Expertise mathématique et informatique : CAMELIA, un logiciel pour raisonner et calculer

Martial Vivet

THÈSE

présentée à

L'UNIVERSITÉ PIERRE ET MARIE CURIE
PARIS VI

pour l'obtention du

GRADE DE DOCTEUR ES SCIENCES

Spécialité : **MATHÉMATIQUES APPLIQUÉES**

Mention : **INFORMATIQUE**

par

M. Martial VIVET

Sujet de la thèse :

EXPERTISE MATHÉMATIQUE ET INFORMATIQUE :

CAMELIA un logiciel pour raisonner et calculer

Soutenue le 25 juin 1984

devant la Commission composée de

Monsieur SIMON JC Président
Monsieur PITRAT J
Monsieur CHEIN M
Monsieur LAURENT JP
Monsieur LAURIERE J.L. examinateurs
Monsieur KHAN G
Monsieur HENNEQUIN P.L.

THESE de DOCTORAT D'ETAT

ès Science Mathématiques

présentée

à l'Université Pierre et Marie Curie

- PARIS 6 -

Par Monsieur VIVET Martial

pour obtenir le grade de DOCTEUR ès SCIENCES

Sujet de la thèse:

EXPERTISE MATHEMATIQUE ET INFORMATIQUE : CAMELIA un logiciel pour raisonner et calculer

soutenue le 25 juin 1984

devant le jury composé de :

MR SIMON JC	President
MR PITRAT J	Rapporteur
MR CHEIN M	Rapporteur
MR LAURENT JP	Rapporteur
MR LAURIERE JL	Examinateur
MR KHAN G	Examinateur
MR HENNEQUIN PL	Examinateur

Je remercie Monsieur le Professeur Jean-Claude SIMON, Professeur à l'Université Pierre et Marie Curie, qui a bien voulu me faire l'honneur de présider le jury.

Que Monsieur Jacques PITRAT, Directeur de recherche au CNRS, trouve ici l'expression de toute ma gratitude et de ma profonde reconnaissance pour l'écoute attentive et critique qu'il a maintenu sans cesse tout au long de ce travail.Les critiques constructives prodiguées avec gentillesse mais sans complaisance,la disponibilité d'esprit dont il a su faire preuve à l'occasion de ce travail ont permis de mener cette réalisation à terme.Il m'est agréable de l'en remercier vivement aujourd'hui.

Monsieur Michel CHEIN, Professeur à l'Université de Montpellier m'a fait l'honneur de bien vouloir accepter d'être rapporteur pour le travail que je présente ici. Je le remercie sincèrement d'avoir accepté de participer à ce jury. Qu'il trouve ici l'expression de ma gratitude et de ma reconnaissance.

Monsieur Jean-Pierre LAURENT, Professeur à l'Université de Savoie, se souvient sans doute que l'idée de réaliser un programme "Taupin" est née il y a une dizaine d'années à partir de travaux voisins que nous avions faits et dont nous avions longuement discutés. Cette idée a servi de support pour affiner les méthodes que nous employons en intelligence artificielle. Je le remercie vivement d'avoir accepté de participer à ce jury.

Monsieur Jean-Louis LAURIERE, Professeur à l'Université Pierre et Marie Curie, m'a guidé et encouragé avec enthousiasme et disponibilité tout au long de ce travail. Les critiques (oh combien constructives !) qu'il a toujours su prodiguer avec gentillesse, mais sans complaisance, l'opiniâtreté et l'exigeance dont il sait faire preuve ont été fondamentales pendant toute la progression de ce travail. Qu'il trouve ici l'expression de ma profonde gratitude.

Monsieur Gilles KHAN Ingénieur de recherche à l'INRIA à SOPHIA-ANTIPOLIS m'a fait l'honneur de bien vouloir accepter de faire partie de ce jury. Qu'il trouve ici l'expression de mes sincères remerciements.

Monsieur Paul-Louis HENNEQUIN, Professeur de mathématiques à l'Université de Clermont est bien connu, en particulier de ceux qui enseignent les mathématiques "de la maternelle à l'université", comme le dit le bulletin de l'Association des Professeurs de Mathématiques qu'il anime. Sa présence dans ce jury témoigne que tous ceux qui ont la délicate mission d'enseigner les mathématiques n'ont pas été oubliés. Pouvoir faire faire des mathématiques à un ordinateur, c'est aussi apprendre à mieux maîtriser des processus fondamentaux pour transmettre la connaissance mathématique. Qu'il accepte ici tous mes remerciements pour avoir accepté d'être membre de ce jury.

L'Université du Maine m'a permis de participer à la mise en place de moyens de calculs sans lesquels ce travail n'aurait pas été possible. Je tiens à lui en être reconnaissant. Mes remerciements s'adressent également à toutes les personnes du service de mathématiques et d'informatique de la Faculté des Sciences qui ont participé, à divers titres, à la réalisation matérielle de ce travail. Merci Danièle, Merci Michèle, Merci Pascale, Merci Josette et Merci Christiane.

Je tiens enfin à remercier tous ceux qui, dans les moments de doute, par leur amitié et leur affection m'ont encouragé, aidé et soutenu. Monique BARON, Jean-Louis LAURIERE, Jacques PITRAT méritent une pensée particulière pour avoir eu la patience de relire et critiquer le manuscrit.

Les crédits nécessaires à la réalisation de CAMELIA (en particulier les crédits de calculs) ont été fournis par le CNRS (ATP 98257). Le travail rapporté ici n'aurait pas été possible sans cette aide. Que le CNRS trouve ici la marque de ma reconnaissance et mes remerciements.

INTRODUCTION

Il y a maintenant une vingtaine d'années que des travaux importants ont commencé pour permettre aux ordinateurs de faire des mathématiques. Précisons qu'il s'agit de l'activité telle que la conçoit le mathématicien : calcul exact et démonstration de théorèmes en écartant les méthodes purement numériques base des débuts du calcul scientifique. Nous discutons dans ce mémoire de l'évolution des idées et des méthodes afin de mieux cerner la nature des travaux faits et l'approche avec laquelle ils ont été faits. Nous décrivons un logiciel qui permet d'aborder de façon homogène les difficultés issues de la séparation trop nette entre CALCUL SYMBOLIQUE, DEMONSTRATION AUTOMATIQUE, ALGORITHMES ET HEURISTIQUES.

Le chapitre I intitulé "MATHEMATIQUES ET ORDINATEURS" met en évidence que deux grandes voies ont été utilisées conjointement, s'ignorant pratiquement l'une l'autre : il s'agit de la voie du calcul algébrique et de la voie de la démonstration automatique de théorèmes. La voie du calcul algébrique a amené le développement de systèmes importants qui sont rappelés et commentés dans une première partie. La deuxième partie traite de la démonstration automatique de théorèmes, du rôle qu'elle a joué en mathématiques et surtout en dehors des mathématiques.

Le chapitre II intitulé "LES VOIES DE L'ALGORITHMIQUE / LES VOIES DE LA PROGRAMMATION HEURISTIQUE" insiste sur les différences entre les approches utilisées. Une première partie reprécise le rôle de la programmation d'algorithmes et la nécessité d'utiliser des heuristiques en particulier pour permettre le choix de l'algorithme approprié (pour traiter un problème ou amener la décision de non application d'un algorithme). L'on montre alors que ces deux approches ont été utilisées tant dans la voie du calcul algébrique que dans la voie de la démonstration de théorèmes. Les sujets supports ont d'ailleurs été souvent les mêmes : l'intégration formelle, le calcul de limites, la manipulation de polynômes ont été des sujets de prédilection. Il apparaît alors que chacune de ces approches ou de ces voix butte parce qu'elle ne prend pas en compte ce que l'autre approche ou l'autre voie permet : ainsi l'on observe des échecs dans les systèmes de démonstration automatique parce que ces systèmes faits pour raisonner savent mal calculer ! Réciproquement, les systèmes de calculs algébriques travaillent de façon aveugle, sans mécanisme d'inférence leur permettant de raisonner les calculs qu'ils fonts. Dans le même ordre d'idées, les algorithmes qui mettent en oeuvre des mécanismes ignorant les données effectives du problème, ne peuvent pas prendre en compte les heuristiques. Par contre, les systèmes qui raisonnent sur les données avec des jeux d'heuristiques adaptées ont parfois du mal à pouvoir déclencher l'exécution d'un algorithme (alors qu'ils peuvent avoir parfaitement caractérisé qu'ils sont dans un cas où il faut utiliser cet algorithme).

Le chapitre III part du fait que le mathématicien est complet en ce sens qu'il calcule et raisonne, emploie des heu-

ristiques et des algorithmes, tous ces aspects coopérant lors de la résolution d'un problème. Il amène alors à l'idée de système expert pour "faire des mathématiques avec un ordinateur", système expert devant satisfaire des contraintes fortes à savoir : permettre le calcul et les mécanismes d'inférence, permettre la manipulation d'heuristiques et le déclenchement de l'exécution d'algorithmes (dont le choix et les conditions d'applications sont justement déterminées de façon heuristique). Cette approche a le mérite essentiel de traiter de façon homogène les aspects de preuve et les aspects opératoires de l'activité du mathématicien. La coopération entre ces aspects est donc facilitée. Les règles de système expert décrivent les heuristiques. En prévoyant la possibilité de déclenchement d'algorithmes au sein de ces règles nous atteignons la possibilité effective de pouvoir contrôler le choix et les conditions d'application d'un algorithme. Ce travail est généralement laissé à l'utilisateur dans les systèmes de calcul algébrique.

Le chapitre IV présente le logiciel CAMELIA que nous avons réalisé pour montrer la faisabilité de ces idées. CAMELIA est un système expert essentiel dont le moteur travaille avec des connaissances organisées en chapitres dans une base de connaissances. Ces connaissances expriment le savoir faire, les démarches du mathématicien. Elles constituent pour l'essentiel une représentation de l'aspect opératoire des théorèmes : coder un théorème, un algorithme,... est une chose, coder la façon de l'utiliser en est une autre. Nous faisons dans ce chapitre une présentation du langage d'expression des connaissances et du moteur d'inférence que nous avons réalisé. La partie qui traite des conditions est importante : la richesse d'un tel système dépend très fortement de la richesse des conditions que l'on peut écrire. CAMELIA doté de mécanismes de preuves étant la notion de condition au sens habituel (expression logique) au champ des propriétés "prouvables" par le système. Nous avons utilisé le terme de "conditions problèmes" pour désigner cette possibilité. Le problème essentiel du contrôle de l'usage des connaissances par le moteur est abordé et nous décrivons l'usage que nous avons fait de la méta-connaissance. Une analyse des divers aspects de l'interface homme-machine avec un tel logiciel est présentée.

Le chapitre V plus prospectif décrit les extensions envisageables de CAMELIA et explicite des voies de recherche qui se sont précisées à l'occasion de ce travail.

Les annexes A,B,C précisent le langage et les outils utilisables actuellement avec CAMELIA. L'annexe D montre des exemples de codages de connaissances. Les annexes F et G donnent la liste des problèmes résolus et des exemples de solutions.

Expertise mathématique et informatique : CAMELIA un logiciel pour raisonner et calculer.

TABLE DES MATIERES

TABLEAUX ET FIGURES

BIBLIOGRAPHIE

CHAPITRE 1

MATHEMATIQUES ET ORDINATEURS

MATHEMATIQUES ET ORDINATEURS

I-1 ASPECTS HISTORIQUES.

Du point de vue historique, les premiers usages des ordinateurs concernaient le calcul scientifique. Il faut se souvenir que pendant les années 1950-1960 on utilisait surtout le terme de calculateur et le travail confié aux machines comportait uniquement du calcul numérique. Cette utilisation relevait essentiellement de scientifiques qui voyaient dans ces calculateurs le bon moyen de dépasser les performances de règles à calcul ou tables numériques diverses.

L'idée d'avoir avec un calculateur une activité de nature mathématique non seulement réduite au calcul numérique est apparue vers les années 1960. On a alors pris conscience qu'un ordinateur pouvait manipuler de l'information symbolique : traitement logique effectué sur des codes. C'est alors le début de l'informatique de gestion mais aussi le début des idées en Intelligence Artificielle, la naissance de LISP (1961), l'époque du rêve de la traduction automatique des langues...

Les difficultés pour écrire des programmes montrent l'importance de la logique. On étudie plus finement les liens entre ordinateurs et logique. C'est l'époque où l'on voit apparaître des travaux très importants sur ces aspects. Cela a donné naissance à des courants de pensée forts : c'est toute l'histoire de la Résolution (ROBINSON 65) avec les raffinements qu'elle a connus dans les années 1970, et qu'elle connait encore ; c'est le courant de pensée qui a permis la naissance de PROLOG. C'est l'époque où l'on prend conscience de l'importance des mécanismes d'inférence. C'est le début de la démonstration de théorèmes avec la thèse de PITRAT (1966) et justement comme support les théories logiques. Cette époque fort riche a dégagé des outils fondamentaux comme l'algorithme d'unification (PITRAT 1966) et (ROBINSON 65). Il s'agit du mécanisme fondamental qui permet aux systèmes de production de fonctionner : il donne avec toute la généralité souhaitée des substitutions de variables permettant de rendre deux formules logiques identiques, lorsque c'est possible.

Cette voie de la logique, de l'inférence, de la déduction, de la démonstration de théorèmes n'a été qu'une facette des mathématiques faites par ordinateurs. Une deuxième voie complètement indépendante est consacrée au calcul algébrique. C'est l'époque durant laquelle fut acceptée l'idée qu'un ordinateur pouvait faire autre chose que du calcul numérique. La disponibilité de FORMAC (BOND 64) dès 1964 est significative de cette approche. Il s'agissait de disposer effectivement de moyens pour conduire un calcul formel. L'idée de réduire les incertitudes de calcul liée à la propagation des erreurs d'arrondis n'était pas étrangère à ce type de développement. Il était (et reste) séduisant de disposer comme résultat d'un calcul d'une formule litté-

TABLEAU I : PRINCIPAUX SYSTEMES ET PROGRAMMES DE CALCUL FORMEL

Nom du Logiciel et Auteur	S=Syst. SD=syst. diffusé P=prog.	Domaine d'application	Langage de Base	Technique de Programmation I.A. uniquement I.A localement Algorith. unique.	Usage B : trait. par lot I : interactif	Références Biblio.	Observation
SAC 1	SD	Polynômes et fractions rationnelles	FORTRAN	ALGO	B	(COLLINS 71-73)	Accessible Strasbourg
SAC 2	SD	"	ALDES (sur PASCAL)	ALGO	B		Nelle version de SAC-1
MACSYMA	SD	Général	MAC-LISP	IA localement	I + B	(MARTIN 71) (FATEMAN 71)	accessible INRIA
FORMAC	SD	Général	FORTRAN puis PL 1	Algo.	B	(BOND 64)	accessible CIRCE
FORDECALL	S	Général	PL 1	Algo.	I	(LAPLACE 73)	accessible CIRCE
REDUCE (A. HEARN)	SD	Général	R-LISP	Algo.	I + B	(HEARN 71)	Version nelle en cours de diffusion accessible CIRCE
SCRATCHPAD	SD	Général				(JENK 79/ DAVENPORT 80)	Version IBM 370
AMP	S	Général	Assembleur IBM 370	Algo.	I	(DROUFFE 81)	Accessible CIRCE
SYCOPHANTE (Bergman.kanoui)	S	surtout calcul intégral	PROLOG			(BERGMAN 78)	
MU-MATH 79	S	général restreint	MU-LISP	Algo.	I	(RICH/STOUTMEYER 79)	version micro-ordinateurs
DI SCALA	S	SAC-1 restreint	PASCAL	Algo.	I	(DI SCALA 82)	version réduite de SAC-1 sur Apple II

rale permettant de mieux apercevoir le rôle des différents paramètres.

Ainsi l'on constate du point de vue historique que l'activité mathématique avec les ordinateurs s'est faite suivant deux voies qui se sont pratiquement ignorées l'une l'autre : celle de la démonstration de théorèmes et celle du calcul symbolique. Ce que l'on peut voir aussi c'est que chacune de ces deux voies a buté et bute encore sur des problèmes relevant de l'autre voie. Les outils et méthodes dégagés dans chaque voie sont devenus étrangers et difficilement utilisables d'une voie dans l'autre.

Un premier travail consacré à la vérification d'identités en raisonnant par récurrence <VIVET 73> nous a permis sur un domaine restreint de prendre conscience de la nécessité d'allier raisonnement et calcul, des possibilités ainsi ouvertes et des problèmes à résoudre pour réaliser un système mathématicien.

I-2 LA VOIE DU CALCUL ALGEBRIQUE

Cette voie a été marquée par la création d'une famille de logiciels spécifiques. L'objet de ce paragraphe est uniquement de faire une synthèse des idées essentielles dans ce domaine. Au plan du vocabulaire l'on distinguera entre SYSTEMES qui correspondent à un ensemble de programmes généraux, intégrés, diffusés et PROGRAMMES qui correspondent à des logiciels plus spécifiques composés d'un programme et des sous-programmes associés pour résoudre un type de problème particulier. Ces PROGRAMMES correspondent généralement à un logiciel de recherche et sont très rarement diffusés et accessibles.

I.2-1 POINTS DE VUE sur les logiciels existants

Le tableau I fait une synthèse des programmes existants. Selon le critère que l'on prend pour observer ces logiciels, les analyses suivantes peuvent être faites.

a) DEGRE DE GENERALITE DU LOGICIEL

i) Systèmes généraux. Il s'agit de systèmes capables de manipuler des objets mathématiques généraux, acceptant parfois pour cela divers types de représentations internes (avec les problèmes de conversion que cela peut poser), ou n'admettant qu'une structure générale de listes (avec les problèmes d'inefficacité que cela peut amener).
Citons FORMAC(BOND 64) , MACSYMA(MARTIN 71), REDUCE(HEARN 71), SCRATCHAPD(GRIESMER 75), AMP(DROUFFE 81)

Dans le chapitre IX (textes d'accompagnement) est joint un arti-

SIN	P	Calcul intégrales (MACSYMA)	lisp .	IA	I	(MOSES 71)	
SAINT	P	Calcul d'intégrales		IA	I	(SLAGLE 63)	
NORMAN	P	Calcul d'intégrales algo. de RISCH	REDUCE	Algo	I	(NORMAN 77)	
BELOVARI	P	Contour d'intégrales (méthode résidus	PROLOG				
DALI (Laurent)	P	Calcul de limites	FORTRAN	IA uniquement	B	(LAURENT 72-73)	
WANG	P	Calcul de limites				(WANG 71)	
HARRINGTON	P	Calcul de limites	REDUCE	algo uniquement		(HARRINGTON 79)	
CLAYBROOK	P	Factorisation de Polynômes	FORTRAN	IA uniquement	B	(CLAYBROOK 76)	
SEME (M. BARON)	P	Sommations formules combinatoires	FORTRAN	IA uniquement	B	(BARON 81-82)	
GRANDBASTIEN	P	Résolution d'équations trigonométriques	FORTRAN	IA uniquement	B	(GRANDBASTIEN 74)	

cle que nous avons présenté au colloque du GR22 à Caen (sept 80) et qui fait le point sur les possibilités effectives des principaux systèmes généraux de calcul algébrique.

ii) Systèmes spécialisés, dans les manipulations concernant les polynômes et les fractions rationnelles. Citons SAC1/SAC2 <COLLINS.71>
Le fait de pouvoir disposer de représentations internes parfaitement adaptées pour des objets bien formalisés permet d'atteindre de bonnes performances dans la spécialité grâce à des algorithmes puissants adaptés aux objets et représentations. Par exemple dans SAC1 et SAC2 il est fait largement appel à l'arithmétique modulaire pour travailler sur les nombres entiers nécessitant beaucoup de chiffres significatifs. Cela permet par exemple des algorithmes de PGCD de polynômes très performants.

iii) Programmes spécialisés. Citons les travaux et auteurs suivants :

CLAYBROOK(CLAYBROOK 76) spécialisé dans la factorisation de polynômes multivariés
GRANDBASTIEN (GRANDBASTIEN 74) programme spécialisé dans la résolution d'équations trigonométriques
LAURENT (LAURENT 72, LAURENT 73) HARRINGTON(HARRINGTON 79) WANG (WANG 71) programmes spécialisés sur le calcul de limites de fonctions numériques.
MOSES (MOSES 67, MOSES 71), HARRINGTON (HARRINGTON 79), DAVENPORT(DAVENPORT 79)
SLAGLE (SLAGLE 63), RISCH(RISCH 69 et RISCH 70) BERGMAN et KANOUI(KANOUI 73) NORMAN(NORMAN 77) pour le calcul intégral
BELOVARI(BELOVARI 81-82a-82b) pour un programme spécialisé sur la détermination des contours d'intégration, pour la méthode des résidus.
BARON (BARON 81-82a-82b). Le programme général SEME a été testé en particulier pour des calculs de sommation de formules combinatoires.

b) nature du langage de base

Pratiquement tous les systèmes utilisent des structures de listes à des degrés divers. Les langages FORTRAN, PL1, PASCAL, LISP sont généralement utilisés :
- Pour la représentation des nombres en précision infinie (cas de FORMAC écrit en FORTRAN pour PL1 (nombres entiers de 2295 chiffres), SAC1/SAC2 , MACSYMA, REDUCE).

- Pour la représentation des expressions.
  . SAC1 listes en FORTRAN avec compteurs de références (pas de "récupération de miettes" ou garbagge collector)

  .SAC2 listes en ALDES (sur langage de PASCAL) avec récupération de miettes.

- Pour la représentation des données et des programmes : programmes écrits en LISP (MACSYMA, MU - MATH pour les micro-ordinateurs) ou en R-LISP (REDUCE).

.autres langages

- AMP, écrit en assembleur IBM 370, est difficile à classer selon ces critères.
- SYCOPHANTE (BERGMAN78) est écrit en PROLOG avec l'idée d'utiliser la logique du 1er ordre comme langage de programmation.

c) Mode d'utilisation : traitement par lots /traitement interactif

Pratiquement pour tous les systèmes, l'utilisateur peut travailler de façon interactive ; c'est dire que généralement les concepteurs ont été conscients de la nécessité de faire conduire un calcul par l'usager. La seule exception est la première version de FORMAC conçue pour le traitement par lots à une époque où les systèmes d'exploitation des ordinateurs étaient certes encore pauvres ; La raison est surtout que l'on croyait pouvoir conduire des calculs formels par des voies purement algorithmiques comme l'on conduit des calculs numériques. Les problèmes propres au calcul symbolique (non unicité d'un résultat liée au problème de simplification, problèmes de forme externe la mieux adpatée à la poursuite d'un calcul...) étaient encore mal cernés. Ce n'est que plus tard (MOSES 71a) que ces problèmes se sont clarifiés, que le rôle du contrôle par l'utilisateur a été perçu. Ce contrôle concerne les décisions à prendre à propos de la simplification, de la présentation finale d'un calcul ne serait ce que pour en permettre la lisibilité à divers points de vue (lisibilité humaine, lisibilité par le système : expressions "mises à plat" : dans un fichier pour relecture ultérieure, lisibilité par un compilateur (FORTRAN par exemple) pour une évaluation numérique éventuelle,....)

d)Interaction calcul formel - calcul numérique

Des travaux intéressants ont été faits permettant de relier calcul formel et calcul numérique. L'idée de départ est de constater que le calcul numérique dans certains cas se ramène à l'évaluation d'une fonction numérique transcrite sous forme d'une expression arithmétique compilable dans un langage adapté, disons FORTRAN. Il peut y avoir intérêt à manipuler formellement la fonction pour améliorer son écriture et parfois réduire de façon importante les ressources nécessaires pour son évaluation numérique.

Exemple La résolution d'une équation $f(x) = 0$ par la méthode itérative de NEWTON amène à itérer un calcul de la forme

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Le calcul suppose que l'on puisse calculer la valeur de la dérivée f(xi) au point xi connaissant l'expression de f (x) . Si f(x) est formellement compliquée il peut y avoir quelques difficultés à calculer les f'(xi). Même si l'on parvient manuellement à coder l'expression arithmétique représentant cette dérivée, le code devient dépendant de l'équation à résoudre et l'on perd en généralité. Les problèmes apparaissent s'il y a un

grand nombre d'équations à résoudre.

Des possibilités faibles existent en REDUCE pour résoudre ce type de problème : la commande ON FORT permet d'assurer la sortie des expressions dans une forme FORTRAN syntaxiquement correcte. Ce qui permet avec des ordres d'écriture adaptés de construire des fichiers contenant des programmes FORTRAN corrects. Pour l'exemple de l'itération de NEWTON, il est ainsi possible de donner formellement f(x), de calculer formellement f'(x) et de générer le programme FORTRAN calculant la racine. La jonction entre calcul numérique et calcul formel se fait par fichier de code interposé.

La même idée peut être traitée de façon plus intéressante en intégrant directement dans un langage type FORTRAN les outils de calcul formel nécessaire. Il suffit d'ajouter les possibilités de manipuler un type ALGEBRAIC dans lequel on pourrait faire le calcul symbolique avec des fonctions appropriées. Cela suppose l'écriture des programmes d'entrée-sortie adaptés, de fonctions de conversions....

L'on peut regretter que ce type de possibilité ne soit pas disponible dans les langages. Ceci fait que des utilisateurs peuvent être amenés à faire de mauvais choix au niveau méthode. Dans le cas de l'itération de NEWTON, le mauvais choix typique consiste à tabuler f(x), calculer des valeurs de f'(x) en des points pivots par des méthodes purement numériques (différences finies...) puis travailler par interpolations successives pour avoir les f(x) et f'(xi) intéressants. Ce qui est grave, c'est que cette solution peut tenter un utilisateur qui possède dans sa bibliothèque de sous-programmes FORTRAN les programmes de dérivation numérique et d'interpolation nécessaires.

Les systèmes de calcul algébrique intègrent généralement tous les éléments nécessaires mais ont de mauvaises performances en matière de calcul numérique et il n'est pas actuellement envisageable de faire traiter des calculs numériques importants par un sytème de calcul formel.

e) Micro-ordinateurs et calcul formel

La plupart des systèmes ont été écrits sur des ordinateurs puissants (PDP10, IBM 370,....) accessibles en temps partagé. Des systèmes restreints sont apparus récemment pour des micro-ordinateurs. Citons MU-MATH(RICH 79) pour les machines à base de Z80 et le travail de M. di SLALA(di SLALA 82) pour une implémentation de SAC-1 en PASCAL sur APPLE II;
L'idée de faciliter l'accès à un système de calcul formel en travaillant avec des machines portables et de grande diffusion est intéresssante. On peut rêver de pouvoir un jour disposer dans sa poche de calculatrices permettant le calcul symbolique. Qui pensait il y a une vingtaine d'années qu'on disposerait de machines programmables en BASIC sous cette forme ?. Il faudra résoudre des problèmes de capacité mais surtout le problème des moyens d'affichage adaptés : le résultat d'un calcul symbolique peut nécessiter beaucoup de symboles pour être écrit avec une typographie délicate (exposants, fractions,). L'affichage d'un résultat numérique pose évidemment beaucoup moins de problèmes.

Les micro-ordinateurs 8 bits actuels manquent de puissance de calcul (vitesse, capacité mémoire limitée à 64 Ko) pour faire correctement du calcul formel. A brève échéance, les micro-ordinateurs conçus avec des processeurs 16 bits vont permettre d'atteindre une vitesse suffisante et une capacité mémoire raisonnable pour ce type de travail (500 Ko à 1000 Ko sont dans les possibilités de ce type de machine. Cela correspond à l'espace que l'on occupe couramment sur un système en temps partagé pour ce type de travail).
Ce type de machine nous apparaît comme la bonne voie pour diffuser des systèmes de calcul performants auprès des utilisateurs qui en ont besoin (laboratoire, bureau d'étude,....).

Notons que REDUCE que nous avons utilisé pour réaliser CAMELIA (cf chapire IV) vient d'être annoncé sur une telle machine Apollo DOMAIN Aegis 5.0 basée sur le micro-processeur MC 68000 de Motorola.
Une version portable de R-LISP (appelée PSL pour Portable Standard LISP) est maintenant disponible ce qui permet tous les espoirs pour réaliser assez facilement ce type d'installations.

Une autre voie possible avec le même type de matériel serait de miser sur des machines dédiées : partir d'une machine LISP pour implanter une réduction de MACSYMA ou REDUCE ou concevoir des machines microprogrammées avec des instructions adaptées au calcul formel.
Là encore, les technologies maintenant disponibles laissent beaucoup d'espoir. Des études sont commencées (DAVENPORT 83a-83b) pour réaliser des processeurs spécialisés pour le calcul formel avec des techniques VLSI. On conçoit bien tout le gain de puissance qui pourrait être obtenu avec des coprocesseurs réalisant du PGCD de polynomes formels par exemple. Il apparaît que les travaux s'orientant dans cette voie pourraient déboucher à terme vers la production de machines dédiées au calcul formel.
Alors que la première idée ne relève que d'un développement de logiciel calqué sur des logiciels existants, la seconde nécessite une recherche du point de vue de l'architecture du processeur et des instructions dont il faudrait le doter.

f) Type de programmation : Algorithmes/programmation heuristique

Indépendamment du langage de programmation utilisé, les systèmes et programmes peuvent être observés par rapport à l'utilisation ou la non utilisation d'heuristiques pour guider la conduite des calculs et réduire les ressources nécessaires. Ainsi, ne relèvent à notre sens de l'intelligence artificielle que les systèmes et programmes utilisant de façon explicite des heuristiques permettant d'améliorer, parfois de rendre possible, la conduite d'un calcul. Cette réduction de ressources peut intervenir sur les deux paramètres essentiels :

- du temps : apercevoir un cas particulier autorisant une simplification importante peut faire gagner beaucoup de temps.
- de l'espace : il est parfois possible de contrôler avec des heuristiques l'explosion des expressions intermédiaires qui est souvent catastrophique avec des méthodes purement algorithmiques.

Citons quelques exemples :(Ceux-ci seront détaillés plus loin - cf II.2.2)

- M. GRANDBASTIEN <GRANDBASTIEN 74> a traité la résolution d'équations trigonométriques par une approche heuristique alors qu'une méthode algorithmique est possible. Ce travail est tout à fait significatif de l'intérêt d'utiliser des heuristiques.

- Pour le calcul intégral, le programme SIN <MOSES 71> utilise des heuristiques pour analyser la forme de l'intégrande et choisir la méthode appropriée. Ce programme qui opère lors de la commande d'intégration de MACSYMA est à comparer au programme de NORMAN <NORMAN 77> qui utilise une version de l'algorithme de RISCH Ce programme accessible avec les versions récentes de REDUCE n'intègre que les fonctions de la classe admise à l'entrée de cet algorithme. Cette classe est heureusement assez large pour traiter un grand nombre des intégrales usuelles.
Une comparaison de performances (ressources, classes d'intégrales possibles,...) mériterait d'être faite afin de juger effectivement de l'intérêt de chaque approche.

- Pour le calcul de limites, le programme DALI <J.P. LAURENT 72-73> a eu de bonnes performances avec des méthodes heuristiques pour lever des indéterminations. Les performances seraient à comparer aux programmes de HARRINGTON <HARRINGTON 79> écrit sous REDUCE mais non disponibles dans le système, et de WANG <WANG 71>.

- Pour la factorisation de polynômes, CLAYBROOK <CLAYBROOK 76> a obtenu de bons résultats pour les polynômes multivariés avec un programme FORTRAN utilisant abondamment des heuristiques propres au domaine. Dans le système SAC 1/2, COLLINS utilise l'algorithme de BERLEKAMP pour obtenir des résultats apparemment moins riches.

On peut observer que dans les systèmes actuels, ce sont essentiellement des algorithmes qui sont implantés. Les

méthodes heuristiques sont le fait de programmes qui restent isolés. La seule exception significative semblant être le programme SIN <MOSES 71> intégré à MACSYMA.
MACSYMA est d'ailleurs le seul système important a avoir pris en compte les idées de l'intelligence artificielle. On y trouve par exemple des outils de reconnaissance de modèles (Pattern matching) non disponibles dans les autres systèmes.

L'évolution historique d'un système comme MACSYMA est intéressante à ce stade et montre bien l'évolution de l'équilibre entre ce qui est traité avec des algorithmes et ce qui est traité avec des heuristiques. Ainsi MACSYMA est né dans un laboratoire d'intelligence artificielle, les premiers programmes ont pris en compte abondamment la démarche heuristique. L'exemple de SIN <MOSES 67 - MOSES 71> est significatif et ce n'est que plus tard (et pour cause !) que MACSYMA a utilisé une implantation de l'algorithme de RISCH. Il serait faux de penser que MACSYMA s'est développé en écartant les aspects heuristiques au profit des aspects algorithmiques. En fait MACSYMA a pu assimiler la puissance des algorithmes là où c'était possible mais parallèlement les outils typiquement intelligence artificielle comme la reconnaissance de formes ont été développés.

Pour les programmes relevant de l'intelligence artificielle, les heuristiques utilisées sont plus ou moins formalisées. Très souvent programmées dans les réalisations plus anciennes, elles deviennent de plus en plus formalisées, isolées des programmes et codées sous forme de règles. L'on arrive ainsi aux idées actuelles des systèmes experts en calcul algébrique. Le programme SEME de M. BARON <BARON 81-82a-82b> est une première réalisation intéressante avec ces idées testées sur un domaine limité : l'évaluation de formules sommatoires en analyse combinatoire.

I-2-2 CONCLUSIONS

Le calcul symbolique s'avère un domaine difficile. Les approches par des méthodes algorithmiques et par des méthodes de type heuristique ont permis d'obtenir un certain nombre de résultats qui peuvent rendre de bons services aux utilisateurs. Il reste que les systèmes existants n'ont pas l'audience qu'ils pourraient avoir auprès des utilisateurs potentiels sans doute par manque d'information mais aussi parce que ces systèmes sont délicats à utiliser.

a) Problème de simplification

Une des difficultés majeures vient du fait qu'en calcul symbolique, le résultat d'un calcul n'est jamais unique contrairement à ce qui se passe en calcul numérique. Ce résultat peut prendre plusieurs formes équivalentes du point de vue mathématique mais qui ne le sont pas du point de vue de l'usage qui doit en être fait. On peut choisir le résultat le plus simple seulement le concept de simplification n'est pas absolu !
La formule la plus simple est bien souvent celle qui met le mieux en évidence ce que l'on a envie d'y voir (termes importants, structure...). Ainsi il n'est pas possible dans l'absolu de dire qu'en sortie il faut mettre en facteur les termes qui peuvent l'être ou bien distribuer les produits sur les sommes : selon l'usage que l'on fera du résultat, c'est l'une ou l'autre

des formes qui apparaîtra comme la plus simple. La fonction $f(x)=tg\ x$ peut apparaître de façon plus intéressante sous la forme $f(x) = \frac{\sin x}{\cos x}$ si l'objectif est de calculer $\int f(x)dx$.

La deuxième forme présente l'avantage de montrer la combinaison d'une fonction (cos x) et de sa dérivée (sin x) : Combinaison que l'on sait bien intégrer !.

b)Problèmes de lisibilité.

Lisibilité par l'utilisateur.

A cet aspect relatif du concept de simplification s'ajoute un problème de lisibilité. Il n'est pas rare qu'un résultat nécessite plusieurs milliers de symboles. Les problèmes de mise en page (bien résolus maintenant dans les bons systèmes) de typographie (place des exposants, empilements de dénominateurs...) méritent qu'on leur prête beaucoup d'attention, en particulier en mode intéractif où l'utilisateur doit voir rapidement ce qu'il cherche. De bonnes idées ont vu le jour à ce sujet : par exemple en REDUCE, il est possible d'afficher une somme de termes à raison d'un terme par ligne (si les termes sont nombreux et volumineux cette possibilité est très pratique).

Lisibilité par le système

Le système peut être amené à relire des expressions stockées temporairement dans des fichiers. Ces expressions ne doivent pas être stockées sous forme "naturelle" c'est-à-dire telles qu'affichées à un écran ou sur une feuille de papier.
Ainsi le résultat :

$\frac{e^{x} + e^{-x}}{2}$ sera stocké sous la forme (e**x+e**-x)/2

Lisibilité par un compilateur

Un système comme REDUCE permet de sortir des expressions sous une forme FORTRAN syntaxiquement correcte. Ces expressions mises en fichier peuvent alors être compilées directement. Ce travail est surtout intéressant si l'on manipule de grosses expressions. Se posent alors des problèmes de découpage de l'expression, dans des cartes suites (débordement éventuel du nombre de cartes suites admises par le compilateur, non découpage des identificateurs de fonctions...).
Tous ces problèmes de lisibilité ne concernent que l'aspect extérieur des expressions et sont traités par des programmes adaptés travaillant sur la représentation interne fixe. Les décisions et choix se manifestent pour l'utilisateur par le positionnement d'indicateurs qui sont nombreux avec les systèmes riches. Il est parfois difficile de prévoir ce que leur combinaison va produire.

c) Problème d'explosion des expressions intermédiaires

Un problème parfois difficile a contrôler est celui de la croissance des expressions : un calcul peut en effet avoir un résultat de taille raisonnable et avoir

nécessité des formules de tailles beaucoup plus grande pour l'établir (cascades de simplifications...). On peut être amené dans certains cas à utiliser des règles explosives (SIKLOSSY 71-73) (par exemple :

$1 = \cos^2 u + \sin^2 u$, $u = u.1$, $u = u + 0$,...)

L'usage de telles règles ne peut être fait que sous contrôle d'heuristiques qui en limitent l'usage, l'application permanente donne une infinité d'expressions amenant une saturation de l'espace de travail par des expressions gigantesques et bien souvent inutiles. Les systèmes qui ne travaillent qu'avec des algorithmes écartent généralement tout usage de ce type de règles. Celles-ci sont parfois bien utiles.

I-3 LA VOIE DE LA DEMONSTRATION DE THEOREMES

La démonstration de théorèmes a joué un rôle central dans l'évolution de l'intelligence artificielle. C'est très tôt que les besoins de systèmes capables de conduire des preuves se sont exprimés.

I.3.1 La démonstration de théorèmes pour faire autre chose que des mathématiques

Il faut noter que ces besoins n'étaient généralement pas ressentis pour "faire des mathématiques". Citons quelques exemples :

a) Ce sont les chercheurs en robotique, qui exprimaient que pour qu'un robot exécute une tâche, on peut être amené à conduire la preuve que la situation finale (cible) est accessible à partir de la situation initiale (source) à l'aide des seules actions de base (opérateurs) exécutables par le robot. Dans ce contexte, la situation initiale servait d'hypothèse, la situation finale servait de conclusion, les actions de bases étaient les théorèmes que l'on avait le droit d'appliquer. La preuve, une fois établie donnait le séquencement des actes élémentaires nécessaires pour exécuter la tâche.

b) Ce sont les chercheurs qui travaillaient en programmation automatique et preuves de programmes qui révélaient des besoins. L'idée initiale de notre travail sur la récurrence (VIVET 73) est ainsi née d'un article de MANNA WALDINGER (MANNA 71) exprimant le besoin de programmes raisonnant par récurrence pour prouver des programmes comportant des boucles avec indices entiers (boucles DO en FORTRAN, FOR....NEXT en BASIC...).

c) Les chercheurs qui travaillaient sur la synthèse automatique en chimie exprimaient des préoccupations de cet ordre : la situation initiale (hypothèse) était la description de produits disponibles, la situation finale (conclusion) était la description d'un produit dont on cherchait une synthèse. Les théorèmes utilisables, opérateurs applicables, étaient les réactions chimiques connues. La preuve du théorème fournissait l'ordre d'exécution d'un certain nombre de réactions permettant de synthétiser le produit attendu à partir des produits disponibles.

d) Dans le domaine des jeux, on cherchait a prouver qu'une situation était accessible à partir d'une situation donnée en utilisant les opérateurs "coups légaux" etc...

I.3.2 La démonstration de théorèmes en mathématiques

Les besoins exprimés de programmes capables de conduire des preuves pour des applications hors du champ des mathématiques ont conduit à tout un travail en démonstration de théorèmes en mathématique. Les raisons possibles sont liées au fait que les mathématiques travaillent avec des objets généralement mieux formalisés que c'est dans ce domaine que l'on perçoit le mieux ce qu'est une preuve,...Il faut dire aussi que dans ce domaine on disposait de résultats théoriques (HERBRAND 30) que l'on espérait pouvoir utiliser.

a) La logique. La logique mathématique a été un domaine particulièrement étudié avec ce type de perspective. Nous reviendrons (cf II-3-1 et II-3-2) sur ce qui a été fait dans ce domaine avec une approche de type algorithmique ou une approche de type heuristique (preuves naturelles). Citons seulement deux noms essentiels : celui de ROBINSON (ROBINSON 65) et celui de PITRAT (PITRAT 66).

b) Analyse, vérification de preuves. Un traitement important a été fait dans le domaine de l'analyse ou de la vérification de pqreuves déjà établies. Citons les travaux de BUNDY (BUNDY 75) DE BRUIJN (DE BRUIJN 68) et de ARNOLD (ARNOLD 70). L'idée de A. ARNOLD a été par exemple de concevoir un langage permettant de décrire la preuve d'un théorème et d'écrire un programme permettant de vérifier que la preuve écrite dans ce langage est juste, le programme fournissant les étapes intermédiaires et les justifications nécessaires à chaque pas.

c) Variété des domaines mathématiques dans lesquels un travail a été fait en Démonstration de Théorèmes.
Des domaines mathématiques très variés ont servi de support à ces travaux en démonstration de théorèmes. La description de certains sera faite en II-3. Contentons nous pour l'instant d'une énumération (sans doute non exhaustive) pour faire apparaître l'ampleur et la diversité des travaux.

