, cette implantation occupe 80 Koctets. Son utilisation requiert une machine virtuelle de 512 Koctets (au moins) de mémoire centrale.

1.3. Utilisation de l'implantation de Tropic sur IBM 360/67

On entre sous CMS la commande TROPIC(f), où f est le nom d'un fichier qui contient une séquence d'instructions de la forme (RCM C ==> D), (CNTR C ==> D), (CNSL C ==> D), (DIAG C ==> D), (PB-INIT p₀) ou (STRUCT p v) :

- . (RCM C ==> D) indique que (C ==> D) est une rcm du problème à résoudre,
- . (CNTR C ==> D) indique que (C ==> D) est une contrainte,
- . (CNSL C ==> D) indique que (C ==> D) est un conseil,
- . (DIAG C ==> D) indique que (C ==> D) est un diagnostic,
- . (PB-INIT p₀) indique que p₀ est le problème initial,
- . (STRUCT p v) indique que v est la valeur de la propriété structurante de p.

La commande TROPIC(f) charge en mémoire centrale les programmes qui sont nécessaires au fonctionnement de Tropic, puis lance leur exécution

avec f comme argument. Une trace de la résolution du problème décrit dans f est alors imprimée sur le terminal de l'utilisateur.

Commentaires

- . Le fichier f est créé et mis à jour à l'aide de l'éditeur intégré à LISP/CMS. Cet éditeur offre sensiblement les mêmes facilités de mises à jour que l'éditeur de CMS.
- . Des fonctions de LISP/CMS, dites de "jolie impression" (Lux 1975), impriment le contenu du fichier f en mettant en évidence sa structure arborescente (cf. Appendice A, § 1).
- . Aucun ordre n'est imposé sur les instructions contenues dans le fichier f .
- . RCM, CNTR, CNSL, DIAG, PB-INIT et STRUCT sont des fonctions de Tropic. Leur évaluation, à la lecture du fichier f , crée les structures de données sur lesquelles travailleront d'autres programmes de Tropic pendant la résolution du problème.
- . Si la partie droite d'une rcm, d'une contrainte, d'un conseil ou d'un diagnostic est une fonction, celle-ci doit être écrite en LISP. Toutefois, plusieurs fonctions et macros définies dans Tropic facilitent son écriture (cf. Appendice A, § 1).
- . On peut faire appel à Tropic en entrant la commande $TROPIC(f_1 f_2 \dots f_n)$, où f_1, f_2, \dots, f_n sont des noms de fichiers. Cette commande est équivalente à la commande $TROPIC(f)$, où f serait le fichier obtenu en concaténant les fichiers f_1, f_2, \dots, f_n .

2. Expérimentation de Tropic

Nous avons utilisé Tropic pour résoudre des problèmes de conception artificiels (dérivés de l'exemple de la tour qui illustre les chapitres précédents) pour lesquels le volume des informations à décrire (fichier f) est assez petit. Nous avons imaginé ces problèmes de façon à pouvoir expérimenter les différents programmes de Tropic et vérifier leur bon fonctionnement. Dans l'appendice A, nous montrons le fichier f et la trace de la résolution imprimée par Tropic, pour le problème utilisé comme exemple au paragraphe 1.2.2 du chapitre 6 (à titre indicatif, il faut environ 35 secondes d'unité centrale pour exécuter la commande $TROPIC(f)$ correspondant à ce problème).

Bien que certains problèmes ainsi imaginés ne soient pas triviaux (pour l'un d'eux, les rcm permettant de construire plus de 30000 modèles différents, parmi lesquels un seul est une solution), leur "complexité" est très inférieure à celles de problèmes de conception réels. C'est pourquoi nous avons aussi utilisé Tropic pour concevoir des transformateurs triphasés de moyenne puissance (entre 10 et 2000 KVA). L'intérêt de cet exemple réside principalement dans les deux points suivants :

- . sa complexité est suffisante pour permettre de vérifier l'applicabilité de Tropic à des problèmes de conception réels : à titre indicatif, un élève-ingénieur ayant reçu une formation théorique et pratique en Electrotechnique, et utilisant le système ESPACE (dans lequel sont programmés les calculs d'analyse) a besoin d'une vingtaine d'heures pour concevoir un tel transformateur (déduction faite du temps nécessaire à l'apprentissage du système) (Bolopion 1975),
- . il met en jeu plusieurs disciplines (électricité, électromagnétisme, thermodynamique, mécanique) et est ainsi caractéristique d'une classe importante de problème de conception.

Nous décrivons cet exemple au paragraphe 2.1. Nous discutons ensuite les aspects positifs et négatifs de Tropic mis en évidence par cette expérimentation.

2.1. L'exemple du transformateur

2.1.1. La description des problèmes

Nous avons soumis à Tropic plusieurs problèmes de conception de transformateur. La description de chacun de ces problèmes comporte environ 230 règles de production (rcm, contraintes, conseils et diagnostics) qui occupent approximativement 25 Koctets de mémoire. Toutefois, seulement 1,5 % de cette description est spécifique du problème, le reste étant commun à tous les problèmes que nous avons soumis.

a/ Les rcm

Le même ensemble de rcm est commun à tous les problèmes de conception de transformateur que nous avons soumis à Tropic. Ces rcm sont énoncées dans l'appendice B. Elles permettent de construire les modèles d'un grand nombre

de transformateurs de moyenne puissance aujourd'hui en fonctionnement. Bien qu'elles ne permettent qu'une seule forme de circuit magnétique (trois noyaux et deux culasses dans le même plan), qu'une seule disposition d'enroulement (disposition concentrique) et qu'une seule forme de cuve (cuve à ailettes à circulation naturelle), on pourrait sans difficulté majeure éviter ces limitations en ajoutant de nouvelles rcm.

b/ Les contraintes

Nous avons regroupé les contraintes en trois sous-ensembles contenant respectivement :

- Les contraintes qui imposent les restrictions spécifiques du problème à résoudre, par exemple : la puissance apparente du transformateur doit être égale à 700KVA, sa fréquence à 50 Hz, sa tension primaire à 5,5 KV, sa tension secondaire à 1,2 KV, ...

[Ces contraintes forment la seule partie de la description d'un problème qui soit spécifique de ce problème.]

- Les contraintes qui imposent des restrictions symboliques et/ou numériques déclaratives non spécifiques d'un problème, par exemple :

- . si l'huile est le réfrigérant, son échauffement maximal doit être inférieur à 45°C,
- . si les tôles du circuit magnétique sont de qualité 1, l'induction magnétique dans les noyaux doit être comprise entre 1,55 T et 1,7 T ; si elles sont de qualité 2, l'induction dans les noyaux doit être comprise entre 1,5 et 1,6 T [les qualités 1 et 2 de tôles sont fictives et ne correspondent à aucune norme en vigueur],
- . la tension aux bornes d'une galette doit être inférieure à 8 KV,
- . si l'huile est le réfrigérant, l'épaisseur des canaux de refroidissement entre empilage et circuit magnétique et entre empilages doit être comprise entre 5 mm et 10 mm ; si l'air est le réfrigérant, l'épaisseur de ces canaux doit être comprise entre 7 et 13 mm.

- Les contraintes qui imposent les restrictions procédurales liant par exemple :

- l'intensité à vide (en pour cent de l'intensité nominale) $I_0\%$ à la puissance apparente P du transformateur, aux pertes fer P_{fer} dans le circuit magnétique et à la puissance magnétisante Q_{fer} dans ce circuit ($I_0\% = 100 \frac{\sqrt{P_{fer}^2 + Q_{fer}^2}}{P}$),
- le rendement ρ du transformateur à la puissance secondaire P_2 et aux pertes totales p dans le transformateur ($\rho = \frac{P_2}{P_2 + p}$),
- l'échauffement superficiel $\Delta\theta$ du réfrigérant aux pertes totales p dans le transformateur et à la surface utile S_u de la cuve ($\Delta\theta = 445(\frac{p}{S_u})^{0,8}$).

c/ Les conseils

Nous avons fourni à Tropic un nombre assez important de conseils. Ci-dessous, nous exprimons en français quelques uns d'entre eux :

- Si la puissance apparente du transformateur à concevoir est supérieure à 50 KVA, il est recommandé de choisir l'huile pour réfrigérant.
- Si la puissance apparente du transformateur à concevoir est comprise entre 10 et 100 KVA, il est recommandé de choisir un nombre de gradins égal à 3 pour les noyaux du circuit magnétique ; si la puissance apparente est comprise entre 100 et 500 KVA, il est recommandé de choisir ce nombre égal à 4 ; si la puissance apparente est comprise entre 500 et 1000 KVA, il est recommandé de le choisir égal à 5 ; si la puissance apparente est supérieur à 1000 KVA, il est recommandé de le choisir égal à 6.
- Dès qu'un conducteur est utilisé, demander sa forme en conseil (cf. commentaire b, § 3.1, chapitre 3) ; si la forme méplat est conseillée, demander alors en conseil le nombre de brins axiaux et le nombre de brins radiaux du conducteur, ainsi que l'épaisseur et la hauteur du brin nu.

d/ Les diagnostics

Nous avons fourni à Tropic un seul diagnostic que nous énonçons ci-dessous dans le formalisme des chapitres précédents :

```

{{{(TYPE $TRSF TRANSFORMATEUR)}}}
→ Fonct: DEMANDER(DIAGNOSTIC ?))