- Géométrie (GELERNTER 59), (BUTHION 75) pour la construction de figures en géométrie.
- Arithmétique (DALLARD 74), (BOURGOIN 78), (LENAT 75 à 77) pour la découverte en mathématique, (BUNDY 73)
- Algèbre (DURAND 75), (HERZ 75), (GILLET 79) pour des travaux sur groupes, anneaux, corps.
- Théorie des ensembles (MERIALDO 79), (PASTRE 76)
- Analyse (BLEDSOE 72), (PASTRE 82b)
- Combinatoire, jeux : (LAURIERE 76), PITRAT

I-4 CONCLUSION

On constate qu'un travail important a été fait en mathématique avec les ordinateurs, en dehors du domaine particulier du calcul numérique. Deux voies principales caractérisent ce travail : celle du calcul algébrique et celle de la démonstration de théorèmes. L'analyse des travaux faits laisse apparaître que ces voies se sont déroulées de façon parallèles et

indépendantes et l'on observe aujourd'hui la situation suivante

On dispose de systèmes de calculs algébriques apparemment performants mais qui sont incapables d'inférer quoi que ce soit sur les calculs qu'ils font. Il font une conduite aveugle des calculs qu'on leur confie. Cela oblige le plus généralement l'utilisateur à conduire lui-même le calcul en usage intéractif du système. C'est l'utilisateur qui décide de l'enchaînement des actions essentielles.
On dispose de systèmes de démonstration de théorèmes performants au plan du raisonnement dans le domaine pour lequel ils ont été conçus, mais ces systèmes sont toujours très pauvres au plan des capacités calculatoires. La plupart sont incapables de conduire un calcul algébrique ou numérique dont la nécessité a pu apparaître au cours d'un raisonnement.
Ces deux observations font à notre avis apparaître une des raisons d'un certain plafonnement des performances dans chacune des voies. Il est certain que le mathématicien expert doit être complet et avoir la double compétence et intégrer les deux voies : le calcul est un outil au service d'un raisonnement et ne pas être capable de calculer à certains stades d'une chaîne d'inférence est un handicap profond. Réciproquement le raisonnement est une aide à la conduite d'un calcul et calculer de façon aveugle, sans tenir compte du raisonnement en cours, des raisons qui ont amené la nécessité du calcul, de l'usage qui sera fait du résultat du calcul, est souvent une gêne importante.N'oublions pas que si pour un calcul numérique, le résultat à souvent une expression unique, pour un calcul algébrique symbolique, le résultat peut prendre des formes mathématiquement équivalentes mais formellement très différentes. C'est généralement le contexte d'usage qui permet de choisir celle qui est la plus appropriée.

CHAPITRE II.-

LES VOIES DE L'ALGORITMIQUE

LES VOIES DE LA PROGRAMMATION HEURISTIQUE

II - LES VOIES DE L'ALGORITHMIQUE, LES VOIES DE LA PROGRAMMATION HEURISTIQUE.-

II - 1 - GENERALITES

Il s'agit dans cette partie de faire apparaître que deux démarches sont possibles pour résoudre avec un ordinateur un problème accompagné de ses données.

II - 1 - 1 - La démarche de l'algorithmique.

a) Cette démarche consiste à caractériser les données par leurs aspects structuraux, organisationnels. Eventuellement par généralisation conservant les structures, il s'agit de définir une classe de situations semblables à la situation fournie. Cette classe aussi large que possible admet le jeu initial de données comme cas particulier. Le problème est alors de concevoir un algorithme général qui fonctionne de façon absolue pour tout représentant de la classe d'objets ainsi définie.

- Le fonctionnement absolu est caractérisé par la garantie d'obtention d'un résultat sitôt que les entrées sont choisies dans la classe des objets admissibles et par la possibilité d'estimer de façon correcte l'efficacité du processus amenant le résultat (appréciation des bornes de ressources,(espace mémoire , temps) nécessaires).

b) Exemple : Soit (Pb1) à calculer la valeur de la fonction

$f(x) = 4x^3 + 3x^2 + 5x + 9$ pour $x = 7$

La situation de ce problème peut être caractérisée par le fait qu'il s'agit de calculer la valeur d'un polynôme de degré 3 à coefficients entiers pour la valeur entière x = 7.Le problème est à ce stade structurellement bien reconnu.

- Par généralisations successives, extensions des ensembles de nombres concernés, il est possible, sans modifier profondément la structure,

  i) d'étendre le degré 3 à tout entier naturel

  ii) d'étendre l'ensemble (4, 3, 5, 9) des coefficients à l'ensemble des entiers N, puis des nombres réels R

  iii) d'étendre l'ensemble (7) auquel appartient la valeur pour laquelle on calcule la fonction à N puis à R.

. On obtient alors la classe C de problèmes pouvant se ramener à un énoncé de la forme :

(PBG) Calculer la valeur de la fonction

$$f(x) = a0\, x^n + a1\, x^{n-1} + \ldots + an$$

(avec $\forall i,\ ai \in R$) pour la valeur xo ($\forall xo \in R$)

. Le problème (Pb1) fait évidemment partie de C .

. Il est possible de fabriquer un algorithme général A (schéma de HORNER par exemple) permettant de résoudre tout problème de la Classe C. Cet algorithme permet d'atteindre la solution de Pb1.

. L'efficacité de A peut être mesurée de façon correcte par une fonction e(n) (où n est le degré du polynôme)

- l'espace mémoire nécessaire au stockage du polynôme (si on le stocke !) est de la forme (n + 1). e où e est l'encombrement pour un coefficient.

- le temps est de l'ordre de n*(tM + tA) où tM est le temps élémentaire d'une multiplication et tA le temps élémentaire d'une addition.

c) Intérêt de cette démarche.-

La démarche de l'algorithmique est intéressante toutes les fois où elle est possible pour la sécurité qu'elle donne sur un problème particulier et la généralité qu'elle permet souvent d'obtenir : possibilité de résoudre toute une classe de problèmes alors qu'on cherchait à résoudre un problème particulier.

d) Limite de la démarche algorithmique.

- Pour certains problèmes, il est difficile de fabriquer une classe de problèmes beaucoup plus grande que celle réduite au problème initial ; ou alors cela se fait au prix d'une complication des structures telle que l'algorithme fabriqué est beaucoup trop complexe (et coûteux en ressources le plus souvent) pour résoudre le problème initial. Il n'est d'ailleurs pas certain que généralement les autres problèmes de la classe fabriquée présentent quelque intérêt.

Il est admis que seuls les problèmes de complexité polynomiale ($\Theta(n^{**}p)$) relèvent du champ de l'algorithmique et des chercheurs comme J.L. LAURIERE n'hésitent pas à définir ainsi la limite entre programmation heuristique (caractéristique de l'intelligence artificielle) et programmation d'algorithmes (caractéristique de la programmation traditionnelle).

Exemple : le problème du placement des 8 dames sur un échiquier peut être généralisé toujours en dimension 2 au placement de n dames sur un échiquier n x n, puis au placement de n dames sur un échiquier m x p (avec n = min(m,p)). En étendant ensuite la dimension on pourrait se poser le problème de placer n dames dans un cube (m, p, q) avec n = min (m, p, q) puis dans des hypercubes

$\prod_{i=1}^{d} Ci$ avec n = min (Ci) $i \in (1,d)$.

On définit ainsi une classe de problèmes pour laquelle la découverte d'un algorithme général permettrait de résoudre le problème initial des 8 dames. Il ne semble pas raisonnable de se lancer dans la recherche d'un tel algorithme pour résoudre ce problème.

- Réciproquement, étant donné un problème, on peut connaître un algorithme général admettant ce problème dans sa classe d'entrée. On a alors la certitude de pouvoir résoudre le problème avec des ressources évaluables. Pourtant on peut dans certains cas préférer ne pas appliquer l'algorithme pour tenter d'utiliser moins de ressources.

Exemple : Soit (Pb2) à calculer la valeur de la fonction

$$f(x) = x^{1000} + 1 \text{ pour } x = 1$$

L'algorithme A précédent permet de résoudre le problème (Pb 2) avec certitude : il n'est pas raisonnable d'utiliser l'algorithme de HORNER pour résoudre Pb2 : l'aspect particulier des données permet de trouver 2 sans se lancer dans 1000 additions et 1000 multiplications. On peut conclure que pour résoudre un problème particulier, il faut tenir compte des données ; une analyse des données permettant parfois de conclure rapidement.

II - 1 - 2 - La démarche de la programmation heuristique.

a) La deuxième démarche consiste à considérer qu'un problème est un tout et que les données particulières sont caractéristiques du problème à résoudre et ne doivent pas être oubliées par le mécanisme des extensions, généralisations caractéristiques de la démarche algorithmique. Une analyse de ces données peut alors amener une simplification importante au point de pouvoir conclure sans avoir à lancer l'exécution d'un algorithme général qui peut être très lourd par rapport au problème courant. Le problème (Pb2) précédent est caractéristique à ce point de vue. L'analyse du problème, du jeu particulier de données pour lequel est posé le problème, est importante avant toute décision d'action. Le phénomène est visible sur des exemples simples qui peuvent faire apparaître des économies de ressources importantes. Le phénomène est plus apparent encore pour les problèmes complexes. Pour de tels problèmes la généralisation de situations, la recherche d'algorithme général peut être difficile voire impossible. Pourtant, le problème peut rester soluble si l'on se place dans le contexte d'un jeu particulier de données.

Rappelons les problèmes d'élaboration d'emplois du temps que les proviseurs d'établissements scolaires résolvent chacun dans leur cas, par adaptation aux situations locales, propres à chaque établissement. Pourtant l'on sait (LAURIERE 71) la difficulté d'écrire un programme général basé sur un algorithme spécifique pour faire des emplois du temps. La solution de tels problèmes peut par contre être apportée par des programmes généraux de résolution de problèmes capables de manipuler des contraintes formellement (LAURIERE 76).

Cette seconde démarche où l'on tente une analyse de la situation en tenant compte des données actuelles du problème est sans doute caractéristique de l'intelligence artificielle. La reconnaissance de situations particulières se fait en utilisant des critères qui permettent d'observer tel ou tel aspect particulier des données. Ces heuristiques qui permettent de filtrer les aspects positifs ou négatifs de la situation fournie favorisent les prises de décision, les choix.

b) Lors de la résolution d'un problème, les décisions et choix relèvent en particulier des aspects suivants :

i) - Choix de l'algorithme à appliquer : si plusieurs algorithmes existent le problème du choix de l'algorithme à utiliser se pose.

Les critères d'appréciation sont divers et nombreux :
. Ressources utilisées : espace/temps
. Généralité de la classe de données autorisées à l'entrée
. disponibilité en bibliothèque d'un programme tout fait codant cet algorithme

ii) - Choix du moment où l'on applique l'algorithme.

Les données telles qu'elles sont fournies peuvent ne pas se prêter directement à l'exécution d'un algorithme efficace. Il peut y avoir intérêt à faire subir un prétraitement à ces données pour se ramener dans un cas d'application d'algorithme plus favorable.

Exemple : Soit à calculer la valeur d'un polynôme P(x) de degré n élevé (disons n>10), dont tous les coefficients sont non nuls pour un ensemble V de 100 valeurs de la variable. S'il s'avère que V est symétrique par rapport à l'origine, ce qui peut se voir facilement en analysant les valeurs dans V, il peut être judicieux de décomposer P(x) en une somme de deux polynômes P1(x) et P2 (x) : P1(x) (resp.P2 (x)) ne contenant que les termes de degré impair (resp. pair).

Il est alors possible d'exploiter le fait que si
P(v) = P1(v) + P2(v) alors P(-v) = -P1(v) + P2(v)
Un calcul séparé, avec l'algorithme de HORNER de P1 (v) et P2 (v)permet d'obtenir immédiatement la valeur de P(-v) quand le calcul a été fait pour P(v).

Le calcul peut être amélioré pour P1(v) et P2(v) en évitant les opérations avec les coefficients nuls intermédiaires.
Il suffit d'observer que :

$$P1(x) = a1\,x + o\,x^2 + a3\,x^3 + \ldots + o\,x^{2k} + a_{2k+1}\,x^{2k+1}$$

s'écrit

$$x\,(a1 + a3\,x^2 + \ldots + a_{2k+1}\,x^{2k}$$ c'est à dire $x * Q1(x^2)$

avec

$$Q1(u) = a1 + a3\,u + \ldots + a_{2k+1}\,u^k$$

et

$$P2(x) = ao + ox + a2\,x^2 + \ldots + o\,x^{2k-1} + a_{2k}\,x^{2k}$$

s'écrit

$$Q2(x^2)$$

avec

$$Q2(u) = ao + a2\,u + \ldots + a_{2k}\,u^k$$

Le calcul de P(v) et de P(-v) se ramène alors au calcul de

$Q1(v^2)$ et $Q2(v^2)$ qui ne font pas faire, globalement pour chaque v, plus d'opérations que le calcul de P(v) seul.

Pour l'ensemble V des valeurs pour lequel il faut faire le calcul on gagne donc pratiquement la moitié des opérations en utilisant toujours le même algorithme de base : le schéma de HORNER.

Il est clair sur cet exemple que le fait d'avoir analysé les données, d'avoir différé l'application d'un algorith-

me pourtant directement applicable a permis une réduction importante des ressources nécessaires pour effectuer le calcul. L'analyse des données s'est faite en utilisant des critères simples : la symétrie de V a permis de déclencher des connaissances sur la parité qui ont pu être exploitées utilement.

Cet exemple permet de saisir ce que l'on entend par conduite intelligente d'un calcul ; par opposition à la conduite aveugle qui aurait utilisé l'algorithme de HORNER comme une boîte noire appliqué directement sur le polynôme fourni en entrée.

iii) - Choix de ne pas appliquer d'algorithme.

L'analyse des données peut dans un certain nombre de cas amener à la décision de ne pas appliquer d'algorithme. La situation particulière courante permettant de conclure immédiatement. L'exemple du calcul de

$x^{1000} + 1$ pour $x = 1$ est significatif à ce sujet.

c) Les limites de la démarche heuristique

. limitations dues aux ressources nécessaires pour effectuer l'analyse.

Il est possible d'apercevoir les limites de cette démarche en imaginant des situations où il faudrait dépenser plus de ressources pour détecter une situation particulière permettant une amélioration du calcul que pour l'exécution d'un algorithme général (s'il en existe un qui soit praticable).

Les ressources nécessaires à l'analyse sont de deux types :

.temps : il faut un certain temps pour effectuer une analyse des données du problème et cette analyse peut ne pas aboutir à la découverte d'un raccourci intéressant. Notons que ce temps n'est pas borné comme celui d'un algorithme. Alors que l'on peut évaluer le temps d'exécution d'un algorithme on peut considérer d'une certaine façon qu'une analyse n'est jamais terminée. Il faut alors des critères, pour décider de l'arrêt de l'analyse, autrement dit, apprécier le moment où l'on a trouvé des choses suffisamment intéressantes ou considérer que l'on a passé assez de temps avec des élements sans intérêt exploitable).

Le temps n'évolue pas de la même façon dans la démarche algorithmique et la démarche heuristique. Le travail à fournir, l'efficacité ne progressent pas de la même façon. Avec un algorithme on peut considérer qu'il y a une certaine régularité dans la progression du travail. Avec des heuristiques l'approche du résultat final se fait beaucoup plus irrégulièrement : on peut passer un certain temps à faire une analyse inefficace (qui ne fait pas progresser la solution pratiquement) et tout d'un coup apercevoir le raccourci qui va permettre une bonne efficacité.

. espace. Les connaissances nécessaires pour faire une analyse fine des situations-problèmes qui se présentent à l'entrée doivent être stockées. Cela nécessite une place non négligeable qui s'ajoute à la place nécessaire pour les données du problème et le programme de traitement.

Notons que les connaissances requises concernent aussi bien les caractérisations de situations particulières que les conditions de mise en oeuvre des algorithmes disponibles.

II-1-3 - CONCLUSION

Il semble finalement que c'est un compromis entre les deux démarches qu'il convient d'adopter. Ce compromis consiste à essayer de profiter au mieux des avantages de chaque approche. Vouloir tout traiter en n'utilisant que des algorithmes ou que des heuristiques semble voué à l'échec : l'approche algorithmique pure est inopérante dans un certain nombre de situations complexes. L'approche heuristique doit prendre en compte la sécurité et l'efficacité disponible avec la plupart des algorithmes connus. Le compromis peut s'énoncer ainsi : "Etant donné un problème, le résoudre suppose d'analyser son objectif, son jeu de données particulières. Cette analyse, faite avec des connaissances appropriées, doit permettre de prendre les décisions fondamentales : choix d'un algorithme, choix du moment où l'on applique cet algorithme, nature des modifications de données nécessaires. La décision peut être de ne pas appliquer d'algorithme si l'analyse permet de conclure".

II - 2 - LES VOIES DE L'ALGORITHMIQUE/LES VOIES DE LA PROGRAMMATION HEURISTIQUE EN CALCUL ALGEBRIQUE.

Les deux voies de l'algorithmique ou de la programmation heuristique ont été utilisées pour aborder l'exécution de calcul algébrique par ordinateur. Il est remarquable d'observer que pratiquement aucun auteur n'a travaillé en combinant les deux approches mais que les sujets importants (manipulation de polynômes, calcul intégral, calcul de limites, ..) ont été abordés parallèlement dans les deux voies.

II - 2 - 1 Algorithmique en calcul algébrique

Cette voie privilégiée pour aborder des problèmes simples (dérivation formelle, substitution, calcul de séries,) a permis d'obtenir de bons résultats dans des domaines plus complexes.

1) Domaine de la manipulation de polynômes multivariés

Le travail qui a permis la réalisation des systèmes SAC-1 puis SAC-2 (COLLINS 71), nous apparaît comme particulièrement significatif.

Ce logiciel comprend :

-/- un module d'arithmétique à précision infinie

-/- un module de manipulation de polynômes (multivariés, coëfficients entiers en précision infinie)
algorithmes pour . + - /
- . Entrées sorties
- . substitution/évaluation
- . différenciation
- . PGCD

-/- un module de manipulations de fractions rationnelles
algorithmes pour . + - /
- . Entrées/sorties
- . subsitution
- . Différenciation

-/- un module d'arithmétique modulaire
Ce module fait appel a des résultats mathématiques sophistiqués(coëfficients dans les corps de GALOIS, théorème du reste chinois, représentations non positionnelles des nombres (algorithme de GARNER)).
Il fournit des outillages
- de calcul de PGCD de grands nombres
- de factorisation de polynômes univariés modulaires (algorithme de BERLEKAMP)

Ces outillages sont fondamentaux pour :
- le calcul de PGCD de polynômes multivariés
- les calculs sur les entiers
- la factorisation de polynômes multi-

variés à coëfficients entiers

-/- un module d'intégration des fractions rationnelles (univariées).
algorithme d'intégration : algorithme d'HOROWITZ
sans décomposition en éléments simples
. de décomposition des fractions rationnelles en éléments simples
. résolution de systèmes linéaires à coëfficients entiers

-/- un module de calcul des zéros réels de polynômes (utilisation du théorème de STURM)

-/- un module de calcul de PGCD de polynômes multivariés

-/- un module de factorisation de polynômes univariés à coëfficients entiers.

-/- un module d'algèbre linéaire
algorithme de résolution d'un système d'équations linéaires dont les coëfficients sont des polynômes multivariés à coëfficients entiers de précision infinie.
Le texte "Calcul formel et intelligence artificielle" annexé au chapitre IX des textes d'accompagnement donne plus de détails sur ce système. L'on peut se reporter également aux textes de G.E. COLLINS (COLLINS 71).

CONCLUSION PARTIELLE : Cette voie de l'algorithmique de manipulation des polynômes a été et reste encore très étudiée. Elle permet d'atteindre des résultats intéressants dans un domaine restreint des mathématiques mais qui concerne de nombreux utilisateurs.

Ce domaine est un des centres d'intérêt des chercheurs français regroupés dans le GREGO "Calcul formel" du CNRS. Citons les travaux de D. LAZARD, MIGNOTTE (MIGNOTTE 76), LAFOND (LAFOND 80).

2) Domaine du calcul intégral.

Le calcul intégral a beaucoup attiré les spécialistes de calcul algébrique qui essaient par ce domaine difficile de montrer la validité de leur approche. Nous devons citer à cet endroit le travail important fait par J.H. DAVENPORT (DAVENPORT 79b-79c). L'essentiel de ce travail est rassemblé dans un excellent livre (DAVENPORT 79a) qu'il est difficile de résumer en peu de place mais auquel doit être renvoyé le lecteur intéressé par ce domaine. Les mathématiciens se sont de leur côté attachés à définir des classes de fonctions pour lesquelles un algorithme permet le calcul effectif d'une intégrale. Les travaux de RISCH (RISCH 69-70) ont permis d'obtenir un algorithme qui fonctionne pour une classe assez large de fonctions. NORMAN (NORMAN 77) a réussi à implanter une variante de cet algorithme dans le système REDUCE. Son programme est diffusé avec les versions récentes du système.

Cet algorithme assez complexe est caractéristique de la démarche algorithmique pour résoudre un problème.Il fournit un programme capable certes de produire le résultat, si la fonction donnée est acceptable.Mais ce programme est à utiliser comme une "boîte noire" ; il ne fournit aucun résultat intermédiaire, aucune justification ou explication sur ce qui est fait.

3) CONCLUSION

La démarche de l'algorithmique a permis d'obtenir des résultats très satisfaisants dans des domaines spécifiques.
Il faut observer que cette approche n'est acceptable que dans les domaines où les objets sont bien structurés (polynômes, fractions rationnelles). Les programmes produits n'ont aucun moyen de raisonner sur les calculs qu'ils font. Le pilotage est complètement aveugle. Des idées intéressantes comme celle d'utiliser l'arithmétique modulaire permettent de limiter certains problèmes d'explosion combinatoire (taille des coëfficients entiers ou rationnels par exemple). L'usage de tels algorithmes nécéssite de la part de l'utilisateur une grande expertise. Le système ne dispose d'aucun moyen pour guider les choix : par exemple, les ressources nécessaires au calcul d'une intégrale peuvent être variables suivant la forme que l'on donne à la fonction à intégrer. Il faut pratiquement savoir comment travaille l'algorithme ou avoir une grande habitude du système pour donner la forme la plus propice conduisant à un calcul efficace. Les systèmes de calcul algébrique s'avèrent pour ces diverses raisons, difficiles à exploiter par des personnes non spécialistes. On peut voir là une raison de leur sous-utilisation.

II - 2 - 2.- PROGRAMMATION HEURISTIQUE EN CALCUL ALGEBRIQUE.

Les chercheurs en Intelligence Artificielle ont été très tôt attirés par le calcul algébrique et ce domaine a été l'objet de travaux importants qui ont permis de prendre conscience de problèmes essentiels : simplification, normalisation (MOSES 71a), problèmes d'explosion combinatoire des expressions manipulés et des espaces de recherche (danger lié aux opérations commutatives et/ou associatives), équivalence à 0 d'une expression symbolique...).

La plupart de ces problèmes ne sont d'ailleurs pas encore résolus de façon totalement satisfaisante.

Presque tous les travaux qui ont marqué cette approche sont restés isolés et MACSYMA est le seul système important à avoir intégré des programmes relevant de la programmation heuristique.

1) Domaine de la manipulation de polynômes.

CLAYBROOK (CLAYBROOK 76) a réalisé un programme FORTRAN qui obtient des résultats très satisfaisants pour la factorisation de polynômes multivariés. Il utilise un jeu d'heuristiques qui lui permettent de prévoir la forme des facteurs (variables devant y figurer, analyse des degrés...) et isole de proche en proche les facteurs. Le programme raisonne comme on le ferait "à la main". La lecture de l'article de CLAYBROOK peut être recommandée a qui doit factoriser de tels polynômes à la main. Les heuristiques nécéssaires sont clairement exprimées et pratiquement directement utilisables. Ces heuristiques sont programmées.

2) Domaine du calcul intégral.

SLAGLE (SLAGLE 63) et MOSES (MOSES 67, MOSES 71) ont été à la base de travaux essentiels dans l'approche du calcul algébrique avec la programmation d'heuristiques. Les programmes SAINT (SLAGLE 63) et SIN (MOSES 67-71) qui a fourni l'essentiel du module d'intégration de MACSYMA ont été des pionniers dans le domaine et ont eu des résultats satisfaisants. Il serait intéressant de disposer des moyens qui permettraient de comparer les perfomances de SIN (sous MACSYMA) et de l'algorithme de RISCH (implanté par NORMAN en REDUCE) sur un même ensemble d'intégrales. Ce type de travail n'a pas été fait à notre connaissance.

3) Domaine de la résolution d'équations trigonométriques.

M. GRANDBASTIEN (GRANBASTIEN 74) a réalisé un programme qui permet de résoudre des équations trigonométriques. Un algorithme est possible pour résoudre nombre de ces équations : On pose $t = tgx/2$ et l'on se ramène en utilisant les formules exprimant sin x, cos x, tg x en fonction de $tgx/2$ à des équations de $P(t)=0$ (où $P(t)$ est un polynôme en t). La factorisation du polynôme (algorithme de BERLEKAMP par exemple) permet de résoudre le problème. La difficulté vient du fait que dans beaucoup de cas, le degré du polynôme $P(t)$ devient élevé. M. GRANDBASTIEN

utilise une approche heuristique pour détecter des caractéristiques particulières de l'équation à résoudre et utilise au mieux les formules trigonométriques (passant aux arcs doubles pour abaisser les degrés par exemple). Cela lui permet de ramener l'équation à résoudre à des formes connues pour lesquelles la solution est immédiate (produit de termes simples, etc).

4) Résolution d'équations et d'inéquations algébriques.

Un certain nombre de travaux ont été réalisés sur les équations/inéquations. BLEDSOE dès 1972 avait fait un travail important dans ces domaines.

Concernant les inéquations, BLEDSOE (BLEDSOE 79) a été amené à implémenter un programme capable de conduire des raisonnements sur des chaines d'inégalités et de conduire des calculs algébriques dans ces domaines. Son programme a permis d'établir des preuves non évidentes (La somme de deux fonctions continues est continue). Le principe de résolution est à la base du mécanisme d'inférence mais les idées fondamentales sont très proches de celles que nous développons en préconisant l'alliance de mécanismes de preuves et de mécanismes de calculs.

Concernant les équations, le travail de BUNDY (BUNDY 79b-81a) à EDIMBOURG nous paraît particulièrement significatif et proche des démarches que nous avons utilisées lors de la conception de CAMELIA. BUNDY partant d'une analyse du comportement du mathématicien identifie un certain nombre de stratégies utilisées pour résoudre des équations f(x) = 0

a) Isolement : si l'équation ne comporte qu'une occurence de l'indéterminée x, on l'isole en utilisant les fonctions inverses des fonctions utilisées pour définir l'expression f(x).

b) Collection : l'objectif est de réduire le nombre d'occurences de l'indéterminée (si possible se ramener à une occurence pour utiliser la stratégie de l'isolement). Pour cela, BUNDY applique des règles dites "COLLECTANTES" qui comportent moins d'occurences d'une variable dans le membre droit que dans le membre gauche.
par exemple : 2 sin(x) * cos(x) ==> sin 2*x

x * y + x * z ==> x * (y + z) sont collectantes en x

c) Attraction : l'objectif est de réduire la distance entre les occurences de l'indéterminée de façon à favoriser leur COLLECTION. Pour cela, BUNDY applique des règles "ATTRACTIVES" qui permettent de rapprocher des variables.
Ainsi : expt(e,f(x)) *expt(e,g(x)) ==>expt(e,f(x)+g(x))

f(x) * (g(x)*h) ==> (f(x)*g(x)) * h

A. f(x) + A. g(x) ==> A. (f(x) + g(x))

sont "attractives" en x.

Ce travail nous paraît intéressant car il finalise le rôle des

règles utilisées lors d'un travail. La même règle peut être utilisée à des moments différents pour des finalités différentes mais on sait pourquoi on l'utilise.
Ainsi, la distributivité peut être utilisée à un moment parce que l'on veut collecter des termes et que l'on sait que cette règle peut être collectante. A un autre moment, on l'utilisera pour "rapprocher" des termes parce qu'on peut l'interpréter comme attractive....
Le savoir faire d'"expert en résolution d'équations" décrit par BUNDY pourrait être rédigé dans le langage que nous avons défini pour CAMELIA (cf chapitre IV). Il y a là un travail qu'il serait intéressant de faire avec notre système.

5) Détermination de limites de fonctions numériques.-

Ce sujet a été l'objet de diverses réalisations par des auteurs qui ont travaillé indépendamment. Citons P.S. WANG(WANG 71), LAURENT J.P.(LAURENT 72),HARRINGTON (HARRINTON 79).

Le problème essentiel consiste à lever les indéterminations, lorsque utilisant les divers théorèmes sur les limites, l'on aboutit à une forme indéterminée.

Le programme DALI de LAURENT (LAURENT 72) utilise délibérément une démarche heuristique pour choisir les transformations de la forme de la fonction ; ces transformations susceptibles de lever l'indétermination sont choisies en fonction de la nature de l'indétermination détectée. Le module TRANSS chargé d'effectuer les transformations renvoie alors l'expression modifiée de la fonction à un module (POLLIM) algorithmique qui vérifie si l'indétermination est levée ; POLLIM calcule effectivement la limite si c'est possible ou renvoie l'expression courante à TRANSS en vue d'une nouvelle transformation formelle. Un mécanisme de contrôle d'application des règles de transformation permet d'interdire certaines séquences de transformations. Cela permet, par exemple, d'éviter des problèmes de bouclage par réalisation de la transformation inverse de celle faite lors d'un appel précédent de TRANSS.

Les résultats obtenus sont intéressants par la variété des indéterminations qui ont pu être levées.

Remarque : Dans la plupart des travaux précédents, les heuristiques utilisées sont intégrées aux programmes et parfois difficile à comprendre clairement. Les programmes ainsi obtenus sont complètement spécialisés et il est difficile de leur adjoindre de nouvelles heuristiques. Des programmes, généralement complexes atteignent des perfomances plafond, difficiles à faire progresser. Des programmes écrits par des auteurs différents, ne présentent aucune homogénéité dans l'écriture et ne peuvent pratiquement pas être intégrés dans un système cohérent. On peut considérer que tous ces travaux ont permis de montrer la faisabilité d'un certain nombre d'idées et méthodes de la programmation heuristique dans ce domaine.

6) Evaluation de sommations en analyse combinatoire.-
Programme SEME DE M. BARON (BARON 81, BARON 82)

L'évolution des idées en Intelligence Artificielle

s'est faite dans le sens d'une séparation de plus en plus nette entre les heuristiques utilisées et les programmes qui les manipulent. Toute l'évolution de l'usage des règles de réécriture/-règles de production est significative de ce sujet. Les systèmes experts actuels séparent complètement les connaissances spécifiques d'un domaine des procédures qui les manipulent.

Le programme SEME de M. BARON est caractéristique de cette évolution. Conçu comme un système expert, SEME comporte une base de connaissances dans laquelle est donné le savoir faire utile pour évaluer des sommes finies dans lesquelles interviennent des fonctions combinatoires (façon de manipuler les indices, les bornes des sommations, façon d'éclater des termes pour se ramener à des sommations dont le résultat est connu...)

Ce savoir-faire est exprimé sous forme de règles (plans) définissant des objectifs à réaliser à partir de situations types observées dans l'expression à calculer. La réalisation des objectifs peut faire appel à des tâches dites "heuristiques" qui sont des tâches pour lesquelles est reconnu un droit à l'échec. Les tâches actions permettent le déclenchement contrôlé, adapté à la situation présente d'algorithmes appropriés. Ce savoir-faire est manipulé par un programme général (moteur d'inférence) indépendant du domaine.

On peut considérer que ce programme est une première réalisation intégrant des idées fondamentales en direction d'un système mathématicien :

. La séparation de la base de connaissances de l'interprêteur permettra une extension à d'autres domaines mathématiques : il suffira d'écrire les chapitres de connaissances nécessaires.
Le système verra sa compétence augmenter tout en restant homogène dans sa structure -(SEME a été également testé sur des problèmes de dérivation (BARON 82 b).

. Le langage d'expression des connaissances permet une expression modulaire des connaissances : items indépendants pouvant être donnés "en vrac" sans ordre préétabli.

. Le langage permet le déclenchement contrôlé d'algorithmes

SEME reste cependant limité par divers aspects :

. le langage d'expression des connaissances n'est sans doute pas assez structuré.

. le mécanisme d'inférence est figé et l'on voit mal comment étendre les possibilités de démonstration automatique.

. le moteur ne gère pas de méta-règles ce qui représente une gène fondamentale pour gérer des bases de connaissances importantes.

7) Conclusion.-

On constate que les travaux ayant utilisé une approche heuristique en calcul algébrique ont été nombreux et variés. Les résultats sont généralement bons parce que chaque programme a travaillé dans un domaine bien précis. Les méthodes utilisées, les organisations de programme retenues ne sont pas souvent portables et ces logiciels s'avèrent rarement exploitables, et à fortiori extensibles par toute personne autre que l'auteur.

La difficulté essentielle reste dans la possibilité de détecter rapidement les éléments caractéristiques, les pattern importants des expressions de façon à pouvoir prendre de façon satisfaisante les décisions significatives. En particulier, lorsqu'un algorithme est possible, il ne faut pas prendre le risque de passer plus de temps à utiliser des critères pour reconnaître des situations particulières qu'à exécuter l'algorithme.

II- 3 - LES VOIES DE L'ALGORITHMIQUE/LES VOIES DE LA PROGRAMMATION HEURISTIQUE EN DEMONSTRATION DE THEOREMES.

La démonstration automatique de théorèmes a également été un terrain où se sont développés des travaux relevant de la démarche algorithmique et de la programmation heuristique.

II - 3 - 1 - Approche algorithmique des preuves automatiques.

L'essentiel du travail dans cette voie a été fait autour de la logique dès les années 1960-1965. Les travaux de HERBRAND (HERBRAND 30) ont été à la base des idées de départ.- Mais PRAWITZ (PRAWITZ 60) puis surtout ROBINSON (ROBINSON 65) ont joué un rôle déterminant en introduisant le principe de Résolution.

1) LA RESOLUTION permet effectivement de disposer d'un algorithme de preuve. Elle suppose que le problème soit posé en terme de clauses (formes normales conjonctives débarrassées des quantificateurs). Les travaux de DAVIS-PUTNAM (DAVIS 60) et l'usage des fonctions de SKOLEM permettent de présenter en termes de clauses tout problème spécifié en termes de logique des prédicats du 1er ordre.

Le principe de Résolution fournit un mécanisme absolu pour résoudre des problèmes posés dans le langage des prédicats du 1er ordre. Il permet des preuves par réfutation et est complet au sens logique du terme. (Rappelons qu'un système de règles d'inférence est complet si toute formule bien formée qui se déduit logiquement d'un ensemble donné des formules est aussi un théorème dérivable de cet ensemble de formules). Le principe est simple : Etant donné un problème Hyp -> Conc, il s'agit de montrer que l'ensemble des clauses fabriquées à partir de l'hypothèse Hyp et du contraire de la conclusion neg (Conc) est contradictoire. En terme de Résolution cela s'exprime par la génération de la clause vide (NIL) c'est-à-dire que l'on peut produire à un stade donné une clause C et la clause contraire non C. On dispose pour cela d'un mécanisme permettant

d'engendrer une clause R (appelé résolvante) à partir de deux clauses mères C1 et C2.

exemples : . si C1 s'écrit P V Q1
et C2 s'écrit ¬ P V Q2 alors R s'écrit Q1 V Q2
. si C1 s'écrit P
et C2 s'écrit ¬ P alors R s'écrit NIL:clause vide

L'algorithme d'unification (ROBINSON 65, PITRAT 66), permet l'identification des termes contraires P et - P de C1 et C2.

Un seul problème de principe se pose :

Si l'énoncé n'est pas un théorème, on peut engendrer des clauses à l'infini sans jamais engendrer la clause vide : le processus ne s'arrête pas.

De plus, pratiquement le problème de l'explosion combinatoire de l'ensemble des clauses rend la méthode inapplicable dans la plupart des cas. Le problème du choix des clauses mères C1 et C2 pour fabriquer les résolvantes se pose et il n'est pas possible de le régler par des voies purement algorithmiques. C'est à ce niveau que les gains que l'on pourrait espérer par un bon guidage heuristique manquent.

Ce problème du choix des clauses mères a été l'objet de nombreux travaux dont l'objectif est de définir des stratégies permettant de réduire les espaces de recherches. Parmi les stratégies développées, citons :

. la stratégie de l'ensemble support (SET OF SUPPORT pour les anglo-saxons) : au moins une des clauses mères de chaque résolvante est choisie parmi les clauses descendantes de la clause fabriquée par la négation de la conclusion.

. la stratégie de "préférence des clauses unitaires" (UNIT-PREFERENCE STRATEGY) : dans ce cas, pour clause mère d'une résolvante on essaie de choisir une clause unitaire (avec un seul littéral).

L'intérêt est de diminuer la longueur des clauses ce qui représente un bon espoir de s'approcher de la clause vide).
En effet, la résolvante de $\begin{cases} C1 = P \\ C2 = \neg P \vee Q \end{cases}$ est Q

. la stratégie " forme d'entrée en ligne " (pour LINEAR-INPUT form Strategy) pour laquelle systématiquement une clause mère est choisie dans l'ensemble initial des clauses fournies en entrée. Cette stratégie n'est pas complète : on peut fabriquer des ensembles de clauses réfutables et dont la réfutation ne peut être faite avec cette seule stratégie. Cette stratégie est cependant souvent utilisée car simple et efficace.

. La stratégie "filtrage par les ancêtres" pour laquelle on ne fabrique la résolvante R de C1 et C2 que si C1 ou C2 est dans

l'ensemble de départ ou si C1 et C2 sont l'une ancêtre de l'autre.

Ces travaux théoriques pour la plupart se sont intéressés à des aspects fondamentaux comme la conservation de la complétude ... En fait, ils buttent parce qu'ils ne s'intéressent qu'aux aspects syntaxiques des problèmes afin de rester généraux.

2) LA RESOLUTION DE PROBLEMES ET LA LOGIQUE MATHEMATIQUE

La logique des propositions et la logique des prédicats fournissent un moyen d'expression des connaissances intéressant et constituent des outils de représentation riches en matière de résolution de problème. L'intérêt est évidemment de disposer de résultats théoriques importants pour manipuler les formules que l'on est amené à écrire. En particulier, le principe de Résolution fournit un moyen d'inférence clair dont le seul inconvénient est de mal résister aux phénomènes d'explosion combinatoire.

Des travaux importants ont été entrepris depuis plus de dix ans dans cette voie surtout marquée par le développement de PROLOG parti du travail de A. COLMERAUER (1971) au groupe d'intelligence artificielle (GIA) de Marseille. PROLOG a maintenant une envergure internationale incontestée et est un élément important du dispositif japonais dit de "5è génération" d'ordinateurs. PROLOG est un interpreteur de formules de la logique du premier ordre. Pour des raisons d'efficacité, PROLOG ne prend en compte que les clauses de HORN (clauses qui n'ont qu'un littéral positif : forme générale $\neg h1, \neg h2, \neg h3, C$).

A ce stade, la logique des prédicats devient un langage de programmation et un programme PROLOG est constitué d'une suite de clauses ordonnées. L'intérêt de programmer en PROLOG est triple :

a) les clauses traduisent directement les assertions, les théorèmes sur l'univers du travail.

b) l'interpreteur fournit un algorithme d'unification tout programmé.

c) l'espace de recherche (arborescence, retour-arrière,...) est géré par le système.

Plusieurs stratégies peuvent être prise en compte pour la gestion des clauses appariées : les unes complètes sont généralement peu efficaces, d'autres plus efficaces ont l'inconvénient de ne pas être complètes. Le problème essentiel n'est pas là cependant. Si PROLOG permet dans un formalisme simple à la fois de décrire et de résoudre des problèmes, il reste à l'utilisateur l'essentiel : à savoir, trouver les prédicats convenables pour exprimer son problème (cf LAURIERE 82).

3) CONCLUSION :

Cette approche de la démonstration automatique de théorèmes par des voies algorithmiques, en particulier avec la résolution, a été très prolifique quant au nombre d'articles publiés dans les revues spécialisées. Elle a permis de nombreux travaux théoriques de qualité. On peut cependant regretter que du point de vue pratique, elle reste peu efficace et peu utilisée.

Les problèmes de choix, d'adaptation aux données réelles du problème courant ne peuvent se résoudre par la définition de stratégies basées sur des aspects purement syntaxiques - si judicieuses soient elles - Les formalismes introduits sont sans doute bons. Les systèmes de preuve par réfutation ont eux aussi besoin de connaissances caractéristiques du domaine du problème pour faciliter les choix. L'espoir peut alors se situer dans une intégration de ces idées et de celle des systèmes experts. A notre connaissance, il n'y a pas de travaux engagés dans cette voie.

II- 3 - 2 - PROGRAMMATION HEURISTIQUE ET DEMONSTRATION AUTOMATIQUE DE THEOREMES.-

La démonstration automatique de théorèmes a été dès le début une préoccupation importante des chercheurs en Intelligence Artificielle. Beaucoup d'espoir se situait au niveau des applications en robotique, en interrogation de bases de données, en génération ou en vérification de programmes.

Les mathématiques ont été un terrain favorable à la vérification d'un certain nombre d'idées et les travaux dans ce domaine ont été nombreux et variés. Les chercheurs se sont surtout intéressés aux domaines où le "raisonnement" est important et relativement pur : la logique, la géométrie, l'arithmétique, l'algèbre dans des structures simples, l'analyse. Plus récemment la théorie des ensembles et la topologie ont été plus particulièrement étudiés. Le travail original de D. B. LENAT avec le programme AM qui découvre des concepts en mathématiques sera également décrit dans cette partie.

1) Les travaux en logique. Les travaux essentiels dans ce domaine datent de l'époque où l'on attendait beaucoup de la logique formelle. Les travaux de PITRAT (PITRAT 66, PITRAT 70) ont été faits en même temps que ceux de ROBINSON sur la Résolution et représentent un travail de pionnier important et toujours d'actualité par de nombreux aspects. Le programme de PITRAT accepte en entrée le jeu d'axiomes d'une théorie logique et établit en les démontrant les théorèmes intéressants de la théorie. L'heuristique fondamentale utilisée peut s'exprimer en disant qu'un théorème est intéressant s'il permet de produire des résultats intéressants. Le programme génère des théorèmes en éliminant ceux qui ne sont pas intéressants (ceux qui ne produisent rien ou n'ont pas un intérêt suffisant au sens de la fonction d'évaluation de l'intérêt) et en conservant ceux qui présentent le plus d'intérêt .

Ce programme crée donc des théorèmes, développe des théories. Il utilise pour cela des outils puissants permettant le contrôle de l'explosion combinatoire du nombre de théorèmes engendrés. En particulier, il génère des Méta-théorèmes qui sont des règles qui manipulent les théorèmes. Les méta-théorèmes permettent d'établir directement un théorème à partir d'un théorème déjà établi sans nécessiter de preuve directe à partir des axiomes. Ces méta-théorèmes sont eux aussi utilisés en fonction de leur intérêt et un niveau de méta-méta-théorèmes facilite leur manipulation. Ce programme a eu des résultats spectaculaires avec plusieurs axiomatiques. Il a redécouvert effectivement les théorèmes intéressants de chacune. Il a aussi établi que certaines preuves données dans les manuels de logique sont maladroites. Ce travail a été l'occasion de la mise au point de l'algorithme d'unification si fondamental en intelligence artificielle.

2) La découverte en mathématique (Travail de D. B. LENAT)

Le programme AM de D.B. LENAT (LENAT 76, LENAT 77) a été écrit pour faire des découvertes en mathématique. Il connait a priori un certain nombre de concepts qui sont décrits dans des structures de données par des "facettes", attributs caractéris-

tiques qui sont par exemple : le nom du concept, des définitions (originale, exprimée en langage du calcul des prédicats, itérative..), des exemples et contre-exemples illustrant le concept, des généralisations ou particularisations du concept, des conjectures, des analogies, des centres d'intérêt, une estimation de la validité. L'objectif est d'enrichir des concepts existants (en complétant des facettes inconnues) ou de découvrir de nouveaux concepts. Les éléments actifs pour favoriser l'enrichissement de cet espace sont des règles qui contiennent dans la partie condition des tests sur des concepts existants et dans la partie action des directives sur le type de recherche qu'il peut être intéressant de faire ou d'action à lancer lorsqu'on est dans une situation donnée.

Au niveau supérieur le contrôle est basé sur la gestion d'un agenda des tâches qu'il peut être intéressant d'envisager (ces tâches sont ordonnées en fonction d'une valeur d'intérêt recalculée après chaque étape, la tâche la plus "prometteuse" étant envisagée en premier). Un exemple de tâche peut être "remplir la facette "EXEMPLES" de "NOMBRE PREMIERS". Une fois que la tâche envisagée est choisie dans l'agenda, AM cherche les règles heuristiques qui peuvent être activées pour accomplir la tâche. Ces règles sont alors toutes appliquées et AM cherche une nouvelle tâche. Chaque exécution d'une règle peut avoir trois types d'effet :

a) les facettes d'un concept se remplissent
C'est le cas avec une règle comme : "si on désire des exemples de X et que X est une sorte de Y (Y étant un concept quelconque plus général que X) alors vérifier avec les définitions de X les exemples de Y (certains d'entre eux peuvent être des exemples de X)".

b) un nouveau concept est créé
Une règle typique de ce cas est la suivante :"si quelques uns (mais pas la plupart) des exemples de X sont aussi exemples de Y (pour tout concept Y) alors créer un nouveau concept défini comme l'intersection de ces 2 concepts (X et Y)".
Elle permet à partir du concept Y de "nombres représentables de façon unique comme la somme de 2 nombres premiers" de créer le concept de "nombres premiers représentables de façon unique comme la somme de 2 nombres premiers "lorsque AM trouve que quelques exemples de X (nombres premiers) sont dans Y.

c) une nouvelle tâche est ajoutée à l'agenda. La règle suivante est typique de ce type de possibilité :

"Si très peu d'exemples de X sont trouvés alors ajouter la tâche suivante à l'agenda "généraliser le concept X".

Une caractéristique importante de l'agenda est d'associer à chaque tâche une explication, une raison pour laquelle il est important de prendre en compte cette tâche.
Ce sont les règles qui véhiculent ces raisons. Au regard de la gestion de l'agenda, il faut noter que si une tâche déjà dans l'agenda est reproposée pour une nouvelle raison alors sa priorité, son intérêt augmente alors que si la tâche est reproposée pour la même raison, l'intérêt n'augmente pas. La qualité des raisons annoncées pour une tâche est utilisée pour évaluer les

ressources (espace-temps) que AM alloue à la tâche et qu'il autorisera de dépenser avant de changer de tâche. Ces raisons sont utilisées pour expliquer et justifier à l'utilisateur pourquoi AM estime important de se concentrer sur cette tâche.

Les centaines de concepts que possède AM sont organisés de différentes façons. Une hiérarchie importante est liée aux facettes "généralisations", "spécialisations". Les concepts peuvent apparaître comme les noeuds d'un graphe orienté dont les arêtes sont étiquetées "généralisation" ou "spécialisation". Ce graphe permet de représenter facilement nombre de propriétés héréditaires (la relation "spécialisation" est transitive). Cela permet de réduire beaucoup le nombre de règles : chaque règle est attachée au concept le plus général pour lequel elle est appliquable et est considérée comme automatiquement applicable pour toutes les spécialisations de ce concept. Ainsi une méthode générale pour inverser une fonction est certainement capable de rendre service pour inverser une permutation. Si pour accomplir une tâche "Inverser la permutation ..." on ne dispose pas de méthode spécifique, alors remonter les flèches "généralisation" à partir du concept "permutation" permettra de rencontrer le concept de "bijection" puis de "fonction". On pourra alors tenter une méthode pour établir l'inverse d'une bijection. Si une telle méthode n'est pas disponible, on tentera la méthode générale pour inverser une fonction etc... Le graphe sur les concepts induit donc un graphe sur les règles. En fait, plus le concept est général plus ses heuristiques sont faibles (plus exigeant en temps de calcul, moins de chance de succès) aussi, AM examine les règles dans un ordre croissant de généralisation.

Résultats : AM a travaillé en connaissant au départ, une centaine de concepts élémentaires sur la théorie des ensembles finis. La plupart des concepts évidents de la théorie des ensembles et relations a été rapidement trouvée (lois de MORGAN, singletons) mais aucune théorie sophistiquée des ensembles n'a été élaborée. Par contre, AM a découvert les nombres naturels et a exploré une théorie élémentaire des nombres. Les opérations arithmétiques ont été rapidement élaborées et AM a eu des résultats surprenants en théorie de la divisibilité (les paires de nombres premiers, les équations diophantiennes, unicité de la factorisation des entiers en produit de nombres premiers, conjecture de Goldbach, sont quelques unes des bonnes découvertes de AM). Par contre, AM n'a rien découvert sur des domaines importants comme la théorie des restes, le PGCD, les relations d'ordre, l'infini... Toutes ces découvertes ont été faites en 1 heure de CPU (PDP 10 KI-Interlisp -100 K).

Conclusion.

Ce travail apparait comme très important et il est difficile d'en donner toute l'essence dans un résumé aussi bref. Il montre bien les performances que l'on peut atteindre avec un système de règles de réécritures pour coder des connaissances. D. B. LENAT a tiré des conclusions méthodologiques quant aux contraintes à satisfaire pour écrire des systèmes d'inférences basés sur des règles de production et destinés à "faire des découvertes", inventer, .. Ces analyses (LENAT 77 D) constituent une référence essentielle dans ce domaine.

3) <u>Les travaux en arithmétique</u>

Un certain nombre de travaux ont été faits dans le domaine de l'arithmétique. Citons (BUNDY 73), (BOURGOIN 78-79), (DALLARD 74) et (LENAT 76-77) à propos de la découverte.

Le programme ARITH de R.DALLARD a obtenu de bons résultats sur des théorèmes de niveau terminale C. Le programme dispose d'un certain nombre de moyens fournis en entrée :

a) des règles de dérivation et le mécanisme :

$$\left\{\begin{array}{l}(P,Q)\\ T\end{array}\right. \quad \texttt{<>---->} \quad T_P \implies T_Q$$

qui exprime que si T est un théorème et (P,Q) une règle, on peut substituer Q à P dansT.

R.DALLARD distingue entre deux types de règles :

- les règles simplifiantes comme :
  (FAUX ET x , FAUX)
  (X = Y et $X \neq Y$, FAUX)
  (X et YZ non premiers entre eux,
  (X et Y non premiers entre eux) ou
  (X et Z non premiers entre eux))

- les règles productrices qui expriment ce que l'on suppose connu du système par exemple (théorème de Gauss) :
  (Y divise ZX, Y n'est pas premier avec X ou Y divise Z)

b) deux théorèmes supposés connus-"axiomes" opératoires :

$$\forall x\ y\ z \qquad x^y z^y = (xz)^y$$

$$\forall x\ y\ z \qquad (x^y)^z = x^{yz}$$

c) la conjecture à démontrer.

<u>Fonctionnement général d'ARITH</u>

- Comme dans la Résolution, ARITH prend la négation de la conjecture à démontrer et cherche à démontrer qu'il y a une contradiction (les règles permettent de réécrire FAUX).

- Le programme fonctionne en 3 phases :

<u>Phase 1</u> : A tout théorème (ou conjecture) on applique les règles simplifiantes. C'est là que l'on espère voir FAUX arriver pour terminer la preuve.

<u>Phase 2</u> : Génération de règles de dérivations (simplifiantes ou productrices) à partir des théorèmes obtenus par la phase 1.

Exemple :

T amène les règles(T,VRAI), (NON T, FAUX)
X = Y amène les règles (X,Y), (Y, X)

<u>Phase 3</u> : Par association des axiomes ou théorèmes avec les règles productrices, on fabrique de nouveaux théorèmes et l'on reprend la phase 1. Il y a aveu d'échec si dans cette phase on ne parvient pas à générer de nouveaux théorèmes.

<u>Résultats obtenus</u> :

A - Résolution de 13 équations diophantiennes par

exemple :

. l'équation $x^2 + y^2 = z^2$ a des solutions

. si y est un impair donné $x^2 + y^2 = z^2$ a des solutions

B - Résolution de 3 exercices sur les progressions arithmétiques :
par exemple : on ne peut trouver 3 nombres x, y, z en progression arithmétique de raison non nulle tels que

$$x^3 + z^3 = 2y^3$$

C - Raisonnement par récurrence : la conjecture est posée sous la forme

$$P(o) \text{ ET } \forall x \, (P(x) \Longrightarrow P(x + 1))$$

par exemple :

z^n divize $2^{(z^n)} + 1$

D - Résolution de 3 exercices de divisibilité :
par exemple : si dans une division le quotient n'est pas nul, le dividende est supérieur au double du reste.

E - Résolution de 11 exercices sur les nombres premiers entre eux :
par exemple : si a et b sont premiers entre eux

$$a^2 \neq 2b^2$$

F - Résolution de 4 exercices sur les nombres premiers par exemple : il existe des nombres premiers qui sont somme et différence de nombres premiers.

G - Divisibilité par 3, 5 ou 7 : 6 exercices.

Principales caractéristiques de ARITH

. L'ensemble des règles de dérivation n'est pas figé. Le programme peut en créer à partir des résultats intermédiaires trouvés.

. Le traitement des quantificateurs : DALLARD rejetant la Skolemisation conserve les quantificateurs le plus près possible de l'expression quantifiée.

. L'algorithme d'unification fait intervenir les règles d'associativité, de commutativité, de distributivité.

. ARITH n'a aucune possibilité numérique : (11 est connu comme 1 + 1.... écrit 11 fois !). C'est une faiblesse im-

portante même pour un programme conçu pour résoudre des problèmes de preuves.

4)Les travaux en géométrie relèvent exclusivement de la géométrie plane traditionnelle. En particulier, il n'y a pas eu de travaux en géométrie analytique qui relèveraient beaucoup du calcul algébrique et seraient un bon terrain pour intégrer des mécanismes d'inférence et des mécanismes de calcul algébrique "conduit intelligemment". Le rôle de la figure en géométrie traditionnelle est important, il a été l'objet de beaucoup d'attention.

i) Les travaux de Gelernter (GELERNTER 59) ont utilisé la figure comme un réservoir d'information de type heuristique : la figure est codée dans une représentation interne permettant la vérification rapide d'un certain nombre de propriétés (alignement de points, intersections de droites, mesures de distances).

Lorsque le programme se propose de démontrer une propriété, il commence par vérifier sur la figure qu'elle est plausible avant toute preuve formelle. C'est une façon élégante d'éviter au programme de perdre du temps à essayer d'établir des propriétés manifestement fausses.

ii) Le travail de M. Buthion (BUTHION 75) a été consacré à la construction de figures de géométrie
en utilisant la règle et le compas. Dans ce domaine la figure joue le rôle de réservoir d'information. Les types de raisonnements utilisés concernent la "méthode des lieux" (définition de points comme des intersections de courbes) et l'usage de transformations géométriques (homothétie). M. Buthion a été amené à distinguer entre phase active qui permet une construction effective et phase passive qui utilise des théorèmes permettant à la phase active de conclure.

Les problèmes rencontrés sont liés à l'introduction d'objets "nouveaux" dans la figure et trois types d'introduction ont été isolés :

- introductions "automatiques" d'un objet. Par exemple, dès qu'un cercle apparait, lui attribuer un centre et un rayon

- introductions "naturelles" obtenues par effets conjugués de règles et canonisation. Par exemple, la règle

ISO XYZ -------> PER DRO X MIL YZ DRO YZ

appliquée à : ISO ABC
permet l'introduction du milieu I du côté BC d'un triangle isocèle de sommet A.

- introductions "artificielles". Ces créations font l'objet d'un module spécialisé du programme.

Les heuristiques utilisées sont du type :

- Heuristique SOMME-DIFFERENCE : si un problème fait intervenir des sommes ou des différences connues de longueurs de segments inconnus, il faut matérialiser cette information.(repérer la SOMME-DIFFRENCE, choisir un support, une origine, cela amène généralement la création d'un point).

- Transformation :
  - Définir une transformation géométrique T suggérée par la figure
  - par T, substituer au problème initial (Pb) un problème dual (Pb*) plus facile
  - résoudre Pb*
  - revenir à Pb par la transformation inverse.

Le programme dispose des moyens donnés à un élève "anciennes mathématiques élémentaires" sous forme de constructions de bases (médiatrices d'un segment, cercle de diamètre donné, symétrique d'un point par rapport à un autre point ...) ou sous forme de règles. Par exemple, le théorème "la hauteur et la médiane principales d'un triangle isocèle sont confondues" était fourni par la règle

ISO XYZ ------> PER DRO X MIL YZ DRO YZ.

M. BUTHION a travaillé sur un problème important apparu comme fondamental dans son domaine : il s'agit de la manipulation de différentes représentations pour un même objet. Ainsi, si 4 points A, B, C, D, sont alignés, la droite support peut s'appeler d mais aussi AB, AC, BC, BD, CD, BA ... La désignation même de l'objet pose problème et son identification avec l'algorithme d'unification peut être ambigue. Une modification de l'algorithme d'unification est nécessaire pour que les instanciations précises d'une variable et d'un objet soient retardées dans le temps. On conserve la classe des noms possibles pour désigner l'objet identifié et l'on diffère la décision de désignation précise.

5)Les travaux en théorie des ensembles et topologie

Des travaux essentiels ont été réalisés dans ces domaines difficiles (MERIALDO 79), (PASTRE 76), (PASTRE 78 a), (PASTRE 78 b), (PASTRE 79), (PASTRE 82 a), (PASTRE 82 b).

Ainsi D. PASTRE a réalisé un premier programme DATTE (PASTRE 76) consacré à la démonstration de théorèmes en théorie des ensembles. Il s'agit de "preuve naturelle" et les règles de déduction et représentations sont proches de celles que l'homme utilise. A ce propos, D. PASTRE a été amené à saisir et analyser des protocoles de mathématiciens en train de résoudre des problèmes. Il apparait que, (je cite), "une difficulté profonde réside dans le fait que, si le mathématicien écrit et communique dans un certain langage propre et bien formalisé, il cherche et raisonne d'une tout autre manière et dispose d'une quantité énorme de savoir-faire et de connaissances qui ne figurent pas dans sa tête sous forme d'axiomes bien formés" (PASTRE 82 a - p 243).

Le démonstrateur utilise des méthodes basées sur :

- le découpage d'un théorème en plusieurs théorèmes plus simples dont la conjonction est équivalente au théorème initial.

- le "chainage avant" qui consiste à chercher les propriétés que l'on peut déduire des hypothèses et les ajouter comme nouvelles hypothèses.

- Utilisation des définitions des prédicats (avec ou sans remplacement)

- création d'objets et éventuellement "Dessin"

- traitement des propriétés existentielles.

Concernant ces deux derniers points, D. PASTRE est proche de M. BUTHION (cf 3 . ou (BUTHION 75)) : elle a manipulé des diagrammes servant de réservoir d'information, dans lesquels une relation peut être rapidement vérifiée ; lors de la recherche d'un élément, on travaille d'abord avec les objets déjà construits mais le diagramme peut "suggérer" la création de nouveaux objets,.. Les mêmes problèmes d'unification avec des objets à représentations multiples se retrouvent.

D. PASTRE travaille maintenant à la réalisation d'un programme destiné à démontrer des théorèmes difficiles nécessitant une grand nombre de connaissances. Le domaine support choisi est celui des espaces vectoriels topologiques et les ensembles convexes. Les méthodes sont aussi générales que possibles. Une démarche type systèmes experts a été adoptée et les connaissances sont fournies dans un langage Ad-hoc !

Plusieurs directions de travail sont menées en parallèle : l'amélioration du "moteur", ainsi que celle de la base de connaissances. Ceci nécessite, outre la mise au point d'un bon système de règles mathématiques générales, une réflexion approfondie sur le domaine mathématique particulier considéré, et l'observation de mathématiciens confrontés aux problèmes proposés au système.

- Le moteur est actuellement spécialement conçu pour les mathématiques. La base de faits est constituée par la représentation interne d'un "théorème à démontrer", éventuellement de plusieurs sous-théorèmes. Il s'agit d'une liste d'items sur lesquels les règles s'appliquent (objets, hypothèses, relations, variables à instancier, substitutions possibles, conclusion, règles déjà appliquées avec instanciations, règles actives, agenda d'actions). Actuellement le programme a un certain nombre de connaissances sur les notions mathématiques représentées par ces items. (Par exemple : il sait que plutôt d'ajouter une hypothèse qui est une conjonction, on ajoute chacun des termes de la conjonction. D'autres actions sont beaucoup plus compliquées). Une phase de travail actuellement en cours consiste à décrire aussi ces connaissances sous forme de règles. Une règle mathématique de haut niveau pouvant faire exécuter une action dont le mode d'emploi est donné par des règles de plus bas niveau. Le moteur peut alors être moins dépendant des mathématiques d'une part, et d'autre part, cela permet avec le même moteur, de travailler indifféremment, même au niveau élémentaire, dans diffé-

rents types de domaines mathématiques (par exemple : la théorie axiomatique des ensembles privilégie la notion de relations alors que les mathématiques classiques privilégient les notions d'éléments et de sous-ensembles).

- L'amélioration des règles consiste à affiner les règles exprimant des méthodes pour que celles-ci s'appliquent à bon escient, tout en n'étant pas trop particularisées. L'observation de mathématiciens est une aide précieuse pour ce travail. Les discussions avec eux sur leur propre comportement, souvent inconscient, sont nécessaires.

- Outre la nécessité d'avoir de bonnes règles, il faut pour certaines d'entre elles les appliquer dans un certain ordre (par exemple les plus particulières avant les plus générales). Quelques méta-règles ont été testées (le même moteur les applique). Elles permettent d'"activer" uniquement les règles pertinentes, parmi celles-ci de placer en priorité certains types de regles. Enfin des méta-règles d'initialisation commencent par modifier les règles données en ajoutant des "méta-actions" (par exemple activer une autre règle ou mettre une règle à la fin de la liste des règles actives).

II - 4 - CONCLUSION

L'exposé des travaux que nous venons de relater est loin d'être exhaustif. Nous pensons malgré tout qu'il est suffisant pour faire apparaître la diversité des travaux qui ont été réalisés en mathématiques avec des ordinateurs. Cette diversité peut être appréciée tant en ce qui concerne les domaines mathématiques abordés que les méthodes utilisées. Il est remarquable d'observer que les deux démarches de l'algorithmique et de la programmation heuristique ont été utilisées parallèlement, sans se rencontrer bien que travaillant parfois dans le même domaine. Les cas du calcul intégral et de la factorisation de polynômes sont particulièrement significatifs. La question essentielle qui se pose alors n'est pas de savoir laquelle des deux démarches est la meilleure mais est de savoir s'il est possible de trouver une voie intégrant au mieux les acquis de chacune de ces approches. Nous sommes en effet convaincu que c'est une combinaison de démarche algorithmique et de démarche heuristique qui permettra de dépasser les performances actuelles.

CHAPITRE 3

APPROCHE SYSTEME EXPERT POUR FAIRE DES MATHEMATIQUES

III - APPROCHE SYSTEME EXPERT POUR FAIRE DES MATHEMATIQUES

III- 1 - NATURE DE LA CONNAISSANCE DU MATHEMATICIEN

La compétence du bon mathématicien est basée sur deux types de connaissances qui se conjuguent pour atteindre l'efficacité :les connaissances de contenu et le savoir-faire.

III.1.1. - Les connaissances de contenu :

Elles concernent pour des théories diverses la connaissance des axiomatiques de la théorie, la connaissance de l'énoncé exact des théorèmes importants de la théorie (y compris l'énoncé des conditions d'application). Dans le cadre d'un enseignement de mathématiques, cela correspond à ce qui se transmet en cours.

III.1.2. - Les savoir-faire :

Ils concernent l'habileté de l'individu à combiner des objets et des résultats liés à une théorie pour établir de nouvelles propriétés. Deux aspects apparaissent comme essentiels : savoir calculer, savoir raisonner.

i) Le "savoir calculer" est lié à des connaissances opératoires, des mécanismes qui permettent d'obtenir de façon certaine un résultat. La connaissance d'algorithmes de base est nécessaire et fondamentale dans presque tous les domaines mathématiques. Citons : en numération la pratique de l'addition, soustraction ..., en algèbre linéaire, le calcul d'un déterminant ..., en géométrie, le tracé du centre de gravité ou du cercle circonscrit à un triangle, en analyse la détermination des racines du trinôme ou de la solution générale d'une équation différentielle linéaire du second ordre à coefficients constants, ...

L'acquisition de ces "savoir calculer" ne peut valablement se faire que par une pratique personnelle et fait donc appel à un enseignement par exercices dont le déroulement est bien contrôlé. L'acquisition de certains mécanismes peut se faire très jeune (dès 5/6 ans). Notons qu'il est toujours indispensable d'atteindre une certaine efficacité afin que cet aspect calculatoire ne soit pas un écran, un frein pour le raisonnement.

Les "savoir-calculer" évoluent au cours du temps et la méthode, l'algorithme à mettre en oeuvre à un instant donné peut varier en fonction de deux paramètres :

- l'adaptation aux données qui nécessite un raisonnement élémentaire pour détecter une situation particuliè-

re permettant de mettre en oeuvre un mécanisme "moins coûteux" en ressources (moins de mémoire, moins de temps). Tout l'enseignement des règles du calcul mental qui constituent un affinement remarquable des mécanismes opératoires de base relève de cette idée.

- l'évolution technologique : depuis longtemps on s 'est aperçu que l'on pouvait confier à des machines une partie de ces activités calculatoires afin de soulager en partie l'esprit de cette charge et le laisser plus disponible pour d'autres tâches. Le boulier chinois, la règle à calcul, la table de logarithmes, la machine de PASCAL, les calculateurs ...relèvent tous de cette idée. Observons que pour chaque technologie, s'est développé un "savoir faire" opératoire adapté.

Retenons que quel que soit le domaine, quelle que soit l'époque, la technologie disponible, une partie de l'activité mathématique relève de l'exécution de mécanismes donnant de façon certaine un résultat élaboré à partir de données. Il est toujours nécessaire de savoir, à certaines étapes d'un raisonnement, qu'un tel mécanisme existe. Il est souvent utile de disposer d'un moyen pratique adapté pour fabriquer effectivement le résultat.

Il est remarquable de constater que l'essentiel de l'histoire du calcul scientifique automatique a été marqué par un développement prépondérant des méthodes de calcul numérique au détriment des systèmes de calcul algébrique.

ii) Le "savoir raisonner" : lié directement aux mécanismes d'inférence, induction, déduction qui sont la base du cheminement logique de la pensée. D'une approche plus délicate, la notion de démonstration par exemple n'est accessible qu'à un âge plus avancé (12/14 ans). Là encore, on peut supposer que l'on dispose d'un certain nombre de techniques (raisonnement direct, raisonnement indirect, raisonnement par l'absurde, preuve par récurrence, raisonnement par analogie ...). Ces techniques sont plus difficiles à maîtriser : nombre d'étudiants ayant eu un baccalauréat de mathématique ne maîtrisent pas bien le raisonnement par récurrence par exemple.

III-1.3. L'art de combiner "contenu" et "savoir-faire":

C'est à ce niveau que peut se juger le degré d'expertise d'un mathématicien. Cela nécessite :

i) de savoir l'usage que l'on peut faire d'un théorème : ainsi, il n'est pas suffisant de savoir que la distributivité du produit sur la somme s'énonce par la relation (a+b)*c = a*c+b*c. Il faut être pleinement conscient que cette propriété est utile dans un sens, toutes les fois où l'on a besoin de ramener une somme à un produit (factorisation ...) et dans l'autre sens, toutes les fois où l'on a besoin d'éclater un produit pour faire apparaître une somme.

ii) savoir utiliser les théorèmes. Il faut pour cela être capable de détecter que l'on est dans les condi-

tions d'application d'un théorème ou exprimer comment l'on espère se ramener à ces conditions d'application. Cela suppose toujours une habileté pour détecter des situations particulières, faire des choix sur lesquels on peut revenir, prévoir des étapes que l'on envisage, que l'on affine progressivement ...

Ainsi, dans le domaine du calcul des limites de fonctions numériques, il ne suffit pas de savoir par coeur la formule : $\lim_{x\to a} (f(x) + g(x)) = \lim_{x\to a} f(x) + \lim_{x\to a} g(x)$

Il faut faire de la connaissance de l'énoncé de ce théorème un moyen de raisonner, d'agir sur des situations mathématiques. Le "savoir utiliser" de ce théorème pourrait s'énoncer ainsi :

. SI j'ai à calculer la limite : $\lim_{x\to a} F(x)$

. SI F(x) apparaît comme une somme f(x)+g(x)
ALORS calculer $\lim_{x\to a} f(x)$,
calculer $\lim_{x\to a} g(x)$
puis faire la somme des résultats obtenus.

. SI F(x) n'est pas une somme (et que je ne sais pas quoi faire d'autre) :
- tenter de faire apparaître F(x) comme une somme (cf.ce qui a été dit sur la distributivité ci-dessus) puis
- appliquer la démarche précédente.

Le théorème énoncé valide effectivement la démarche adoptée.

III-1-4. Le problème de la méta-connaissance.

L'expert possède au moment d'aborder un problème des connaissances beaucoup plus larges que celles requises pour traiter le problème. Ces connaissances sont relativement organisées, des liens existent entre différents concepts qui permettent de passer d'un type d'activité à un autre. La prise de conscience de ces liens est une connaissance sur les connaissances. On parlera alors de méta-connaissance.

Donnons un exemple en géométrie : on peut décrire des théorèmes et des savoir faire en géométrie descriptive, en géométrie analytique, en géométrie pure. Un même problème peut être traité dans l'une ou l'autre de ces approches. Le problème du choix de la représentation la mieux adaptée nécessite une connaissance externe , décrivant les avantages ou les inconvénients de chacune d'entre elles, caractérisant les types de problèmes pour lesquels chacune est plus convenable, ...

Le terme de méta-connaissance s'applique pour cette forme de connaissances sur les connaissances. La méta-connaissance a pour but de faciliter des choix, de favoriser des enchaînements, éventuellement d'interdire des approches vouées à l'échec, ou de déclarer que l'on ne sait rien sur le problème.

Derrière le terme de méta-connaissance il y a l'idée qu'il faut "savoir ce que l'on sait" et "savoir ce que l'on ne sait pas". Les heuristiques qui permettent un travail "en finesse" se cachent à ce niveau et l'on peut penser que la qualité d'un spécialiste est directement liée à la qualité de ses méta-connaissances. Apprendre c'est probablement se forger de la méta-connaissance pour manipuler de la connaissance brute. On peut sans doute regretter que l'enseignement se contente généralement de transmettre des connaissances, un peu de savoir faire et laisse de côté (comme pratiquement tous les manuels d'ailleurs) l'aspect méta-connaissance. Le travail de POLYA (POLYA 57,POLYA 58) n'en a que plus de mérite et l'on peut regretter qu'il n'ait pas eu de suite pour des domaines autres que la géométrie.

III- 2 - LES CONTRAINTES POUR FAIRE UN SYSTEME MATHEMATICIEN

III.2.1. Expression des connaissances.

. La première contrainte consiste à être capable de représenter les connaissances issues de théories diverses et de les stocker dans des bases de connaissances (BDC). Il s'agit de représenter tant les connaissances de type contenu et savoir faire que la méta-connaissance. Tout cela doit être fait de façon la plus homogène possible. Ce critère d'homogénéité est primordial dès que l'on souhaite que les connaissances d'un domaine soient utilisables dans un autre domaine.

. D'autre part, il faut pouvoir disposer de mécanismes de calcul efficaces. Cela suppose un langage puissant pour exprimer des algorithmes de calcul symbolique et numérique et des procédés simples pour décider des conditions d'appel de ces algorithmes. Sont à prendre en compte, à ce stade, les éléments permettant de décider de l'algorithme précis qu'il y a lieu de lancer et du meilleur moment pour le lancer.

. Le langage d'expression des connaissances doit rester simple, facile à utiliser pour les experts et être extensible. Un expert doit pouvoir y définir facilement des mots nouveaux, correspondant à de nouveaux concepts et des procédures associées aux aspects opératoires de son domaine.

. Les connaissances doivent pouvoir être données "EN VRAC". Ce type de système pour atteindre une compétence raisonnable nécessite beaucoup de connaissances. La construction, la mise à jour, les ajouts à la base de connaissances doivent être facilités au maximum. Le système doit être insensible à l'ordre dans lequel sont fournies les connaissances. Dans un certain sens le problème est plus lié à des aspects humains qu'a des aspects systèmes : la gestion "du vrac" doit rester de taille humaine afin que l'on puisse savoir facilement si le système dispose de telle ou telle connaissance. On peut toujours fabriquer des outils simples d'interrogation de la base de connaissances. La construction d'outils fins pose des problèmes bien connus en documentation automatique et interrogation de bases documentaires : problème de questions à poser, problème de bruit et de silence documentaire, ...

Pour ces raisons humaines essentiellement, une certaine organisation de la base de connaissances est souhaitable : organisation en chapitres avec commentaires par exemple.

III.2.2. Expression du mode d'inférence : le moteur d'inférence.

Il est indispensable de disposer d'un mécanisme de raisonnement capable, au vu d'un problème à résoudre, d'extraire de la base de connaissances les éléments utiles et d'activer ceux qui sont les plus adéquats : cela suppose d'être apte à analyser l'objectif d'un problème, de choisir les connaissances permettant d'approcher sinon d'atteindre cet objectif. Le moteur d'inférence doit manipuler la méta-connaissance afin de pouvoir adopter des stratégies adaptées au problème en cours. Ces stratégies doivent pouvoir être décrites de façon externe afin de

rendre le moteur indépendant du domaine dans lequel il travaille.

Le mode de raisonnement du système doit être simple de façon à pouvoir être suivi facilement par l'utilisateur. Il serait intéressant de pouvoir disposer d'un moteur offrant plusieurs modes d'inférence (raisonnement par l'absurde, preuve descendante des hypothèses vers la conclusion ou ascendante de la conclusion vers les hypothèses, ...). Le choix du mode d'inférence se faisant lui-même suivant des règles données au système.

III-2.3. Interface Homme-Machine

III-2.3.1. - Dialogue avec l'utilisateur final.

Pour atteindre un réel système de "Mathématiques Assistées par Ordinateurs" (doit-on parler de MAO ?) il est indispensable d'atteindre un dialogue de qualité et l'interface homme-machine doit satisfaire des contraintes d'ergonomie logicielle sévères.

Ces contraintes peuvent se situer sur plusieurs plans :

i) Plan du contrôle du système par l'utilisateur. L'utilisateur doit disposer de moyens pour contrôler l'autonomie qu'il laisse au système. Trois degrés d'autonomie apparaîssent comme indispensables.

- Autonomie globale : le système est entièrement libre et conduit seul la résolution du problème posé.
- Autonomie locale : l'utilisateur conduit les grandes étapes de la résolution mais laisse le système autonome pour résoudre des sous-problèmes bien définis. Ainsi l'utilisateur peut garder la liberté d'imposer une démarche, une méthode et laisser le système libre sur certaines étapes.
- Autonomie restreinte : l'utilisateur prend toute décision au plan du contrôle du séquencement des étapes et ne sous-traite au système que les parties calculatoires (mise en oeuvre d'algorithmes appropriés). Le système dans ce cas peut suggérer des étapes en fonction des connaissances méthodologiques qu'il a, mais, chaque étape est soumise à approbation de l'utilisateur.

Notons que les systèmes de calcul algébrique actuels ont une autonomie nulle en ce sens, que tout le contrôle se fait par l'utilisateur et que le système ne dispose d'aucune connaissance même pour suggérer à l'utilisateur des poursuites possibles au travail en cours (ce qui serait un minimum).
La partie IV-3-2-3 met en évidence comment nous avons abordé ce problème dans CAMELIA.

ii) Plan des facilités d'enrichissement du système.

Ce type de système doit a priori être considéré comme ouvert et présenter toute facilité pour que l'utilisateur puisse incorporer des connaissances caractéristiques de son domaine

d'activité. Cela concerne :

* les connaissances de type contenu (résultats connus),
* les possibilités pour donner accès à de nouvelles connaissances opératoires (définition de nouveaux algorithmes - description des conditions d'application), * la méta-connaissance.

iii) Plan des justifications données par le système à l'utilisateur.

Deux types de justifications sont éventuellement (et à sa demande) à fournir à l'utilisateur :

* Justification des connaissances utilisées. Il est essentiel que le système puisse justifier les items de connaissances qu'il a utilisés pour établir une solution. Cela permet de conforter l'utilisateur à l'égard de la validité de la solution fournie. En cas de solution douteuse, cela permet de remettre en cause certaines connaissances qui peuvent être fausses ou dont les conditions d'applications n'ont pas été suffisamment précisées.
* Justification des étapes intermédiaires qui ont permis d'aboutir : raisonnements utilisés, choix faits et justification de l'usage des algorithmes mis en oeuvre.

III.2.3.2. Dialogue avec l'expert ayant à fournir les connaissances.

La base de connaissances initiales du système peut être de mise au point laborieuse et il est indispensable de disposer de facilités adaptées pour faire ce travail.

* Le langage d'expression des connaissances doit être aussi simple et proche que possible du langage des spécialistes du domaine. En particulier, il doit leur permettre d'utiliser leur vocabulaire.

* La connaissance doit être codée en items (nodules) indépendants de façon à pouvoir ajouter, enlever, modifier des items en ne travaillant que localement. Le codage doit au maximum être de type déclaratif. Le critère de modularité (nodularité !) est essentiel.
* On doit disposer d'outils d'interrogation de la base de connaissances afin de savoir facilement si le système connaît un résultat ou un savoir-faire et surtout savoir comment il le sait.
* On doit disposer d'outils de modification de la base de connaissances. En particulier une voie est ouverte pour développer des éditeurs intelligents adaptés à des langages spécialisés. Ainsi modifier la valeur d'un attribut, modifier les étapes d'un item décrivant un savoir-faire,... devrait pouvoir se faire avec des commandes adéquates. Un éditeur de texte fait l'affaire ; un éditeur de règles adapté au formalisme utilisé serait sûrement plus pratique.

* On doit disposer d'outils qui permettent de créer facilement un jeu de tests pour valider un nodule de

connaissances qui vient d'être modifié ou ajouté. Il s'agit de pouvoir créer un problème caractéristique et surtout d'imposer l'intervention du nodule mis à jour afin de voir son comportement de façon isolée. La difficulté en effet qui apparaît avec un système disposant de connaissances en vrac et gérées par le système est qu'il peut devenir difficile d'imposer le choix d'un nodule sur un problème donné. Deux solutions sont envisageables pour lever cette difficulté. Concevoir le système de contrôle pour que le pilotage à la main soit possible. On retrouve une possibilité jugée intéressante en III.2.3.1.i point 3) contrôle du système en autonomie restreinte pour l'utilisateur. L'autre solution consiste à pouvoir rajouter temporairement une métaconnaissance qui du point de vue du contrôle impose l'application de la seule connaissance mise à jour. Cette solution plus satisfaisante intellectuellement semble plus lourde du point de vue pratique puisque la mise à jour d'un nodule de connaissance nécessite la mise en place temporaire d'une méta-règle très spécialisée du type :

SI test en cours (nodule i) ALORS imposer (nodule i)

III.2.3.3. Conclusion

Ces éléments permettant d'atteindre un dialogue riche entre le système et les utilisateurs (expert ou utilisateur final) sont fondamentaux et constituent une grande partie de l'intérêt d'un système. Notons qu'ils doivent être une préoccupation permanente du rédacteur du système. Ces caractéristiques lui sont fort utiles pendant la phase de développement et de mise au point. Il est le premier concerné par cet aspect des choses.

III.2.4.CONCLUSION

Le problème tel que nous l'évoquons fait penser aux caractéristiques maintenant bien dégagées de ce qu'il est convenu d'appeler les "systèmes experts". Notre travail a contribué à préciser dès 1977 (VIVET 77) les idées liées à ce type encore nouveau à l'époque d'usage des ordinateurs. Cette voie des systèmes experts pour faire des mathématiques reste à approfondir. Des résultats encourageants ont été obtenus par D.B.LENAT (cf.II.3.3.3) avec ce type de système pour piloter des découvertes en mathématique, la formation de concepts. Il était urgent de se doter de moyens et d'outils permettant d'expérimenter ces idées en vue de conduire des preuves et des calculs algébriques.

Le chapitre suivant fait apparaître deux voies possibles pour aborder ce travail.

III - 3 - DEUX APPROCHES POSSIBLES POUR ABORDER L'ECRITURE D'UN SYSTEME EXPERT EN MATHEMATIQUES.

Ce type de système nécessite un volume de code important et il ne se justifie pas pour expérimenter ces idées d'écrire un système dans son ensemble. Des systèmes sont disponibles pour aborder cette démarche de travail : citons PROLOG comme système général présentant des mécanismes d'inférence. Citons les systèmes généraux pour faire du calcul symbolique MACSYMA et REDUCE.

III.3.1 Première approche.

La première approche consiste à partir d'un système présentant des possibilités d'inférence comme PROLOG et à écrire les éléments concernant les mathématiques. Le système SYCOPHANTE (BERGMAN 78) écrit à Marseille relève de ces idées pour faire du calcul algébrique. Ce système a été écrit en restant trop près de PROLOG qui s'avère ne pas etre très adapté pour écrire des algorithmes performants en calcul algébrique.

Un inconvénient majeur vient également de la rigidité du mode d'inférence de PROLOG. En particulier il est délicat d'intervenir au niveau du contrôle. Des travaux plus récents de DINCBAS et GALLAIRE ont cependant abordé ce problème important.

L'accessibilité difficile du système au moment où nous avons commencé notre étude fait que cette voie n'a pas été choisie pour expérimenter nos idées.

III.3.2. Seconde approche.

L'autre approche consiste à partir d'un système de calcul algébrique existant et à greffer dessus un système expert lui donnant les capacités d'inférence et de contrôle adaptées.

Cette approche présente des avantages divers :

- Certains de ces systèmes sont très distribués et accessibles et les utilisateurs sont nombreux dans des domaines d'application variés. On peut ainsi espérer que le travail fait puisse être effectivement utilisé et diffusé. L'autre intérêt est aussi de pouvoir être en liaison avec ces utilisateurs et à l'écoute de leurs problèmes. Il y a là une opportunité à saisir pour qu'un travail de recherche fondamentale réponde effectivement à un besoin.

- Ces systèmes restent bien adaptés pour écrire des algorithmes efficaces en calcul algébrique et il n'y a pas lieu de se priver de cette puissance de base si l'on souhaite atteindre des performances effectives.

- Certains de ces systèmes sont écrits sur un noyau LISP qui reste un langage bien adapté pour écrire des applications en Intelligence Artificielle. L'essentiel est de disposer des moyens pour que les différentes couches logicielles (lisp, calcul algébrique, système expert) puissent dialoguer.

Deux systèmes de calcul algébrique nous sont apparus intéressants pour aborder cette voie système expert intégré à un système de calcul algébrique. Il s'agit de MACSYMA et REDUCE.