```


L'instance par $\{(\$TRSF TRSF1)\}$ de ce diagnostic devient active dès le début de la résolution d'un problème, car (cf. la première rcm de l'appendice B) la décomposition du problème formel (CONCEVOIR-TRANSFORMATEUR) a alors pour effet d'ajouter le littéral (TYPE TRSF1 TRANSFORMATEUR) à l'état courant. Tropic évaluera donc la partie droite de cette instance de diagnostic chaque fois qu'il subira un échec résultant de l'incompatibilité de restrictions numériques actives (cf. § 3.1.2.c, chapitre 6). L'effet de cette évaluation sera l'impression de la question "DIAGNOSTIC ?" sur le terminal ; son résultat sera la réponse fournie. Celle-ci devra être un ensemble de littéraux . si cet ensemble est vide, alors Tropic choisira un point de retour à l'aide d'une expression causale formée comme nous l'avons indiqué au paragraphe 3.1.1.g du chapitre 6, . si cet ensemble n'est pas vide, alors Tropic choisira un point de retour à l'aide de l'expression causale qui est déterminée par cet ensemble (cf. § 3.1.2.c, chapitre 6).

Le diagnostic énoncé ci-dessus nous a ainsi permis de participer interactivement et de façon importante au traitement de tous les échecs résultant de l'incompatibilité de restrictions numériques actives, sans qu'il ait été nécessaire de les prévoir.

2.1.2. La résolution des problèmes

Nous avons utilisé Tropic pour résoudre complètement deux problèmes de conception de transformateur :

- . pour un problème, il s'agissait de concevoir un transformateur ayant une puissance apparente de 700 KVA, une fréquence de 50 Hz, des montages primaire et secondaire en étoile, une tension primaire de 5,5 KV et une tension secondaire de 1,2 KV pour $\cos\phi = 0,8$.
- . pour l'autre problème, il s'agissait de concevoir un transformateur ayant une puissance apparente de 200 KVA, une fréquence de 60 Hz, un montage primaire en triangle, un montage secondaire en étoile, une tension primaire de 10 KV et une tension secondaire à vide de 0,23 KV.

Nous avons soumis à Tropic d'autres problèmes dérivés de ces deux problèmes principalement pour vérifier certains aspects du fonctionnement de Tropic, mais nous ne les avons pas traités complètement.

Les deux problèmes traités avaient déjà été proposés en projet à des élèves de 2ème année de l'ENSEGP*. Grâce aux comptes rendus de ces projets, et par l'intermédiaire des conseils et du diagnostic fournis à Tropic, nous avons pu contrôler les choix importants de leur résolution. Les solutions ont été engendrées après avoir rencontré peu d'échecs (4 à 5 dans chaque cas), correspondant pour la plupart aux difficultés classiques de la conception d'un transformateur (tension de court-circuit trop élevée et/ou échauffement dans les galettes trop grand). Grâce au diagnostic, nous avons corrigé ces échecs en modifiant suivant les cas les dimensions des conducteurs des galettes primaires et secondaires, la composition des cylindres isolants entre les empilages. Nous avons aussi causé volontairement d'autres échecs en conseillant une qualité de tôle ou un réfrigérant inapproprié.

La résolution de chaque problème a nécessité entre 10 et 20 mn d'unité centrale (nous avons résolu chaque problème plusieurs fois avec des conseils différents). La durée du travail sur le terminal (un téléimprimeur IBM 2741) a été de l'ordre de 3 h pour chaque problème. Cette durée pourrait être assez nettement réduite en utilisant un terminal plus rapide (écran de visualisation par exemple) car la trace de la résolution comporte environ 3000 lignes.

2.2. Les aspects positifs et les aspects négatifs mis en évidence par cette expérimentation.

Nous discutons ci-dessous les aspects positifs et les aspects négatifs mis en évidence par cette expérimentation, suivant trois points de vue.

2.2.1. Le fonctionnement de Tropic

L'expérimentation de Tropic a mis en évidence le bon fonctionnement des programmes de Tropic avec des temps d'exécution raisonnables. Ces temps pourraient toutefois être réduits (dans un rapport que nous estimons à 2 ou 3 en fonction de notre expérience) en réécrivant avec plus de soin certains programmes fréquemment exécutés ; en effet :

- . Le temps d'exécution d'un programme écrit en LISP est particulièrement sensible aux maladroesses de programmation. De plus, celles-ci peuvent aussi

* Ecole Nationale Supérieure d'Electrotechnique et de Génie Physique.

causer inutilement la génération de listes qui nécessitent l'exécution plus fréquente du "ramasse-miettes" ("garbage collector").

- . Les implantations successives de Tropic sur DEC PDP-10 et sur IBM 360/67 nous ont conduit à ajouter plusieurs programmes d'interface fréquemment exécutés. Les parties correspondantes de Tropic pourraient être simplifiées.
- . Nous avons consacré relativement peu d'effort à l'écriture du programme d'optimisation numérique. Or, dans l'exemple du transformateur, ce programme est responsable d'environ 40 % du temps d'unité centrale.

L'aspect négatif le plus important mis en évidence concerne la place de mémoire qui est nécessaire au fonctionnement de Tropic. En effet, nous avons dû travailler sur une machine virtuelle ayant 1024 Koctets de mémoire centrale pour mettre en oeuvre l'exemple du transformateur. Le programme d'activation des connaissances est le principal utilisateur de cette mémoire à cause des nombreuses arborescences d'activation qu'il engendre. Une écriture plus astucieuse de ce programme permettrait sans doute de réduire la place de mémoire qu'il utilise. Toutefois, nous pensons qu'un gain substantiel de place ne pourra être obtenue qu'en améliorant l'algorithme du chapitre 7. Ainsi, dans l'exemple du transformateur, il est possible de former plusieurs sous-ensemble de contraintes, conseils et diagnostics dont les conditions d'activation diffèrent assez peu. La mise en commun d'une partie du travail d'activation et des arborescences d'activation pour ces contraintes, conseils et diagnostics résulterait probablement en une réduction importante de la place de mémoire utilisée.

2.2.2. La description des problèmes

A l'exception de quelques défauts que nous énumérerons plus loin, nous avons trouvé commode l'utilisation des rcm et des contraintes pour énoncer les problèmes de conception que nous avons soumis à Tropic. Par ailleurs, en nous permettant de contrôler certaines décisions, les conseils et les diagnostics nous ont fourni un moyen souple de participer à leur résolution sans avoir à envisager par avance une méthode complète de résolution. Ainsi, les rcm, les contraintes, les conseils et les diagnostics constituent les éléments d'un langage de programmation nouveau en CAO et convenant bien à la résolution d'une classe de problèmes de conception. De plus, l'expression de ces éléments sous la forme de règles de production encourage un style de travail graduel de la part du concepteur.

A côté de ces aspects positifs, nous avons relevé les aspects négatifs suivants :

- Les programmes de détection d'erreurs de Tropic sont nettement insuffisants. Dans le futur, il sera nécessaire de compléter ces programmes car leurs insuffisances peuvent entraîner des pertes de temps sérieuses pour l'utilisateur du système.

- Certaines contraintes peuvent être assez difficiles à exprimer. C'est par exemple le cas de la contrainte "les hauteurs des empilages d'un enroulement concentrique doivent être égales à 1 % près", que nous avons exprimée sous la forme suivante :

```
((TYPE $ENR ENROULEMENT)(DISPOSITION $ENR CONCENTRIQUE)(COMPOSANT $ENR $EMP1)
  (COMPOSANT $ENR $EMP2)(TYPE $EMP1 EMPILAGE)(TYPE $EMP2 EMPILAGE))
```

```
→ (((DIFF-HAUTEURS-EMP% $ENR) ← {f (HAUTEUR $EMP1 $EMP2)}))
```

```
(PP DIFF-HAUTEURS-EMP% $ENR 1))
```

où $f(x y)$ s'évalue comme $100 \frac{|x - y|}{x}$.

- La structure hiérarchique des parties gauches des contraintes, conseils et diagnostics, qui résulte de la propriété imposée au paragraphe 1.1.a du chapitre 7, entraîne parfois une écriture assez lourde de ces règles de production (cf. par exemple le dernier conseil énoncé à la fin du § 1 de l'appendice A).

- La fonction %ECHEC définie au paragraphe 3.1.2.b du chapitre 6 est souvent insuffisante pour caractériser précisément l'incompatibilité de restrictions numériques.

2.2.3. La résolution des problèmes

Grâce aux heuristiques spécialisées (conseils et diagnostics) et aux possibilités d'interaction, nous avons pu utiliser Tropic pour résoudre des problèmes de conception de transformateurs, vérifiant ainsi son applicabilité pratique en tant que système de CAO. Toutefois, il est clairement apparu que Tropic n'est pas capable de résoudre des problèmes de conception réels dans des temps raisonnables, s'il travaille de façon complètement autonome, c'est-à-dire sans conseil, ni diagnostic, ni interaction avec le concepteur (notre ambition initiale n'était d'ailleurs pas d'atteindre un tel objectif) ; les seuls problèmes qu'il ait pu résoudre ainsi sont des

problèmes de conception de tour simples.