MACSYMA développé au MIT (MARTIN71) à cause de sa richesse en général et de ses possibilités de pattern-matching en particulier était le meilleur candidat. En 1978 il ne semblait pas transportable et il ne nous a pas été possible de l'installer en France (il n'est disponible sous Multics à l'INRIA que depuis 1981-82). Aussi après l'avoir étudié de près, il nous a fallu abandonner l'idée de l'utiliser.

REDUCE (HEARN71), certes moins riche que MACSYMA, a présenté deux avantages essentiels. Il était d'abord facilement transportable, installé sur de nombreux types de machines et outil de travail d'un grand nombre d'utilisateurs. Le deuxième avantage est que les possibilités de dialogue entre la couche algébrique et la couche R-Lisp semblaient plus faciles à mettre en oeuvre. La documentation n'était pas très riche en particulier pour le mode symbolique mais nous avions accès aux sources.

Ces différentes raisons nous ont fait choisir REDUCE pour écrire un système expert en mathématiques appelé CAMELIA (Calcul Algébrique, Mathématiques Elémentaires et Intelligence Artificielle). Il a d'abord été nécessaire d'acquérir le système que nous avons installé sur CIRCE en 1980. Il a fallu ensuite se familiariser avec son utilisation (sans aucune aide extérieure!).

CHAPITRE 4

UNE REALISATION : CAMELIA

CHAPITRE 4

UNE REALISATION : CAMELIA

Calcul Algébrique
Mathematiques Elémentaires
et Intelligence Artificielle

GENERALITES

Nous avons discuté (II.1-2) de l'importance de l'examen des conditions d'application des algorithmes : disons que les algorithmes expriment COMMENT effectuer un calcul. Ainsi dans MACSYMA nous disposons de commandes pour prendre la transformée de Laplace et la transformée inverse d'une équation cela permet, étant donné une équation, d'obtenir l'expression de sa transformée. Mais ce que ne fait pas MACSYMA, c'est dire QUAND, pour quel type d'équations il est judicieux de chercher une solution en utilisant ces transformations. Cette compétence est nécessaire pour résoudre le problème. Actuellement, c'est l'utilisateur qui possède et utilise cette connaissance de type QUAND. C'est lui qui, en mode interactif le plus souvent décide des enchaînements, prend les décisions au plan méthodologique. Notre travail se place délibérément dans une voie qui consiste à représenter le savoir-faire d'un tel utilisateur, à chercher des méthodes de représentation des connaissances mathématiques caractérisant les situations et permettant de décider de l'enchaînement, du contrôle de l'application d'algorithmes. La solution aboutissant parfois à la décision de non application d'un algorithme. Ces connaissances doivent permettre un raisonnement sur le calcul en cours, la conduite de la preuve d'une propriété.

CAMELIA est un système qui a été conçu pour expérimenter ces idées d'intégration de compétences calculatoires et inférentielles. Le résultat est un système dans lequel sont présents et coopèrent des connaissances de type COMMENT et des connaissances de type QUAND. Il permet d'atteindre effectivement des solutions relevant du calcul algébrique et de la démonstration de théorèmes.

IV.I. PRESENTATION GENERALE

IV.1.1. Architecture du système : CAMELIA apparait comme un système ESSENTIEL pour faire des mathématiques ; établir des preuves et conduire des calculs symboliques. Ce système est caractérisé d'essentiel puisqu'il est inopérant si on ne lui fournit pas une base de connaissances dans laquelle est inscrite l'expertise mathématique nécessaire pour travailler dans le domaine pour lequel on souhaite l'utiliser. Dès 1977 (VIVET 77) nous avons donné sa structure, caractéristique de ce

que l'on appelle aujourd'hui les systèmes experts : à savoir, un moteur d'inférence qui raisonne sur une base de faits en utilisant des connaissances, fournies en VRAC dans une "base de connaissances. La figure 1 donne l'architecture générale du système.

IV.1.2. Domaines d'expérimentation : CAMELIA a été testé dans des domaines relevant de l'analyse élémentaire niveau classes préparatoires ou 1er cycle d'Université. Il a été testé sur des problèmes de preuve de parité de fonctions numériques, des calculs raisonnés de développements limités et de limites, des calculs de primitives et des évaluations de sommes finies portant sur des expressions contenant des fonctions combinatoires. Dans aucun de ces domaines les bases de connaissances constituées ne sont exhaustives quant aux savoir-faire manipulés, mais un critère important de ce type de système est d'être extensible : l'essentiel est de disposer des outils nécessaires pour incorporer facilement de nouvelles connaissances. L'intérêt d'avoir travaillé avec des domaines variés a été :

1°) de montrer l'aspect essentiel du système aucune connaissance mathématique ne figure (sous quelle que forme que ce soit) dans le moteur d'inférence.

2°) de montrer la souplesse du langage d'expression de connaissances qui s'est montré apte à exprimer des connaissances mathématiques dans des domaines variés.

IV.1.3. Implantation dans REDUCE

Nous avons déjà vu (III.3) que deux approches sont possibles pour aborder l'écriture d'un tel système. Nous avons choisi la deuxième, celle qui consiste à prendre comme base un système de calcul algébrique existant et à le doter des connaissances mathématiques nécessaires.

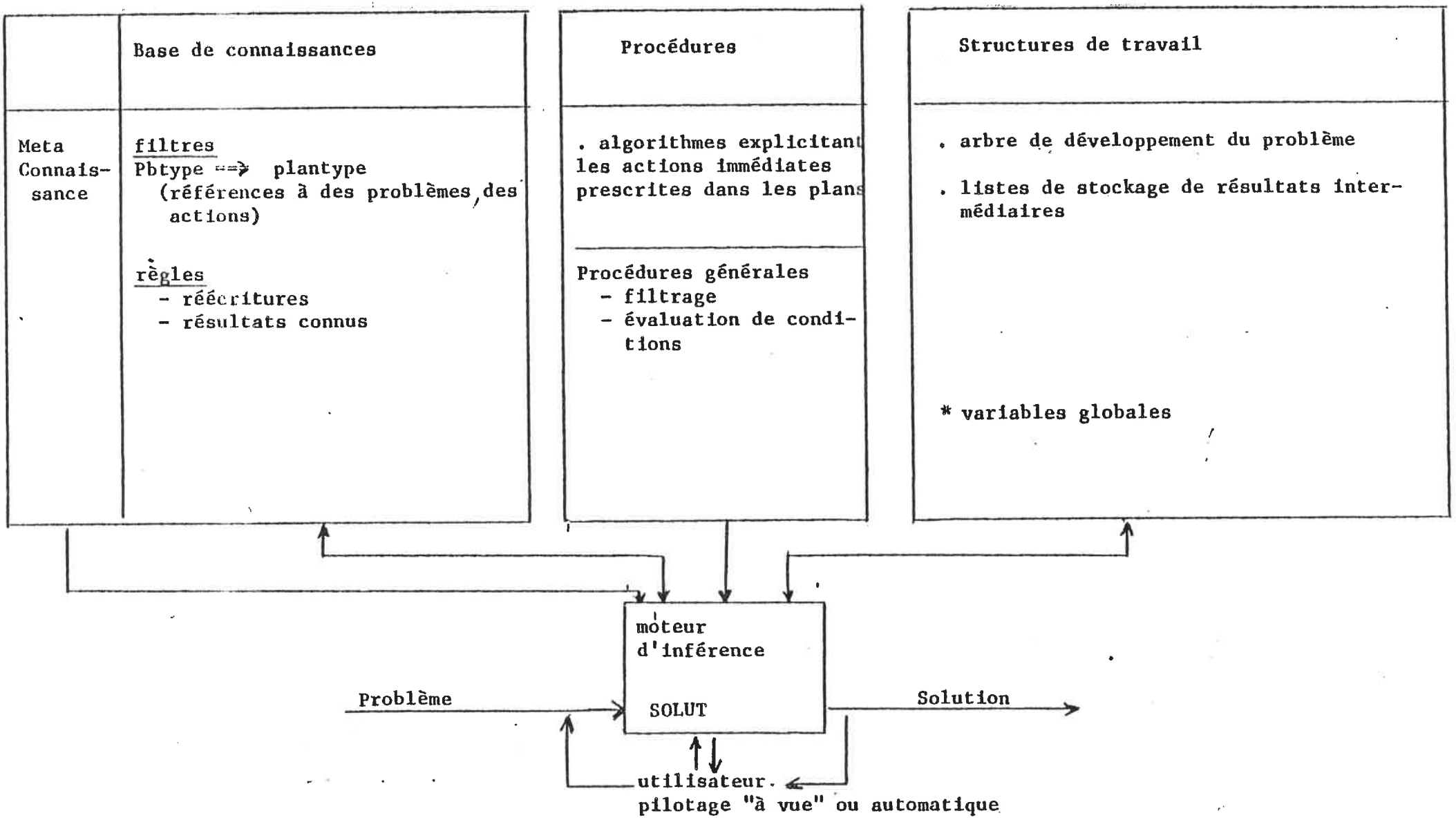
Après avoir vainement tenté d'accéder à MACSYMA en 1977/78 nous avons choisi REDUCE. Ce système de calcul algébrique est très diffusé et n'est pas trop spécialisé. Il a été développé dans un dialecte de LISP appelé R-LISP qui permet une écriture plus aisée que LISP traditionnel (on y dispose de procédures, des structures de contrôle usuelles en algorithmique. Une syntaxe type ALGOL permet d'éviter beaucoup de parenthèses dans l'écriture des programmes).

Une caractéristique intéressante est le fait que deux modes de travail sont accessibles à l'utilisateur :

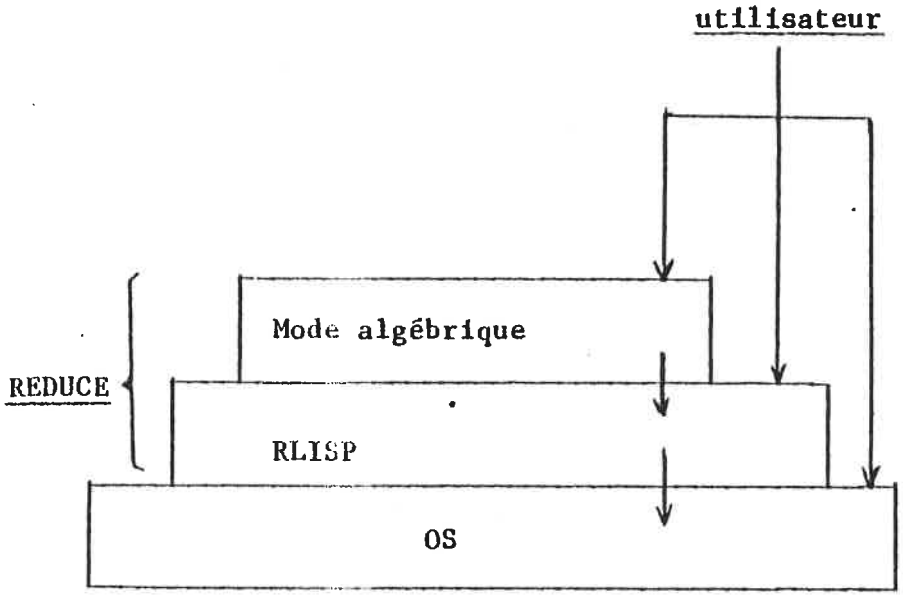
a) le mode algébrique qui permet les opérations algébriques formelles usuelles et dont l'utilisateur occasionnel peut souvent se contenter.

b) le mode symbolique ou mode LISP qui permet de travailler avec des listes, donne accès aux fonctions importantes de LISP, aux représentations internes du mode algébrique. Nous avons trouvé là tous les outils pratiques nécessaires pour écrire les aspects base de connaissances et moteur d'inférence de CAMELIA.

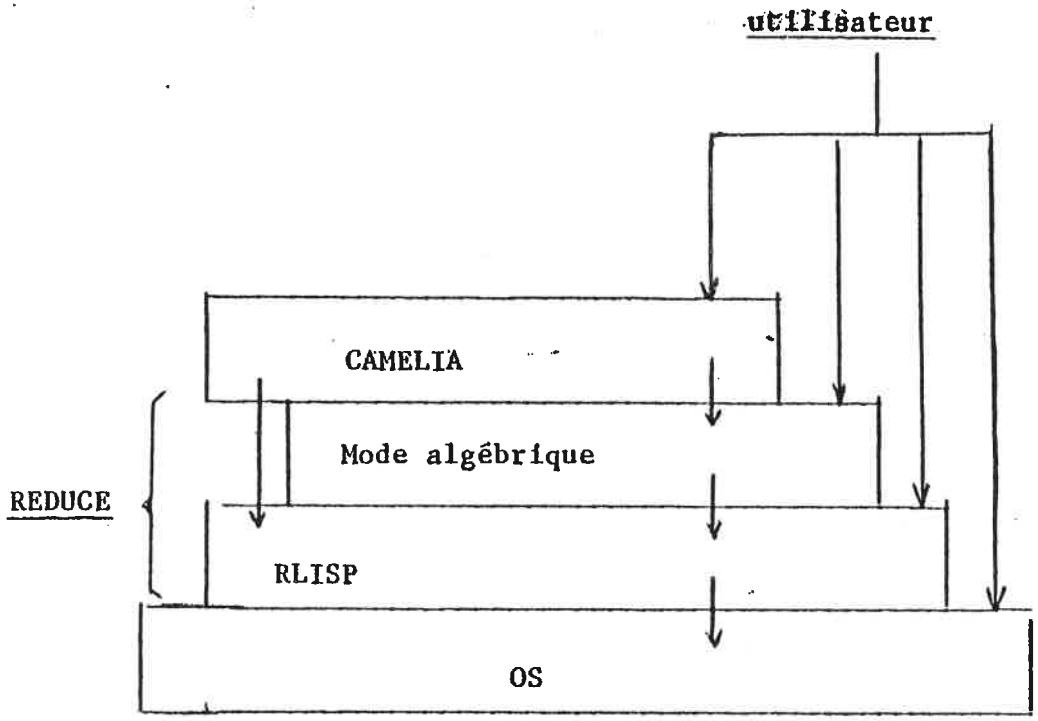
La communication entre les deux modes n'est pas trop difficile à mettre en oeuvre. Cela permet d'écrire en mode algébrique des procédures spécialisées en calcul algébrique. Ces



ARCHITECTURE GENERALE DE CAMELIA
Figure 1.



USAGE NORMAL
DE REDUCE



USAGE AVEC
CAMELIA

Figure 2 : la place relative de REDUCE et de CAMELIA

procédures sont appelables en mode symbolique et peuvent être des actions dont un plan demande l'exécution. De façon schématique on peut dire que certaines procédures algébriques représentent un savoir-faire opératoire de type COMMENT, le savoir-faire de type QUAND (contrôle de l'appel) étant défini par les plans exprimés comme des listes en mode symbolique. La figure 2 situe CAMELIA comme sur système de REDUCE.

IV.1.4. Forme des connaissances

Les connaissances pour assurer le pilotage, les raisonnements peuvent être données sous forme de plans qui représentent des démarches usuelles, des "savoir-faire" du mathématicien. Les plans peuvent être vus commes des "programmes flous" : sans que nous ayons cherché à formaliser cette idée; nous entendons par là, la description d'enchaînements d'actions plus ou moins déterministes, plus ou moins exécutables, qui explicitent une démarche tout en laissant de côté le détail des opérations qui seront effectivement exécutées pour résoudre un problème précis. Le flou, le droit à l'échec associé aux actions décrites dans les plans imposent de mettre en place des possibilités de choix et de retour-arrière au cours de la résolution d'un problème. Nous avons discuté dans (VIVET 77) du concept général de plan en mathématique et de quelques modèles de l'usage qui peut en être fait.

Quelles connaissances fournir ?

On peut constater que pour le mathématicien, un théorème est un opérateur qui permet d'agir sur une situation mathématique ; connaître un nouveau théorème, c'est se donner un nouveau moyen d'agir, de déduire une situation à partir d'une autre. Dès lors, il nous apparaît important de fournir les connaissances liées aux aspects opérateurs des théorèmes et ne pas se contenter de fournir les théorèmes sous une forme brute. Disons que "connaître un théorème c'est bien, savoir l'utiliser c'est mieux !".

Exemple :

Dans le domaine du calcul de primitives, l'énoncé du théorème :

(E1) $\int(f(x)+g(x))\,dx = \int f(x)dx + \int g(x)\,dx$

peut se traduire en une connaissance opératoire liée à l'usage de ce théorème par un énoncé.

(E2) Pour calculer $R = \int F(x)\,dx$

si $F(x)$ est une somme $f(x)+g(x)$, envisager les étapes suivantes:

1) Calculer $R1 = \int f(x)\,dx$
2) Calculer $R2 = \int g(x)\,dx$
3) Attribuer à R le résultat R1 + R2.

L'énoncé (E1) est manipulable tel quel par les systèmes de démonstration de théorèmes mais est peu efficace s'il s'agit de conduire un calcul de primitive. L'énoncé (E2) exprime une façon d'utiliser le théorème qui peut être utile pour calculer effectivement une primitive. La différence d'écriture entre ces deux énoncés fait apparaître ce que nous entendons par "savoir un théorème" (éventuellement "par coeur") et "savoir utiliser" un théorème, savoir l'usage qui peut en être fait.

CAMELIA est un système expert utilisant des connaissances mathématiques exprimées de façon opératoire dans l'esprit de l'énoncé E2. Une étape ultérieure serait évidemment de concevoir un système qui se contente en entrée des énoncés de type E1, utilisant éventuellement une génération d'énoncés de type E2. Cette étape est plus difficile puisqu'elle relève de l'apprentissage : à notre sens apprendre à être mathématicien, apprendre les mathématiques c'est pour une grande part être capable de passer de la lecture d'énoncés de type E1 (appris en cours ou lus dans un livre !) à la mise en oeuvre d'énoncés de type E2. Le rôle essentiel des exercices et problèmes dans l'enseignement des mathématiques ne relève-t-il pas de l'entraînement à percevoir les situations où un énoncé s'applique et comment il s'applique ?

IV.1.5. Un exemple simple pour voir la démarche

Soit à calculer l'intégrale $I = \int \frac{1 + x^2}{1 + x} dx$

Plusieurs méthodes sont disponibles pour effectuer ce calcul. On peut décomposer la fraction rationnelle en éléments simples ; on peut utiliser l'algorithme de RISCH (RISCH 69-70). Cet algorithme permet d'intégrer une classe importante de fonctions mais le résultat n'est pas toujours d'une exploitation facile (résultat très développé, introduisant des imaginaires, à la limite du lisible,...). Un article récent de P.SMITH et L.STERLING (ON INTEGRATION BY MAN AND MACHINE) paru dans ACM-SIGSAM, (nov-dec.83 - p.21-24) est éloquent à ce sujet. Nous pouvons au plan méthodologique, pour des raisons de lisibilité de résultats, réserver l'usage de cet algorithme à des cas où l'utilisateur sait que le calcul ne pose pas de problème, sait qu'il sait faire. Le plan (34) ci-dessous est une écriture qui permet de voir comment dans CAMELIA on peut exprimer cet usage contrôlé d'un algorithme puissant (on réserve ici l'algorithme de RISCH à l'intégration de fractions rationnelles.

```
prettyprint plan(34);
(PLAN34
   (DESCRIPTEURS VARSUBST (U V X R) OTOPEX (QUOTIENT) COUTMETHODE 15
      MSGSUCCES
      "FRACTION RATIONNELLE PAR RISCH "
      MSGECHEC
      "PAS SU INTEGRER LA FRACTION")
   POUR
   (CALCUL (PRIM (QUOTIENT U V) X) R)
   ESSAYER
   (SI
      (ET (POLYNOMEP U X) (POLYNOMEP V X))
      (SUC 2 (!%ATTRIB R (!%ALGEVAL (INT (QUOTIENT U V) X))) T)))
```

L'approche que nous avons utilisée nécessite de disposer d'un langage permettant d'exprimer la méthode retenue, en particulier il faut disposer des moyens d'expression des conditions. CAMELIA comporte une base de connaissances dans laquelle sont exprimées de telles démarches. Pour résoudre le problème, il convient alors de trouver, par filtrage entre l'énoncé du problème et les méthodes disponibles dans la base, un module de connaissances adapté pour aborder le problème en cours. Par ex-

emple, pour l'intégrale I ci-dessus, le plan (34) peut être choisi. Ce choix fait, l'exécution du plan est engagée. Dans ce cas, il n'y a pas de mauvaise surprise : le numérateur et le dénominateur de l'intégrande sont des polynômes en x (reconnus par le prédicat POLYNOMEP F X), la condition requise étant vraie on déclenche l'algorithme de RISCH (commande INT en mode algébrique de REDUCE grâce à l'action (!%ALGEVAL (INT F X))). Cet algorithme retourne une valeur attribuée au résultat RR. La trace de la solution fournie par CAMELIA a l'allure suivante :

```
prettyprint pb(181)$
    9: (SUC
          2
          (CALCUL (PRIM (QUOTIENT (PLUS 1 (EXPT X 2)) (PLUS 1 X)) X) RR)
          (!%ECRIRE (!%ALGEB RR)))


solut pb(181);
    10: " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (QUOTIENT (PLUS 1 (EXPT X
2))
    (PLUS 1 X)) X) RR)
    NO DU PLAN A UTILISER ?
(34)
    "J'ENVISAGE LE PLAN " PLAN34
    "???????????????? VOYONS SI " "FRACTION RATIONNELLE PAR RISCH "
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%ALGEVAL (INT (QUOTIENT (
PLUS
     1 (EXPT X 2)) (PLUS 1 X)) X)))
    *** RR DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " T
    "LE PLAN " 34 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "FRACTION RATIONNELLE PAR RISCH "
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

                      2
    (4*LOG(X + 1) + X  - 2*X)/2

    T
```

```
prettyprint pb(181)$
    11: (SUC
           2
           (CALCUL (PRIM (QUOTIENT (PLUS 1 (EXPT X 2)) (PLUS 1 X)) X) RR
)
           (!%ECRIRE (!%ALGEB RR)))


solut pb(181);
    12: " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (QUOTIENT (PLUS 1 (EXPT X
2))
    (PLUS 1 X)) X) RR)
    NO DU PLAN A UTILISER ?
999

    "J'ENVISAGE LE PLAN " INT44
    "??????????????????? VOYONS SI " "FRACTION AVEC DENOMINATEUR CONSTANT
"
    "J'ENVISAGE LE PLAN " INT28
    "??????????????????? VOYONS SI " "MONOME X**N INTEGRE / X"
    "J'ENVISAGE LE PLAN " INT50
    "??????????????????? VOYONS SI " "INT(C/F(X),X) EST C*INT(1/F(X),X"
    "J'ENVISAGE LE PLAN " INT27
    "??????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
    "J'ENVISAGE LE PLAN " INT39
    "??????????????????? VOYONS SI " "INT PAR PARTIE A PARTIR DE 1/(A+B)"
    "J'ENVISAGE LE PLAN " INT20
    "??????????????????? VOYONS SI " "RESULTAT USUEL "
    "##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR "
    "J'ENVISAGE LE PLAN " INT56
    "??????????????????? VOYONS SI " "INTEGRATION D'UN POLYNOME (RISCH)"
    "J'ENVISAGE LE PLAN " INT25
    "??????????????????? VOYONS SI " "(A+B)**N / C "
    "J'ENVISAGE LE PLAN " INT23
    "??????????????????? VOYONS SI " " DEVELOPPEMENT DU BINOME"
    "J'ENVISAGE LE PLAN " PLAN34
    "??????????????????? VOYONS SI " "FRACTION RATIONNELLE PAR RISCH "
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%ALGEVAL (INT (QUOTIENT (
PLUS
     1 (EXPT X 2)) (PLUS 1 X)) X)))
    " "
    ".... PROBLEME A RESOUDRE " T
    "LE PLAN " 34 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "FRACTION RATIONNELLE PAR RISCH "
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

                        2
    (4*LOG(X + 1) + X  - 2*X)/2

    T
```

Cet exemple montre :

i) comment nous pouvons envisager un contrôle heuristique de l'application d'algorithmes par l'écriture de règles (de plans) décrivant les conditions d'application.

ii) la nature des difficultés à surmonter, à savoir :

- la définition d'un langage d'expression de connaissances,
- la conception d'un moteur d'inférence permettant de gérer ces connaissances. En particulier, il doit effectuer les choix des items de connaissances utiles pour résoudre le problème proposé, gérer le dialogue avec l'utilisateur, gérer les retours arrière lorsque les heuristiques l'ont conduit à faire de mauvais choix sur lesquels il faut revenir,
- ce travail suppose en particulier de se définir des outils de filtrage, d'expression de conditions suffisantes.

IV.2. LA BASE DE CONNAISSANCES DANS CAMELIA

IV.2.1. Le langage d'écriture des connaissances

La base de connaissances permet de stocker des items de savoir qui constituent les règles du système expert.Le vocabulaire manque encore un peu dans la terminologie des systèmes experts pour désigner cette notion d'item de connaissances. Nous proposons de parler de "nodules" de connaissances ou de "molécules" de connaissances, le terme de molélcule présentant l'avantage d'évoquer des objets capables de se combiner, pas "durs" ni rigides mais flous. Ces items ont la forme générale de relations pouvant être décrites par un jeu extensible de descripteurs.

i) un premier type de règles appelées "filtres" a la forme générale suivante :

Pb-type = = > Plan-type

dans laquelle Pb-type est le "pattern" (modèle) d'un problème et Plan-type est un plan d'actions à envisager pour résoudre le problème type associé. Un jeu de descripteurs (liste d'attributs-valeurs) fournit de l'information sur l'utilisation de l'item. L'annexe A2 donne la syntaxe actuelle de ces règles destinées à exprimer des "savoir-faire".

ii) un deuxième type de règles permet d'exprimer des résultats connus ou des règles de réécritures connues dans un domaine donné. Ces règles décrivent des savoirs surs. L'annexe A3 donne la syntaxe actuelle de ces règles destinées à exprimer des "contenus".

Toutes ces règles, indépendantes entre elles, peuvent être entrées en VRAC, mises à jour, complétées de façon isolée. Elles sont évoquées par leur contenu. Ainsi seuls sont évoqués les filtres dont le problème caractéristique du Pb-type coïncide avec celui du problème en cours. Ces règles sont évoquées par paquets, rassemblées sous un nom qui caractérise le rôle qu'elles peuvent jouer.

Exemple :

i) le filtre

POUR (PRV (PAIRE (plus f g) x)) ESSAYER ...

a pour caractéristique (PRV PAIRE).

ii) Les règles de réécritures régissant la limite d'une somme de deux termes avec tous les cas particuliers où un (ou 2) terme(s) est (sont) fini(s) ou infini(s) ou indéterminé(s) seront toutes évoquées par leur nom caractéristique LIMPLUS. Lorsque, voulant calculer une intégrale, on décide de consulter l'ensemble des primitives connues, les intégrales dont le résultat est connu par coeur sont évoquées par leur nom caractéristique PRIM.

IV.2.2. Les Plans

IV.2.2.1. Les opérateurs : SUC, CHOIX, POURUN, POURTOUT

. Deux opérateurs n-aires s'avèrent essentiels pour exprimer des plans : ce sont

(SUC n (e1) (e2)...(en)) exprime la commande en séquence de n étapes -toutes doivent aboutir.

et

(CHOIX m (e1) (e2)...(em)) exprime le choix, l'alternative entre m étapes : une au moins doit aboutir.

Note: Nous avons retenu le seul opérateur SUC pour exprimer que les étapes doivent être toutes faites avec succès dans l'ordre des indices. On peut remarquer que l'ordre n'est pas toujours impératif et que si des étapes sont indépendantes entre elles certaines pourraient être traitées en parallèle puisqu'alors le seul objectif est qu'elles soient toutes faites. Nous n'avons pas tenu compte des possibilités de parallèlisme dans le moteur d'inférence et de fait c'est un traitement séquentiel des étapes qui est fait. Avec un moteur permettant des traitements parallèles, il y aurait lieu de se doter d'un opérateur exprimant qu'un ensemble d'étapes doit être exécuté et que l'ordre n'a pas d'importance. Une solution meilleure à ce problème consisterait à considérer qu'il n'y a pas d'ordre a priori et que c'est au moteur d'inférence de faire les choix (c'est ce qui se passe avec l'utilisation de " variables notations" cf.IV.2.2.4.).

Pour le cas où aucun ordre ne s'impose à priori, le moteur ferait un choix de nature heuristique. Il conviendrait, pour conserver un moteur général, que ces heuristiques soient elles-mêmes mises sous forme de règles.

Exemple :

Pour calculer "primitive (f(x) - g(x)) dx" avec le filtre : POUR (CALCUL (INT (Difference F G) x) R)

```
ESSAYER
(SUC 3 (CALCUL (INT F x ) R1 )
       (CALCUL (INT G x ) R2 )
       (R <==  R1 - R2 ))
```

On voit que les calculs de f(x) dx et g(x) dx sont indépendants. S'il s'avère que le calcul de f(x) dx aboutit et que le calcul de g(x) échoue, le temps passé à calculer f(x) est perdu. Avec une interprétation de SUC qui ferait commencer par le problème le plus difficile lorsque des problèmes sont indépendants, ce temps n'aurait pas été perdu. La seule difficulté est de savoir apprécier les ressources nécessaires pour établir la décision !.

. Les opérateurs POURTOUT et POURUN peuvent être vus comme des extensions de SUC et CHOIX. Ils ont surtout été utilisés pour écrire des plans mettant en oeuvre des opérateurs variaires.

Le plan s'écrivant (POURTOUT x L <Plantype>) exprime un enchaînement de plans qui doivent tous aboutir lorsque x par-

court l'ensemble fini L (décrit comme une liste de ses éléments).

Le plan s'écrivant (POURUN x L <Plantype>) exprime que pour au moins un x de l'ensemble fini L,<Plantype> conduit à un succès.

Du point de vue de l'interprétation, le moteur d'inférence fait parcourir l'ensemble L à x, dans l'ordre d'écriture des éléments. Ces éléments étant indépendants, il est clair que là encore un choix heuristique devrait permettre d'exhiber au plus vite une valeur de x faisant échouer un plan (POURTOUT...) si celui-ci doit échouer. De façon duale, un choix heuristique devrait permettre d'exhiber plus rapidement un élément x de L pour lequel un plan (POURUN ...) est vrai.

Exemples :

L'opérateur PLUS est n-aire et l'on ne sait pas a priori combien une somme aura de termes

i) Soit à exprimer la connaissance suivante : pour établir qu'une fonction F est paire en x, si F est une somme, il suffit d'établir que chacun des arguments de cette somme est une fonction paire. Indiquons que :

!%OP F renvoie l'opérateur principal de F

!%ORDROP F renvoie l'ordre de l'opérateur principal de F

!%INTERVAL a b (où a et b sont entiers) renvoie l'ensemble des nombres entiers de l'intervalle fermé (ab)

!%ARGI I F renvoie le i-ème argument de F. Le filtre pour exprimer cette connaissance peut alors s'écrire :

```
POUR (PRV (PAIRE F X)) ESSAYER
      (SI ((!%OP F) EST PLUS)
          (POURTOUT I (!%INTERVAL 1 (!%ORDROP F))
                      (PRV (PAIRE (!%ARGI I F) X)))) $
```

ii) Exprimons maintenant la méthode qui consiste, pour intégrer une somme, à cumuler les primitives de chaque terme de cette somme.Il est possible d'utiliser une notation (R1) qui note successivement les différentes primitives trouvées. Le résultat R st initialisé à 0. Puis chaque terme est intégré (avec résultat noté R1), R1 est cumulé à R et "montré" à l'utilisateur. Le filtre pour exprimer cette connaissance peut alors s'écrire :

```
POUR (CALCUL (PRIM F X) R) ESSAYER
        (SI ((!%OP F)  EST PLUS)
           (SUC 4
              (!%CREER R1) (!%ATTRIB R 0)
              (POURTOUT I  (!%INTERVAL 1 (!%ORDROP F))
                 (SUC 3   R1))  (CALCUL  (PRIM  (!%ARGI  I  F)  X)
                 (!%VAL R1))
                       (!%ATTRIB R   (PLUS   (!%OBJ   R)  (!%VVAL
                       R1)))
                       (!%MONTRER R1)) )
              (!%LIBERER R1) ) )  $
```

Les actions (!%CREER R1) (!%LIBERER R1) font apparaître l'aspect local de la notation. L'action (!%VAL R1) désigne la notation R1. Ainsi (CALCUL PRIM (!%ARGI I F) x) (!%VAL R1)) indique que je note R1 le résultat

de la primitive du i-ème argument de F. L'action(!%VVAL R1) permet d'accéder à la valeur notée R1. L'action (!%OBJ R) désigne l'accès à la valeur attribée à R.

IV.2.2.2. Les actions.

Les plans décrivent des enchaînements d'actions à envisager pour résoudre des problèmes.
Ces actions peuvent être :

1) Des actions problèmes : schémas de problèmes contenant des variables qui sont instanciées avec des éléments du problème initial au moment de la sélection du plan. Rappelons (cf.IV.2.1.) que les plans sont dans notre approche les membres droits de règles de réécritures (filtres) et que les variables substituables qui y figurent sont définies lors de l'unification du problème avec le problème-type du filtre. Ces actions sont caractérisées par le fait qu'elles peuvent échouer, c'est-à-dire qu'on peut ne pas réussir dans la résolution du problème associé.

2) Des actions immédiates : ces actions sont de nature purement algorithmique et produisent toujours un résultat.

Un usage intensif de ces actions immédiates est fait pour accéder aux parties d'un problème, d'une expression, créer des notations, etc. L'annexe C donne la liste des actions immédiates actuellement disponibles.

Un usage beaucoup plus important peut être fait pour déclencher l'exécution d'algorithmes écrits en mode algébrique de REDUCE. Dans ce cas nous retrouvons la possibilité effective de déclencher un algorithme après avoir décidé, lors des actions précédentes du plan, que ce déclenchement est impératif et justifié.

ASPECTS LIES A L'IMPLEMENTATION

i) . Extensibilité Il est essentiel de pouvoir enrichir facilement l'ensemble des actions immédiates. Ceci permet d'ajouter l'accès à des algorithmes qui peuvent être importants (si l'on travaille dans un contexte nouveau en particulier).

Actuellement :

- les actions immédiates ont un nom qui commence par les symboles !%,

- le lien entre le nom de l'action et la procédure qui exécute le travail se fait par l'intermédiaire d'une table appelée ACTIMMFONCT. Elle est constituée d'une liste de couples (nom de l'action - nom de la procédure). Pour des raisons pratiques, beaucoup de noms d'actions immédiates sont le nom de la procédure précédé de !%. Ce n'est pas une obligation et l'on a la possibilité de définir des actions synonymes : noms différents d'actions appelant la même procédure. En phase de mise au point, pour des exécutions différentes, on peut sans changer le nom de l'action dans le plan, accéder à des procédures différentes en changeant seulement le nom de la procédure dans le couple concerné.

- une procédure AJOUTACTIMM a été créée pour ajouter un couple (nom d'action -nom de procédure) à cette table. Cela permet de compléter de façon interactive des liens de ce type s'ils ont été oubliés au chargement.

ii). La procédure EXECACTIMM est chargée de l'interprétation des actions contenant des actions immédiates. Ces actions peuvent être composées à des profondeurs quelconques.

Exemple : DLSIN (X,PT,N) étant une procédure algébrique permettant de donner le développement limité de SIN(X) à l'ordre N autour du point PT, la mise dans R du coefficient du 2e terme du développement de SIN(X) à l'ordre 5 autour de 0 peut être commandée par :

```
(!%ATTRIB R (!%ARG1 (!%ARG2 (!%ALGEVAL DLSIN X 0 5))))
```

dans lequel : !%ATTRIB est l'action d'affectation,
!%ARG1 et !%ARG2 sont les accès aux arguments d'une expression,
!%ALGEVAL est l'action qui lance l'exécution de l'algorithme DLSIN avec les arguments (X, 0, 5).

iii). Cas particulier : actions retournant NIL.

Certaines actions peuvent renvoyer NIL ou au contraire un résultat si le calcul associé a abouti. Ces actions peuvent être utilisées comme des conditions ou dans des affectations.

Exemple : L'action !%MONOMEP (E,X)
renvoie - NIL si E n'est pas un monôme en X
- la liste ((A coef)(N exposant)))
si E est un monôme (c'est à dire
une des formes suivantes : A * X ** N, X ** N (A = 1), A * X (N = 1), A (N = 0).

Exploitation du résultat : Elle peut se faire sous deux formes :

- Comme condition : on peut écrire dans un plan :

```
(SI (!%MONOMEP F (X))
    (action))
```

- Comme pseudo-condition : (cf IV.2.2.3) la commande

```
(!%ATTRIB R (!%MONOMEP F (X))
```

permet de mettre dans R les éléments permettant d'accéder au coefficient et à l'exposant de F si F est effectivement un monôme en X.

Ce qui est intéressant, c'est que si F n'est pas un monôme en X, la tentative d'attribuer NIL à R va faire échouer l'affectation !%ATTRIB qui retourne NIL elle-même. L'interprétation de l'affectation dans les plans interdit en effet d'attribuer NIL à une variable. L'échec de cette affectation peut forcer un retour arrière (si on se trouve sous un SUC) ou la prise en compte d'un autre choix (sous CHOIX).

IV.2.2.3. Les conditions

Les conditions ont un rôle fondamental dans un système comme CAMELIA et beaucoup de la puissance du système vient de la richesse des moyens d'expression des conditions.

a) USAGE : Les conditions interviennent directement :

1) dans les plans de type (SI (<cond>) (<plantype>) pour lesquels l'exécution de <plantype> n'est envisagée que si la condition est interprétée comme vraie.

2) dans les règles conditionnelles (cf IV.2.3.3.) elles sont alors placées comme expression logique située derrière l'attribut CONDIT dans le descripteur de la règle.

b) Forme des règles :

- L'annexe A5 décrit la syntaxe détaillée des conditions.
- La forme (<Prédicat> <Args>) est intéressante parce qu'elle permet d'exprimer des conditions extrêmement variées. En effet, <Predicat> peut être le nom d'une fonction définie dans le système ou par l'utilisateur ; la valeur de la condition est alors VRAI ou FAUX suivant que cette fonction appliquée aux arguments renvois NIL ou pas.

L'annexe A5 donne la liste des prédicats usuels actuellement disponibles.

c) Les conditions problèmes :

Un cas intéressant de la forme (<Predicat> <Args>) est celui des conditions problèmes : si dans cette écriture, <Predicat> n'est pas associé à une procédure, alors la condition est interprétée comme une propriété qu'il faut chercher à prouver. Dans ce cas l'évaluation de la condition se ramène à la preuve d'une propriété. Si la preuve aboutit alors la condition est considérée comme vraie sinon elle est considérée comme fausse.

L'évaluation se fait dans ce cas par un appel récursif du moteur d'inférence et l'on peut dire que toute propriété dont la preuve peut être conduite dans CAMELIA est susceptible de devenir une condition. On voit là tout l'intérêt d'avoir un système intégrant des compétences en matière de preuves.

Exemple :

Si une méthode de travail marche pour des fonctions paires en X on peut écrire :

```
(SI (PAIRE F X)
    (méthode))
```

Dans ce cas le moteur ne lancera l'exécution de la méthode qu'après avoir établi que F est paire en X c'est-à-dire avoir résolu le problème (PRV (PAIRE F X)).

Interprétation :

Si CP est une condition problème, on peut considérer que les deux écritures suivantes sont équivalentes :

```
(SI (CP)                    et        (SUC 2
    (méthode))                            (PRV CP)
                                          (méthode))
```

La première semble plus naturelle et plus usuelle dans la façon de s'exprimer chez les mathématiciens.

d) Trois prédicats particuliers :

Pour renforcer la puissance d'expression des conditions, trois prédicats ont été définis : ils ont la syntaxe suivante :

```
TOUSVRAI    <L>  <cond>
UNVRAI      <L>  <cond>
AUCUNVRAI   <L>  <cond>
```

dans lesquels . <L> est une liste linéaire d'objets
. <cond> une condition quelconque (éventuellement une condition problème).

La condition <cond> comporte une variable (que nous appelons joker) marquée par le symbole ? .
Les interprétations sont les suivantes :

- TOUSVRAI <L> <cond> est vrai si <cond> est vraie pour toute valeur du joker prise dans <L>

- UNVRAI <L> <cond> est vrai si <cond> est vraie pour au moins une valeur du joker prise dans <L>

- AUCUNVRAI <L> <cond> est vrai si <cond> est fausse pour toute valeur du joker prise dans <L>.

Exemple :
Pour montrer qu'une somme S de fonctions paires est paire, on peut écrire une condition comme :

```
(SI (TOUSVRAI (QUEUE S) (PAIRE ? X)) T)
```

((QUEUE S) est la fonction qui renvoie ce qu'il reste de S si on enlève son premier élément : CDR en LISP, SAUFPREMIER LOGO...).
En prenant :

```
S = (PLUS (COS X)(ABS X)(EXPT X 2)(EXPT (SIN X)2))
```

on pourra aboutir à la preuve que S est paire en x en prouvant que tous les arguments de la somme sont des fonctions paires.

NOTES :

1) Ces trois prédicats permettent d'exprimer des conditions puissantes et peuvent être combinés avec les opérateurs ET, OU, NON aux autres formes de conditions.

2) Nous n'avons pas prévu que ces trois formes se combinent elles-mêmes ou entre elles. Cela permettrait d'atteindre des conditions comme (TOUSVRAI <l1> (AUCUNVRAI <l2> <cond ?1 ?2>)) ; cela suppose de disposer de plusieurs jokers -c'est une extension qui reste à faire.

e) Les pseudo-conditions :

Il s'agit de conditions qui peuvent faire échouer un plan mais qui ne sont pas exprimées derrière un SI (et sont en ce sens moins faciles à voir).

Le mécanisme est basé sur le droit à l'échec des actions immédiates (cf.IV.2.2.2.iii). L'algorithme se termine mais ne détermine pas ce que l'on cherchait.

Dans ce cas l'action immédiate renvoie la valeur NIL. Lorsque cette action est étape d'un plan, nous avons un dispositif qui permet de forcer un retour arrière (vers le dernier choix fait par exemple). L'échec d'une étape est en effet repéré par le renvoi de NIL. Le cas couramment utilisé est celui de l'affectation (!%ATTRIB) et de l'action !%RESULTATCONNU qui permet de chercher si un résultat est connu dans une table.

Exemple :

Soit (SUC n =
(!%ATTRIB R (!%RESULTATCONNU F INT))
=)

(!%RESULTATCONNU F INT) va consulter la table des intégrales dont le résultat est connu.

. Si F est dans cette table, le résultat est mis dans R et le travail à faire sous SUC se poursuit en exploitant R par exemple.

. Si F n'est pas dans cette table, !%RESULTATCONNU retourne NIL, !%ATTRIB refuse d'affecter NIL à R et retourne NIL. Cela suffit pour faire échouer SUC.

IV.2.2.4. Les variables de notations

Dans les plans destinés à conduire un calcul (cf.plan 21 de l'annexe D2 pour intégrer une différence) il est souvent utile d'utiliser une notation pour repérer un résultat qui sera repris ultérieurement. Le mathématicien qui numérote des équations au cours d'un calcul ne fait rien d'autre. Le problème qui se pose alors est celui de la durée de vie d'une notation. La solution de ce problème n'est pas évidente si l'on observe que les plans peuvent s'appeler les uns les autres, y compris récursivement. Ces variables de notation doivent être locales aux plans.

Pour conserver l'indépendance entre les filtres et toute la liberté, nécessaire pour fixer une notation au cours de l'expression d'une démarche, nous avons programmé un mécanisme de création et de libération de notations et nous avons défini des fonctions d'accès aux objets repérés par une notation.

Exemple:

Si l'on décide pour intégrer $\int (f(x) - g(x))\,dx$ de noter R1 : $\int f(x)\,dx$
R2 : $\int g(x)\,dx$

Les notations R1 et R2 peuvent être déjà utilisées pour noter quelque chose, éventuellement dans le même contexte. Ainsi si l'on calcule :

$$\int u(x) - (v(x) - w(x)) \; dx$$

Une première fois : R1 peut noter $\int u(x)\,dx$
R2 $\int v(x) - w(x)\,dx$

Une deuxième fois : R1 note $\int v(x)\,dx$
R2 note $\int w(x)\,dx$

et les deux calculs s'imbriquent, la même notation servant à identifier des objets différents au cours du temps.

Solution actuellement retenue

Pour chaque notation N, nous associons une pile de ses représentants actuels qui sont des noms symboliques (créés par un générateur de noms) et c'est à ce nom symbolique que nous attribuons la valeur du résultat.

Les actions que nous avons définies sont alors les suivantes :

!%CREER N — permet de créér la notation N par appel du générateur de noms. Actuellement, les noms ont la syntaxe SERV n° où n° est un numéro sur 4 chiffres. CREER admet comme argument un identificateur ou une liste d'identificateurs.
L'utilisation d'une notation N ainsi créée se fait avec deux fonctions d'accès.

!%VAL N — qui donne le nom symbolique (SERV...) associé (lié) à N. Cela représente le nom actuel de la notation.

!%VVAL N — qui donne la valeur de l'expression affectée à (SERV...) actuellement liée à N

La libération des notations se fait par :

!%LIBERER N — qui libère la notation en dépilant le dernier nom symbolique associé à N.

Remarques 1.

i) - pour les problèmes de type "changement de variable" la commande !%VAL permet de créer le nom de la nouvelle variable.

ii) - généralement, !%VAL N permet de "noter" N un objet et !%VAL N permet d'accéder à la valeur associée à la notation N.

Le schéma type de l'usage d'une notation temporaire R1 dans un plan est alors le suivant :

Création	(!%CREER R1)	
	.....	
Définition de la notation	(!%ATTRIB (!%VAL R1) exp)	on note R1 l'expression
	ou	
	(CALCUL exp (!%VAL R1))	on note R1 le résultat du calcul
	ou	
	(EVALUER exp (!%VAL R1))	ou de l'évaluation

```
                .....
Utilisation
                (!%ATTRIB R f(!%VVAL R1))        on utilise
de la            ou                              l'expression
                (CALCUL  f(!%VVAL R1)    )       notée R1
notation        ou
                (EVALUER f (!%VVAL R1)  )
                (SI (condition dépendant de !%VVAL R1)
                etc... (   plan))

Libération      (!%LIBERER R1)
```

ii) Les noms choisis !%VAL et !%VVAL ne nous paraissent plus très appropriés : il apparaît que le choix !%NOT et !%VNOT serait plus parlant puisque permettant mentalement des idées comme : "noter R1 l'expression" ou "attribuer à la notation R1 l'expression ..." qui se traduirait par (!%ATTRIB (!%NOT R1) exp). L'usage se faisant alors en lisant (!%VNOT R1) comme "Valeur Notée R1".

iii) Un tel usage des notations alourdit sensiblement l'écriture des plans mais il nous a semblé important d'implémenter ce dispositif qui fait réellement partie du mode d'expression du mathématicien.

Remarques 2 - Aspect implémentation.

i) Nous avons retrouvé les problèmes classiques liés à la durée de vie des variables locales à un bloc.

L'implémentation actuelle pourrait être revue afin d'éviter l'explicitation de la création et de la libération des notations à l'entrée et à la sortie d'un bloc. Une déclaration "NOTATION n" dans le descripteur du plan permettrait alors de typer ces objets (qui ne doivent pas être confondues avec les variables substituables (descripteur VARSUBST) par exemple). La difficulté vient du fait que nous sommes dans des plans et qu'une action peut échouer entre la création et la libération de la notation.

ii) Si dans un plan une notation a été créée et qu'un échec survient sans que l'on soit parvenu à l'étape de libération, il y a lieu malgré tout de libérer la notation créée pour mettre la pile au niveau convenable. Cela est traité dans le moteur d'inférence : dans l'interprétation d'un SUC par exemple, si l'on constate, lors d'un échec que des notations ont été créées, il faut les libérer avant de permettre le retour arrière.

Exemple :

```
(SUC i
(!%CREER R1)
...
(Action qui échoue)
...
(!%LIBERER R1)).
```

Il y a lieu de libérer effectivement R1 à l'arrivée de l'échec. L'implémentation que nous avons choisie permet de trou-

ver assez facilement les notations qu'il convient de libérer lors d'un échec. Une solution de création/libération implicite des notations, poserait beaucoup plus de problèmes au niveau du moteur d'inférence.

IV.2.2.5. Deux filtres particuliers PLANNIL-PLANT

a) La démarche que nous utilisons nous a conduit à définir deux savoir-faire de type universel :

i) L'ignorance universelle (notée PLANNIL) qui garantit l'échec : le plan échoue quel que soit le problème auquel on cherche à l'appliquer.

ii) Le savoir-faire universel (noté PLANT) qui garantit un succès : le plan aboutit à un succès quel que soit le problème auquel on cherche à l'appliquer.

L'annexe D1 montre l'écriture de ces plans universels.

b) Usages.

i) Ces deux filtres jouent un rôle dans l'interface du système avec l'utilisateur (Cf.IV.4) : dans la conduite d'une session, on peut paramètrer le système pour qu'il nous demande l'item de connaissance à utiliser. Lui indiquer l'un ou l'autre de ces filtres universels peut être l'occasion d'écarter un sous-problème parce que l'on sait que le système ne peut pas le résoudre, ou bien parce que l'on sait qu'il peut le résoudre mais que l'on voit bien la preuve. Ce peut être l'occasion de forcer un retour arrière quand le système semble mal parti ... etc. Ces filtres sont de bons outils pour la mise au point de la base de connaissances mais seront également utilisés dans la phase d'utilisation du système.

ii) Un usage intéressant de PLANNIL a été fait dans les méta-règles lors de la détection d'un "problème absurde" (Cf.IV.2.4.2). Un problème est déclaré "absurde" si la propriété qu'il faut prouver est visiblement fausse (soit parce que contradictoire avec un résultat connu dans la base de connaissance, soit parce que ne vérifiant pas une propriété simple requise (vérification sémantique) pour que la propriété à prouver puisse être vraie).

Exemples :

- Le problème (PRV (IMPAIRE (COS X) X)) est absurde car la propriété est contradictoire avec le résultat connu (PAIRE (COS X) X).

- De même le problème (PRV(PAIRE (EXPT E X) X)) est absurde car f(x) = E * * x ne vérifie même pas f(1) = f(-1).

Si un problème est reconnu comme absurde les méta-règles, au lieu de declencher l'évaluation des plans permettant d'aborder le problème, laissent PLANNIL comme seul plan à exécuter. Ceci permet d'être assuré d'un échec relançant un retour arrière.

IV.2.3. Expressions de contenus

Il s'agit à ce niveau de disposer des outils permettant de stocker facilement les résultats usuels connus par coeur du mathématicien.

Les résultats et propriétés connues peuvent s'exprimer sous forme de règles de réécritures. Pour cela nous avons défini un opérateur algébrique binaire REEC dont le 1er argument est le membre gauche de la réécriture et le 2ème argument est le membre droit. Ceci permet d'écrire les formules dans leurs notations usuelles (sans les mettre sous forme de listes préfixées). La forme polonaise préfixée est également admise par le système.

Les règles sont décrites par une liste de descripteurs. Par exemple la liste des variables substituables suit le mot VARSUBST.

IV.2.3.1. Les résultats connus

Le langage permet d'exprimer et de stocker facilement les propriétés connues.

Exemple :

```
LISP ;
REG (7) := '(PAIRE7 (DESCRIPTEURS VARSUBST (X))) ;
ALGEBRAIC ;
PAIRE7 := REEC (PAIRE (COS (X), X) , T) ;
```

exprime que la fonction x ===> cos (x) est paire en x (T signifiant TRUE : constante VRAI en LISP).

On peut aussi écrire :

```
LISP ;
REG (7) := '(PAIRE7 ...) ;
PAIRE7 := '(REEC (PAIRE (COS X) X) T) ;
```

IV.2.3.2. Les formules connues : les règles

Nous donnons en annexe A3 la syntaxe des règles. Le même opérateur REEC est utilisé.

Exemple :

$$\int \frac{1}{x}\,dx \;==>\; LOG\,|x|$$

s'écrit sous la forme :

```
LISP ;
REG (21) := '(PRIM21 (DESCRIPTEURS VARSUBST (X)) ) ;
ALGEBRAIC ;
PRIM21 := REEC (INT (1/X,X) , LOG(X)) ;
```

Les règles dans CAMELIA peuvent être beaucoup plus riches que ces formes simples. En particulier nous avons repris les idées de règles conditionnelles (VIVET 73) et nous avons étendu les possibilités de contraintes sur les instanciations avec ce que nous avons appelé les règles paramétrées.

IV.2.3.3. Les règles conditionnelles

- On peut être amené à n'utiliser une règle que si une certaine condition est satisfaite. Dans ce cas la condition est

écrite derrière le descripteur CONDIT. La richesse d'un système basé sur l'usage de règles de production ou de réécriture dépend largement de la richesse des conditions qui peuvent être exprimées. Nous avons vu en IV.2.2.3 la richesse des conditions que nous pouvons exprimer : toutes ces formes (cf.annexe A5 pour récapitulation) sont utilisables y compris les conditions problèmes. La condition peut porter sur des variables qui seront instanciées au moment de l'unification :

- Exemples : . Le résultat connu en calcul intégral :

$$\int (ax+b)^m \, dx \Longrightarrow \frac{1}{a} \frac{(ax+b)^{m+1}}{m+1}$$

peut s'écrire :

```
REG(46) := '(PRIM46 (DESCRIPTEURS VARSUBST (X A B M)
                     CONDIT (ET (NONVUS X A) (NONVUS X
                     M) (NONVUS X B)(M NESTPAS -1) ))
                     ;
ALGEBRAIC ;
PRIM46 := REEC (PRIM((A*X+B)**M,X), (1/A) * ((A*X+B)-
**(M+1))/(M+1)) ;
```

On observera que la réécriture exprimée en mode algébrique par l'affectation de PRIM46 est décrite par le jeu de descripteurs visibles dans REG(46). Le descripteur VARSUBST est suivi des variables substituables de la règle, le descripteur CONDIT est suivi de la condition d'application de le règle dans ce cas :

$x \notin A$ et $x \notin M$ et $x \notin B$ et $M \neq -1$

. En calcul de limites :

Deux cas différents qui peuvent se produire pour calculer lim P/Q si l'on sait que P tend vers plus l'infini Notons
PINF pour plus l'infini
MINF pour moins l'infini
IND pour indétermination.

Lorsqu'on calcule une limite par rapport à une variable x, on peut réécrire :

. PINF /A comme PINF uniquement si $x \notin A$ et $A \neq$ PINF et $A \neq$ MINF et $A > 0$.

. PINF/A comme MINF uniquement si $x \notin A$ et $A \neq$ PINF et $A \neq$ MINF et $A < 0$. Ceci nous permet d'écrire les 2 règles ci-dessous :

```
 REG (181):='(LIMQUOT181 ( DESCRIPTEURS VARSUBST (A X)
                          PARAM (X)
                          CONDIT (ET (NONVUS X A)
                                     (A NESTPAS PINF)
                                     (A NESTPAS MINF)
                                     (A NESTPAS IND )
                                     (POSITIF A) ))) ;
REG (182)=:'(LIMQUOT182 ( DESCRIPTEURS VARSUBST (A X)
                         PARAM (X)
                         CONDIT (ET (NONVUS X A)
```

```
                                    (A NESTPAS PINF)
                                    (A NESTPAS MINF)
                                    (A NESTPAS IND )
                                    (NEGATIF A)     ))) ;
ALGEBRAIC ;
  LIMQUOT181 := REEC (PINF / A , PINF ) ;
  LIMQUOT182 := REEC (PINF / A , MINF ) ;
```

On observera l'écriture du descripteur PARAM qui indique la liste des variables substituables (ici (X)) auxquelles on impose une substitution au moment du filtrage : ce symbole x doit être ici la variable par rapport à laquelle on recherche la limite.

- Processus d'évaluation :

. Pour appliquer une règle conditionnelle à un problème P1 :

1) On lance l'unification de P et du membre gauche de la règle.

2) Si succès on applique la substitution obtenue à la condition.

3) On évalue la condition ainsi instanciée.

4) Si la condition est vraie, on renvoie le membre droit après lui avoir appliqué la substitution.

- On peut observer que si la condition fait intervenir des conditions-problèmes, on peut être amené à lancer le moteur d'inférence sur des preuves - pour seulement vérifier qu'une règle s'applique - Il est difficile d'apprécier à quel niveau de récursion le moteur travaille dans ce cas mais on se rend compte de toute la puissance d'expression que l'on gagne à avoir inclus des mécanismes de preuve dans un système expert.

IV.2.3.4 Les règles avec paramètres

Il peut être important d'imposer des couplages de variables lors de la recherche d'une substitution permettant d'instancier un problème et le membre gauche d'une règle.

La solution que nous avons retenue permet d'imposer des substitutions lors de la reconnaissance d'une forme.
Les variables dont la subsitution est imposée sont alors appelées paramètres formels et leur liste est donnée derrière le descripteur PARAM dans la règle.

La fonction qui reconnait la forme est alors un algorithme d'unification (procédures EXUNIFIP, EXUNIFIE, FORMEXPLICITE, dans CAMELIA (cf annexe B)) auquel il faut fournir la liste des paramètres actuels. L'algorithme doit alors vérifier la concordance entre les paramètres actuels et les paramètres formels (décrits derrière PARAM dans la règle).

Exemple 1 :

La primitive d'un monôme en x intégré par rapport à se fait facilement sitôt que l'on a reconnu qu'il s'agit d'un monome en x.

Si l'on cherche à intégrer $f = p^4 . t^2$ par rapport à t, il faut imposer l'instanciation (x:t) pour trouver (p**4)* (t**3 /3). Ce résultat est évidemment différent si l'on intègre par rapport à p. Pourtant f peut être vu comme monôme en p ou monôme en t.

Exemple 2 :

La fonction MONOMEP (ex, x) renvoie NIL si ex ne peut pas ête considéré comme monome en x, ou bien le couple (instanciation de A, instanciation de n) si ex peut être considéré comme un monome A * x ** N.

Pour cela MONOMEP exploite la famille de règles suivantes :

```
EXPMONX    : = '((X) (DESCRIPTEURS VARSUBST (X) PARAM (X)));

EXP1MONAX  : = '((TIMES A X) (DESCRIPTEURS VARSUBST (A X) PARAM
               (X) CONDIT (NONVUS X A)));

EXPMONXN   : = '((EXPT X N) (DESCRIPTEURS VARSUBST (X N) PARAM
                            (X) CONDIT (NONVUS X N)));

EXP1MONAXN : = '((TIMES (EXPT X N) A)
                        (DESCRIPTEURS VARSUBST (A X N) PARAM (X)
                        CONDIT (ET (NONVUS X A) (NONVUS X N)) ));

EXP2MONAX  : = '((TIMES X A) (DESCRIPTEURS VARSUBST (A X) PARAM
               (X) CONDIT (NONVUS X A)));

EXP2MONAXN : = '((TIMES A (EXPT X N))
                        (DESCRIPTEURS VARSUBST (A X N) PARAM (X)
                        CONDIT (ET (NONVUS X A) (NONVUS X N)) ));
```

pour voir si ex est une forme x, ax, x^n, ax^n ou xa^n $x^n a$.

(Le cas ex indépendant de x, c'est-à-dire monôme ex*x**0 est traité directement dans MONOMEP. On aurait pu lui associer la règle

```
EXPMONC : = '((U) (DESCRIPTEURS VARSUBST (U) PARAM (X)
                          CONDIT (NONVUS X U)) );
```

mais le test direct est plus rapide que l'application d'une règle).

Si l'on reprend l'exemple 1 ci-dessus, le passage de la variable d'intégration se fait alors par son passage en paramètre actuel à l'action MONOMEP. Décrivons le filtre pour intégrer un monome (en mettant le résultat dans R) ; Ce filtre (n° 29) a pour problème type (CALCUL (PRIM F X)R).
Le plan-type (membre droit) comporte 4 étapes :

- Définition d'une notation R1

- Par le mécanisme des pseudo-conditions (cf IV 2 2 3) on attribue à la notation R1, la liste (coefficient, exposant) de F si F est un monôme de la forme A*X**N. Si F n'est pas un monôme en x, MONOMEP renvoie NIL, l'action !%ATTRIB échoue et le plan échoue (x paramètre transmis aux règles qui détectent que F est un monôme en x doit bien sûr être la variable d'intégration.

- La troisième action affecte R le résultat a*x**(n+1)/(n+1) en polonaise préfixée à partir du couple (a,n) noté R1 (accès à la valeur notée R1 par (!%VVAL R1). L'action (!%TETE (!%VVAL R1) permet alors d'accéder au coefficient du monôme alors que l'action (!%TQ (!%VVAL R1), permet d'accéder à l'exposant.

La dernière étape est la libération de la notation R1 utilisée localement dans ce plan. L'écriture complète du filtre peut alors être comme suit.

```
%INTEGRER UN MONOME ;
PLAN (29) : = (PRIM29 (DESCRIPTEURS VARSUBST (F X R)
                      COUTMETHODE 6
                      MSGSUCCES "INTEGRATION D'UN MONOME")

              POUR (CALCUL (PRIM F X) R) ESSAYER
                      (SUC 4
                      (!%CREER R1)
                      (!%ATTRIB (!%VAL R1) (!%MONOMEP F (X))
                      (!%ATTRIB R (!%SIMPLIF (QUOTIENT
                                  (TIMES (!%TETE (!%VVAL R1))
                                  (EXPT X
                                  (PLUS (!%TQ (!%VVAL R1))
                                  1)))

                                  (PLUS (!%TQ  (!%VVAL  R1))  1)  ))
                                  )

                      (!%LIBERER R1)

                ));
```

Le descripteur COUTMETHODE 6 est une indication sommaire donnée par l'expert pour indiquer qu'il s'agit d'une méthode simple. Le descripteur MSGSUCCES permet d'accéder au message caractéristique qui est transmis à l'utilisateur :
i) lors de la tentative d'application de la méthode sur un problème :
"VOYONS SI INTEGRATION D'UN MONOME"

ii) lors d'un succès d'application de la méthode :
"JUSTIFICATION : INTEGRATION D'UN MONOME".

Remarques :

1) autres possiblités d'écriture
i) Le plan-type pourrait aborder le problème avec une condition comme (SI (MONOMEP F (X))
(action))

- Cette écriture qui pourrait paraître plus intéressante aurait l'inconvénient de ne pas conserver le coefficient et l'exposant de F lorsque F est effectivement un monôme. Ces éléments sont évidemment indispensables pour fabriquer le résultat : MONOMEP

(action couteuse) ne doit pas être rappelée pour accéder à ces objets.

ii) Une autre solution consisterait à avoir autant de filtres qu'il y a de formes possibles de monômes en x (a,x, a*x, a*x**n). Cette solution permettait pour certaines formes l'usage de descripteurs destinés à faciliter la sélection des plans (cf IV 2.4.1). Ainsi le plan utilisant la dernière forme pourrait être décrit par OTOPEX (TIMES EXPT) pour dire qu'il n'a de chance d'être applicable que si * ou ** apparaissent dans l'expression. Cette solution cacherait le concept de monôme que le mathématicien manipule globalement dans ce genre de problèmes.

2) Il est clair que nous pouvions mettre dans la liste des résultats connus en calcul intégral une liste supplémentaire de règles pour faire ce travail :

Ces règles seraient :

```
(PRIM (A * x ** N,x) ==> A * x ** (N+1)/(N+1)
 avec condition  (ET(NONVUS  x  A)(NONVUS  x N)(N NESTPAS
 -1))

(PRIM (A * x   , x) ==> A * x ** 2   /2
 avec condition (NONVUS x A)

etc.
```

Cette solution est tout à fait réalisable dans CAMELIA. Mais elle amène un certain nombre de règles pour le calcul intégral, puis à peu près autant pour le calcul de limites, puis autant pour des développements limités,.... Le concept de monôme est suffisamment universel en mathématiques pour que nous l'ayons implémenté. Avec un seul plan, nous intégrons tous les monômes. Nous avons cherché à définir des outils puissants liés aux concepts manipulés par le mathématicien. Nous voyons bien, sur un tel exemple, comment conceptualiser différents niveaux de règles dans un système expert : nous avons là un plan (règle filtre) qui manipule le concept de MONOME, la procédure monome associée à l'identification des constituants du monome faisant elle-même appel à un jeu de règles décrivant des modèles différents de monomes.

IV.2.4 Expression du contrôle

La base de connaissances comporte un certain nombre d'informations destinées à faciliter le travail du moteur d'inférence en particulier pour ce qui concerne les choix d'items de connaissances significatifs pour aborder un problème.

Nous avons déjà vu (cf IV-2-1.ii) que les règles sont évoquées par paquets en utilisant un nom caractéristique du rôle de ce paquet. De même, les seuls filtres évoqués sont ceux dont le problème caractéristique coïncide avec le problème à traiter. Généralement, ces filtres sont encore beaucoup trop nombreux et un certain nombre de critères peuvent être utilisés pour écarter ceux pour lesquels on peut avoir une certitude d'échec.

L'application de ces critères se fait en deux temps. D'abord une sélection grossière sur critères rapides à mettre en oeuvre. Puis une sélection plus fine, basée sur l'usage de méta-règles, permettant le choix de ce que le système pense être la premiere chose à faire sur le problème donné.

IV.2.4.1 La sélection grossière

a) Des descripteurs de plans peuvent être codés par l'expert décrivant ses connaissances derrière le mot DESCRIPTEURS. Actuellement, de tels descripteurs sont gérés par le moteur pour permettre une sélection rapide :

- OTOPEX derrière lequel on donne la liste de tous les opérateurs qui doivent être effectivement tous présents dans l'expression du problème pour que le plan soit applicable.

- O1OPEX derrière lequel on donne une liste d'opérateurs dont un au moins doit être présent dans l'expression du problème pour que le plan soit applicable.

Le seul usage de ces deux descripteurs permet généralement d'écarter beaucoup de filtres non significatifs sans avoir fait la moindre opération de filtrage (toujours coûteuse).

- COUTMETHODE est un descripteur dans lequel l'expert code ce qu'il pense de la difficulté de mise en oeuvre de la méthode qu'il décrit.

Exemples :

i) nous savons que le SIN, la TG ou la COTG d'une fonction impaire est impaire.
Le plans (17) ci-dessous permet d'exprimer cette connaissance. Le descripteur O1OPEX (SIN TG COTG) exprime que l'un au moins de ces trois opérateurs doit être présent dans l'expression pour que le plan mérite attention lorsqu'on cherche à prouver qu'une fonction est impaire. Le coût méthode fixé à 6 par l'expert indique que la méthode est assez facile à mettre en oeuvre.

```
02590 %LE SIN LA TG OU LA COTG D'UNE FONCTION IMPAIRE EST IMPAIRE     $
02600 PLAN(17):='(IMPAIRE17 ( DESCRIPTEURS VARSUBST (F X)
02610                          O1OPEX (SIN TG COTG)
02620                          COUTMETHODE 6
02630                          MSGSUCCES "TRIGO COMPOSEE AVEC FCT IMPAIR
E"
02640                        )
02650           POUR (PRV (IMPAIRE F X)) ESSAYER
02660             (SI (OU ((!%OP F) EST SIN) ((!%OP F) EST TG)
02670                     (((!%OP F ) EST COTG)  )
02680                  (PRV (IMPAIRE (!%ARG1 F) X))))$
```

ii) Pour calculer le développement limité d'une fonction de la forme LOG u - LOG v, il peut être intéressant de calculer le développement à partir de log (u/v). Le plan (94) ci-dessous exprime cette méthode. Le descripteur OTOPEX (LOG DIFFERENCE) indique que la méthode n'a de sens que pour des fonctions où apparaissent au moins un LOG et une différence.

```
17006 PLAN(94):='(DEVLIM94 (DESCRIPTEURS VARSUBST (A B X N R)
17008                        OTOPEX (LOG DIFFERENCE)
17010                        COUTMETHODE 8
17012                        MSGSUCCES "DEV LOG A-LOG B = DEV LOG(A/B)"
17014                      )
17016        POUR (CALCUL (DEVLIM (DIFFERENCE (LOG A)(LOG B)) X 0 N)R)
17017       ESSAYER
17018        (SUC 4 (!%CREER R1)
17020           (CALCUL (DEVLIM (QUOTIENT A B) X 0 N ) (!%VAL R1))
17022           (CALCUL (DEVLIM (LOG (!%VVAL R1)) X 0 N ) R)
17024           (!%LIBERER R1) )) $
```

b) Certains descripteurs sont calculés par le système et incorporés aux filtres lors du premier calcul de leur valeur. Il s'agit le plus souvent de caractéristiques fixes qui peuvent être coûteuses à établir (à la main en particulier).

Par exemple, les méta-règles (cf IV-2-4-2) font intervenir des indicateurs comme :

- CUMULCOUTACTIMM : cumul du coût des actions immédiates d'un plan.

- RAROTO : somme des raretés relatives des opérateurs figurant dans OTOPEX.

- VOLCONDACT1 : volume de la condition de la première action du plan si c'est un SI.

- VOLPBTYPE : volume du problème type du filtre.

Ces deux derniers descripteurs permettent d'apprécier si la mise en oeuvre du filtre est très contraignante. Ceci est à mettre en relation avec l'heuristique qui consiste à dire qu'il faut commencer par mettre en jeu les règles les plus contraintes car si l'une s'applique, on est sûrement sur une bonne voie.

- VOLPLANTYPE : volume du plan type du filtre : appréciation possible par le système du poids du travail à faire quand on lance une méthode.

Exemples: Le plan 6 ci-dessous exprime que le cas où la valeur absolue d'une fonction paire ou impaire est paire. Le plan 14 exprime que pour prouver qu'une fonction est impaire, il peut suffire de consulter la table des fonction impaires.
L'écriture 1 est celle fournie en entrée (texte listé à l'éditeur). L'écriture 2 (listée en fin de session alors que les plans ont servi) fait apparaître les descripteurs calculés par le système.
Ainsi pour plan (6), le système a calculé que le problème type est de volume 4 (VOLPBTYPE 4) ce qui signifie que le problème s'exprime avec 4 symboles. Il a déterminé que la première action du plan est un SI, que la condition est de volume 9 (VOLCONDACT1 9) et que le volume du plan est 22 (VOLPLANTYPE 22). Ces descripteurs donnent un moyen d'apprécier le poids du plan associé au problème type, le degré de contraintes associés à l'usage de la méthode.

```
02200 % VOIR SI LE RESULTAT EST CONNU                                    $
02210 PLAN(14):='(IMPAIRE14 ( DESCRIPTEURS VARSUBST (F X)
02220                          COUTMETHODE 1
02230                          MSGSUCCES " PROPRIETE EVIDENTE "
02240                          MSGECHEC "JE NE SAIS PAS TOUT PAR COEUR"
02250                        )
02260            POUR (PRV (IMPAIRE F X)) ESSAYER
02270               (SI (RESULTATCONNU (IMPAIRE F X) IMPAIRE)
02280                      (!%ECRIRE "RESULTAT CONNU"))
02290                  )$
02300 %$
```

```
prettyprint plan(6);
    (PAIRE6
       (DESCRIPTEURS VOLPBTYPE 4 VOLCONDACT1 9 CUMULCOUTACTIMM 1
          VOLPLANTYPE 22 VARSUBST (F X) OIOPEX (COS ABS) COUTMETHODE 5
          MSGSUCCES "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST PAIRE")
       POUR
       (PRV (PAIRE F X))
       ESSAYER
       (SI
          (OU ((!%OP F) EST COS) ((!%OP F) EST ABS))
          (CHOIX
             2
             (PRV (PAIRE (!%ARG1 F) X))
             (PRV (IMPAIRE (!%ARG1 F) X)))) )

prettyprint plan(14)$
    24: (IMPAIRE14
          (DESCRIPTEURS VOLPBTYPE 4 VOLCONDACT1 5 CUMULCOUTACTIMM 0
             VOLPLANTYPE 8 VARSUBST (F X) COUTMETHODE 1 MSGSUCCES
             " PROPRIETE EVIDENTE "
             MSGECHEC
             "JE NE SAIS PAS TOUT PAR COEUR")
          POUR
          (PRV (IMPAIRE F X))
          ESSAYER
          (SI
             (RESULTATCONNU (IMPAIRE F X) IMPAIRE)
             (!%ECRIRE "RESULTAT CONNU")))
```

Remarques:

- Ce jeu de descripteurs est parfaitement extensible. Lors de l'écriture de nouvelles méta-règles, on peut être amené à incorporer des actions faisant appel à des procédures utilisant de nouveaux descripteurs. Ces procédures généralement très simples gèrent elles-mêmes la mise en place du descripteur dans les plans lors du premier appel.

- On voit que la base de connaissance n'est pas statique et peut être enrichie par le système au fur et à mesure que celui-ci résout des problèmes.

- La solution qui consiste à ne calculer de tels éléments que lors du premier appel de leur besoin nous paraît bien préférable à celle qui consisterait à un calcul systématique de tous ces éléments pour tous les filtres de la base avec une procédure de pré-traitement. Les plans sont loin de toujours tous servir !.

IV.2.4.2 La sélection fine Nécessité de la méta-connaissance

Les programmes d'intelligence artificielle (en particulier les systèmes experts) sont de plus en plus amenés à manipuler de grandes quantités de connaissances. Pratiquement ces programmes ne sont effectifs que si l'on dispose de moyens pour leur faire manipuler de la méta-connaissance : connaissances sur les connaissances manipulées. La méta-connaissance est fondamentale dès lors qu'on envisage un programme qui non seulement utilise une base de connaissances mais sache aussi raisonner sur elle, la structurer, l'enrichir, décider de ses usages. Bref, un tel système doit savoir utiliser son savoir, savoir ce qu'il sait, savoir ce qu'il ne sait pas....Notons que cette hiérarchie dans les connaissances se répète et que l'on peut envisager un niveau méta-méta-connaissances qui représente les connaissances pour manipuler de la méta-connaissance. A ces stades le système sait qu'il sait ce qu'il sait ...

La base de connaissances de CAMELIA comporte un certain nombre d'items de méta-connaissance fournie par un jeu de méta-règles (30 actuellement).Nous discuterons avec précision de la forme et de l'usage des méta-règles lors de la description du moteur d'inférence (cf IV -3.2). L'annexe A4 donne leur syntaxe exacte. L'annexe D4 donne la liste des méta-règles actuellement utilisées.

IV.2.5 Structure des chapitres de connaissances

IV.2.5.1 Organisation en chapitres

- La base de connaissances est organisée en chapitres qui regroupent les items de connaissances relevant d'un même domaine mathématique.

Actuellement, nous avons réalisé 4 chapitres dans les domaines suivants :

PARITE : preuves de parité
LIMITES : calcul de limites et de développement limités

INTEGR : calcul de primitives
COMBINES : évaluation de sommes comportant des combinaisons.

- Les connaissances d'un chapitre peuvent faire appel aux connaissances d'autres chapitres. L'organisation en chapitres permet de gagner de la place en memoire centrale lorsqu'on travaille dans un domaine limité mais les connaissances peuvent cohabiter lors de la résolution d'un problème sans détérioration des temps de réponse du système.

- La lecture d'un chapitre de connaissances se fait avec une commande LIRCHAP : par exemple LIRCHAP 'INTEGR donne à CAMELIA des compétences en calcul Intégral. LIRCHAP assure également le prétraitement de la base pour structurer l'accès aux connaissances que l'on vient d'entrer.

- Réciproquement, une commande IGNORE (par exemple : IGNORE 'INTEGR) permet de dire au système que l'on ne souhaite plus utiliser un chapitre de connaissances : cela permet de récupérer la place occupée par les plans et les règles correspondants.

IV.2.5.2 Gestion des chapitres

Actuellement, c'est l'utilisateur qui décide des chapitres qu'il y a lieu de charger ou d'ignorer. On pourrait imaginer que le système choisisse les chapitres dont il a besoin en fonction du problème qui est posé. Il suffirait de disposer d'une table (table des matières) qui fournisse le nom du chapitre à lire en fonction de l'objectif problème. Nous n'avons pas implémenté cette possibilité qui ne pose aucun problème.

IV.2.5.3 Structure d'un chapitre

La structure type peut être la suivante :

a) déclaration des opérateurs et connectives utilisés dans le domaine mathématique concerné.

i) les déclarations CONNEC1, CONNEC2,... CONNEC4, CONNECV suivies d'une liste de connectives permettent de déclarer des connectives respectivement unaires, binaires,... 4-aires, variaires.

exemple : CONNEC2 '(EVALUER RAMENER EFFECTUER);

ii) si ces opérateurs doivent être utilisés en mode algébrique, (notation usuelle) il faut les déclarer comme opérateurs algébriques :

exemple: ALGEBRAIC OPERATOR CNP, SIG;

b) déclaration des filtres : règles associant des plans types à des problèmes types. Ces éléments sont mis successivement comme composantes d'un tableau appelé PLAN. L' annexe A2 donne la syntaxe exacte utilisable et l' annexe D les exemples d'écriture- Cette partie est la plus longue et la plus délicate à mettre au point : elle représente effectivement le savoir faire de l'expert, ses démarches.

c) déclaration des résultats/règles connus dans le domaine. Ces règles (tableau REG) peuvent être groupées en paquets, chaque paquet étant repéré par un nom caractéristique (1er élément de la liste; avant les descripteurs). Ces paquets peuvent être évoqués par les actions immédiates !%RESULTATCONNU ou !%APPLIPAQUET1 figurant dans les plans.

d) éventuellement des procédures spécifiques dans le domaine et des déclarations d'actions immédiates appelant ces procédures.

exemple : la définition d'une procédure symbolique PROC et la déclaration AJOUTACTIMM ('!%TRUC, 'PROC) permet de lancer l'exécution de PROC par utilisation de l'action !%TRUC dans un plan.
Note : si on désire que le nom de l'action soit le nom de la procédure précédé de !%, il suffit de passer NIL comme 1er argument à AJOUTACTIMM.

* Si la procédure est de type algébrique, elle doit être évoquée dans les plans en utilisant la fonction de conversion ALGEVAL.
exemple :

. Définition de la fonction donnant les polynômes de Legendre en mode algébrique :

```
ALGEBRAIC PROCEDURE LEGENDRE (N, x );
SUB(Y=0, DF((Y**2 - 2* x *Y + 1)**(-1/2),Y,N)) /
(FOR I : = 1 : N  PRODUCT I) ;
```

ce qui représente directement la définition :

$$Pn(x) = \frac{1}{n!}\frac{d^n}{dy^n}\left[\frac{1}{(y^2 - 2xy + 1)^{1/2}}\right]_{y=o}$$

. AJOUTACTIMM (NIL, 'LEGENDRE) déclare alors que l'on dispose dans les plans d'un polynôme de Legendre avec l'action !%LEGENDRE.

. Un travail avec le 5ème polynôme de Legendre en R dans un plan peut alors être décrit avec
.....(!%ALGEVAL (!%LEGENDRE 5 R)).....

e) déclaration des méta-règles caractéristiques du domaine. Ces méta-règles (tableau M-REG) peuvent fournir de l'information sur la façon de manipuler les connaissances du domaine (cas où des filtres peuvent être écartés à coup sûr, type de vérifications "sémantiques" qui peuvent être faits sur des problèmes avant toute tentative "sérieuse" de résolution, détection de "problèmes-absurdes,... On trouvera en IV.3.2 la description et le mode d'usage des méta-connaissances dans CAMELIA.

Notons que ces méta-règles sont évoquées par la procédure REAR-

RANGE qu'il y a donc lieu de modifier pour que les nouvelles méta-règles puissent être prises en compte. Cette modification est nécessaire car nous n'avons pas mis en place de niveau Méta-méta règles. Le niveau Méta-méta est programmé très simplement dans REARRANGE. (La partie à corriger fait à peine une vingtaine de lignes d'instructions et il suffit d'injecter la liste portant les numéros des nouvelles méta-règles!).

f) le mot END; termine le chapitre pour redonner le contrôle à l'utilisateur en fin de lecture.

IV.2.6. Pré-traitement de la base de connaissances

IV.2.6.1 Idées générales

La difficulté essentielle lors de la mise au point d'un système expert se situe au niveau de la définition, de la constitution, de la mise au point de la base de connaissances. Il est couramment admis maintenant (LAURIERE 82) que les connaissances doivent être exprimée au maximum de façon déclarative, dans des items indépendants, non classés à l'entrée, à l'aide d'un langage directement accessible à l'expert.

Cette démarche qui a pour but de faciliter le travail de l'utilisateur lors de l'introduction de nouvelles connaissances et la vérification de l'information dans la base ne facilite pas le travail du moteur d'inférence qui utilisera cette base.

IV.2.6.2 Type de pré-traitement réalisé

Un pré-traitement de la base peut être fait afin de faciliter le travail du moteur d'inférence tout en conservant les facilités décrites précédemment pour aider l'usager.

Ce pré-traitement peut comporter plusieurs aspects :

- amener les connaissances à une forme plus appropriée au niveau de chaque item. Dans un cas comme celui que nous traitons, il serait intéressant d'envisager de fournir l'énoncé déclaratif d'un théorème et de faire générer automatiquement les plans lors d'un pré-traitement.

- structurer la base pour faciliter sa relecture (regroupement d'items traitant d'un même domaine en chapitres,-....).

- préparer la méta-connaissance : faire en sorte, par analyse de la base, que le système sache ce qu'il sait par exemple.

- inférer à partir de connaissances fournies, de nouvelles connaissances non explicitées.

Exemple

Le fait que deux concepts ou connaissances s'excluent est souvent une méta-connaissance importante. Il serait possible à partir de la donnée d'une méta-connaissance du type (SEXCLUENT C1 C2) et d'une famille de règles disant : " la propriété P1 est

vraie au regard de C1" de générer une famille de règles disant "la propriété P2 est fausse au regard de C2".

Ainsi la donnée de
(SEXCLUENT (PAIRE fx) (IMPAIRE fx))
et
(PAIRE (COS X) X) vrai
(PAIRE (ABS X) X) vrai

permettrait d'engendrer lors du pré-traitement les règles

(IMPAIRE (COS X) X) faux
(IMPAIRE (ABS X) X) faux

.....

- établir des contradictions éventuelles dans la base.

Ce type de traitement qui ne se fait qu'à l'occasion d'une modification de la base semble intéressant. Il y a là une voie à développer.
Avec cette démarche, dans CAMELIA, nous avons écrit une procédure (PLANOBJ) qui établit la liste caractéristique des objectifs des plans ce qui revient à dire que l'on analyse les objectifs pour lesquels on dispose de connaissances dans la base. Etant donné un problème, le moteur d'inférence par consultation de cette liste, vérifiera, avant toute activité, qu'il dispose de connaissances adaptées pour essayer d'atteindre l'objectif du problème. La détection du manque de connaissances adaptées pouvant forcer un retour arrière. Ces éléments servent lors de la sélection des plans. De même, c'est lors du prétraitement que les règles sont classées par objectif ou contexte d'application. Nous établissons pour cela une liste qui asocie pour chaque caractéristique de règle, la liste des règles concernées.