L'incapacité de Tropic à résoudre seul des problèmes de conception réels réside principalement dans le fait qu'il ne "voit" pas (ou très peu) l'interdépendance des données symboliques et des données numériques susceptibles de figurer dans un modèle. Les conseils et les diagnostics permettent de combler cette lacune, mais d'autres solutions peuvent être envisagées (cf. § 3.3).

L'aspect le plus positif mis en évidence par l'expérimentation de Tropic concerne la représentation hybride utilisée par Tropic pour mémoriser ses décisions (à l'exception du défaut énoncé plus loin). Cette représentation permet d'effectuer des modifications réduites à la suite des échecs, et il est en général satisfaisant d'observer les parties correspondantes de la trace de résolution imprimée par Tropic (cf. § 2, appendice A). Cette qualité de Tropic est essentielle pour résoudre des problèmes de conception réels, car ces problèmes mettent en jeu un volume considérable d'information.

L'expérimentation de Tropic a aussi montré les aspects négatifs suivants :

- Des rcm maladroitement peuvent conduire Tropic à effacer inutilement certaines parties de son travail à la suite d'un échec. C'est par exemple le cas de la rcm de l'appendice B qui décompose le problème formel (ENGENDRER-COMPOSANTS-TRANSFO \$TRSF \$REF). Cette rcm conduit Tropic à effacer inutilement la décomposition des problèmes (CONCEVOIR-CIRC-MAGNETIQUE \$CM) et (CONCEVOIR-BOBINAGE \$BOB \$REF), lorsqu'il modifie le réfrigérant du transformateur.

- La méthode de Hookes et Jeeves utilisées pour vérifier la compatibilité des restrictions numériques n'est pas assez fiable. En se bloquant sur une arrête, elle a été responsable de l'abandon par Tropic de la résolution d'un problème qui pourtant avait une solution.

3. Quelques nouvelles directions de recherche

Nous énumérons ci-dessous quelques nouvelles directions de recherche sur Tropic, possibles ou en cours.

3.1. Réalisation d'un interface de communication en langue naturelle avec Tropic

Jusqu'à présent, les communications en langue naturelle n'ont fait que très peu l'objet d'études ou de réalisations en CAO. Nous pensons que Tropic offre une occasion propice à une recherche dans ce sens.

M. Lopez (Lopez 1976) a entrepris la réalisation d'un interface permettant d'exprimer la description de problèmes dans un sous-ensemble du français. Il utilise le système PIAF (Grandjean 1975, Courtin 1977) pour effectuer l'analyse morphologique et syntaxique des règles de production énoncées en français, et transforme les arborescences qui résultent de cette analyse en "figures" (petits réseaux) qu'il intègre une à une à un réseau sémantique "partitionné" (Hendrix 1975). Ce réseau sémantique lui permet de conserver les avantages de la modularité des règles de production, tout en éliminant certains de leurs inconvénients : les parties communes des règles de production n'y sont représentées qu'une fois, ce qui permet par exemple d'avoir à chaque instant un accès direct à l'ensemble des formes possibles d'un toit (ce qui est impossible lorsque les règles de production sont mémorisées séparément).

La mise en oeuvre de communications en langue naturelle avec Tropic est un objectif ambitieux, car il est difficile de prévoir la nature et la difficulté des ambiguïtés qu'il sera nécessaire de traiter. Ce travail peut toutefois contribuer fortement à la définition d'un langage formel mieux adapté au concepteur que le langage actuel, et à la définition d'un nouveau modèle de représentation des connaissances pour Tropic.

3.2. Mise en oeuvre d'autres modèles de résolution de problème

A un modèle de résolution de problème correspond une classe de problèmes qu'il est possible de résoudre de façon satisfaisante suivant ce modèle. On pourrait donc étendre le domaine d'application de Tropic en mettant en oeuvre de nouveaux modèles de résolution de problème. Les travaux de l'équipe de G.J. Sussman (Laboratoire d'Intelligence Artificielle du MIT) nous conduisent à envisager les modèles de "propagation de contraintes (initialement étudié par Waltz (1972)) et de "mise au point de plans presque corrects" (aussi suggéré dans Westerberger (1974)) comme des candidats possibles.

3.3. Définition d'un modèle de représentation des relations de causes à effets.

Nous avons vu (cf. § 2.2.3) qu'une lacune importante de Tropic est de ne pas "voir" l'interdépendance des données symboliques et numériques susceptibles de figurer dans un modèle. Actuellement, cette lacune est comblée par les conseils et les diagnostics fournis par le concepteur. Toutefois nous pensons qu'en nous inspirant des travaux de l'équipe de C. Rieger (Rieger 1977), il serait possible de définir une représentation dans laquelle on pourrait décrire explicitement les relations de causes à effets entre ces données, de telle sorte que Tropic puisse en tenir compte lors de ses décisions.

3.4. Définition d'un modèle d'apprentissage.

Considérons à nouveau l'exemple du paragraphe 1.2.2 du chapitre 6. Dans cet exemple, Tropic subit deux échecs (§ 1.2.2.b et d) avant d'obtenir une solution du problème qui lui est soumis. Lors de la détermination du deuxième point de retour, il associe l'indicateur d'échec $G = \{(TYPE\ TR1\ TOUR\ (COMPOSANT\ TR1\ COL1)(COMPOSANT\ TR1\ TT1)(TYPE\ COL1\ COLONNE)(TYPE\ TT1\ TOIT)\ (NOMBRE-DE-BLOCS\ COL1\ 2)\}$ au sommet-OU qui correspond à $rcm14:\{(\$TT\ TT1)\}$ (cf. figure 6.9). Cet indicateur d'échec signifie que l'on est certain que l'application de $rcm14:\{(\$TT\ TT1)\}$ conduit à un échec lorsque tous les littéraux contenus dans G ont la valeur VRAI dans l'état courant, car on n'a utilisé aucun diagnostic dans cet exemple (cf. § 2.4.2 et 2.4.3).

L'indicateur d'échec G et le littéral $(FORME\ TT1\ PRISMATIQUE)$ qui serait engendré par l'application de $rcm14:\{(\$TT\ TT1)\}$ permettent de construire la contrainte suivante en substituant des variables aux constantes [nous avons déjà écrit et expérimenté un programme qui effectue ces transformations $(\{(TYPE\ \$TR\ TOUR)(COMPOSANT\ \$TR\ \$COL)(COMPOSANT\ \$TR\ \$TT)(TYPE\ \$COL\ COLONNE)\ (TYPE\ \$TT\ TOIT)(NOMBRE-DE-BLOCS\ \$COL\ 2)\} \rightarrow \{(NON\ FORME\ \$TT\ PRISMATIQUE)\})$

On note que si cette contrainte avait fait partie du problème original, Tropic aurait immédiatement choisi un toit pyramidal, et aurait ainsi pu engendrer une solution sans subir d'échec.

Si des diagnostics avaient été utilisés, on aurait pu alors définir l'expression ci-dessus comme un conseil.

Dans la mesure où de nombreux problèmes de conception ont des énoncés qui diffèrent assez peu (cf. l'exemple du transformateur, § 2.1), un modèle d'apprentissage inspiré de celui que nous venons d'introduire permettrait à Tropic d'accumuler une certaine expérience pendant leur résolution.

3.5. Modification interactive de la description d'un problème.

La trace qui est imprimée par Tropic au cours de la résolution d'un problème peut conduire le concepteur à découvrir de nouveaux aspects du problème ouvert de conception (cf. § 1.1, chapitre 1). Il peut alors désirer modifier des rcm, des contraintes, des conseils et/ou des diagnostics du problème fermé en cours de résolution, sans attendre que celle-ci soit terminée. Une telle possibilité nécessiterait que Tropic puisse interrompre la résolution, puis la reprendre en tenant compte des modifications effectuées par le concepteur. Compte tenu du fait que le programme d'activation des connaissances est assez indépendant des autres programmes de Tropic, nous pensons qu'elle ne constitue pas un objectif impossible.

3.6. Utilisation de rcm conditionnelles.

Au paragraphe 3.3 du chapitre 2, nous avons introduit la notion de rcm conditionnelle et nous l'avons illustrée sur un exemple.