IV.2.6.3 Conclusion

Ce type de pré-traitement s'avère très utile et efficace. Il a cependant ses limites : il peut en effet nécessiter des ressources importantes et il n'est pas forcément judicieux de pré-traiter une grosse base de connaissances alors que peut-être tous les items qui y figurent ne seront pas utilisés. L'idée qui s'impose est donc de tenter de ne faire a priori que les pré-traitements rapides qui s'imposent : ce sont en particulier ceux qui établissent des liens entre les éléments. Par contre, il faut différer les pré-traitements qui peuvent attendre : par exemple, le calcul d'éléments caractéristiques d'un nodule de connaissances peut n'être fait que lors de la première évocation de ce nodule. La base de connaissances devient alors évolutive et s'enrichit au fur et à mesure qu'on l'utilise. Des mécanismes de ce type sont actuellement implémentés dans CAMELIA. Cette idée de pré-traitement peut être largement étendue et nous discuterons au chapitre V de ce que nous pourrions appeler "compilation de connaissances". Ces idées et extensions, en terme de compilation de connaissances, ne sont pas actuellement prises en compte dans CAMELIA.

IV.3 Le moteur d'inférence

IV.3.1 Rôle

Le moteur d'inférence (procédure SOLUT dans CAMELIA) a le rôle essentiel dans un tel ,système : c'est lui le chef d'orchestre qui coordonne le séquencement des tâches, évoque les items de connaissances significatifs pour aborder le problème qu'on lui (ou qu'il se) confie. Il met en oeuvre les stratégies de choix, gère les structures de travail, les retours arrière toujours possible, déclenche les actions prévues dans les plans.

IV.3.2 Fonctionnement

IV.3.2.1 Le fonctionnement global est le suivant :

* SOLUT accepte un problème PB en entrée. Il renvoie NIL pour un échec et T pour un succès.

* Si PB est une succession de n sous problèmes à traiter (forme (SUC n e1 e2...en)), il envisage ceux-ci récursivement, successivement, dans l'ordre des indices. Si un sous-problème ek échoue, SOLUT renvoie NIL sur ce sous- problème et considère qu'il a échoué sur le problème PB initial. Il remonte alors au dernier choix fait ou reconnait son échec à l'utilisateur si tous les choix ont été envisagés.

* Si PB est un choix entre n étapes possibles (forme (CHOIX n e1 e2....en)) il envisage ces étapes récursivement jusqu'au succès de l'une d'entre elles. Si toutes échouent, SOLUT retourne NIL pour avouer son échec sur ce problème et assure un retour arrière si possible. Actuellement, les étapes sont tentées dans l'ordre des indices. On pourrait envisager de faire traiter en premier l'étape qui a le plus de chances d'aboutir à priori. Cette évaluation peut être coûteuse si l'on souhaite faire intervenir des critères assez fins.

* Si PB est un problème conditionné de la forme (Si (cond) (PBC)), la condition est évaluée. Notons que s'il s'agit d'une condition problème (cond est une propriété dont il faut prouver qu'elle est vraie) (cf IV.2.3.3) le moteur peut être rappelé récursivement par l'évaluateur de condition.
Si la condition est vraie, SOLUT tente de résoudre le PBC qui suit la condition. Le retour de SOLUT PB est alors le retour de SOLUT PBC. Si la condition est fausse, SOLUT renvoie NIL.

* Si PB est de la forme (POURTOUT x L PBx), SOLUT tente de résoudre récursivement tous les PBx lorsque x parcourt L. Si un seul sous-problème échoue, SOLUT renvoie NIL. Si tous ont amené un succès, SOLUT renvoie T.

* Si PB est de la forme (POURUN x L PBx), SOLUT tente de résoudre les PBx, x parcourant L. Dès qu'il renvoie un succès sur un sous-problème, il renvoie T pour PB.

* Pour les autres problèmes, SOLUT lance l'exécution des actions immédiates éventuelles et il cherche dans la base de

connaissances un plan adapté pour résoudre le problème. Le problème de la sélection des plans sera vu en détail en IV.3.2.4. Si SOLUT ne trouve pas de plan convenable, il avoue son échec. Sinon, il exécute le plan retenu. Si ce plan échoue, il revient faire le choix d'un autre plan.

IV.3.2.2 Forme des problèmes

Les problèmes sont fournis sous forme d'une liste et envoyés comme argument à la procédure SOLUT.

exemples : PB : = '(PRV (PAIRE (TIMES X (SIN X)) X)); SOLUT PB;

SOLUT '(CALCUL (PRIM (QUOTIENT 1 X) X) RES);

Il est également possible de fournir les problèmes en mode algébrique avec les notations usuelles :

exemples : SOLUT PRV PAIRE (X * SIN(X), X);

SOLUT CALCUL (PRIM (1/X, X), RES);

Les formes de problème actuellement admises sont les suivantes :

(PRV (PAIRE f x)) pour prouver que f(x) est paire en x

(PRV (IMPAIRE f x))

(CALCUL (LIM F x PT) R) pour R <== Lim (f(x), x--> PT)

(CALCUL (DEVLIM F x PT N) R) pour mettre dans R le développement limité de F(x) au tour de PT à l'ordre N

(CALCUL (PRIM F x) R) pour R <== $\int f(x)\,dx$

(EVALUER (SIG f K K1 K2) R) pour R <== $\sum_{K=K1}^{K2} f$

(RAMENER (APFORBIN (SIG f K K1 K2)) R) pour mettre dans R ce que l'on obtient en se ramenant à l'application de la formule du binôme de NEWTON sur un SIGMA. Cette formule est donnée sous forme de règles du type

SIG C(n,p) x^{p} *y^{n-p} p 0 n ====> (x+y)^{n}

IV.3.2.3 Pilotage automatique/pilotage utilisateur

a) Une variable globale, NAVIGAVUE, est définie dans le système et permet à l'utilisateur de choisir son mode de fonctionnement.
Les choix se font alors selon le schéma suivant : (fig.3)

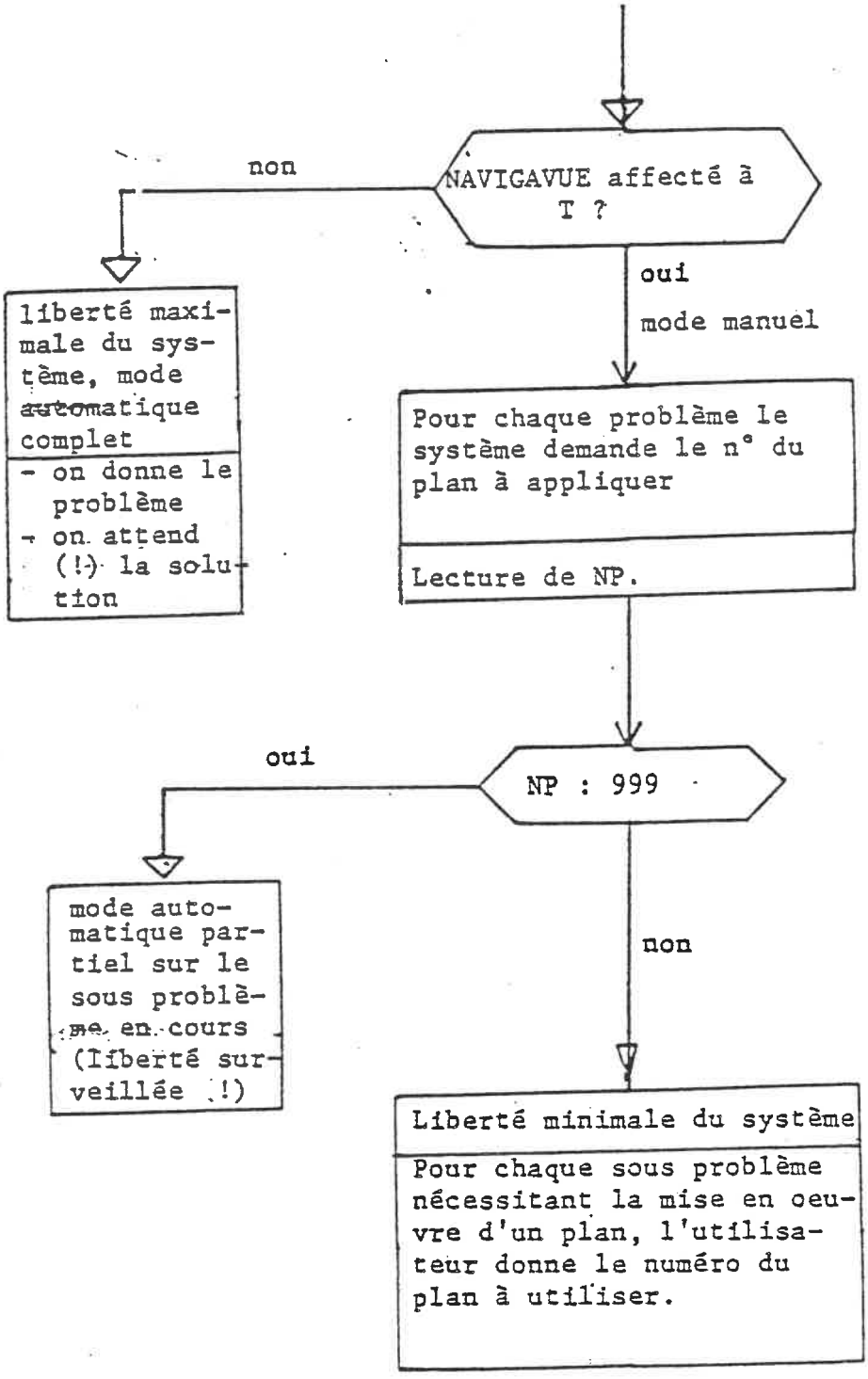


Figure 3. CONTROLE DU DEROULEMENT PAR L'UTILISATEUR

b) Rôle des plans universels PLANT, PLANNIL

Les deux plans de savoir faire universel peuvent être utiles en cas de contrôle manuel du déroulement d'une session. Il est en effet possible, sans détruire les structures de données correspondant au graphe de résolution du problème de bloquer le développement d'une voie qui nous semble conduire à un échec : Il suffit d'envoyer comme numéro de plan la valeur 0 (qui est l'indice de PLANNIL) ce qui revient pour le système à être assuré de l'échec sur le sous-problème en cours et à forcer un retour arrière. De même, si l'utilisateur ne souhaite pas que le système s'engage dans une preuve ou un calcul (soit évident, soit hors de portée pour le système, soit pour voir ce qui se passerait ensuite si l'étape était résolue) il suffit d'envoyer comme numéro de plan la valeur 1 (indice de PLANT) : Le sous problème est alors considéré comme résolu et l'on passe à la suite.

c) Messages délivrés par le système

Le système délivre en console un certain nombre de renseignements qui permettent à l'utilisateur de suivre la démarche adoptée, les étapes envisagées, l'énoncé du problème en cours. Ainsi si un plan contient un SUC ou un CHOIX, on peut faire apparaître (après que les instanciations sont faites) les listes d'étapes associées aux messages :

"IL ME FAUT REUSSIR TOUTES LES ETAPES SUIVANTES"

"J'AI LE CHOIX ENTRE LES ETAPES SUIVANTES".

De même les conditions évaluées peuvent être affichées. Tout ceci permet de vérifier comment se font les choses, comment sont utilisées les connaissances de la base de connaissances. Cela permet de voir quelles connaissances manquent dans la base pour résoudre un problème donné. Indépendamment de l'intérêt pour l'utilisateur de voir le système justifier ce qu'il fait, il y a là un outil précieux pour améliorer la qualité de la base de connaissances.

d) Pilotage du niveau de messages

Les messages sont liés à un niveau. Le volume de messages affichés est variable et conditionné par la valeur de la variable globale MSGLEVEL qui prend ses valeurs dans l'intervalle (09). La règle retenue est la suivante : plus MSGLEVEL est grand, plus on a de messages en console. Seuls, les messages qui ont un numéro inférieur à MSGLEVEL sont envoyés à la console. L'utilisateur peut ainsi, en affectant MSGLEVEL, rendre le système plus ou moins bavard sur ce qu'il fait, les problèmes qu'il se pose etc. La valeur moyenne 5 correspond à une valeur courante raisonnable, c'est la valeur par défaut.
Notons que même lorsque les messages ne sont pas envoyés à la console, ils sont stockés avec leur numéro de niveau dans un tableau interne (limité en taille mais un stockage sur fichier aurait été plus coûteux).
Il est ainsi possible en "temps différé" d'avoir l'ensemble des

messages qui ont été délivrés (et dans l'ordre!) ce qui donne une possibilité de trace complète de la solution. L'appel procédure JUSTIFIER (à un argument x) permet alors de sortir dans l'ordre chronologique les messages de niveau inférieur à n trace de ce que l'on aurait eu en console si MSGLEVEL avait été fixé à n lors de la résolution du problème.
Il y a là un outil de "solution plus ou moins complètement rédigée" intéressant qui permet de faire résoudre un problème au système et de faire justifier sa solution plus ou moins finement ensuite.
La procédure SOLUTION est une version améliorée du moteur SOLUT qui propose à l'utilisateur de justifier la solution à un niveau que celui-ci désire.
Nous pensons avoir par ce type de solution répondu de façon intéressante aux contraintes de qualité fixées dans le dialogue avec le système (cf III.2.3).

IV.3.2.4 Le choix des plans

En mode automatique complet ou partiel (cf fig. 3), le choix du plan applicable à un problème se fait en deux temps.

1) une sélection "grossière" dont l'objectif est d'extraire de la base de connaissance une liste de candidats "raisonnables".

2) une sélection "fine" faisant appel à des méta-connaissances codées sous forme de méta-règles. Plus coûteuse en ressources, cette sélection permet d'établir un ordre de préférence parmi les candidats retenus lors de la sélection grossière.

IV.3.2.4.1 La sélection grossière

Elle se fait par niveaux successifs d'analyse du problème et des connaissances disponibles. L'objectif est de fabriquer une liste (LPP) des plans possibles, "raisonnables" pour approcher la résolution du problème. Cette liste sera alors l'objet de l'analyse fine ultérieure.

Différents niveaux de sélection.

Niveau 1 de sélection

Un prétraitement (cf IV.2.6) de la base permet de classer les filtres par les objectifs des problèmes qu'ils abordent (établissement de la liste PLANOBJ : couplage d'un objectif donné à la liste des plans susceptibles d'atteindre cet objectif). Le premier travail consiste à isoler dans LPP1, la liste des plans susceptibles d'atteindre l'objectif du problème en cours.

Niveau 2 de sélection

Une analyse est faite et la liste des Opérateurs Effectivement Présents dans l'Expression courante (OEPEX) est établie. Cette liste est confrontée à la liste d'opérateurs (OTOPEX : Opérateurs Tous Obligatoirement Présents dans l'Expression)

qui est éventuellement associée au plan dans la liste des descripteurs du filtre. Rappelons (cf IV.2.1., IV.2.4.1) que la liste OTOPEX représente la liste des opérateurs dont la présence est obligatoire pour que le plan s'applique. (Notons qu'elle pourrait être établie automatiquement par analyse du plan, dans une phase de prétraitement. Pour l'instant cette liste est établie à "la main").
Au niveau 2 de la sélection, nous ne retenons dans LPP2 que les plans de LPP1 pour lesquels OTOPEX est inclus dans OEPEX ou OTOPEX : nil.

Niveau 3 de sélection

Nous travaillons ensuite en comparant OEPEX à une liste O1OPEX, fournie comme attribut de filtre et qui est une liste d'Opérateurs dont un (1) au moins doit être obligatoirement présent dans l'expression pour que le plan s'applique.
Au niveau 3 de la sélection nous ne retenons dans LPP3 que les plans de LPP2 pour lesquels la condition

$$O1OPEX = nil \text{ ou } OEPEX \cap O1OPEX \neq \emptyset$$

est vérifiée. Actuellement, nous retenons LPP3 comme liste LPP des plans possibles. Il est clair que l'analyse peut être affinée avec des niveaux supplémentaires et l'intervention d'attributs nouveaux dans les plans.

IV.3.2.4.2 La sélection fine :usage de la méta- connaissance

Nous avons rappelé en IV-2-4-2 la notion de méta-connaissance et la nécessité d'utiliser ce type de connaissances dans les systèmes experts. Cette partie a pour objectif de présenter comment la méta-connaissance est utilisée dans CAMELIA.

A- POSITION DU PROBLEME

1) contexte du travail. La base de connaissances de CAMELIA est constituée d'un ensemble de règles de la forme : POUR <pbtype> ESSAYER <plantype> dans lesquelles <pbtype> sont des modèles de problèmes et <plantype> sont des modèles de plans, démarches usuelles pour résoudre le problème décrit par <pbtype>. Etant donné un problème à résoudre, CAMELIA détecte par filtrage une liste de plans applicables au problème. Généralement plusieurs plans sont applicables, le problème du choix se pose alors. Ces plans peuvent échouer, ce qui pose, lorsque le premier choix n'a pas été judicieux, le problème du retour arrière et du changement de méthode pour un problème donné.

2) La méta-connaissance est utilisée dans CAMELIA pour définir la stratégie d'application des plans et faciliter les choix. Une liste de méta-règles (cf annexe D4) est fournie. Le système les utilise pour effectuer des évaluations permettant de trier les plans par ordre d'intérêt décroissant. Ceci se produit à chaque noeud (problème) de l'espace de recherche qui se développe progressivement.

3) Pour un problème donné, une liste LPP de plans possibles est disponible pour aborder le problème. Il s'agit d'établir un jugement de ces différents plans, éventuellement d'en éliminer ou d'en imposer certains. Pour cela on peut NOTER (au sens de mettre une NOTE) chaque plan en estimant un COUT et un ESPOIR pour sa mise en oeuvre pour résoudre le problème. Les différents plans peuvent alors être classés par "ordre de mérite" dans LPP. C'est typiquement un problème de type fonction d'évaluation. L'avantage va être ici que les heuristiques sont formalisées dans des règles bien individualisées.

B -METHODE RETENUE

1) notations : Nous utilisons quatre variables globales : LPP (liste des plans possibles), NEWLPP (nouvelle liste de plans possibles après application des méta-règles), COUT (coût estimé pour appliquer le plan courant au problème actuel), ESPOIR (espoir associé à l'application du plan courant pour résoudre le problème actuel). En principe PB sera utilisé pour désigner le problème actuel (noeud à développer dans l'arborescence), PL sera le plan courant en cours d'estimation, LISMREG sera la liste des méta-règles qu'il convient d'appliquer pour juger de l'intérêt d'utiliser le plan courant PL en vue de résoudre le problème actuel PB alors que l'on pourrait aussi utiliser aussi les plans de LPP.

2)Démarche - Procédures utilisées

Les procédures suivantes ont été définies :

- REARRANGE (LPP, PB) qui, pour le problème actuel PB, réordonne LPP (modifié) par ordre décroissant des notes et ne conserve que les 12 meilleurs plans candidats.

- JUGEMENT (PL, PB, LPP, LISMREG) qui établit un jugement du plan PL, face au problème PB, dans le contexte des autres plans possibles LPP en utilisant les seules méta-règles dont le numéro figure dans LISMREG.
Par "Etablir un jugement", il faut entendre affecter une valeur à COUT et ESPOIR, éventuellement modifier LPP.
Les modifications de LPP peuvent concerner l'ajout, la suppression de plans de LPP. Cela permet d'obtenir des comportements du type : "pour tel problème, inutile de considérer le plan P1"... ou bien "pour tel problème, considérer absolument le plan P2",-... ou bien "Pour tel problème, aucun plan n'est convenable, le problème est absurde"...

Remarque : On peut à ce stade ne laisser dans la liste des plans possibles que le savoir-faire universel (PLANT) ou l'ignorance universelle (PLANNIL), qui jouent un rôle particulier.

- NOTEFINALE (COUT, ESPOIR) est la fonction qui attribue la note finale à PL à partir des valeurs de COUT et de ESPOIR;
. (Actuellement NOTE = 1000*ESPOIR/COUT).

C- FORME DES META REGLES

1 -Syntaxe

```
<M-Reg> : =' (SI <cond1> <estimation 1>)
           ' (SI <cond2> <estimation 2>)
           ' (SI <cond3> <estimation 3>)

 <estimation 1> :=(!%ATTRIB NEWLPP < f(LPP, PB>)
 <estimation 2> :=(!%ATTRIB  COUT  <fct cout 2>)
                  (!%ATTRIB ESPOIR <fct espoir 2>)
 <estimation 3> :=(!%ATTRIB COUT   <fct cout 3>)
                  (!%ATTRIB ESPOIR <fct espoir3>)
```

Dans lesquelles :

<cond1> est une condition quelconque,
<cond2>, <fct cout2>, <fct espoir2> ne dépendent que du plan courant PL à juger;
<cond3>, <fct cout3>,(fct espoir3> dépendent à la fois du plan courant PL et du problème à résoudre PB.

2 -Différents types de méta-règles

Les méta-règles sont classées en trois types suivant les éléments qu'elles font intervenir :

type 1 : fonction NEWLPP :=f (LPP, PBB) qui enlèvent, ajoutent, imposent des plans (quel que soit le problème PB).

type 2 : ⌈COUT = f(PL) qui représentent des jugements à priori

ESPOIR
sur les plans (quel que soit le problème à résoudre).

type 3 : COUT = f(PL,PB) qui analysent des caracéristiques
ESPOIR
présentes ou absentes dans le plan PL et le problème PB pour juger de l'intérêt d'utiliser ce plan.

Exemples :

. La méta-règle MREG(9) s'applique pour calculer des intégrales : elle indique que si le volume du problème (surtout lié au nombre de symboles de l'intégrande) est grand (disons supérieur à 16 symboles) alors il y a peu de chances pour que nous soyons dans le cas de l'intégration d'une constante ou dans le cas d'une primitive connue. Cette méta-règle de type 1 (car ne dépendant que du problème et de LPP) s'écrit :

```
%SI LE PROBLEME EST TROP GROS,IL Y A PEU DE CHANCES QUE CE SOIT;
%UN RESULTAT CONNU OU UNE CONSTANTE                              ;
%CALCUL INTEGRALES                                               ;
MREG(9):='(SI (GREATERP (!%VOL PB) 16)
              (!%ATTRIB NEWLPP (!%ENLEVELIST LPP (20 27 28))
              ));
```

Elle indique qu'il y a lieu pour un tel problème d'enlever de la liste LPP des plans possibles les plans 20, 27, 28 qui traitent des cas peu probables.

On voit bien la nature heuristique de cette règle puisqu'on peut envisager des intégrations de constantes formelles dont l'écriture est volumineuse parce que contenant beaucoup de paramètres différents de la variable d'intégration.
On voit bien l'aspect méta de cette règle qui indique qu'il y a dans ce cas des connaisances qu'il n'y a pas lieu d'envisager.

. La méta-règle 22 de type 2 (car ne dépendant que du plan PL) permet d'augmenter l'espoir que l'on peut mettre dans un plan fortement contraint. Le critère utilisé dans cette estimation est le volume en nombre de symboles de la condition de la première étape du plan si celui-ci est de la forme :

(SI (condition) (action))

Cette méta-règle s'écrit :

```
MREG(22):='(SI (GREATERP (!%!% VOLCONDACT1 PL) 5)
               (!%ATTRIB ESPOIR (PLUS (!%OBJ ESPOIR) 5 )));
```

Cette méta-règle relève de l'heuristique générale bien connue indiquant qu'il est généralement judicieux d'appliquer d'abord les connaissances les plus contraignantes.

. La méta-règle 30 est de type 3. Elle permet d'augmenter l'espoir que l'on peut attribuer aux plans qui imposent (par OTOPEX : liste des opérateurs tous obligatoirement présents dans l'expression pour que le plan s'applique) la présence de l'opérateur principal de l'expression. Autrement dit si on travaille sur une expression qui est un quotient, on augmente l'espoir attribué aux plans imposant la présence de quotients.

Cette méta-règle s'écrit :

```
%META-REGLES DE TYPE 3 : (COUT,ESPOIR)=F(PB,PL);
%ESPOIR LIE A LA PRESENCE DE L'OPERATEUR PRINCIPAL DANS OTOPEX;
MREG(30):='(SI (VUS (!%OPRINCEX PB) (!%!%OTOPEX PL))
               (!%ATTRIB ESPOIR (PLUS (!%OBJ ESPOIR) 50)) );
```

3-Interprétation des méta-règles instanciées comme des problèmes

i) Les calculs de coût et d'espoir permettant d'établir la note jugeant l'intérêt d'un plan pour résoudre un problème sont des calculs arithmétiques que l'on peut confier au système lui-même. Le moteur d'inférence SOLUT de CAMELIA qui peut raisonner ou calculer suivant les connaissances qu'on lui fournit apparaît ainsi comme profondément récursif : alors qu'il est en train de conduire un calcul formel ou de raisonner une preuve, on l'interpelle pour conduire le calcul arithmétique qui lui permet d'estimer les différents plans qui s'offrent à lui pour poursuivre.

ii) Cette possibilité vient du fait que les méta-règles prennent une fois les instanciations faites, une forme problème directement traitable par SOLUT. Les problèmes ainsi amenés à SOLUT sont généralement simples actuellement : la partie "estimation" des méta-règles se ramène à l'exécution d'actions immédiates (!%ATTRIB) et d'appel de fonctions d'accès aux caractéristiques de la situation courante.

D. PROBLEME DES META-ACTIONS

. Les méta-règles peuvent faire intervenir des actions agissant sur le plan à évaluer ou le problème en cours. En particulier certaines actions peuvent avoir à agir sur les actions immédiates des plans (par exemple : compter le nombre d'actions du plan, évaluer leur poids,...). Ces actions, pour lesquelles nous parlons de META-ACTIONS (notées M-actions) doivent s'exécuter à un niveau supérieur. Pour éviter les interactions entre les différents niveaux, elles ont reçu une notation particulière au plan syntaxique : leur nom est précédé de !%!% (alors que les actions immédiates n'ont leur nom précédé que de !%).
. Cela nous a amené à définir un outillage de manipulation des méta-actions. Le traitement est formellement homogène avec celui du traitement des actions : ainsi les deux procédures EXECACTIMM (exécution d'une action immédiate) et MEXECACTIMM (exécution d'une M-action) sont identiques au type d'action mis en oeuvre près. Ce qui les diffère c'est l'usage qui en est fait : le moment d'appel en particulier.

Exemple : La méta-règle 22 vue ci-dessus fait intervenir la méta-action (!%!% VOLCONDACT1 PL) qui permet d'évaluer le volume de la condition éventuelle de l'action 1 du plan PL. La condition peut évidemment elle-même contenir des actions. Par exemple l'évaluation du plan (30) pour calculer la primitive d'un produit n-aire F commence par :

```
(SI (ET (!%OP F) EST TIMES) (PGE (!%ORDROP F) 3 ))
```

(actions))

L'évaluation du volume de la condition porte sur des actions et doit donc être traité au niveau méta. (Les actions !%OP et !%ORDROP ne sont bien sûr pas encore exécutables, aucune instanciation n'est faite (pour F en particulier) : nous n'en sommes qu'à juger de l'intérêt du plan !).

Ainsi c'est MEXECACTIMM qui exécute la M-action !%!%VOLCONDACT 1. Les actions !%OP et !%ORDROP seront exécutées ultérieurement par EXECACTIMM, après les instanciations faites et seulement si ce plan est choisi.

E. ASPECTS META-META

1. L'ordre d'application des méta-règles n'est pas indifférent, en particulier :

i) Les méta-règles de type 2 qui ne dépendent que du plan peuvent être appliquées très tôt, en principe dès le prétraitement de la base de connaissances. Les valeurs de COUT et d'ESPOIR à priori ainsi calculées pourraient alors accompagner le plan dans la base de connaissances. Par hypothèse, la base est importante et l'on risque au prétraitement de calculer ainsi beaucoup d'informations qui ne seront peut-être jamais utilisées : tous les plans sont loin d'etre utilisés à chaque session.

La technique actuellement retenue consiste à appliquer les méta-règles de type 2 systématiquement mais les procédures associées à la détermination des éléments fixes, caractéristiques d'un plan ne font le calcul que lors du premier appel de ce plan : la philosophie utilisée pourrait s'énoncer : "si la valeur cherchée à déjà été calculée (elle figure dans les descripteurs du plan) alors la prendre ; sinon la calculer, la stocker dans les descripteurs du plan et la retourner pour usage". On voit donc qu'il y a une certaine dynamique à la base de connaissances qui s'enrichit au fur et à mesure que des problèmes sont résolus.

Exemple : La méta-règle fait appel à COUTMETHODE ; le coût méthode est la somme de 2 termes : le coût méthode donné par l'expert et le cumul du coût des actions immédiates du plan. Celui-ci est calculé à partir d'une table COUTACTIMM "tarif des actions immédiates" (cf Annexe C) fournie par le concepteur du système : le tarif d'une action immédiate est une estimation directement liée à la complexité de la procédure à lancer pour exécuter cette action. Ce cumul du coût des actions pour un plan donné n'est évidemment fait qu'une fois et stocké (sous le descripteur CUMULCOUTACTIMM) la première fois qu'il est nécessaire d'en disposer. La figure 4 dans l'exemple 1 (cf.IV.3.3.) fait apparaître cet aspect.

ii) Les méta-règles de type 1 : NEWLPP = f(PB, LPP) doivent s'appliquer une seule fois pour un problème donné. Il reste alors à appliquer les méta-règles de type 3 uniquement pour les plans figurant dans NEWLPP (LPP modifié : les M-règles de type 1 ont pu imposer ou supprimer des plans dans LPP).

2. On voit qu'il est donc possible de définir une stratégie claire d'application des 3 paquets (un par type) de méta-règles.

Pour les méta-règles de type 1, il est même possible de calculer, en fonction du problème,la liste LISMREG des méta-règles qu'il y a lieu d'appliquer.

Actuellement, c'est la procédure REARRANGE qui gère l'ordre d'application des méta-règles. On pourrait imaginer un niveau META-META règles qui fasse l'analyse des types des méta-règles (fait "à la main" actuellement) et qui gère l'ordre d'application de ces méta-règles. Compte-tenu de la simplicité de l'écriture de la procédure REARRANGE qui fait ce travail, nous

n'avons pas jugé utile de mettre cette possibilité dans CAMELIA.

F. UN USAGE PARTICULIER DES META-REGLES : ELIMINATION DES PROBLEMES ABSURDES

1) Un usage particulier des méta-règles a pu être fait pour détecter qu'un problème est "absurde" par rapport aux connaissances disponibles pour essayer de le résoudre. Cela permet d'écarter, dès le traitement au niveau méta, des problèmes que l'on sait ne pas pouvoir résoudre. Ceci évite de lancer le moteur d'inférence sur de multiples tentatives d'applications de plans qui ne peuvent qu'échouer.

i) Soient les problèmes :

pb1 : (PRV (PAIRE x x)) (prouver que x->x est paire)
pb2 : (PRV (PAIRE (EXPT x) x)) (prouver que x->e **x est paire).

Ces problèmes (que le système peut être amené à se poser) sont absurdes. Deux critères peuvent être utilisés pour s'en apercevoir :

. pour pb1 : on dispose dans la base de connaissances d'un résultat connu : (IMPAIRE x x) et l'on sait qu'une fonction non nulle ne peut être à la fois PAIRE et IMPAIRE.

. pour pb1 et pb2 : on peut utiliser une vérification sémantique de la parité : le fait que f(1) soit différent de f(-1) est une raison suffisante pour que les fonctions ne soient pas paires.

ii) Nous avons défini (cf.annexe D4 MREG (5) et MREG(7)) deux méta-règles pour exprimer ces idées lorsque se présente un problème de type (PRV (PAIRE f x)). Ces méta-règles s'écrivent :

```
%GESTION DE PROBLEME ABSURDE:NE LAISSE QUE LE PLAN NIL A TENTER;
% PRV PAIRE                                                    ;
 MREG(5):='(SI (!%RESULTATCONNU (!%REMPL PAIRE IMPAIRE
                                  (!%TQ PB)) IMPAIRE)
               (SUC 2 (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")
                      (!%ATTRIB NEWLPP (0)) ) ) ;

%VERIFICATION SEMANTIQUE POUR LA PARITE;
% PRV PAIRE                                                    ;
 MREG(7):='(SI (PGQ (!%SIMPLIF (ABS (VALNUM (DIFFERENCE
                  (!%REMPL (!%TQQTQ PB) 1 (!%TQTQ PB) )
                  (!%REMPL (!%TQQTQ    PB)    -1    (!%TQTQ   PB))))))
                  0.0000001)
               (SUC 2 (!%ECRIRE "F(X) NON PAIRE:F(1) F(-1)")
                      (!%ATTRIB NEWLPP (0) ) ));
```

. La première (MREG(5)) indique que pour un problème (PRV (PAIRE f x)) si l'on connaît le résultat (IMPAIRE f x), alors le problème est absurde : il suffit de le dire et de forcer un échec en ne laissant dans NEWLPP que le plan de numéro 0 (PLANNIL). C'est une façon de piloter un retour arrière dans l'arborescence qui nous a amené à ce problème absurde.

. La deuxième (MREG (7)) indique que pour un problème (PRV (PAIRE f x)), si le résultat de la simplification de f(1)-f(-1) est > 10**-7, alors la fonction n'est pas paire. Le pilotage du retour arrière se fait de la même façon après avoir signalé que la fonction n'est sûrement pas paire puisque ne coïncidant pas en 1 et -1.

iii) La page suivante fournit des résultats obtenus sur les pb1 et pb2 précédents. On y voit l'affichage de la (ou des) raison(s) suffisante(s) pour ne pas tenter la preuve.

```
solut pb(11);                                    (PRV (PAIRE X x)
   NO DU PLAN A UTILISER ?
999

   J'AI PU APPLIQUER 7                           Règle 7 : résultat connu "impaire x ↦ x"

   ".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")
   PB ABSURDE:F(X) IMPAIRE                       } application de MREG(5)

   ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))

   ".... PROBLEME A RESOUDRE " (!%ECRIRE "F(X) NON PAIRE:F(1)#F(-1)")
   F(X) NON PAIRE:F(1)#F(-1)                     } application MREG(7)

   ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))
   "J'ENVISAGE LE PLAN " PLANNIL

   ***** ECHEC SUR (PRV (PAIRE (X) X))

   NIL

solut pb(13);                                    (PRV (PAIRE (EXPT E x) x))
   NO DU PLAN A UTILISER ?
999

   ".... PROBLEME A RESOUDRE " (!%ECRIRE "F(X) NON PAIRE:F(1)#F(-1)")
   F(X) NON PAIRE:F(1)#F(-1)

   ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))
   "J'ENVISAGE LE PLAN " PLANNIL

   ***** ECHEC SUR (PRV (PAIRE (EXPT E X) X))

   NIL
```

2) -limite du travail précédent

i) Telle que nous l'avons implantée, la séparation entre niveau connaissansse et méta-connaissance est artificielle. Par exemple, la vérification f (1) = f (-1) avant de prouver que f (x) est paire est une connaissance du niveau méta : c'est une connaissance qui permet de piloter l'usage des connaissances de la base (en l'occurence, d'interdire l'usage de toute connaissance de cette base). On pourrait aussi considérer que cette méta-connaissance est du domaine de la connaissance du mathématicien.

ii) autre approche possible : une approche plus intéressante consisterait à mettre dans la base de connaissances tout ce qui concerne le domaine mathématique et de ne laisser au niveau méta que ce qui concerne le contrôle de l'usage de ces connaissances. Les exemples donnés en F 1 pourraient alors se traduire ainsi :

* mettre dans la base de connaissances, à côté des résultats connus des déclarations : par exemple (SEXCLUENT (PAIRE f x) (IMPAIRE f x)).

Ces déclarations, manipulées comme les réécritures (REEC) des résultats connus, devraient pouvoir être conditionnées (cf règles conditionnelles (VIVET 82 c)).

* mettre dans la base de connaissances, des plans permettant d'exprimer la vérification sémantique : par exemple :

```
POUR (PRV (PAIRE f x)) essayer
(SUC 2 (SASSURER (f (1) est f (-1))
       (PRV (PAIRE f x))
```

dans lequel l'action SASSURER (cond)
. renvoie VRAI si cond est VRAIE
. force le retour arrière dans l'arborescence qui a conduit à ce problème sinon. La prise en compte de cette possibilité nécessite d'intervenir sur la gestion faite par le moteur d'inférence. Nous retrouvons ce que fait la commande ECHEC dans les plans de M. BARON (BARON 82a).

* Il resterait alors à gérer de telles connaissances au niveau méta avec des méta-règles qui ne feraient plus intervenir le domaine mathématique. Elles pourraient prendre l'allure suivante :

```
exclusion
. si le problème est de la forme (PRV P1)
  si l'on a une connaissance (SEXCLUENT P1 P2)
  si (résultat connu P2)
Alors
(suc 2 (écrire "PB absurde")
       (%ATTRIB NEWLPP (0)))

 Coïncidence en 1 et -1
 . si (et (VUS 'SASSURER P1)
          (NONVUS 'SASSURER P2))
   Alors (APPLIQUERAVANT P1 P2)
```

Cette méta-règle générale indiquerait qu'il faut appliquer les plans comportant une vérification (pouvant forcer un retour ar-

rière) avant les autres plans.

iii) Cette approche ii) nous paraît plus intéressante et être une voie dans laquelle il faudrait travailler pour atteindre une bonne séparation entre les niveaux connaissances et méta-connaissances. Cette possibilité n'est pas actuellement implémentée dans CAMELIA.

G. PROBLEMES D'EFFICACITE

La mise en place des méta-règles dans CAMELIA a permis d'atteindre des solutions beaucoup plus satisfaisantes quant au comportement du système face à un problème. On constate que pratiquement, il n'y a plus d'essais ridicules qui sont tentés. Disons qu'on peut constater un comportement "beaucoup plus intelligent". Les performances au niveau temps de calcul ont par contre baissé : les temps sont pratiquement multipliés par 10 avec la première implantation faite. La mise en place d'une politique systématique de calcul d'un attribut stable uniquement lors de sa première utilisation doit permettre d'améliorer cette situation. Nous n'avons pas refait de mesures pour vérifier qu'actuellement la chute de performances n'est pas aussi grave.

H. POURSUITE DU TRAVAIL

.. Le travail tel que nous l'envisageons maintenant comporte plusieurs facettes :

1) amélioration de l'écriture d'un certain nombre de points permettant de réduire la chute de performance vue en G. Ce travail est en cours.

2) augmentation du nombre de méta-règles utilisées pour prendre plus conscience encore de tout ce qui peut être traité à ce niveau. Il y a là un travail d'expérimentation assez long à faire.

3) reprendre avec les idées décrites en F. 2-ii) l'écriture des méta-règles en séparant plus finement connaissance de méta-connaissance. Cela peut amener à revoir un peu le moteur d'inférence, la gestion intelligente des retours arrières en particulier.

CONCLUSION

Les expériences faites avec les méta-règles dans CAMELIA montrent clairement qu'un apport d'intelligence important peut être fait en utilisant la méta-connaissance . Il est certain qu'un système expert ne doit pas se contenter de manipuler les connaissances de l'expert mais doit aussi manipuler les connaissances qu'à l'expert pour manipuler des connaissances. L'intérêt est évidemment de pouvoir manipuler ces différents niveaux de connaissances de façon homogène dans la forme. Par contre, les moments d'interventions de ces différents niveaux doivent être bien définis.

IV. 3.2.5 Exécution des plans

Pour un problème donné, le moteur établit une liste de plans possibles. Ceci est fait, soit en demandant à l'utilisateur le numéro du plan à appliquer (cf IV.3.2.3 pour le pilotage par l'utilisateur), soit en mettant en oeuvre les mécanismes de sélection de plans. Le "meilleur" plan étant choisi, il reste à exécuter effectivement les étapes qu'il préconise. Ces étapes peuvent être des actions immédiates ou des sous-problèmes. Les actions immédiates sont exécutées par appel des procédures associées sous le contrôle d'un évaluateur EXECACTIM. Le résultat de cette évaluation peut être "achevé" : l'étape a pu se résoudre par seulement les appels de procédures, l'on passe alors à l'étape suivante du plan en cours. Le résultat peut aussi être "inachevé" : dans ce cas l'étape se ramène à un sous-problème que l'on résout récursivement avant de poursuivre.