L'emploi de rcm conditionnelles simplifierait l'écriture de certaines rcm de l'appendice B, en nous évitant d'utiliser l'argument \$REF des problèmes formels (CONCEVOIR-BOBINAGE \$BOB \$REF), (CONCEVOIR-ENROULEMENT \$ENR \$REF), ..., pour transmettre la nature du réfrigérant à un problème tel que (CONCEVOIR-DISQUE \$DISQ \$REF). En effet, on pourrait alors remplacer l'avant dernière rcm de l'appendice B par la rcm conditionnelle suivante :

```
((CONCEVOIR-DISQUE $DISQ) {(TYPE $TRSF TRANSFORMATEUR )(REFRIGERANT $TRSF $REF)
→ ((TYPE $DISQ DISQUE-ISOLANT)(MATERIAU $DISQ $REF)}
NIL))
```

Toutefois, le traitement des rcm conditionnelles présente plusieurs difficultés :

- (1) leur activation nécessite l'exécution d'un algorithme semblable à celui que nous avons décrit au chapitre 7,

(2) l'ordre d'application des rcm peut avoir une influence sur les modèles que l'on peut construire (ce qui remet en question le principe du traitement des échecs décrit au § 3 du chapitre 3),

(3) les opérations du cheminement arrière sont sensiblement plus compliquées.

Seules les difficultés (1) et (3) sont résolues dans l'implantation actuelle de Tropic, et les rcm conditionnelles ne peuvent donc être utilisés qu'avec de nombreuses précautions. La difficulté (2) nécessite de nouvelles recherches.

3.7. Utilisation de plusieurs quantifications de variables.

Il serait utile d'offrir au concepteur la possibilité d'utiliser diverses quantifications de variables, notamment dans les contraintes, les conseils et les diagnostics. La notion de variable plurale brièvement introduite en commentaire au paragraphe 1.1.1 du chapitre 6 correspond à un tel besoin.

3.8. Réalisation d'un logiciel général de traitement d'échec.

Plusieurs algorithmes pour le traitement non aveugle d'échecs ont été développés récemment, notamment en déduction automatique (Doyle 1977). L'analyse de ces algorithmes et leur comparaison avec l'algorithme de Tropic devraient permettre la réalisation d'un logiciel général utilisable par un grand nombre d'applications. Un tel logiciel pourrait être inclus dans un langage tel que PLANNER ou QLISP.

CONCLUSION

Pendant les années 60 et au début des années 70, l'objectif principal de la recherche en CAO a été de montrer les avantages offerts par un certain nombre d'outils directement issus des progrès du logiciel et du matériel informatiques. La reconnaissance de ces avantages par le monde industriel ayant été un fait majeur de ces dernières années (cf. notamment les conférences CAD 74 et 76), une nouvelle tâche importante de la recherche en CAO est de créer des outils plus généraux et plus avancés pour les systèmes du futur (Hatvany 1977), tout en élaborant les bases théoriques propres à la CAO qui seront indispensables pour maîtriser la complexité de ces systèmes. La conception et la réalisation de Tropic constitue un effort dans cette direction.

Nous avons orienté notre recherche vers l'élaboration d'outils de résolution de problème, car nous avons trouvé que ces outils étaient insuffisamment développés en CAO. La résolution de problème étant un thème central en Intelligence Artificielle, nous avons été conduits à adapter à la conception des modèles de résolution de problème et de représentation des connaissances issus de cette discipline. Tropic a ainsi été conçu progressivement, ses implantations successives nous servant d'outils pour imaginer de nouveaux concepts.

Du point de vue de la CAO, le résultat le plus intéressant de cette recherche est la définition d'un langage dont les éléments sont les rcm, les contraintes, les conseils et les diagnostics:

- . les rcm et les contraintes permettent au concepteur d'énoncer des problèmes de conception,
- . les conseils et les diagnostics lui permettent de participer, éventuellement de façon interactive, aux choix importants de leur résolution, sans qu'il ait à définir une méthode de résolution.

Ce langage n'impose aucun ordre de description. Il permet d'envisager des communications et une interaction homme-machine d'un type nouveau en CAO.

Du point de vue de l'Intelligence Artificielle, le résultat le plus important de cette recherche est la définition d'un modèle de résolution de problème qui exploite la décomposabilité des problèmes tout en tenant compte des dépendances entre les sous-problèmes engendrés, et qui inclut un traitement des échecs fondé sur l'analyse des conditions des échecs subis. Le principe de ce modèle est général et pourrait s'appliquer à d'autres domaines que la conception (génération de plans d'actions en robotique, par exemple).

Nous pensons que dans les années à venir l'application de modèles d'Intelligence Artificielle à la conception sera un thème de recherche important à la fois en CAO et en Intelligence Artificielle. Après une période assez conservatrice, la CAO a besoin de nouvelles méthodes pour progresser, et celles de l'Intelligence Artificielle sont prometteuses. Réciproquement, la conception constitue pour l'Intelligence Artificielle un domaine d'application riche et stimulant. La Conférence du Travail du Groupe de Travail 5.2, de l'IFIP organisée à Grenoble en 1978 sur le thème "Intelligence Artificielle et Reconnaissances de Formes en Conception Assistée par Ordinateur" devrait être une étape décisive de cette évolution.

Appendice A : UN EXEMPLE DE CONCEPTION DE TOUR

1. LE FICHER f

Le contenu du fichier f pour cet exemple est énoncé ci-dessous. Sa présentation est à peu près celle que l'on obtient avec les fonctions de "jolie impression" de LISP/CMS. Pour faciliter sa lecture, nous avons regroupé les instructions d'une même forme.

```
(STRUCT TYPE (D S))
(STRUCT COULEUR (D S))
(STRUCT FORME (D S))
(STRUCT SECTION (D S))
(STRUCT HAUTEUR (D N))
(STRUCT COMPOSANT (A H))
(STRUCT SUR (A T))
(PB-INIT (CONCEVOIR-TOUR))
(RCM (CONCEVOIR-TOUR)
  ==>
  (FONCT (AFFECTER-A $TR (! TR))
    (I? (((TYPE $TR TOUR)
      ((ENGENDRER-COMPOSANTS-TOUR $TR)))))))
```

[dans l'implantation actuelle, les ensembles sont exprimés comme les listes]

```
(RCM (ENGENDRER-COMPOSANTS-TOUR $TR)
  ==>
  (FONCT (AFFECTER-A $COL (! COL))
    (AFFECTER-A $TT (! TT))
    (I? (((COMPOSANT $TR $COL)
      (COMPOSANT $TR $TT)
      (SUR $COL $TT)
      ((CONCEVOIR-COLONNE $COL)
        (CONCEVOIR-TOIT $TT)))))))
(RCM (CONCEVOIR-COLONNE $COL)
  ==>
  (((TYPE $COL COLONNE))
    ((DETERMINER-NOMBRE-DE-BLOCS $COL))))
```

```

(POUR NBL = (1 2)
  (RCM (DETERMINER-NOMBRE-DE-BLOCS $COL)
    ==>
      (((NOMBRE-DE-BLOCS $COL NBL))
        ((ENGENDRER-COMPOSANTS-COLONNE $COL NBL))))))
[POUR est une fonction de Tropic qui permet ici de définir deux rcm]
(RCM (ENGENDRER-COMPOSANTS-COLONNE $COL $N)
  ==>
    (FONCT (SI (EGAL $N 1)
      ALORS (AFFECTER-A $BLC (! BLC))
        (I? (((COMPOSANT $COL $BLC))
          ((CONCEVOIR-BLOC $BLC))))))
      SINON (AFFECTER-A $BLC1 (! BLC))
        (AFFECTER-A $BLC2 (! BLC))
          (I? (((COMPOSANT $COL $BLC1)
            (COMPOSANT $COL $BLC2)
              (SUR $BLC1 $BLC2))
                ((CONCEVOIR-BLOC $BLC1)
                  (CONCEVOIR-BLOC $BLC2))))))))))
(RCM (CONCEVOIR-BLOC $BLC)
  ==>
    (((TYPE $BLC BLOC))
      ((DETERMINER-SECTION-BLOC $BLC)
        (DETERMINER-COULEUR-BLOC $BLC)))
  (POUR SECT = (CARREE RONDE)
    (RCM (DETERMINER-SECTION-BLOC $BLC)
      ==>
        (((SECTION $BLC SECT))
          NIL)))
  (POUR COUL = (BLANCHE VERTE)
    (RCM (DETERMINER-COULEUR-BLOC $BLC)
      ==>
        (((COULEUR $BLC COUL))
          NIL)))

```

```

(RCM (CONCEVOIR-TOIT $TT)
  ==>
  (((TYPE $TT TOIT))
   ((DETERMINER-FORME-TOIT $TT)
    (DETERMINER-COULEUR-TOIT $TT))))
(POUR FORM = (PYRAMIDALE PRISMATIQUE)
  (RCM (DETERMINER-FORME-TOIT $TT)
    ==>
    (((FORME $TT FORM))
     NIL)))
(POUR COUL = (ROUGE NOIRE)
  (RCM (DETERMINER-COULEUR-TOIT $TT)
    ==>
    (((COULEUR $TT COUL))
     NIL)))
(CNTR ((TYPE $TR TOUR))
  ==>
  ((HAUTEUR $TR 200)))
(CNTR ((TYPE $TT TOIT)
  (FORME $TT $F))
  ==>
  (FONCT (SI (EGAL $F PRISMATIQUE)
    ALORS (I? ((HAUTEUR $TT 25)))
    SINON (I? ((HAUTEUR $TT 35))))))
(CNTR ((TYPE $BLC BLOC)
  (SECTION $BLC $S))
  ==>
  (FONCT (SI (EGAL $S RONDE)
    ALORS (I? ((PG HAUTEUR $BLC 45)
      (PP HAUTEUR $BLC 85)))
    SINON (I? ((PG HAUTEUR $BLC 55)
      (PP HAUTEUR $BLC 95))))))

```

```

(CNTR ((TYPE $COL COLONNE)
      (COMPOSANT $COL $BLC1)
      (COMPOSANT $COL $BLC2)
      (TYPE $BLC1 BLOC)
      (TYPE $BLC2 BLOC)
      (SECTION $BLC1 $S))
==>
((SECTION $BLC2 $S)))
(CNTR ((TYPE $BLC BLOC)
      (SECTION $BLC CARREE))
==>
((COULEUR $BLC VERTE)))
(CNTR ((TYPE $COL COLONNE)
      (NOMBRE-DE-BLOCS $COL 2)
      (COMPOSANT $COL $BLC)
      (TYPE $BLC BLOC))
==>
((COULEUR $BLC BLANCHE)))
(CNTR ((TYPE $TR TOUR)
      (COMPOSANT $TR $COL)
      (COMPOSANT $TR $TT)
      (TYPE $COL COLONNE)
      (TYPE $TT TOIT))
==>
(((HAUTEUR $TR)
  <==
  (PLUS (HAUTEUR $COL)
        (HAUTEUR $TT))))))

```

```

(CNTR (($BLC)
  ((TYPE $COL COLONNE)
   (COMPOSANT $COL $BLC)
   (TYPE $BLC BLOC)))
==>
  (((HAUTEUR $COL)
   <==
   (PLUS (HAUTEUR $BLC))))))

(CNSL ((TYPE $TR TOUR)
  (COMPOSANT $TR $COL)
  (TYPE $COL COLONNE))
==>
  (FONCT (SI (BUT (PG HAUTEUR $TR 150))
    ALORS (I? ((NOMBRE-DE-BLOCS $COL 2))))))

(CNSL ((TYPE $TR TOUR)
  (COMPOSANT $TR $TT)
  (TYPE $TT TOIT))
==>
  (FONCT (SI (BUT (PG HAUTEUR $TR 150))
    ALORS (I? ((FORME $TT PRISMATIQUE))))))

(CNSL ((TYPE $TR TOUR)
  (COMPOSANT $TR $COL)
  (COMPOSANT $TR $TT)
  (TYPE $COL COLONNE)
  (TYPE $TT TOIT)
  (FORME $TT PRISMATIQUE))
  (COMPOSANT $COL $BLC)
  (TYPE $BLC BLOC))
==>
  ((SECTION $BLC RONDE)))

```


2. LA TRACE DE LA RESOLUTION

La trace de la résolution imprimée par Tropic pour ce problème est approximativement la suivante (nous avons légèrement comprimé cette trace) :

```

*
(***) ETAPE TR1 0 (***)
(AJOUT (TYPE TR1 TOUR)) [le littéral (TYPE TR1 TOUR) est ajouté à l'état
*
*
*
(<<< VERIFICATION DE LA COMPATIBILITE DES RESTRICTIONS NUMERIQUES ACTIVES)
(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS DECLARATIVES ?)
- oui [réponse entrée sur le terminal]
(HAUTEUR TR1 200)
(<<< VALEUT INITIALES :)
(HAUTEUR TR1 200)
(!!! LES RESTRICTIONS SONT COMPATIBLES) [cas trivial]
(<<< VOULEZ-VOUS L'IMPRESSION DES RESULTATS ?)
- non
*
*
(***) ETAPE TR1 1 (***)
(AJOUT (COMPOSANT TR1 COL1))
(AJOUT (COMPOSANT TR1 TT1))
(AJOUT (SUR COL1 TT1))
*
(***) ETAPE COL1 0 (***)
(AJOUT (TYPE COL1 COLONNE))
(AJOUT (NOMBRE-DE-BLOCS COL1 2)) [cf. un des conseils]
*
(***) ETAPE TT1 0 (***)
(AJOUT (TYPE TT1 TOIT))
(AJOUT (FORME TT1 PRISMATIQUE)) [cf. un des conseils]
(AJOUT (COULEUR TT1 ROUGE))
*

```

*

(<<< VERIFICATION DE LA COMPATIBILITE DES RESTRICTIONS NUMERIQUES ACTIVES)

(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS DECLARATIVES ?)

- oui

(HAUTEUR TR1 200)

(HAUTEUR TT1 25)

(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS PROCEDURALES ?)

- oui

(HAUTEUR-TOUR TR1 COL1 TT1)

(<<< VALEURS INITIALES :)

(VALEUR INITIALE DE HAUTEUR COL1 ?)

- 200

(HAUTEUR COL1 200)

(HAUTEUR TT1 25)

(!!! LES RESTRICTIONS SONT COMPATIBLES)

(<<< VOULEZ-VOUS L'IMPRESSION DES RESULTATS ?)

- oui

(HAUTEUR COL1 175) [Tropic imprime ici un ensemble de D.N. littéraux qui

(HAUTEUR TT1 25) satisfait les restrictions. Les A-termes de ces littéraux

(HAUTEUR TR1 200) sont en fait exprimés par des nombres en virgule flottante]

*

*

(*** ETAPE COL1 1 ***)

(AJOUT (COMPOSANT COL1 BLC1))

(AJOUT (COMPOSANT COL1 BLC2))

(AJOUT (SUR BLC1 BLC2))

*

(*** ETAPE BLC1 0 ***)

(AJOUT (TYPE BLC1 BLOC))

(AJOUT (SECTION BLC1 RONDE)) [cf. un des conseils]

(AJOUT (COULEUR BLC1 BLANCHE)) [cf. une des contraintes]

*

(*** ETAPE BLC2 0 ***)

(AJOUT (TYPE BLC2 BLOC))

(AJOUT (SECTION BLC2 RONDE)) [cf. une des contraintes]

(AJOUT (COULEUR BLC1 BLANCHE)) [cf. une des contraintes]

*

*

(<<< VERIFICATION DE LA COMPATIBILITE DES RESTRICTIONS NUMERIQUES ACTIVES)

(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS DECLARATIVES ?)

- oui

(HAUTEUR TR1 200)

(HAUTEUR TT1 25)

(PG HAUTEUR BLC1 45)

(PP HAUTEUR BLC1 85)

(PG HAUTEUR BLC2 45)

(PP HAUTEUR BLC2 85)

(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS PROCEDURALES ?)

- oui

(HAUTEUR-COLONNE COL1 (BLC1 BLC2))

(HAUTEUR-TOUR TR1 COL1 TT1)

(<<< VALEURS INITIALES :)

(HAUTEUR BLC1 65) [la valeur initiale choisie par Tropic est la moyenne

(HAUTEUR BLC2 65) entre les deux limites (45 et 85) imposées]

(HAUTEUR TT1 25)

(??? LES RESTRICTIONS SONT INCOMPATIBLES)

*

*

(↑↑↑ RETOUR A ETAPE BLC1 0)

(RETRAIT (SECTION BLC1 RONDE)) [le littéral (SECTION BLC1 RONDE) est

retiré de l'état courant]

*

(*** ETAPE BLC1 0 ***)

(AJOUT (SECTION BLC1 CARREE))

(??? LES RESTRICTIONS (COULEUR BLC1 BLANCHE) ET (COULEUR BLC1 VERTE)

SONT INCOMPATIBLES)

*

(↑↑↑ RETOUR A ETAPE TT1 0)

(RETRAIT (FORME TT1 PRISMATIQUE))

*

*

(*** ETAPE TT1 0 ***)
(AJOUT (FORME TT1 PYRAMIDALE))

*

*

(<<< VERIFICATION DE LA COMPATIBILITE DES RESTRICTIONS NUMERIQUES ACTIVES)
(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS DECLARATIVES ?)

- oui

(HAUTEUR TR1 200)
(HAUTEUR TT1 35)
(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS PROCEDURALES ?)

- oui

(HAUTEUR-TOUR TR1 COL1 TT1)
(<<< VALEURS INITIALES :)
(HAUTEUR COL1 175) [Tropic prend comme valeur initiale la dernière valeur
(HAUTEUR TT1 35) calculée avant l'échec]
(!!! LES RESTRICTIONS SONT COMPATIBLES)
(<<< VOULEZ-VOUS L'IMPRESSION DES RESULTATS ?)

- non

*

*

(*** ETAPE COL1 1 ***)

*

(*** ETAPE BLC1 0 ***)
(RETRAIT (SECTION BLC1 CARREE))
(AJOUT (SECTION BLC1 RONDE))

*

(*** ETAPE BLC2 0 ***)

*

*

(<<< VERIFICATION DE LA COMPATIBILITE DES RESTRICTIONS NUMERIQUES ACTIVES)
(<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS DECLARATIVES ?)

- oui

(HAUTEUR TR1 200)
(HAUTEUR TT1 35)

(PG HAUTEUR BLC1 55)
 (PP HAUTEUR BLC1 95)
 (PG HAUTEUR BLC2 55)
 (PP HAUTEUR BLC2 95)
 (<<< VOULEZ-VOUS L'IMPRESSION DES RESTRICTIONS PROCEDURALES ?)