IV.3.3 Traitement d'exemples

IV.3.3.1 Exemple 1 :

* Soit (PB16) à prouver que la fonction Cos (Sin x) est paire en x. Ce problème est posé sous la forme :

```
SOLUT '(PRV (PAIRE (COS (SIN x)) x));
```

* CAMELIA dispose dans sa base de connaissances des règles et des plans suivants :

règles

```
reg(0);
    5:
    (PAIRE0 (DESCRIPTEURS VARSUBST (X)))

algebraic paire0;
    6:
    REEC(PAIRE(COS(X),X),T)

reg(5);
    7:
    (IMPAIRE5 (DESCRIPTEURS VARSUBST (X)))

algebraic impaire5;
    8:
    REEC(IMPAIRE(SIN(X),X),T)
```

plans

```
prettyprint plan(6);
    (PAIRE6
       (DESCRIPTEURS VARSUBST (F X) OIOPEX (COS ABS) COUTMETHODE 5
          MSGSUCCES "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST PAIRE")
       POUR
       (PRV (PAIRE F X))
       ESSAYER
       (SI
          (OU ((!%OP F) EST COS) ((!%OP F) EST ABS))
          (CHOIX
             2
             (PRV (PAIRE (!%ARG1 F) X))
             (PRV (IMPAIRE (!%ARG1 F) X)))) )

prettyprint plan(14);
    10: (IMPAIRE14
          (DESCRIPTEURS VARSUBST (F X) COUTMETHODE 1 MSGSUCCES
             " PROPRIETE EVIDENTE "
             MSGECHEC
             "JE NE SAIS PAS TOUT PAR COEUR")
          POUR
          (PRV (IMPAIRE F X))
          ESSAYER
          (SI
             (RESULTATCONNU (IMPAIRE F X) IMPAIRE)
             (!%ECRIRE "RESULTAT CONNU")))
```

figure 4a

```
msglevel:=5$   pb(16);
    13:
    14:
    (PRV (PAIRE (COS (SIN X)) X))

solut pb(16);
    15:
    "J'ENVISAGE LE PLAN " PAIRE3
    "???????????????????? VOYONS SI " "PROPRIETE EVIDENTE "
    "##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
    "J'ENVISAGE LE PLAN " PAIRE8
    "???????????????????? VOYONS SI " " FONCTION CONSTANTE DONC PAIRE"
    "J'ENVISAGE LE PLAN " PAIRE6
    "???????????????????? VOYONS SI " "COS OU ABS D'UNE FCT PAIRE OU IMPAI
RE E
    ST PAIRE"
    " "
    ".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG1 (COS (SIN X))) X))

    J'AI PU APPLIQUER 5
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

    PB ABSURDE:F(X) IMPAIRE
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))
    "J'ENVISAGE LE PLAN " PLANNIL
    ***** ECHEC SUR (PRV (PAIRE (SIN X) X))

    " "
    ".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG1 (COS (SIN X))) X))

    "J'ENVISAGE LE PLAN " IMPAIRE14
    "???????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
    J'AI PU APPLIQUER 5

    RESULTAT CONNU
    "LE PLAN " 14 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" " PROPRIETE EVIDENTE "
    "LE PLAN " 6 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST P
AIRE
    "

    T
```

```
msglevel:=7;pb(16);
    16:
    17:
    (PRV (PAIRE (COS (SIN X)) X))

solut pb(16);
    18:
    "LPP : " (3 6 8 9 10)
    "TABTRI(" 1 ") : " (27000 3)
    "TABTRI(" 2 ") : " (27000 8)
    "TABTRI(" 3 ") : " (19142 6)
    "TABTRI(" 4 ") : " (7714 10)
    "TABTRI(" 5 ") : " (6000 9)
    "J'ENVISAGE LE PLAN " PAIRE3
    "???????????????????? VOYONS SI " "PROPRIETE EVIDENTE "
    "##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
    "J'ENVISAGE LE PLAN " PAIRE8
    "???????????????????? VOYONS SI " " FONCTION CONSTANTE DONC PAIRE"
    "J'ENVISAGE LE PLAN " PAIRE6
    "???????????????????? VOYONS SI " "COS OU ABS D'UNE FCT PAIRE OU IMPAI
RE E
    ST PAIRE"
    ... J'AI LE CHOIX ENTRE LES ETAPES SUIVANTES
    (PRV (PAIRE (!%ARG1 (COS (SIN X))) X))
    (PRV (IMPAIRE (!%ARG1 (COS (SIN X))) X))
    " "
    ".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG1 (COS (SIN X))) X))

    "LPP : " (3 8 9 10)
    J'AI PU APPLIQUER 5
    ... IL ME FAUT REUSSIR  LES ETAPES SUIVANTES
    (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")
    (!%ATTRIB NEWLPP (0))
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

    PB ABSURDE:F(X) IMPAIRE
    "==> RESULTAT SUC : " T
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))
    "==> RESULTAT SUC : " T
    "TABTRI(" 1 ") : " (51000 0)
    "J'ENVISAGE LE PLAN " PLANNIL
    ***** ECHEC SUR (PRV (PAIRE (SIN X) X))

    "J'AI ESSAYE TOUS LES PLANS " NIL
    "==> RESULTAT CHOIX : " NIL
    " "
    ".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG1 (COS (SIN X))) X))

    "LPP : " (14 17 19)
    "TABTRI(" 1 ") : " (27000 14)
    "TABTRI(" 2 ") : " (16750 17)
    "TABTRI(" 3 ") : " (6000 19)
    "J'ENVISAGE LE PLAN " IMPAIRE14
    "???????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
    J'AI PU APPLIQUER 5

    RESULTAT CONNU
    "LE PLAN " 14 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" " PROPRIETE EVIDENTE "
    "==> RESULTAT CHOIX : " T
    "LE PLAN " 6 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST P
 AIRE
    "

    T
```

figure 4b

La figure 4 donne la solution obtenue pour ce problème avec un niveau de message fixé a 5.La figure 4 donne la solution avec un niveau de message fixé à 7. on y voit la trace du tri effectué pour choisir les plans dans le tableau TABTRI qui contient les couples (note du plan , n° du plan).

* Dans un premier temps, SOLUT va activer son mécanisme de sélection grossière et ses méta-règles pour choisir les plans susceptibles de convenir.

- A ce stade, il vérifie que f(x) : Cos (Sin(x)) n'est pas connu comme une fonction impaire (Méta-règle 5) ce qui serait contradictoire avec le problème et que $|f(1) - f(-1)| <$ 1.E -07 (vérification de type sémantique faite par la méta-règle 7). En effet, il ne sert à rien de prouver que f(x) est paire si déjà $f(1) \neq f(-1)$.

- Tous les plans qui ne répondent pas au problème caractéristique (PRV PAIRE) sont écartés et dans ceux qui restent on élimine ceux qui imposent la présence d'opérateurs (OTOPEX) autres que SIN et COS (seuls opérateurs présents dans le problème).

* SOLUT prend le meilleur candidat : (le plan (6)). Celui-ci est de la forme (Si (cond) (étapes)).

- L'instanciation est faite et la substitution (F : Cos (Sin x) , x : x) est établie.

- La condition comporte des actions immédiates : (!%OP : pour accéder à l'opérateur de F). Celles-ci sont exécutées et la condition passe de la forme :

(OU ((!%OP (COS(SIN x))) est COS) ((!%OP (COS(SIN x)) est ABS))

à

(OU (COS est COS) (COS est ABS))

- Cette condition est reconnue comme vraie puisque le premier terme du OU est vrai.

* SOLUT lance alors l'exécution des étapes du (SI...). Il a le choix entre deux étapes. Il choisit la première :

à savoir : (PRV (PAIRE (!%ARG1 (COS (SIN x))) x))

Cette étape comporte l'action immédiate !%ARG1 (qui permet d'accéder à l'argument de la fonction). EXECACTIM renvoie le problème "inachevé" (PRV (PAIRE (SIN x) x)).

- Celui-ci est examiné par SOLUT qui déclare ce problème comme absurde (la méta-règle 5 faisant apparaître la contradiction avec le fait que Sin(x) est une fonction connue comme impaire). La vérification sémantique MREG (7) est d'ailleurs aussi appliquée et l'on annonce à l'utilisateur deux bonnes raisons pour que ce problème soit absurde. A savoir, Sin(x) est impaire et $f(1) \neq f(-1)$.

- Cette voie est donc un échec et l'on revient au 2ème choix qui était possible dans la méthode préconisée par plan (6) et SOLUT se pose le problème :

(PRV (IMPAIRE (!%ARG1 (COS (SIN x))) x))

EXECACTIM réduit le problème à

(PRV (IMPAIRE (SIN x) x))

problème inachevé traité par SOLUT.

* Ce problème fait l'objet d'un traitement du même type.

- Sélection des plans applicables (on peut observer que plan (6) n'en fait plus partie à cause de OTOPEX (COS ABS)). Le problème n'est pas rejeté comme absurde par les méta-règles.

- Le plan (14) s'avère être le meilleur candidat et est donc choisi. En effet, ce plan (pénalisé uniquement d'un COUTMETHODE 2) est simple : il est facile de voir qu'un résultat est connu.

- L'instanciation amène la substitution (F : (SIN x), x : x) pour les variables substituables et le plan (14) instancié est exécuté.

- Les étapes sont :

(SI (RESULTATCONNU (IMPAIRE (SIN x) x) IMPAIRE)

(!%ECRIRE "Résultat connu"))

La condition (de la forme (prédicat args)) est évaluée. La procédure RESULTATCONNU consulte les règles de caractéristique IMPAIRE et l'instanciation avec la règle 5 est possible. RESULTATCONNU annonce à l'utilisateur la justification "j'ai pu appliquer 5"
et le résultat T (membre droit de la règle) est renvoyé. La condition est donc vraie et l'on exécute la partie action qui se réduit ici au problème (!%ECRIRE "Résultat connu").

* SOLUT prend ce dernier problème. Mais l'évaluateur EXECACTIM suffit à résoudre le problème : ECRIRE affiche le message à l'utilisateur et retourne T. Le plan 14 a donc abouti (l'utilisateur en est prévenu par le message figurant sous MSGSUCCES). Le problème (PRV (IMPAIRE (SIN x) x)) est donc un succès. Ceci rend possible le succès du CHOIX prévu dans plan (6) qui s'achève ainsi. SOLUT prévient l'utilisateur avec le message MSGSUCCES prévu dans ce plan "Le COS ou ABS....".

Remarques :

1) Cet exemple permet effectivement de voir comment

s'articulent : plans - problèmes - conditions - exécutions d'actions immédiates - retour-arrière -problème refusé comme absurde à cause des méta-règles.

2) On peut observer que SOLUT active deux fois la procédure RESULTATCONNU avec les mêmes arguments

expression : (IMPAIRE (Sin x) x) caractéristique : IMPAIRE

- la première fois : c'est la méta-règle (5) pour déclarer que le problème

(PAIRE (SIN x) x)

est absurde.

- la deuxième fois : pour effectivement résoudre le problème

(IMPAIRE (SIN x) x)

On peut s'inquiéter de voir que SOLUT attaque deux fois le même problème sans s'en apercevoir. Mais les raisons de l'activation sont ici très différentes. SOLUT ne gère pas de base de faits et CAMELIA ne garde pas d'historique des problèmes qu'il a résolu. C'est ce qui explique ce comportement.

IV.3.3.2 exemple 2 :

Cet exemple illustre l'usage des conditions problèmes sous les opérateurs UNVRAI, TOUSVRAI, AUCUNVRAI. Les solutions de trois problèmes (PB (17), PB (18), PB (19) sont données en figure 5. La commande RESOUD (16, 19, "T", NIL) permet d'enchaîner les résolutions avec solution écrite sur le terminal (un nom de fichier à la place de NIL permettrait d'avoir la solution dans ce fichier).

Ces exemples sont très simples et destinés à montrer les mécanismes utilisés. Commmentons les.

Pb (17) : le texte est visible sur la figure. Il s'agit d'écrire un message si une des fonctions (Sin x) ou (Cos x) est paire.

* CAMELIA cherche à évaluer la condition

(UNVRAI ((SIN x) (COS x)) (PAIRE ? x))

* Il balaie la liste ((SIN x) (COS x)) avec le joker ? et se pose le premier problème.

Condition problème (PAIRE (SIN x) x)

* Le niveau méta reconnaît alors ce problème comme absurde (l'on sait (règle 5) que SIN (x) est impaire en x.

* CAMELIA attribue à NEWLPP le seul numéro de plan qu'il croit utile d'utiliser : à savoir 0 le numéro de PLANNIL (l'ignorance universelle). CAMELIA sait qu'il ne pourra résoudre ce problème.

* Il passe alors au choix suivant et se pose le problème

Condition problème (PAIRE (COS x) x)

* Il résoud ce problème en consultant sa base de propriétés connues (PLAN (3)) où il trouve la règle 0 lui confirmant que COS (x) est connue comme fonction paire en x.

* Une des conditions sous UNVRAI est donc vraie : la condition (UNVRAI...) est donc vraie et l'action s'exécute. Le message est écrit!.

. Pb (18) : Le problème est de même nature avec un problème de type AUCUNVRAI.

* On voit sur la figure que les méta-règles rejettent deux fois comme absurdes les problèmes envisagés à savoir :

(IMPAIRE (COS x) x)

(IMPAIRE (EXPT x) 2) x)

figure 5-a

```
----------THES193 SUR SY2 ----------
IN CAMELIA;
%#########################################################$

%$

OUT REDUFICH $

CHAPITRE(S) A LIRE? ENTREE DU CHAPITRE PARITE
FIN DU CHAPITRE PARITE  -------> TEMPS CHARGEMENT
TIME: 2749 MS

 %#########################################################$

END$

%$

NAVIGAVUE :=NIL$
MSGLEVEL:=5$

CONTEXTE()$
NAVIGAVUE     : NIL
MSGLEVEL      : 5
PLANOBJ       : (NIMPORTEQUOI (0 1 2) (PRV PAIRE) (3 4 5 6 7 8 9 10 11 12) (PRV IMPAIRE) (14 15 16 17 18 19 20 21))
REGOBJ        : (REGOBJ SOMAB (15) IMPAIRE (9 8 7 6 5 4) PAIRE (3 2 1 0))

WRITE "TEMPS CHARGEMENT : "$
TEMPS CHARGEMENT : SHOWTIME;

TIME: 333 MS

NIL

%$

ON ECHO$

%PREUVES DE PARITE  $

RESOUD (16,19,"T",NIL)$
*******************************
** PB(16) ** =====> (PRV (PAIRE (COS (SIN X)) X))
*******************************


TIME: 14 MS

"J'ENVISAGE LE PLAN " PAIRE3
"???????????????? VOYONS SI " "PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " PAIRE8
"???????????????? VOYONS SI " " FONCTION CONSTANTE DONC PAIRE"
"J'ENVISAGE LE PLAN " PAIRE6
"???????????????? VOYONS SI " "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST PAIRE"
" "
".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG1 (COS (SIN X))) X))

J'AI PU APPLIQUER 5
" "
".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

PB ABSURDE:F(X) IMPAIRE
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))
```

figure 5-4

```
"J'ENVISAGE LE PLAN " PLANNIL
***** ECHEC SUR (PRV (PAIRE (SIN X) X))

" "
".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG1 (COS (SIN X))) X))

"J'ENVISAGE LE PLAN " IMPAIRE14
"??????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
J'AI PU APPLIQUER 5

RESULTAT CONNU
"LE PLAN " 14 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" " PROPRIETE EVIDENTE "
"LE PLAN " 6 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST PAIRE"
TEMPS EN MS :
TIME: 7917 MS
NIL
*******************************
** PB(17) ** =====> (SI (UNVRAI ((SIN X) (COS X)) (PAIRE ? X)) (%ECRIRE EXEMPLE DE CONDITION PROBLEME SOUS UNVRAI))

*******************************


TIME: 2 MS

"CONDITION PROBLEME ===> " (PAIRE (SIN X) X)

J'AI PU APPLIQUER 5
" "
".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

PB ABSURDE:F(X) IMPAIRE
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (O))
"J'ENVISAGE LE PLAN " PLANNIL
***** ECHEC SUR (PRV (PAIRE (SIN X) X))


"CONDITION PROBLEME ===> " (PAIRE (COS X) X)

"J'ENVISAGE LE PLAN " PAIRE3 ⊗
"??????????????????? VOYONS SI " "PROPRIETE EVIDENTE "
J'AI PU APPLIQUER O

RESULTAT CONNU
"LE PLAN " 3 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "PROPRIETE EVIDENTE "

EXEMPLE DE CONDITION PROBLEME SOUS UNVRAI
TEMPS EN MS :
TIME: 4896 MS
NIL
*******************************
** PB(18) ** =====> (SI (AUCUNVRAI ((COS X) (EXPT X 2)) (IMPAIRE ? X)) (%ECRIRE NI COS(X) NI X**2 N'EST IMPAIRE))
*******************************


TIME: 3 MS

"CONDITION PROBLEME ===> " (IMPAIRE (COS X) X)

J'AI PU APPLIQUER O
" "
".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) PAIRE")
```

figure 5-c

```
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (O))
    "J'ENVISAGE LE PLAN " PLANNIL
    ***** ECHEC SUR (PRV (IMPAIRE (COS X) X))


    "CONDITION PROBLEME ===> " (IMPAIRE (EXPT X 2) X)

    J'AI PU APPLIQUER 3
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) PAIRE")

*   PB ABSURDE:F(X) PAIRE
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (O))
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE "F(X) NON IMPAIRE:F(1)#-F(-1)")

*   F(X) NON IMPAIRE:F(1)#-F(-1)
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (O))
    "J'ENVISAGE LE PLAN " PLANNIL
    ***** ECHEC SUR (PRV (IMPAIRE (EXPT X 2) X))


    NI COS(X) NI X**2 N'EST IMPAIRE
    TEMPS EN MS :
    TIME: 2176 MS
    NIL
    ********************************
    ** PB(19) ** =====> (CHOIX 2 (SI (TOUSVRAI ((SIN X) (TG X)) (PAIRE ? X)) (%ECRIRE RIDICULE:NI SIN(X) NI TG(X) NE SO
    NT PAIRES)) (SI (AUCUNVRAI ((SIN X) (TG X)) (PAIRE ? X)) (%ECRIRE BON:NI SIN(X) NI TG(X) NE SONT PAIRES EN X)))
    ********************************


    TIME: 2 MS
    " "
    ".... PROBLEME A RESOUDRE " (SI (TOUSVRAI ((SIN X) (TG X)) (PAIRE !? X)) (!%ECRIRE "RIDICULE:NI SIN(X) NI TG(X) NE
    SONT PAIRES"))

    "CONDITION PROBLEME ===> " (PAIRE (SIN X) X)

    J'AI PU APPLIQUER 5
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

*   PB ABSURDE:F(X) IMPAIRE
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (O))
    "J'ENVISAGE LE PLAN " PLANNIL
    ***** ECHEC SUR (PRV (PAIRE (SIN X) X))

    " "
    ".... PROBLEME A RESOUDRE " (SI (AUCUNVRAI ((SIN X) (TG X)) (PAIRE !? X)) (!%ECRIRE "BON:NI SIN(X) NI TG(X) NE SONT
     PAIRES EN X"))

    "CONDITION PROBLEME ===> " (PAIRE (SIN X) X)

    J'AI PU APPLIQUER 5
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

*   PB ABSURDE:F(X) IMPAIRE
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (O))
```

```
***** ECHEC SUR (PRV (PAIRE (SIN X) X))


"CONDITION PROBLEME ===> " (PAIRE (TG X) X)

J'AI PU APPLIQUER 6
" "
".... PROBLEME A RESOUDRE " (I%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

* PB ABSURDE:F(X) IMPAIRE
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB NEWLPP (O))
"J'ENVISAGE LE PLAN " PLANNIL
***** ECHEC SUR (PRV (PAIRE (TG X) X))


BON:NI SIN(X) NI TG(X) NE SONT PAIRES EN X
TEMPS EN MS :
TIME: 3265 MS
NIL
```

* Aucune des fonctions (COS x) ou (X xx2) n'étant impaire, la condition (AUCUNVRAI...) devient vraie et le message "NI COS (x)...." apparaît.

. Pb (19) : Le problème est un choix entre deux problèmes de même nature que Pb (17) et Pb (18).

* On aperçoit que le premier choix échoue très vite, le TOUSVRAI devenant faux dès que (SIN x) est reconnue comme non paire en x.

Remarques : Ces trois exercices sont été traités au début d'une session. Pb (17), traité en premier, a un temps (4862 ms) plus long à l'exécution que les autres exercices; en particulier, comparer à Pb (19) (3261 ms) qui nécessite plus de travail apparent. La raison vient du fait que pour les plans concernés (qui sont les mêmes dans les trois problèmes) le calcul des descripteurs fixes pour les plans est fait lors de la première évocation des plans. L'excès de temps passé à Pb (17) est donc du temps que l'on n'a pas passé au pré-traitement (cf IV.2.4.1.b).

IV.3.3.3. Exemple 3 :

* Soit (Pb156) à calculer $\int$(tg x) Log (Cos x) dx

* La figure 6 donne le graphe du cheminement fait pour effectuer ce calcul.

- les noeuds sont numérotés et renvoient aux lignes correspondantes du résultat (figure 7)

- les flèches doubles, horizontales indiquent des successions (noeuds SUC des plans).

- les flèches discontinues, horizontales indiquent les choix (noeuds CHOIX des plans).

- Les listes de numéros sur les flèches obliques donnent les numéros des plans envisagés pour résoudre le problème au noeud. Le dernier numéro souligné de deux traits est le plan retenu. Les messages "VOYONS SI..." permettent d'avoir une idée de ce qui a été tenté. L'annexe D contient le détail des connaissances nécessaires.

* La figure 7 donne la solution obtenue avec un MSGLEVEL fixé à 5.

* Notons qu'au cours de cette preuve, CAMELIA ne savait pas que tg (x) est SIN (x) / COS (x) mais seulement que une primitive de tg (x) est -Log (COS x).

Expliquons la démarche

* Après quelques essais sans succès, CAMELIA tente assez rapidement l'intégration par parties (étape 2). Il hésite dans le choix de ce qu'il prend comme u et comme dv : il tente le choix u = tg x et dv = Log (COS x) dx, mais échoue dans le calcul de $\int$ Log (COS x) dx (problème posé à l'étape 3). La seule méthode sérieuse disponible pour aborder ce problème dans la base de connaissance est le plan (41) de changement de variable trigonométrique et ce plan échoue car l'intégrande n'est pas stable par les transformations x ==> -x, x ==> PI+x, x ==> PI-x.

* L'autre choix u = Log (COS x) et dv = tg (x) dx est alors tenté. Le problème (6) $\int$ tg x dx aboutit grâce à la règle 42 (résultat connu) évoquée par le plan (20). L'intégrale (7)
$\int \frac{\text{SIN x Log (COS x)}}{\text{COS x}}$ dx est tentée. Notons que CAMELIA ne voit pas que c'est le problème initial un peu modifié (il ne sait pas que tg (x) est SIN (x) / COS (x)).
Après plusieurs essais, il tente le plan (41). Observons que ce plan était applicable dès le départ. C'est le jeu des coûts et espoirs des méta-règles qui ont fait tenter 31 avant 41 à l'étape 1.

* Ce plan 41 met en jeu une condition problème. Pour établir que la fonction est impaire en x (étape 8), le plan (16) est

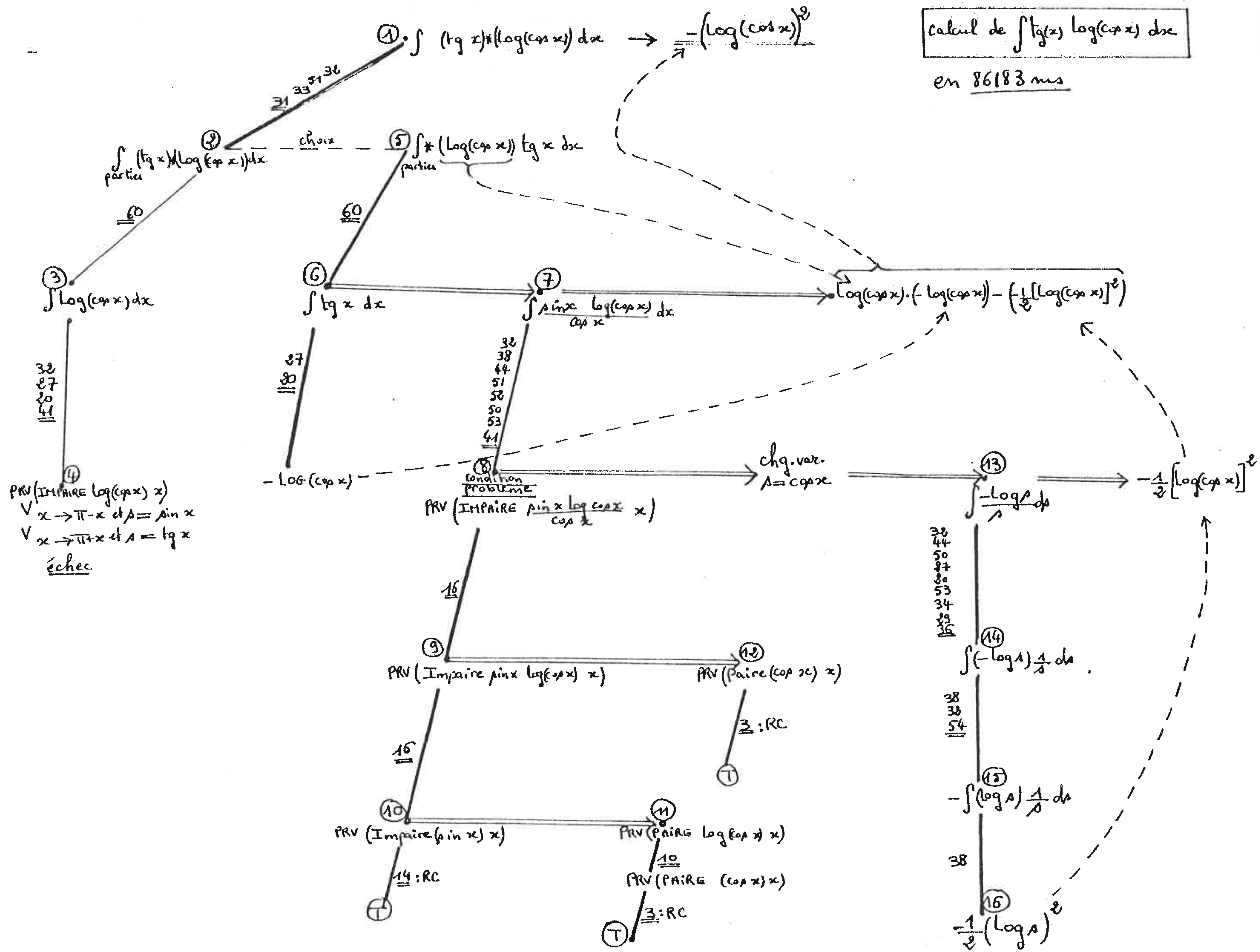

calcul de ∫ tg(x) Log(cos x) dx
en 86183 ms
① ∫ (tg x)*(Log(cos x)) dx → −(Log(cos x))²
② ∫ parties (tg x)*(Log(cos x)) dx
choix
⑤ ∫ parties * (Log(cos x)) tg x dx
③ ∫ Log(cos x) dx
④ PRV (IMPAIRE Log(cos x) x)
V x → π − x et s = sin x
V x → π + x et s = tg x
échec
⑥ ∫ tg x dx
− LOG (cos x)
⑦ ∫ sin x Log(cos x) / cos x dx
Log(cos x).(− Log(cos x)) − (−1/2 [Log(cos x)]²)
⑧ condition problème
PRV (IMPAIRE sin x Log cos x / cos x x)
chg. var. s = cos x
⑨ PRV (Impaire sin x Log(cos x) x)
⑫ PRV (Paire (cos x) x)
3 : RC
⑩ PRV (Impaire (sin x) x)
14 : RC
⑪ PRV (PAIRE Log(cos x) x)
PRV (PAIRE (cos x) x)
3 : RC
⑬ ∫ −Log s / s ds
−1/2 [Log(cos x)]²
⑭ ∫ (−Log s) 1/s ds
⑮ − ∫ (Log s) 1/s ds
⑯ −1/2 (Log s)²

figure 6

figure 1-1

$$\int tg(x)\, Log(\cos x)\, dx$$

```
%RESOUD(145,149,"T",NIL)$
%RESOUD(151,151,"T",NIL)$
%RESOUD(152,153,"T",NIL)$
%RESOUD(154,155,"T",NIL)$
RESOUD(156,156,"T",NIL)$
**********************************
** P8(156) ** =====> (SUC 2 (CALCUL (PRIM (TIMES (TG X) (LOG (COS X))) X) RR) (%ECRIRE (%ALGEB RR)))
**********************************

TIME: 5410 MS
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TIMES (TG X) (LOG (COS X))) X) RR)

"J'ENVISAGE LE PLAN " INT38
"???????????????? VOYONS SI " "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (SI ((!%DERIV (TG X) X) EST (LOG (COS X))) (!%ATTRIB RR (QUOTIENT (EXPT (TG X) 2) 2)))

" "
".... PROBLEME A RESOUDRE " (SI ((!%DERIV (LOG (COS X)) X) EST (TG X)) (!%ATTRIB RR (QUOTIENT (EXPT (LOG (COS X))
2) 2)))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " INT32
"???????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT51
"???????????????? VOYONS SI " "INT(COS(A*X+B),X)  ."
"J'ENVISAGE LE PLAN " INT33
"???????????????? VOYONS SI " "INTEGRATION DU PRODUIT PAR PARTIE"
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (TG X) (X)) (VUS X (LOG (COS X)))) (CHOIX 2 (SI (OU ((!%OP (LOG (COS X
))) EST EXPT) (TRIGO (LOG (COS X)))) (CALCUL (INTEGRERPARPARTIE (TIMES (TG X) (LOG (COS X))) X) RR)) (SI ((!%OP (LO
G (COS X))) EST LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (LOG (COS X)) (TG X)) X) RR))))
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (LOG (COS X)) (X)) (VUS X (TG X))) (CHOIX 2 (SI (OU ((!%OP (TG X)) EST
 EXPT) (TRIGO (TG X))) (CALCUL (INTEGRERPARPARTIE (TIMES (TG X) (LOG (COS X))) X) RR)) (SI ((!%OP (LOG (COS X))) ES
T LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (LOG (COS X)) (TG X)) X) RR))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"##### EXPLICATION :" "ECHEC SUR PRODUIT PAR PARTIES"
"J'ENVISAGE LE PLAN " INT41
"???????????????? VOYONS SI " "PRODUIT INTEGRE PAR PARTIES"
" "
".... PROBLEME A RESOUDRE " (CALCUL (INTEGRERPARPARTIE (TIMES (TG X) (LOG (COS X))) X) RR)

"J'ENVISAGE LE PLAN " INT60
"???????????????? VOYONS SI " "INTEGRATION PAR PARTIES"
" "
".... PROBLEME A RESOUDRE " (!%CREER (VIPP))
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (LOG (COS X)) X) (!%VAL VIPP))

"J'ENVISAGE LE PLAN " INT32
"???????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT27
"???????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " INT20
"???????????????? VOYONS SI " "RESULTAT USUEL "
```

120/2

figure 7-8

```
"#####" EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR "
"J'ENVISAGE LE PLAN " INT41
"??????????????????? VOYONS SI " "CHG VARIABLE FONCTION TRIGONOMETRIQUE"
" "
".... PROBLEME A RESOUDRE " (!%CREER (R1 R2 R3 R4 R5 R6))
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R6) (!%NOUVAR))
*** SERVO095 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (LOG (COS X)))
*** SERVO094 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (CHOIX 3 (SI (IMPAIRE (LOG (COS X)) X) (SUC 2 (!%ATTRIB (!%VAL R5) (COS X)) (!%ATTRIB (
!%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (COS X))))) (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (DI
FFERENCE PI X) (!%VVAL R1)))) (!%MONTRER R2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRI
B (!%VAL R5) (SIN X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (SIN X)))))) (SUC 3 (!%ATTRIB (!
%VAL R2) (!%SIMPLIF (!%REMPL X (PLUS PI X) (!%VVAL R1)))) (!%MONTRER R2) (SI ((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (
!%ATTRIB (!%VAL R5) (TG X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (TG X)))))))
" "
".... PROBLEME A RESOUDRE " (SI (IMPAIRE (LOG (COS X)) X) (SUC 2 (!%ATTRIB (!%VAL R5) (COS X)) (!%ATTRIB (!%VAL R3)
 (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (COS X)))))

4/ "CONDITION PROBLEME ===> " (IMPAIRE (LOG (COS X)) X)

"J'ENVISAGE LE PLAN " IMPAIRE14
"??????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
"#####" EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " IMPAIRE19
"??????????????????? VOYONS SI " "DEFINITION D'UNE FCT IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%SIMPLIF (!%REMPL X (MINUS X) (LOG (COS X)))))
*** SERVO101 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R1)
R1  SERVO101  (LOG (COS X))

LOG(COS(X))
" "
".... PROBLEME A RESOUDRE " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (LOG (COS X)))) EST 0) T)
"******ECHEC SUR " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (LOG (COS X)))) EST 0) T)
"#####" EXPLICATION :" "DEFINITION  DE IMPAIRE INAPPLICABLE"
*****" ECHEC SUR (PRV (IMPAIRE (LOG (COS X)) X))

" "
".... PROBLEME A RESOUDRE " (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (DIFFERENCE PI X) (!%VVAL R1)))) (!%M
ONTRER R2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SIN X)) (!%ATTRIB (!
%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (SIN X))))))
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (DIFFERENCE PI X) (!%VVAL R1))))
*** SERVO095 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R2)
R2  SERVO095  (LOG (MINUS (COS X)))

LOG( - COS(X))
" "
".... PROBLEME A RESOUDRE " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SIN
 X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (SIN X)))))
"******ECHEC SUR " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SIN X)) (!%A
TTRIB (!%VAL R5) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (SIN X)))))
" "
".... PROBLEME A RESOUDRE " (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (PLUS PI X) (!%VVAL R1)))) (!%MONTRER
```

Figure 7-3

```
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R2)
R2  SERV0095  (LOG (MINUS (COS X)))

LOG( - COS(X))
" "
".... PROBLEME A RESOUDRE " (SI (((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (!%ATTRIB (!%VAL R5) (TG X)) (!%ATTRIB (!%VAL
R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (TG X)))))
"******ECHEC SUR " (SI (((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (!%ATTRIB (!%VAL R5) (TG X)) (!%ATTRID (!%VAL R3) (!%CH
GVARINT (LOG (COS X)) X (!%VAL R6) (TG X)))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"******ECHEC SUR " (CHOIX 3 (SI (IMPAIRE (LOG (COS X)) X) (SUC 2 (!%ATTRIB (!%VAL R5) (COS X)) (!%ATTRIB (!%VAL R3)
 (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (COS X))))) (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (DIFFERENCE
PI X) (!%VVAL R1)))) (!%MONTRER R2) (SI (((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL
R5) (SIN X)) (!%ATTRID (!%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (SIN X)))))) (SUC 3 (!%ATTRIB (!%VAL R2)
(!%SIMPLIF (!%REMPL X (PLUS PI X) (!%VVAL R1)))) (!%MONTRER R2) (SI (((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (!%ATTRIB
(!%VAL R5) (TG X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (LOG (COS X)) X (!%VAL R6) (TG X)))))))
"J'ENVISAGE LE PLAN " INT29
"???????????????????? VOYONS SI " "INTEGRATION D'UN MONOME "
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%MONOMEP (LOG (COS X)) (X)))
*** SERV0102 DECLARED FLUID
"******ECHEC SUR " (!%ATTRIB (!%VAL R1) (!%MONOMEP (LOG (COS X)) (X)))
*****ECHEC SUR (CALCUL (PRIM (LOG (COS X)) X) SERV0093)

"******ECHEC SUR " (CALCUL (PRIM (LOG (COS X)) X) (!%VAL VIPP))
***** ECHEC SUR (CALCUL (INTEGRERPARPARTIE (TIMES (TG X) (LOG (COS X))) X) RR)

" "
".... PROBLEME A RESOUDRE " (CALCUL (INTEGRERPARPARTIE (TIMES (LOG (COS X)) (TG X)) X) RR)

"J'ENVISAGE LE PLAN " INT60
"???????????????????? VOYONS SI " "INTEGRATION PAR PARTIES"
" "
".... PROBLEME A RESOUDRE " (!%CREER (VIPP))
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TG X) X) (!%VAL VIPP))

"J'ENVISAGE LE PLAN " INT27
"???????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " INT25
"???????????????????? VOYONS SI " "RESULTAT USUEL "
J'AI PU APPLIQUER 42
*** SERV0105 DECLARED FLUID
"LE PLAN " 25 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "RESULTAT USUEL "
" "
".... PROBLEME A RESOUDRE " (!%MONTRER VIPP)
VIPP  SERV0105  (MINUS (LOG (COS X)))

- LOG(COS(X))
" "
".... PROBLEME A RESOUDRE " (!%CREER (DUIPP))
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL DUIPP) (!%DERIV (LOG (COS X)) X))
*** SERV0106 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (!%SIMPLIF (TIMES (!%VVAL VIPP) (!%VVAL DUIPP))) X) (!%VAL R1))

"J'ENVISAGE LE PLAN " INT32
```

$$\int \frac{\sin x \, \log(\cos x)}{\cos x} \, dx$$

figure 7-4

```
"J'ENVISAGE LE PLAN " INT33
"???????????????????? VOYONS SI " "INTEGRATION DE U*U'  "
"J'ENVISAGE LE PLAN " INT44
"???????????????????? VOYONS SI " "FRACTION AVEC DENOMINATEUR CONSTANT"
"J'ENVISAGE LE PLAN " INT51
"???????????????????? VOYONS SI " "INT(COS(A*X+B),X)  "
"J'ENVISAGE LE PLAN " INT52
"???????????????????? VOYONS SI " "INT(SIN(A*X+B),X)  "
"J'ENVISAGE LE PLAN " INT50
"???????????????????? VOYONS SI " "INT(C/F(X),X) EST C*INT(1/F(X),X"
"J'ENVISAGE LE PLAN " INT33
"???????????????????? VOYONS SI " "INTEGRATION DU PRODUIT PAR PARTIE"
"J'ENVISAGE LE PLAN " INT27
"???????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " IN(41)
"???????????????????? VOYONS SI " "CHG VARIABLE FONCTION TRIGONOMETRIQUE"
" "
".... PROBLEME A RESOUDRE " (!%CREER (R1 R2 R3 R4 R5 R6))
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R6) (!%NOUVAR))
*** SERVO113 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)))
*** SERVO108 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (CHOIX 3 (SI (IMPAIRE (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X) (SUC 2 (!%ATT
RIB (!%VAL R5) (COS X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X (!%VAL
 R6) (COS X))))) (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (DIFFERENCE PI X) (!%VVAL R1)))) (!%MONTRER R2)
(SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SIN X)) (!%ATTRIB (!%VAL R3) (!
%CHGVARINT (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X (!%VAL R6) (SIN X)))))) (SUC 3 (!%ATTRIB (!%VAL R2) (
!%SIMPLIF (!%REMPL X (PLUS PI X) (!%VVAL R1)))) (!%MONTRER R2) (SI ((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (!%ATTRIB (
!%VAL R5) (TG X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X (!%VAL R6) (
TG X)))))))
" "
".... PROBLEME A RESOUDRE " (SI (IMPAIRE (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X) (SUC 2 (!%ATTRIB (!%VA
L R5) (COS X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X (!%VAL R6) (COS
 X)))))

8 "CONDITION PROBLEME ===> " (IMPAIRE (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X)

"J'ENVISAGE LE PLAN " IMPAIRE14
"???????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " IMPAIRE17
"???????????????????? VOYONS SI " "TRIGO COMPOSEE AVEC FCT IMPAIRE"
"J'ENVISAGE LE PLAN " IMPAIRE(15)
"???????????????????? VOYONS SI " "* OU / D'UNE FCT PAIRE ET IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (SUC 2 (PRV (PAIRE (!%ARG1 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X)) (PRV (
IMPAIRE (!%ARG2 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X)))
" "
".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG1 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X))

"J'ENVISAGE LE PLAN " PAIRE3
"???????????????????? VOYONS SI " "PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " PAIRE6
"???????????????????? VOYONS SI " " FONCTION CONSTANTE DONC PAIRE"
"J'ENVISAGE LE PLAN " PAIRE6
"???????????????????? VOYONS SI " "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST PAIRE"
"J'ENVISAGE LE PLAN " PAIRE5
"???????????????????? VOYONS SI " "* OU / DE FONCTIONS PAIRES OU IMPAIRES"
" "
".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG1 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X)) (PRV (PAIRE (!%ARG2 (TIME
```

figure 5

b

```
J'AI PU APPLIQUER 5
" "
".... PROBLEME A RESOUDRE " (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")

PB ABSURDE:F(X) IMPAIRE
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPF (0))
"J'ENVISAGE LE PLAN " PLANNIL
***** ECHEC SUR (PRV (PAIRE (SIN X) X))

"******ECHEC SUR " (PRV (PAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X))
" "
".... PROBLEME A RESOUDRE " (SUC 2 (PRV (IMPAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X)) (PRV (IMPAIRE (!%ARG2 (
TIMES (SIN X) (LOG (COS X)))) X)))
" "
".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X))

"J'ENVISAGE LE PLAN " IMPAIRE(14)
"?????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
J'AI PU APPLIQUER 5

RESULTAT CONNU
"LE PLAN "(14)" A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" " PROPRIETE EVIDENTE "
" "
".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG2 (TIMES (SIN X) (LOG (COS X)))) X))

"J'ENVISAGE LE PLAN " IMPAIRE14
"?????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " IMPAIRE19
"?????????????????? VOYONS SI " "DEFINITION D'UNE FCT IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%SIMPLIF (!%REMPL X (MINUS X) (LOG (COS X)))))
*** SERV0115 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R1)
R1  SERV0115  (LOG (COS X))

LOG(COS(X))
" "
".... PROBLEME A RESOUDRE " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (LOG (COS X)))) EST 0) T)
"******ECHEC SUR " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (LOG (COS X)))) EST 0) T)
"##### EXPLICATION :" "DEFINITION  DE IMPAIRE INAPPLICABLE"
***** ECHEC SUR (PRV (IMPAIRE (LOG (COS X)) X))

"******ECHEC SUR " (PRV (IMPAIRE (!%ARG2 (TIMES (SIN X) (LOG (COS X)))) X))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " PAIRE10
"?????????????????? VOYONS SI " "U(V(X))EST PAIRE SI V(X)L'EST"
"J'ENVISAGE LE PLAN " PAIRE9
"?????????????????? VOYONS SI " "PAR DEFINITION DE PAIRE "
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%SIMPLIF (!%REMPL X (MINUS X) (TIMES (SIN X) (LOG (COS X))))))
*** SERV0116 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R1)
R1  SERV0116  (MINUS (TIMES (SIN X) (LOG (COS X))))
```

figure 7-6

```
 - SIN(X)*LOG(COS(X))
" "
".... PROBLEME A RESOUDRE " (SI ((!%SIMPLIF (DIFFERENCE (!%VVAL R1) (TIMES (SIN X) (LOG (COS X))))) EST 0) T)
"******ECHEC SUR " (SI ((!%SIMPLIF (DIFFERENCE (!%VVAL R1) (TIMES (SIN X) (LOG (COS X))))) EST 0) T)
"##### EXPLICATION :" " MEME PAS AVEC LA DEFINITION   "
***** ECHEC SUR (PRV (PAIRE (TIMES (SIN X) (LOG (COS X))) X))

"******ECHEC SUR " (PRV (PAIRE (!%ARG1 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X))
" "
".... PROBLEME A RESOUDRE " (SUC 2 (PRV (IMPAIRE (!%ARG1 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X)) (PRV
 (PAIRE (!%ARG2 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X)))
" "
9 ".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG1 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X))
plus judicieux
"J'ENVISAGE LE PLAN " IMPAIRE14
"?????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " IMPAIRE17
"?????????????????? VOYONS SI " "TRIGO COMPOSEE AVEC FCT IMPAIRE"
"J'ENVISAGE LE PLAN " IMPAIRE16
"?????????????????? VOYONS SI " "* OU / D'UNE FCT PAIRE ET IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (SUC 2 (PRV (PAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X)) (PRV (IMPAIRE (!%ARG2 (TI
MES (SIN X) (LOG (COS X)))) X)))
" "
".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X))

J'AI PU APPLIQUER 5
" "
".... PROBLEME A RESOUDRE " (!%ECRIRE "P8 ABSURDE:F(X) IMPAIRE")

P8 ABSURDE:F(X) IMPAIRE
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB NEWLPP (0))
"J'ENVISAGE LE PLAN " PLANNIL
***** ECHEC SUR (PRV (PAIRE (SIN X) X))

"******ECHEC SUR " (PRV (PAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X))
" "
".... PROBLEME A RESOUDRE " (SUC 2 (PRV (IMPAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X)) (PRV (PAIRE (!%ARG2 (TI
MES (SIN X) (LOG (COS X)))) X)))
" "
10 ".... PROBLEME A RESOUDRE " (PRV (IMPAIRE (!%ARG1 (TIMES (SIN X) (LOG (COS X)))) X))

"J'ENVISAGE LE PLAN " IMPAIRE14
"?????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
J'AI PU APPLIQUER 5

RESULTAT CONNU
"LE PLAN " 14 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" " PROPRIETE EVIDENTE "
" "
11 ".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG2 (TIMES (SIN X) (LOG (COS X)))) X))

"J'ENVISAGE LE PLAN " PAIRE3
"?????????????????? VOYONS SI " "PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
"J'ENVISAGE LE PLAN " PAIRE8
"?????????????????? VOYONS SI " " FONCTION CONSTANTE DONC PAIRE"
"J'ENVISAGE LE PLAN " PAIRE6
"?????????????????? VOYONS SI " "COS OU ABS D'UNE FCT PAIRE OU IMPAIRE EST PAIRE"
"J'ENVISAGE LE PLAN " PAIRE10
"?????????????????? VOYONS SI " "U(V(X))EST PAIRE SI V(X)L'EST"

"J'ENVISAGE LE PLAN " PAIRE2
```