- oui

(HAUTEUR-COLONNE COL1 (BLC1 BLC2))
 (HAUTEUR-TOUR TR1 COL1 TT1)

(<<< VALEURS INITIALES :)

(HAUTEUR BLC1 75)

(HAUTEUR BLC2 75)

(HAUTEUR TT1 35)

(!!! LES RESTRICTIONS SONT COMPATIBLES)

(<<< VOULEZ-VOUS L'IMPRESSION DES RESULTATS ?)

- non

*

*

(*** MODELE SOLUTION ***)

*

(DONNEES SYMBOLIQUES :)

((TYPE TR1 TOUR) (COMPOSANT TR1 COL1) (COMPOSANT TR1 TT1)
 (SUR COL1 TT1) (TYPE COL1 COLONNE) (NOMBRE-DE-BLOCS COL1 2)
 (TYPE TT1 TOIT) (FORME TT1 PYRAMIDALE) (COULEUR TT1 ROUGE)
 (COMPOSANT COL1 BLC1) (COMPOSANT COL1 BLC2) (SUR BLC1 BLC2)
 (TYPE BLC1 BLOC) (SECTION BLC1 RONDE) (COULEUR BLC1 BLANCHE)
 (TYPE BLC2 BLOC) (SECTION BLC2 RONDE) (COULEUR BLC2 BLANCHE))

*

(DONNEES NUMERIQUES :)

((HAUTEUR BLC1 85) (HAUTEUR BLC2 80) (HAUTEUR COL1 165)
 (HAUTEUR TT1 35) (HAUTEUR TR1 200))

*

Appendice B : LES RCM DE L'EXEMPLE DU TRANSFORMATEUR

```

(RCM (CONCEVOIR-TRANSFORMATEUR)
  ==>
  (FONCT (AFFECTER-A $TRSF (! TRSF))
    (!? (((TYPE $TRSF TRANSFORMATEUR))
      ((DETERMINER-FREQUENCE $TRSF)
        (DETERMINER-REFRIGERANT $TRSF))))))
(POUR FR = (50 60)
  (RCM (DETERMINER-FREQUENCE $TRSF)
    ==>
    (((FREQUENCE $TRSF FR))
      NIL)))
(POUR REF = (HUILE AIR)
  (RCM (DETERMINER-REFRIGERANT $TRSF)
    ==>
    (((REFRIGERANT $TRSF REF))
      ((ENGENDRER-COMPOSANTS-TRANSFO $TRSF REF))))))
(RCM (ENGENDRER-COMPOSANTS-TRANSFO $TRSF $REF)
  ==>
  (FONCT (AFFECTER-A $CM (! CM))
    (AFFECTER-A $BOB (! BOB))
    (SI (EGAL $REF AIR)
      ALORS (!? (((CIRCUIT-MAGNETIQUE-DE $TRSF $CM)
        (BOBINAGE-DE $TRSF $BOB))
          ((CONCEVOIR-CIRC-MAGNETIQUE $CM)
            (CONCEVOIR-BOBINAGE $BOB $REF))))))
      SINON (AFFECTER-A $CUVE (! CUVE))
        (!? (((CIRCUIT-MAGNETIQUE-DE $TRSF $CM)
          (BOBINAGE-DE $TRSF $BOB)
            (CUVE-DE $TRSF $CUVE))
              ((CONCEVOIR-CIRC-MAGNETIQUE $CM)
                (CONCEVOIR-BOBINAGE $BOB $REF)
                  (CONCEVOIR-CUVE $CUVE)))))))

```

```

(RCM (CONCEVOIR-CIRC-MAGNETIQUE $CM)
  ==>
  (((TYPE $CM CIRCUIT-MAGNETIQUE))
  ((DETERMINER-JOINTS $CM)
  (DETERMINER-QUALITE-TOLE $CM)
  (ENGENDRER-COMPOSANTS-CIRC-MAG $CM))))
(POUR J = (45 90)
  (RCM (DETERMINER-JOINTS $CM)
  ==>
  (((JOINTS $CM J))
  NIL)))
(POUR Q = (1 2)
  (RCM (DETERMINER-QUALITE-TOLE $CM)
  ==>
  (((QUALITE-TOLE $CM Q))
  NIL))
(RCM (ENGENDRER-COMPOSANTS-CIRC-MAG $CM)
  ==>
  (FONCT (AFFECTER-A $NOY (! NOY))
  (AFFECTER-A $CUL (! CUL))
  (I? (((NOYAU-DE $CM $NOY)
  (CULASSE-DE $CM $CUL))
  ((CONCEVOIR-NOYAU $NOY)
  (CONCEVOIR-CULASSE $CUL)))))))
(RCM (CONCEVOIR-NOYAU $NOY)
  ==>
  (((TYPE $NOY NOYAU))
  ((DETERMINER-NB-GRADINS $NOY))))
(POUR N = (1 2 3 4 5 6)
  (RCM (DETERMINER-NB-GRADINS $NOY)
  ==>
  (((NOMBRE-DE-GRADINS $NOY N))
  NIL)))

```

```

(RCM (CONCEVOIR-CULASSE $CUL)
  ==>
  (((TYPE $CUL CULASSE))
   NIL)
(RCM (CONCEVOIR-BOBINAGE $BOB $REF)
  ==>
  (((TYPE $BOB BOBINAGE))
   ((DETERMINER-MONTAGE-1 $BOB)
    (DETERMINER-MONTAGE-2 $BOB)
    (ENGENDRER-COMPOSANTS-BOBINAGE $BOB $REF))))
(POUR M1 = (Y D)
  (RCM (DETERMINER-MONTAGE-1 $BOB)
  ==>
  (((MONTAGE-PRIMAIRE $BOB M1))
   NIL)))
(POUR M2 = (Y D)
  (RCM (DETERMINER-MONTAGE-2 $BOB)
  ==>
  (((MONTAGE-SECONDAIRE $BOB M2))
   NIL)))
(RCM (ENGENDRER-COMPOSANTS-BOBINAGE $BOB $REF)
  ==>
  (FONCT (AFFECTER-A $ENR (! ENR))
         (AFFECTER-A $ECR (! ECR))
         (I? (((ENROULEMENT-DE $BOB $ENR)
                (ECRAN-ISOLANT-DE $BOB $ECR))
              ((CONCEVOIR-ENROULEMENT $ENR $REF)
               (CONCEVOIR-ECRAN $ECR $REF))))))
(RCM (ENGENDRER-ENROULEMENT $ENR $REF)
  ==>
  (((TYPE $ENR ENROULEMENT))
   ((ENGENDRER-COMPOSANTS-ENROULEMENT $ENR $REF))))

```



```
(RCM (ENGENDRER-COMPOSANTS-ENROULEMENT $ENR $REF)
```

```
==>
```

```
(FONCT (AFFECTER-A $CALE (! CALE))
  (AFFECTER-A $EMP1 (! EMP))
  (AFFECTER-A $EMP2 (! EMP))
  (AFFECTER-A $CYL1 (! CYL))
  (AFFECTER-A $CYL2 (! CYL))
  (I? (((CALE-DE $ENR $CALE)
    (EMPILAGE-DE $ENR $EMP1)
    (EMPILAGE-DE $ENR $EMP2)
    (CYLINDRE-ISOLANT-DE $ENR $CYL1)
    (CYLINDRE-ISOLANT-DE $ENR $CYL2)
    (AUTOUR $CYL1 $EMP1 $CYL2 $EMP2))
    ((CONCEVOIR-CALE $CALE $REF)
    (CONCEVOIR-CYLINDRE $CYL1 $REF)
    (CONCEVOIR-CYLINDRE $CYL2 $REF)
    (CONCEVOIR-EMPILAGE $EMP1 $REF)
    (CONCEVOIR-EMPILAGE $EMP2 $REF))))))
```

```
(RCM (CONCEVOIR-EMPILAGE $EMP $REF)
```

```
==>
```

```
(( (TYPE $EMP EMPILAGE))
  ((DETERMINER-APPARTENANCE $EMP)
  (DETERMINER-NB-GALETTES $EMP $REF))))
```

```
(POUR APP = (PRIMAIRE SECONDAIRE)
```

```
(RCM (DETERMINER-APPARTENANCE $EMP)
```

```
==>
```

```
(( (APPARTENANCE $EMP APP))
  NIL)))
```

```
(POUR NG = (1 2 3 4 5 6)
```

```
(RCM (DETERMINER-NB-GALETTES $EMP $REF)
```

```
==>
```

```
(( (NOMBRE-DE-GALETTES $EMP NG))
  ((ENGENDRER-COMPOSANTS-EMPILAGE $EMP $REF NG))))
```

```

(RCM (ENGENDRER-COMPOSANTS-EMPILAGE $EMP $REF $N)
==>
(FONCT (AFFECTER-A $GAL (! GAL))
(SI (EGAL $N 1)
ALORS (I? (((GALETTE-DE $EMP $GAL))
((CONCEVOIR-GALETTE $GAL))))
SINON (AFFECTER-A $DISQ (! DISQ))
(I? (((GALETTE-DE $EMP $GAL)
(DISQUE-ISOLANT-DE $EMP $DISQ))
((CONCEVOIR-GALETTE $GAL)
(CONCEVOIR-DISQUE $DISQ $REF))))))
(RCM (CONCEVOIR-GALETTE $GAL)
==>
(((TYPE $GAL GALETTE))
((ENGENDRER-CONDUCTEUR $GAL)))
(RCM (ENGENDRER-CONDUCTEUR $GAL)
==>
(FONCT (AFFECTER-A $COND (! COND))
(I? (((CONDUCTEUR-DE $GAL $COND))
((CONCEVOIR-CONDUCTEUR $COND))))))
(RCM (CONCEVOIR-CONDUCTEUR $COND)
==>
(((TYPE $COND CONDUCTEUR)
(MATERIAU $COND CUIVRE))
(DETERMINER-FORME $COND)))
(RCM (DETERMINER-FORME $COND)
==>
(((FORME $COND ROND))
((DETERMINER-DIAMETRES-CONDUCTEUR $COND)))
(RCM (DETERMINER-FORME $COND)
==>
(((FORME $COND MEPLAT))
((DETERMINER-NB-BRINS-AXIAUX $COND)
(DETERMINER-NB-BRINS-RADIAUX $COND)
(DETERMINER-HAUTEURS-BRIN $COND)
(DETERMINER-EPAISSEURS-BRIN $COND)))

```

```

(POUR DIAM = (1.00E-03 1.10E-03 ...) [liste des diamètres normalisés]
  (RCM (DETERMINER-DIAMETRES-CONDUCTEUR $COND)
    ==>
      (((DIAMETRE-NU $COND DIAM)
        (DIAMETRE-ISOLE $COND ($ (PLUS DIAM 3.00E-04))))
        NIL)))
(POUR NBA = (1 2)
  (RCM (DETERMINER-NB-BRINS-AXIAUX $COND)
    ==>
      (((NOMBRE-DE-BRINS-AXIAUX $COND NBA)
        NIL)))
(POUR NBR = (1 2)
  (RCM (DETERMINER-NB-BRINS-RADIAUX $COND)
    ==>
      (((NOMBRE-DE-BRINS-RADIAUX $COND NBR)
        NIL)))
(POUR HAUT = (3.55E-03 4.00E-03 ...) [liste des hauteurs normalisées]
  (RCM (DETERMINER-HAUTEURS-BRIN $COND)
    ==>
      (((HAUTEUR-BRIN-NU $COND HAUT)
        (HAUTEUR-BRIN-ISOLE $COND ($ (PLUS HAUT 4.00E-04))))
        NIL)))
(POUR EP = (1.25E-03 1.40E-03 ...) [liste des épaisseurs normalisées]
  (RCM (DETERMINER-EPAISSEURS-BRIN $COND)
    ==>
      (((EPAISSEUR-BRIN-NU $COND EP)
        (EPAISSEUR-BRIN-ISOLE $COND ($ (PLUS EP 4.00E-04))))
        NIL)))
(RCM (CONCEVOIR-ECRAN $ECR $REF)
  ==>
    (((TYPE $ECR ECRAN-ISOLANT)
      ((DETERMINER-MATERIAU-ECRAN $ECR $REF))))
(RCM (DETERMINER-MATERIAU-ECRAN $ECR $REF)
  ==>
    (((MATERIAU $ECR $REF))
    NIL))

```

```

(RCM (DETERMINER-MATERIAU-ECRAN $ECR $REF)
  ==>
  (((MATERIAU $ECR PAPIER-BAKELISE))
   NIL))
(RCM (CONCEVOIR-CALE $CALE $REF)
  ==>
  (((TYPE $CALE CALE)
    (MATERIAU $CALE $REF))
   NIL))
(RCM (CONCEVOIR-CYLINDRE $CYL $REF)
  ==>
  (((TYPE $CYL CYLINDRE-ISOLANT)
    ((DETERMINER-COMPOSITION-CYLINDRE $CYL $REF))))))
(RCM (DETERMINER-COMPOSITION-CYLINDRE $CYL $REF)
  ==>
  (((COMPOSITION $CYL SIMPLE)
    (MATERIAU $CYL $REF))
   NIL))
(RCM (DETERMINER-COMPOSITION-CYLINDRE $CYL $REF)
  ==>
  (((COMPOSITION $CYL TRIPLE)
    (MATERIAU1 $CYL $REF)
    (MATERIAU2 $CYL $PAPIER-BAKELISE))
   NIL))
(RCM (CONCEVOIR-DISQUE $DISQ $REF)
  ==>
  (((TYPE $DISQ DISQUE-ISOLANT)
    (MATERIAU $DISQ $REF))
   NIL))
(RCM (CONCEVOIR-CUVE $CUVE)
  ==>
  (((TYPE $CUVE CUVE)
    (FORME $CUVE A-AILETTES))
   NIL))

```


REFERENCES

- ABRIAL(1974)
J.R. ABRIAL: "Data semantics", IFIP WorkingConference, Cargèse, France
North-Holland, 1974.

- ALEXANDER(1964)
C.ALEXANDER: "Notes on the synthesis of form", Harvard University Press,
Cambridge, 1964.

- ALLAN (1977)
J.J. ALLAN III (Editeur): "CAD Systems", Proceedings of the IFIP Working
Conference on "CAD Systems", North-Holland, 1977.

- BAZJANAC (1974)
V. BAZJANAC: "Architectural design theory: Models of the design process",
dans Spillers (1974).

- BOBROW (1974)
D.G. BOBROW, B. RAPHAEL: "New programming languages for Artificial
Intelligence", Computing Surveys, Vol. 6, N°3, Septembre 1974.

- BOLOPION (1975)
A. BOLOPION, M. CAMPMAS, J.C. LATOMBE, J.C. SABONNADIÈRE: "A new approach
to CAI by interactive design of electrical engineering systems", IEEE
Transactions on Education, Vol E-18, N°2, Mai 1975.

- CAD74 et 76
Conférences biennales organisées par IPC Press Limited, Guilford,
Angleterre.

- Computer Aided Design Journal
Revue périodique publiée par IPC Press Limited, Guilford,
Angleterre.

- COURTIN(1977)
J. COURTIN: "Algorithmes pour le traitement interactif des langues
naturelles", Thèse de Doctorat d'état, Université de Grenoble,
Octobre, 1977.

- DAVIS (1975)
R. DAVIS, B. BUCHANAN, E.H. SHORTLIFFE: "Production rules as a representation for a knowledge-based consultation program", Stanford Artificial Intelligence Laboratory, Memo AIM-266, Computer Science Department, Stanford University, Octobre 1975.

- DAVIS (1977)
R. DAVIS, B.G. BUCHANAN: "Meta-level knowledge: Overview and applications", 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, USA, Août 1977.

- Design Automation Conferences 1974-1977
Conférences annuelles organisées par ACM et IEEE.

- DOYLE (1977)
J. DOYLE: "Truth maintenance systems for problem solving", S.M. Thesis, Department of Electrical Engineering and Computer Science Department, MIT, June 1977.

- EASTMAN (1971)
C.M. EASTMAN: "Heuristic Algorithm for Automated Space Planning", 2nd International Joint Conference on Artificial Intelligence, Londres, 1969.

- EASTMAN (1973)
C.M. EASTMAN: "Automated Space Planning", Artificial Intelligence, vol.4, 1973.

- EASTMAN (1978)
C.M. EASTMAN: "The representation of design problems and maintenance of their structure", article sélectionné pour la Conférence de Travail du Groupe 5.2. de l'IFIP, Grenoble, 17-19 Mars 1978.

- ERNST (1969)
G.W. ERNST, A. NEWELL: "GPS: a case study in generation and problem solving Academic Press, 1969.

- FEIGENBAUM (1971)
E.A. FEIGENBAUM, B.G. BUCHANAN, J. LEDERBERG: "On generality and problem solving: A case study using the DENDRAL program", Machine Intelligence 6, Edinburgh University Press, 1971.

- FREEMAN (1971)
P.A. FREEMAN, A. NEWELL: "A model for functional reasoning in design"
2nd International Joint Conference on Artificial Intelligence, Londres,
1969.

- GRANDJEAN (1975)
E. GRANDJEAN: "Conception et réalisation d'un dictionnaire pour un
analyseur interactif de langues naturelles", thèse CNAM, Février 1975.

- HART (1975)
P.E. HART: "Progress on a Computer-Based Consultant", 4th International
Joint Conference on Artificial Intelligence, Tbilissi, URSS, Septembre
1975.

- HATVANY (1977)
J. HATVANY: "Trends and developments in Computer-Aided Design",
IFIP Congress 77, Toronto, Canada, Août 1977.

- HENDRIX (1975)
G.G. HENDRIX: "Partitioned networks for the mathematical modeling
of natural languages", Ph.D. Dissertation, Computer Science Department,
University of Texas at Austin, 1975.

- HEWITT (1971)
C. HEWITT: "Description and theoretical analysis (using schemata) of
PLANNER: a language for proving theorems and manipulating models in a
robot", Ph.D. Dissertation (1971) ; AI-TR-258, MIT AI Laboratory,
Cambridge, USA, 1972.

- HOOKES (1961)
R. HOOKES, T.A. JEEVES: "Direct search solution of numerical and
statistical problems", Journal of the ACM, Avril 1961.

- IWATA (1975)
S. IWATA, S. ISHINO, Y. MISHIMA: "Heuristics in the Alloy Designing",
4th International Joint Conference on Artificial Intelligence, Tbilissi,
Septembre 1975.

- JACQUART (1975)
R. JACQUART, P. REGNIER, F.R. VALETTE, J. FOISSEAU: "Current trends in
the development of integrated general-purpose CAD systems", Proceedings
of the 11th Design Automation Workshop of the ACM and IEEE, Denver,
USA, Juin 1974.

- JIRAUGH (1966)
G. JIRAUGH: "Artificial Intelligence in Automated Design", Proceedings of the SHARE Design Automation Workshop, New Orleans, Mai 1966.

- LANG (1974)
C.A. LANG: "Archivements in Computer Aided Design", IFIP Congress 74, Stockolm, Suède, Août 1974.

- LATOMBE (1972)
J.C. LATOMBE: "Elaboration d'un système pédagogique d'assistance à la conception en Electrotechnique", Thèse de Docteur-Ingénieur, Université de Grenoble, Novembre 1972.

- LATOMBE (1976a)
J.C. LATOMBE: "Artificial Intelligence in Computer-Aided Design: the TROPIC system", dans Allan (1977) ; aussi publié dans SRI Artificial Intelligence Technical Note 125, Février 1976.

- LATOMBE (1976b)
J.C. LATOMBE: "TROPIC: Un système de Conception Assistée par Ordinateur utilisant des techniques d'Intelligence Artificielle", Congrès AFCET, Ecole Supérieure d'électricité, Gif-Sur-Yvette, Novembre 1976.

- LATOMBE (1977a)
J.C. LATOMBE: "TROPIC: a general-purpose design problem solver applying domain-specific heuristics", Rapport de Recherche n° 61, Laboratoire Associé au CNRS n°7, Grenoble, Janvier 1977.

- LATOMBE (1977b)
J.C. LATOMBE: "Application de l'Intelligence Artificielle à la Conception Assistée par Ordinateur", Cahiers du Groupe de Travail AFCET "Analyse et Expérimentation dans les Sciences de l'Homme par les Méthodes Informatiques", Juin 1977 (aussi présenté aux Journées Scientifiques et Techniques de la Production Automatisée, ADEPA, Mai 1977).

- LOPEZ (1976)
M. LOPEZ: "Etude de quelques réseaux sémantiques en vue d'une application à Tropic", Rapport de D.E.A., Université de Grenoble, Septembre 1976.

- LUX (1974)
A. LUX: "Etude d'un modèle abstrait pour une machine LISP et de son implantation", Thèse de 3ème cycle, Université de Grenoble, Mars 1975.

- McCARTHY (1965)
J. McCARTHY et al.: "LISP 1.5 Programmer's Manual", MIT Press, 1965.

- McDERMOTT (1977)
D. McDERMOTT: "Flexibility and efficiency in a computer program for designing circuits", AI Laboratory, MIT, Technical Report AI-TR-402, Juin 1977.

- MILLEY (1972)
M.K. MILLEY: "An interactive computer aid for design parameter and constraint tradeoff analysis", S.M. Thesis, Mechanical Engineering Department, MIT, Cambridge, USA, 1972.

- NEWELL (1972a)
A. NEWELL, H.A. SIMON: "Human Problem Solving", Prentice Hall, 1972.

- NEWELL (1972b)
A. NEWELL: "Production systems: models of control structures", Visual Information Processing, W.G. Chase ed., Academic Press, 1972.

- NILSSON (1971)
N.J. NILSSON: "Problem solving methods in Artificial Intelligence", Mc Graw-Hill, 1971.

- PFEFFERKORN (1972)
C.E. PFEFFERKORN: "The Design Problem Solver: a system for designing equipments or Furniture layouts", rapport interne, Department of Computer Science, Carnegie-Mellon University, Pittsburg, USA, Septembre 1972.

- PFEFFERKORN (1975)
C.E. PFEFFERKORN: "A heuristic problem solving design system for equipment or furniture layouts", Communications of ACM, vol. 18, N° 5, Mai 1975.

- POWERS (1973)
G.J. POWERS: "Non-numerical problem solving methods in Computer-Aided Design", dans Vlietstra (1973).

- POWERS (1974)
G.J. POWERS, D.F. RUDD: "A theory for chemical engineering design", dans Spillers (1974).

- RIEGER (1977)
C. RIEGER, M. GRINBERG: "The declarative representation and procedural simulation of physical mechanisms", 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, USA, Août 1977.

- RITTEL (1972)
H. RITTEL: "On the planning crisis: Systems analysis of the first and second generations", Brediftsøkonomien, N°8, 1972.

- ROXLO (1973)
J. ROXLO: "Application of CADSYS to the design of an inertial navigator", S.M. Thesis, Mechanical Engineering Department, MIT, Cambridge, USA, 1973.

- RULIFSON (1972)
J.F. RULIFSON, J.A. DERKSEN, R.J. WALDINGER: "QA4: a procedural calculus for intuitive reasoning", SRI Artificial Intelligence Center Technical Note 73, Novembre 1972.

- SACERDOTI (1973)
E.D. SACERDOTI: "Planning in a hierarchy of abstraction spaces", 3rd International Joint Conference on Artificial Intelligence, Stanford, USA, Août 1973.

- SACERDOTI (1975)
E.D. SACERDOTI: "A structure for plans and behaviour", SRI Artificial Intelligence Center, Technical Note 109, Août 1975.

- SACERDOTI (1976)
E.D. SACERDOTI, R.E. FIKES, R. REBOH, D. SAGALOWICZ, R.J. WALDINGER, B.M. WILBER: "QLISP: A language for the interactive development of complex systems", National Computer Conference, 1976.

- SHORTLIFFE (1976)
E.H. SHORTLIFFE: "MYCIN: Computer-Based Medical consultations", New-York, American Elsevier (1976).

- SIMON (1969)
H.A. SIMON: "The Sciences of the Artificial", MIT Press, 1969.

- SIMON (1973)
H.A. SIMON: "The structure of ill structured problems", Artificial Intelligence, vol.4, 1973.

- SPILLERS (1974)
W.R. SPILLERS (Editeur): "Basic questions in design theory", North-Holland, American Elsevier, 1974.

- STALLMAN (1976)
R. STALLMAN, G.J. SUSSMAN: "Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis", MIT Artificial Intelligence Laboratory, Memo 380, Septembre 1976.

- SUSSMAN (1975)
G.J. SUSSMAN: "A computer model of skill acquisition", American Elsevier, 1975.

- TEITELMAN (1974)
W. TEITELMAN: "INTERLISP reference manual", XEROX Palo Alto Center, Palo Alto, USA, 1974.

- VLIETRSTRA (1973)
J. VLIETSTRA, R.F. WIELINGA (Editeurs): "Computer-Aided Design", Proceedings of the IFIP Working Conference on "Principles of Computer Aided-Design", North Holland, 1973.

- WALTZ (1972)
D.L. WALTZ: "Generating semantic descriptions from drawings of scenes with shadows", MIT AI Laboratory, Technical Report 271, Novembre 1972.

- WATERMAN (1970)
D.A. WATERMAN : "Generalization learning techniques for automating the learning of heuristics", Artificial Intelligence, vol.1, 1970.

- WESTERBERG (1974)
A.W. WESTERBERG, G. STEPHANOPOULOS, J. SHAH: "The synthesis problem with some thoughts on evolutionary synthesis in the design of engineering system dans Spillers (1974).

- WHITNEY (1974)
D.E. WHITNEY, M.K. MILLEY: "CADSYS: a new approach to Computer-Aided Design", IEEE Transactions on Systems, Man and Cybernetic, Janvier 1974.

- WHITNEY (1975)
D.E. WHITNEY: "Practical Results with CADSYS", 12-th Design Automation Workshop of the ACM and IEEE, Boston, USA, 1975.

- WILBER (1976)
B.M. WILBER: "A QLISP reference manual", SRI Artificial Intelligence Center Technical Note 118, Mars 1976.

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 5 de l'arrêté du 16 Avril 1974,

VU les rapports de Messieurs :

- G. VEILLON, Professeur à l'I.N.P.G.
- N.J. NILSSON, Professeur à l'Université du Massachusetts
U.S.A.
- A. COLMERAUER, Maître de Conférences à l'Université de
MARSEILLE-LUMINY.

Monsieur Jean-Claude LATOMBE

est autorisé à présenter une thèse en soutenance pour l'obtention du
grade de DOCTEUR d'ETAT ES-SCIENCES.

A Grenoble, le 7 Novembre 1977

Le Président de l'U.S.M.G.

Le Président de l'I.N.P.G.

Robert Cau



Ph. Traynard

Ph. TRAYNARD
Président
de l'Institut National Polytechnique