120/7

figure 7-7

```
J'AI PU APPLIQUER 0

RESULTAT CONNU
"LE PLAN "(3)" A PERMIS DE CONCLURE "
"##### JUSTIFICATION :" "PROPRIETE EVIDENTE "
"LE PLAN "(10)" A PERMIS DE CONCLURE "
"##### JUSTIFICATION :" "U(V(X))EST PAIRE SI V(X)L'EST"
"LE PLAN "(16)" A PERMIS DE CONCLURE "
"##### JUSTIFICATION :" "* OU / D'UNE FCT PAIRE ET IMPAIRE"
" "
12 ".... PROBLEME A RESOUDRE " (PRV (PAIRE (!%ARG2 (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X))) X))

"J'ENVISAGE LE PLAN " PAIRE(3)
"???????????????????? VOYONS SI " "PROPRIETE EVIDENTE "
J'AI PU APPLIQUER 0

RESULTAT CONNU
"LE PLAN "(3)" A PERMIS DE CONCLURE "
"##### JUSTIFICATION :" "PROPRIETE EVIDENTE "
"LE PLAN "(16)" A PERMIS DE CONCLURE "
"##### JUSTIFICATION :" "* OU / D'UNE FCT PAIRE ET IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL P5) (COS X))   changement de variable s=cos x
*** SERVO112 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (TIMES (SIN X) (LOG (COS X))) (COS X)) X (!
%VAL R6) (COS X)))
*** SERVO110 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R3) (!%SIMPLIF (!%VVAL R3)))
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R3)
R3  SERVO110  (QUOTIENT (MINUS (LOG SERVO113)) SERVO113)

( - LOG(SERVO113))/SERVO113        ∫ -log s / s ds

13 ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (!%VVAL R3) (!%VAL R6)) (!%VAL R4))

"J'ENVISAGE LE PLAN " INT32
"???????????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT44
"???????????????????? VOYONS SI " "FRACTION AVEC DENOMINATEUR CONSTANT"
"J'ENVISAGE LE PLAN " INT50
"???????????????????? VOYONS SI " "INT(C/F(X),X) EST C*INT(1/F(X),X"
"J'ENVISAGE LE PLAN " INT27
"???????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " INT20
"???????????????????? VOYONS SI " "RESULTAT USUEL "
*** SERVO111 DECLARED FLUID
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR "
"J'ENVISAGE LE PLAN " INT53
"???????????????????? VOYONS SI " "INT(-F,X) EST -INT(F,X) "
"J'ENVISAGE LE PLAN " PLAN34
"???????????????????? VOYONS SI " "FRACTION RATIONNELLE PAR RISCH "
"##### EXPLICATION :" "PAS SU INTEGRER LA FRACTION"
"J'ENVISAGE LE PLAN " INT29
"???????????????????? VOYONS SI " "INTEGRATION D'UN MONOME "
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%MONOMEP (QUOTIENT (MINUS (LOG SERVO113)) SERVO113) (SERVO113)))

*** SERVO117 DECLARED FLUID
"******ECHEC SUR " (!%ATTRIB (!%VAL R1) (!%MONOMEP (QUOTIENT (MINUS (LOG SERVO113)) SERVO113) (SERVO113)))
```

figure 7-8

```
"J'ENVISAGE LE PLAN " INT36
"???????????????????? VOYONS SI " "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (SI ((!%DERIV (MINUS (LOG SERVO113)) SERVO113) EST (QUOTIENT 1 SERVO113)) (!XATTRIB SER
VO111 (QUOTIENT (EXPT (MINUS (LOG SERVO113)) 2) 2)))
" "
".... PROBLEME A RESOUDRE " (SI ((!%DERIV (QUOTIENT 1 SERVO113) SERVO113) EST (MINUS (LOG SERVO113))) (!XATTRIB SER
VO111 (QUOTIENT (EXPT (QUOTIENT 1 SERVO113) 2) 2)))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " INT32
"???????????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT33
"???????????????????? VOYONS SI " "INTEGRATION DU PRODUIT PAR PARTIE"
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (MINUS (LOG SERVO113)) (SERVO113)) (VUS SERVO113 (QUOTIENT 1 SERVO113)
)) (CHOIX 2 (SI (OU ((!%OP (QUOTIENT 1 SERVO113)) EST EXPT) (TRIGO (QUOTIENT 1 SERVO113))) (CALCUL (INTEGRERPARPART
IE (TIMES (MINUS (LOG SERVO113)) (QUOTIENT 1 SERVO113)) SERVO113) SERVO111)) (SI ((!%OP (QUOTIENT 1 SERVO113)) EST
LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (QUOTIENT 1 SERVO113) (MINUS (LOG SERVO113))) SERVO113) SERVO111))))
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (QUOTIENT 1 SERVO113) (SERVO113)) (VUS SERVO113 (MINUS (LOG SERVO113))
)) (CHOIX 2 (SI (OU ((!%OP (MINUS (LOG SERVO113))) EST EXPT) (TRIGO (MINUS (LOG SERVO113)))) (CALCUL (INTEGRERPARPA
RTIE (TIMES (MINUS (LOG SERVO113)) (QUOTIENT 1 SERVO113)) SERVO113) SERVO111)) (SI ((!%OP (QUOTIENT 1 SERVO113)) ES
T LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (QUOTIENT 1 SERVO113) (MINUS (LOG SERVO113))) SERVO113) SERVO111))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"###### EXPLICATION :" "ECHEC SUR PRODUIT PAR PARTIES"
"J'ENVISAGE LE PLAN " INT(34)
"???????????????????? VOYONS SI " "INT(-F*G,X) EST -INT(F*G,X) "
" "
".... PROBLEME A RESOUDRE " (!%CREER R1)
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TIMES (LOG SERVO113) (QUOTIENT 1 SERVO113)) SERVO113) (!XVAL R1))

"J'ENVISAGE LE PLAN " INT(35)
"???????????????????? VOYONS SI " "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (SI ((!%DERIV (LOG SERVO113) SERVO113) EST (QUOTIENT 1 SERVO113)) (!XATTRIB SERVO118 (Q
UOTIENT (EXPT (LOG SERVO113) 2) 2)))
*** SERVO118 DECLARED FLUID
"LE PLAN " (36) " A PERMIS DE CONCLURE  "
"###### JUSTIFICATION :" "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (!XATTRIB SERVO111 (!%SIMPLIF (MINUS (!XVVAL R1))))
" "
".... PROBLEME A RESOUDRE " (!%LIBERER R1)
"LE PLAN " (34) " A PERMIS DE CONCLURE  "
"###### JUSTIFICATION :" "INT(-F*G,X) EST -INT(F*G,X) "
"LE PLAN " (39) " A PERMIS DE CONCLURE  "
"###### JUSTIFICATION :" "QUOTIENT U/V RAMENE A U*(1/V)"
" "
".... PROBLEME A RESOUDRE " (!%MONTRER R4)
R4  SERVO111  (QUOTIENT (MINUS (EXPT (LOG SERVO113) 2)) 2)

( - LOG(SERVO113) )/2

".... PROBLEME A RESOUDRE " (!XATTRIB SERVO107 (!%REMPL (!XVAL R6) (!XVVAL R5) (!XVVAL R4)))
*** SERVO107 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (!%LIBERER (R1 R2 R3 R4 R5 R6))
"LE PLAN " (41) " A PERMIS DE CONCLURE  "
"###### JUSTIFICATION :" "CHG VARIABLE FONCTION TRIGONOMETRIQUE"
" "
".... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%SIMPLIF (DIFFERENCE (TIMES (LOG (COS X)) (!XVVAL VIPP)) (!XVVAL R1))))

*** RR DECLARED FLUID
```

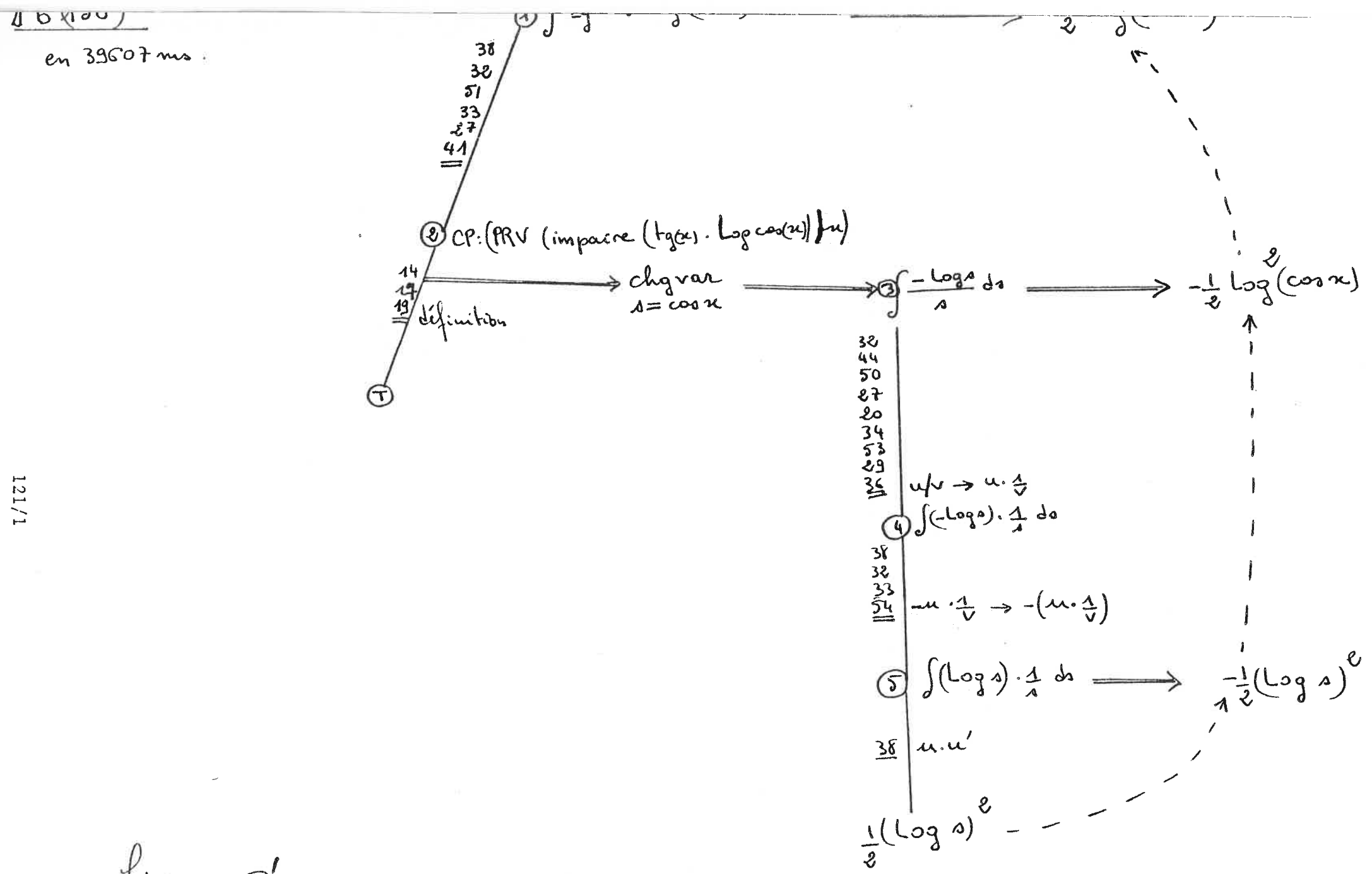
```
"..... PROBLEME A RESOUDRE " (!%LIBERER R1)
"LE PLAN " (6) " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "INTEGRATION PAR PARTIES"
"LE PLAN " (31) " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "PRODUIT INTEGRE PAR PARTIES"
" "
"..... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

                    2
( - LOG(COS(X)) )/2
TEMPS EN MS :
TIME: 66183 MS
NIL

%RESOUD(157,159,"T",NIL)$

%RESOUD(160,162,"T",NIL)$

%RESOUD(163,165,"T",NIL)$

RESOUD(166,169,"T",NIL)$
*******************************
** PB(166) ** =====> (SUC 2 (CALCUL (PRIM (QUOTIENT 1 (PLUS (EXPT (SIN U) 2) (EXPT X 2))) X) RR) (%ECRIRE (%ALGEB R
R)))
*******************************


TIME: 7 MS
" "
"..... PROBLEME A RESOUDRE " (CALCUL (PRIM (QUOTIENT 1 (PLUS (EXPT (SIN U) 2) (EXPT X 2))) X) RR)

"J'ENVISAGE LE PLAN " INT44
"???????????????????? VOYONS SI " "FRACTION AVEC DENOMINATEUR CONSTANT"
"J'ENVISAGE LE PLAN " INT52
"???????????????????? VOYONS SI " "INT(SIN(A*X+B),X)  "
"J'ENVISAGE LE PLAN " INT28
"???????????????????? VOYONS SI " "MONOME X**N INTEGRE / X"
"J'ENVISAGE LE PLAN " INT50
"???????????????????? VOYONS SI " "INT(C/F(X),X) EST C*INT(1/F(X),X"
"J'ENVISAGE LE PLAN " INT27
"???????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " INT39
"???????????????????? VOYONS SI " "INT PAR PARTIE A PARTIR DE 1/(A+B)"
"J'ENVISAGE LE PLAN " INT43
"???????????????????? VOYONS SI " "FORME SIN(X) ** N "
"J'ENVISAGE LE PLAN " INT20
"???????????????????? VOYONS SI " "RESULTAT USUEL "
J'AI PU APPLIQUER 49
"LE PLAN " 20 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "RESULTAT USUEL "
" "
"..... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

ATAN(X/SIN(U))/SIN(U)
TEMPS EN MS :
TIME: 10960 MS
NIL
*******************************
** PB(167) ** =====> (SUC 2 (CALCUL (PRIM (QUOTIENT 1 (RAC2 (PLUS (EXPT (COS FI) 2) (EXPT X 2)))) X) RR) (%ECRIRE (
%ALGEB RR)))
*******************************
```

$$\int \frac{1}{\sin^2 u + x^2}\, dx$$

choisi mais pour commencer le moteur essaie de prouver que (SIN x) . Log (COS x) est paire en x et que (COS x) est impaire en x. SIN (x) n'étant pas paire, mais impaire, il tente de prouver que Log (COS x) est impaire. Il échoue et revient sur son choix malheureux en tentant de prouver que le numérateur SIN x Log (COS x) est une fonction impaire et que le dénominateur (COS x) est une fonction paire. Les étapes 10, 11, 12 permettent d'aboutir sans difficulté dans cette voie. La condition problème du noeud 8 est donc vraie et l'on envisage le changement de variable s = COS x dans l'intégrale. Il est alors nécessaire de calculer (noeud 13) l'intégrale $\int(-\text{Log } s)/s \; ds$. Le changement (-u)/v = (-u) * (1/v) est effectué par le plan (36), le signe - est sorti par le plan (54) et le plan (38) à l'étape 15 permet de reconnaître une intégrale $\int u.u'dx$.
Les notations intermédiaires utilisées permettent de remonter les valeurs en particulier de reprendre en compte le changement de variable.

Remarques. Ce calcul n'est certes pas parfait : Il ne représente pas la meilleure solution pour calculer cette intégrale. La solution obtenue permet de voir ce que nous entendons par calcul raisonné, justifié à chaque étape, avec des interactions entre preuve et calcul algébrique. Le cheminement est typique de ce que pourrait rendre un étudiant sur un tel problème. On peut évidemment regretter que CAMELIA n'ait pas tenté d'emblée le plan (41).

Les figures 6' et 7' donnent une autre solution beaucoup plus élégante du même problème.



121/1

figure 6'

Fig- 7-1

```
----------INT74 SUR SY1 ----------
RESOUD(156,156,"T",NIL)$
*****************************
** PB(156) ** =====> (SUC 2 (CALCUL (PRIM (TIMES (TG X) (LOG (COS X))) X) RR) (%ECRIRE (%ALGEB RR)))
*****************************


TIME: 3792 MS
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TIMES (TG X) (LOG (COS X))) X) RR)

"J'ENVISAGE LE PLAN " INT38
"??????????????????? VOYONS SI " "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (SI ((I%DERIV (TG X) X) EST (LOG (COS X))) (I%ATTRIB RR (QUOTIENT (EXPT (TG X) 2) 2)))

" "
".... PROBLEME A RESOUDRE " (SI ((I%DERIV (LOG (COS X)) X) EST (TG X)) (I%ATTRIB RR (QUOTIENT (EXPT (LOG (COS X))
2) 2)))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " INT32
"??????????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT51
"??????????????????? VOYONS SI " "INT(COS(A*X+B),X)  "
"J'ENVISAGE LE PLAN " INT33
"??????????????????? VOYONS SI " "INTEGRATION DU PRODUIT PAR PARTIE"
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (TG X) (X)) (VUS X (LOG (COS X)))) (CHOIX 2 (SI (OU ((I%OP (LOG (COS X
))) EST EXPT) (TRIGO (LOG (COS X)))) (CALCUL (INTEGRERPARPARTIE (TIMES (TG X) (LOG (COS X))) X) RR)) (SI ((I%OP (LO
G (COS X))) EST LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (LOG (COS X)) (TG X)) X) RR))))
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (LOG (COS X)) (X)) (VUS X (TG X))) (CHOIX 2 (SI (OU ((I%OP (TG X)) EST
 EXPT) (TRIGO (TG X))) (CALCUL (INTEGRERPARPARTIE (TIMES (TG X) (LOG (COS X))) X) RR)) (SI ((I%OP (LOG (COS X))) ES
T LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (LOG (COS X)) (TG X)) X) RR))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"##### EXPLICATION :" "ECHEC SUR PRODUIT PAR PARTIES"
"J'ENVISAGE LE PLAN " INT27
"??????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " INT41
"??????????????????? VOYONS SI " "CHG VARIABLE FONCTION TRIGONOMETRIQUE"
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2 R3 R4 R5 R6))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R6) (I%NOUVAR))
*** SERVO104 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (TIMES (TG X) (LOG (COS X))))
*** SERVO099 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (CHOIX 3 (SI (IMPAIRE (TIMES (TG X) (LOG (COS X))) X) (SUC 2 (I%ATTRIB (I%VAL R5) (COS
X)) (I%ATTRIB (I%VAL R3) (I%CHGVARINT (TIMES (TG X) (LOG (COS X))) X (I%VAL R6) (COS X)))))) (SUC 3 (I%ATTRIB (I%VAL
 R2) (I%SIMPLIF (I%REMPL X (DIFFERENCE PI X) (I%VVAL R1)))) (I%MONTRER R2) (SI ((I%SIMPLIF (PLUS (I%VVAL R1) (I%VVA
L R2))) EST 0) (SUC 2 (I%ATTRIB (I%VAL R5) (SIN X)) (I%ATTRIB (I%VAL R3) (I%CHGVARINT (TIMES (TG X) (LOG (COS X)))
X (I%VAL R6) (SIN X)))))) (SUC 3 (I%ATTRIB (I%VAL R2) (I%SIMPLIF (I%REMPL X (PLUS PI X) (I%VVAL R1)))) (I%MONTRER R
2) (SI ((I%VVAL R2) EST (I%VVAL R1)) (SUC 2 (I%ATTRIB (I%VAL R5) (TG X)) (I%ATTRIB (I%VAL R3) (I%CHGVARINT (TIMES (
TG X) (LOG (COS X))) X (I%VAL R6) (TG X)))))))
" "
".... PROBLEME A RESOUDRE " (SI (IMPAIRE (TIMES (TG X) (LOG (COS X))) X) (SUC 2 (I%ATTRIB (I%VAL R5) (COS X)) (I%AT
TRIB (I%VAL R3) (I%CHGVARINT (TIMES (TG X) (LOG (COS X))) X (I%VAL R6) (COS X)))))

"CONDITION PROBLEME ===> " (IMPAIRE (TIMES (TG X) (LOG (COS X))) X)

"J'ENVISAGE LE PLAN " IMPAIRE14
"??????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR"
```

121/2

fig. 7-8

```
"??????????????????? VOYONS SI " "TRIGO COMPOSEE AVEC FCT IMPAIRE"
"J'ENVISAGE LE PLAN " IMPAIRE(19)
"??????????????????? VOYONS SI " "DEFINITION D'UNE FCT IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (I%CREER R1)
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%SIMPLIF (I%REMPL X (MINUS X) (TIMES (TG X) (LOG (COS X))))))
*** SERVO106 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R1)
R1  SERVO106  (MINUS (TIMES (LOG (COS X)) (TG X)))

 - LOG(COS(X))*TG(X)
" "
".... PROBLEME A RESOUDRE " (SI ((I%SIMPLIF (PLUS (I%VVAL R1) (TIMES (TG X) (LOG (COS X))))) EST O) T)
" "
".... PROBLEME A RESOUDRE " (I%LIBERER R1)
"LE PLAN " (19) " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEFINITION D'UNE FCT IMPAIRE"
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R5) (COS X))
*** SERVO103 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R3) (I%CHGVARINT (TIMES (TG X) (LOG (COS X))) X (I%VAL R6) (COS X)))
*** SERVO101 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R3) (I%SIMPLIF (I%VVAL R3)))
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R3)
R3  SERVO101  (QUOTIENT (MINUS (LOG SERVO104)) SERVO104)

( - LOG(SERVO104))/SERVO104
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (I%VVAL R3) (I%VAL R6)) (I%VAL R4))

"J'ENVISAGE LE PLAN " INT32
"??????????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT44
"??????????????????? VOYONS SI " "FRACTION AVEC DENOMINATEUR CONSTANT"
"J'ENVISAGE LE PLAN " INT50
"??????????????????? VOYONS SI " "INT(C/F(X),X) EST C*INT(1/F(X),X"
"J'ENVISAGE LE PLAN " INT27
"??????????????????? VOYONS SI " "FONCTION CSTE PAR RAPPORT A X "
"J'ENVISAGE LE PLAN " INT20
"??????????????????? VOYONS SI " "RESULTAT USUEL "
*** SERVO102 DECLARED FLUID
"##### EXPLICATION :" "JE NE SAIS PAS TOUT PAR COEUR "
"J'ENVISAGE LE PLAN " PLAN34
"??????????????????? VOYONS SI " "FRACTION RATIONNELLE PAR RISCH "
"##### EXPLICATION :" "PAS SU INTEGRER LA FRACTION"
"J'ENVISAGE LE PLAN " INT53
"??????????????????? VOYONS SI " "INT(-F,X) EST -INT(F,X) "
"J'ENVISAGE LE PLAN " INT29
"??????????????????? VOYONS SI " "INTEGRATION D'UN MONOME "
" "
".... PROBLEME A RESOUDRE " (I%CREER R1)
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%MONOMEP (QUOTIENT (MINUS (LOG SERVO104)) SERVO104) (SERVO104)))

*** SERVO107 DECLARED FLUID
"******ECHEC SUR " (I%ATTRIB (I%VAL R1) (I%MONOMEP (QUOTIENT (MINUS (LOG SERVO104)) SERVO104) (SERVO104)))
"J'ENVISAGE LE PLAN " INT36
"??????????????????? VOYONS SI " "QUOTIENT U/V RAMENE A U*(1/V)"

"J'ENVISAGE LE PLAN " INT38
```

```
"??????????????????? VOYONS SI " "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (SI ((I%DERIV (MINUS (LOG SERVO104)) SERVO104) EST (QUOTIENT 1 SERVO104)) (I%ATTRIB SER
VO102 (QUOTIENT (EXPT (MINUS (LOG SERVO104)) 2) 2)))
" "
".... PROBLEME A RESOUDRE " (SI ((I%DERIV (QUOTIENT 1 SERVO104) SERVO104) EST (MINUS (LOG SERVO104))) (I%ATTRIB SER
VO102 (QUOTIENT (EXPT (QUOTIENT 1 SERVO104) 2) 2)))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " INT32
"??????????????????? VOYONS SI " "INTEGRATION DU LOG PAR PARTIES"
"J'ENVISAGE LE PLAN " INT33
"??????????????????? VOYONS SI " "INTEGRATION DU PRODUIT PAR PARTIE"
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (MINUS (LOG SERVO104)) (SERVO104)) (VUS SERVO104 (QUOTIENT 1 SERVO104)
)) (CHOIX 2 (SI (OU ((I%OP (QUOTIENT 1 SERVO104)) EST EXPT) (TRIGO (QUOTIENT 1 SERVO104))) (CALCUL (INTEGRERPARPART
IE (TIMES (MINUS (LOG SERVO104)) (QUOTIENT 1 SERVO104)) SERVO104) SERVO102)) (SI ((I%OP (QUOTIENT 1 SERVO104)) EST
LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (QUOTIENT 1 SERVO104) (MINUS (LOG SERVO104))) SERVO104) SERVO102))))
" "
".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (QUOTIENT 1 SERVO104) (SERVO104)) (VUS SERVO104 (MINUS (LOG SERVO104))
)) (CHOIX 2 (SI (OU ((I%OP (MINUS (LOG SERVO104))) EST EXPT) (TRIGO (MINUS (LOG SERVO104)))) (CALCUL (INTEGRERPARPA
RTIE (TIMES (MINUS (LOG SERVO104)) (QUOTIENT 1 SERVO104)) SERVO104) SERVO102)) (SI ((I%OP (QUOTIENT 1 SERVO104)) ES
T LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (QUOTIENT 1 SERVO104) (MINUS (LOG SERVO104))) SERVO104) SERVO102))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"##### EXPLICATION :" "ECHEC SUR PRODUIT PAR PARTIES"
"J'ENVISAGE LE PLAN " INT54
"??????????????????? VOYONS SI " "INT(-F*G,X) EST -INT(F*G,X) "
" "
".... PROBLEME A RESOUDRE " (I%CREER R1)
" "
".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TIMES (LOG SERVO104) (QUOTIENT 1 SERVO104)) SERVO104) (I%VAL R1))

"J'ENVISAGE LE PLAN " INT38
"??????????????????? VOYONS SI " "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (SI ((I%DERIV (LOG SERVO104) SERVO104) EST (QUOTIENT 1 SERVO104)) (I%ATTRIB SERVO108 (Q
UOTIENT (EXPT (LOG SERVO104) 2) 2)))
*** SERVO108 DECLARED FLUID
"LE PLAN " 38 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "INTEGRATION DE U*U'  "
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO102 (I%SIMPLIF (MINUS (I%VVAL R1))))
" "
".... PROBLEME A RESOUDRE " (I%LIBERER R1)
"LE PLAN " 54 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "INT(-F*G,X) EST -INT(F*G,X) "
"LE PLAN " 36 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "QUOTIENT U/V RAMENE A U*(1/V)"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R4)
R4  SERVO102  (QUOTIENT (MINUS (EXPT (LOG SERVO104) 2)) 2)

                    2
( - LOG(SERVO104) )/2
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB RR (I%REMPL (I%VAL R6) (I%VVAL R5) (I%VVAL R4)))
*** RR DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2 R3 R4 R5 R6))
"LE PLAN " 41 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "CHG VARIABLE FONCTION TRIGONOMETRIQUE"
" "
".... PROBLEME A RESOUDRE " (I%ECRIRE (I%ALGEB RR))

                    2
( - LOG(COS(X)) )/2
```

IV.3.3.4. Exemple 4

* Soit (PB 200) à évaluer l'expression $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1} C(n,k)$

(Ce problème et les connaissances nécessaires sont extraits du travail de M. BARON (BARON 81-82)).

* Ce problème est posé sous la forme :

```
(SUC 2
   (EVALUER (SIG (TIMES (EXPT (MINUS 1) KK)
                        (TIMES (QUOTIENT 1 (PLUS KK 1))
                               (VCNP NN KK)))
                 KK  0  NN)  RR)
    (!%ECRIRE (!%ALGEB RR)));
```

* Cette écriture signifie que le problème comporte deux étapes :

  - l'évaluation d'une expression, le résultat étant mis dans RR.
  - l'écriture algébrique du résultat trouvé.

* L'expression à évaluer est un SIG à quatre arguments : le terme (en écriture préfixée), l'indice muet, la borne basse et la borne haute du sigma.

* La figure 8 donne le graphe de la solution élaborée et la figure 9 donne le listing complet de la solution.

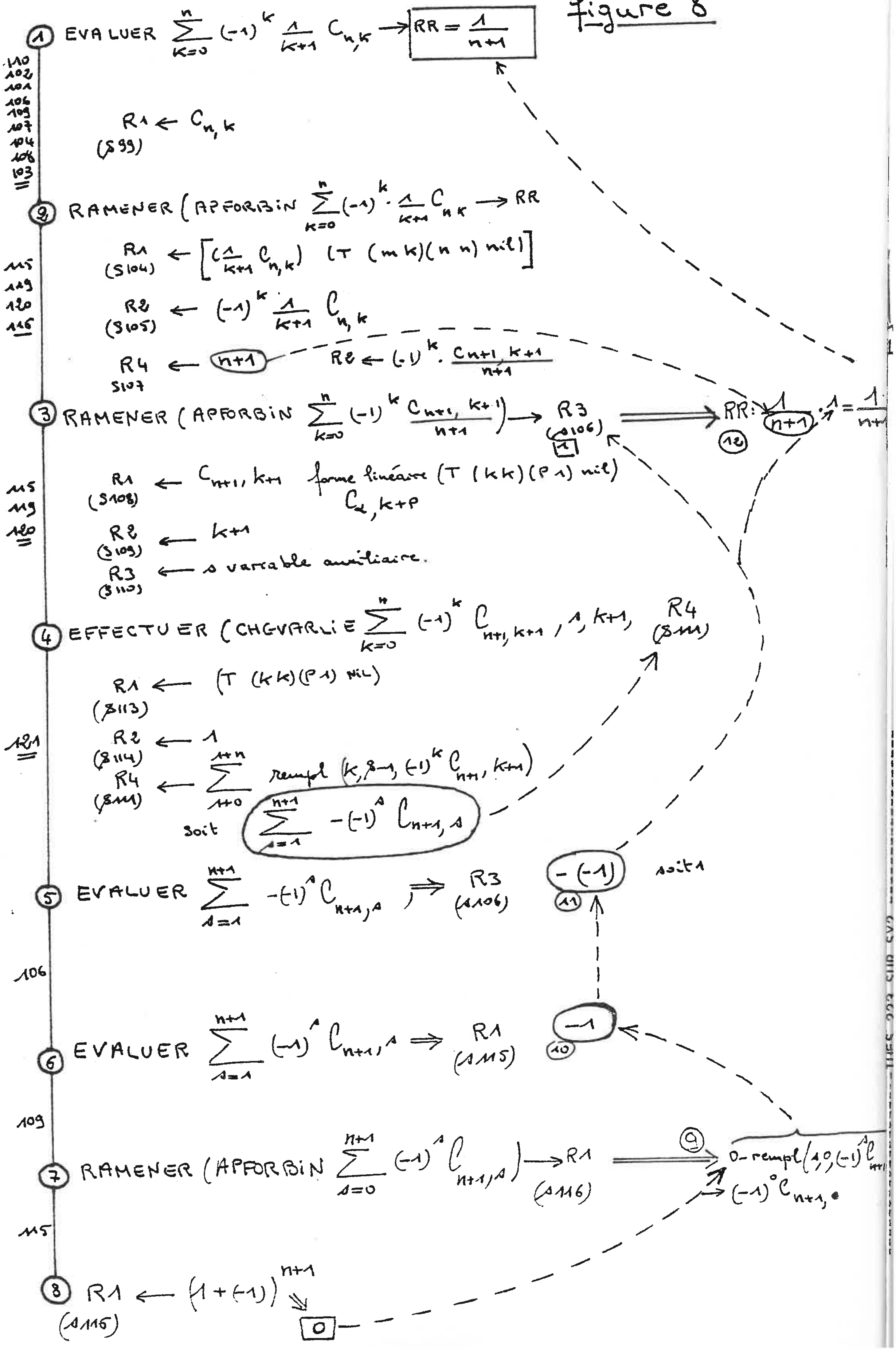
* Le moteur démarre avec l'étape EVALUER, tente quelques plans avant d'essayer le plan 103 qui s'avère être intéressant. La démarche préconisée dans ce plan (méthode 3) est la suivante :

> Pour évaluer SIG exp k k1 k2
> Si il y a un seul terme de la forme C(i,j) dans exp
> Si i ne contient pas k
> et j contient k
> alors se ramener à l'application de la formule du binôme de NEWTON.

* Nous parvenons ainsi au noeud 2 :
(RAMENER (APFORBIN (SIG.....)) RR)

La base de connaissanses dispose de plans pour ramener une expression à une forme sur laquelle il est possible d'appliquer la formule du binôme. Il trouve en particulier la méthode 6 (plan 116) qui peut s'exprimer ainsi :

> Pour se ramener à l'application de la formule du binôme lorsqu'on évalue (SIG exp k k1 k2)
> Si exp contient un terme de la forme
> $exp1 = \frac{1}{m+1} C(n,m)$
> alors remplacer exp1 par $\frac{1}{n+1} . C(n+1, m+1)$
> et se ramener à l'application de la formule du binôme en mettant 1/(n+1) en facteur devant le SIG.



1 EVALUER $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1} C_{n,k}$ → RR = $\frac{1}{n+1}$
R1 ← $C_{n,k}$ (S99)
2 RAMENER (APFORBIN $\sum_{k=0}^{n} (-1)^k \cdot \frac{1}{k+1} C_{n,k}$ → RR
R1 (S104) ← [($\frac{1}{k+1} C_{n,k}$) (T (m k)(n n) nil)]
R2 (S105) ← $(-1)^k \frac{1}{k+1} C_{n,k}$
R4 S107 ← n+1
R2 ← $(-1)^k \cdot \frac{C_{n+1,k+1}}{n+1}$
3 RAMENER (APFORBIN $\sum_{k=0}^{n} (-1)^k \frac{C_{n+1,k+1}}{n+1}$) → R3 (S106)
RR: $\frac{1}{n+1} \cdot 1 = \frac{1}{n+1}$
R1 (S108) ← $C_{n+1,k+1}$ forme linéaire (T (k k)(P 1) nil) $C_{\alpha,k+p}$
R2 (S109) ← k+1
R3 (S110) ← s variable auxiliaire.
4 EFFECTUER (CHGVARLIE $\sum_{k=0}^{n} (-1)^k C_{n+1,k+1}$, s, k+1, R4 (S111)
R1 (S113) ← (T (k k)(P 1) NIL)
R2 (S114) ← 1
R4 (S111) ← $\sum_{s=0}^{s+n}$ rempl (k, s-1, $(-1)^k C_{n+1,k+1}$)
soit $\sum_{s=1}^{n+1} -(-1)^s C_{n+1,s}$
5 EVALUER $\sum_{s=1}^{n+1} -(-1)^s C_{n+1,s}$ ⇒ R3 (S106)
$-(-1)$ soit 1
6 EVALUER $\sum_{s=1}^{n+1} (-1)^s C_{n+1,s}$ ⇒ R1 (S115)
-1
7 RAMENER (APFORBIN $\sum_{s=0}^{n+1} (-1)^s C_{n+1,s}$) → R1 (S116)
0 - rempl (s, 0, $(-1)^s C_{n+1,s}$)
$(-1)^0 C_{n+1,0}$
8 R1 ← $(1+(-1))^{n+1}$ (S115)
0

figure 8

```
%CALCUL DE COMBINAISONS $
 LIRCHAP 'COMBINES $
ENTREE DU CHAPITRE SOMMES DE COMBINES
--->FIN DES COMBINES
END$
*
RESOUD(200,200,"T",NIL)$
*******************************
** PB(200) ** =====> (SUC 2 (EVALUER (SIG (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK)))
 KK O NN) RR) (%ECRIRE (%ALGEB RR)))
*******************************


TIME: 3880 MS
" "
(1) ".... PROBLEME A RESOUDRE " (EVALUER (SIG (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK)))
 KK O NN) RR)

"J'ENVISAGE LE PLAN " SIG110
"??????????????????? VOYONS SI " "SIG K*F,O,N = SIG K*F,1,N"
"J'ENVISAGE LE PLAN " SIG102
"??????????????????? VOYONS SI " "SIG A*F = A*SIG F"
" "
".... PROBLEME A RESOUDRE " (SI (NONVUS KK (EXPT (MINUS 1) KK)) (SUC 2 (EVALUER (SIG (TIMES (QUOTIENT 1 (PLUS KK 1)
) (VCNP NN KK)) KK O NN) RR) (I%ATTRIB RR (I%SIMPLIF (TIMES (I%ALGEB RR) (EXPT (MINUS 1) KK))))))
" "
".... PROBLEME A RESOUDRE " (SI (NONVUS KK (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK))) (SUC 2 (EVALUER (SIG (EXP
T (MINUS 1) KK) KK O NN) RR) (I%ATTRIB RR (I%SIMPLIF (TIMES (I%ALGEB RR) (EXPT (MINUS 1) KK))))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " SIG101
"??????????????????? VOYONS SI " "SIG F  K A A"
"J'ENVISAGE LE PLAN " SIG106
"??????????????????? VOYONS SI " "SIG -F K K1 K2 ==>-SIG..."
"J'ENVISAGE LE PLAN " SIG109
"??????????????????? VOYONS SI " "SIG 1 N+1==> SIG O N"
"J'ENVISAGE LE PLAN " SIG107
"??????????????????? VOYONS SI " "SIG F K 1 K2+1==>SIG F K O K2"
"J'ENVISAGE LE PLAN " SIG104
"??????????????????? VOYONS SI " "METHODE 4"
"J'ENVISAGE LE PLAN " SIG108
"??????????????????? VOYONS SI " "SIG 1 N+1==> SIG O N"
"J'ENVISAGE LE PLAN " SIG103
"??????????????????? VOYONS SI " "METHODE 3 "
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%TETE (I%TERMES VCNP (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT
 1 (PLUS KK 1)) (VCNP NN KK))))))
*** SERVO(99) DECLARED FLUID
" "
(2) ".... PROBLEME A RESOUDRE " (CHOIX 2 (SI (ET (NONVUS KK (I%ARG1 (I%VVAL R1))) (VUS KK (I%ARG2 (I%VVAL R1)))) (RAMEN
ER (APFORBIN (SIG (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK))) KK O NN)) RR)) (SI (ET
(VUS KK (I%ARG1 (I%VVAL R1))) (NONVUS KK (I%ARG2 (I%VVAL R1)))) (RAMENER (APFORCOL (SIG (TIMES (EXPT (MINUS 1) KK)
(TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK))) KK O NN)) RR)))
" "
".... PROBLEME A RESOUDRE " (SI (ET (NONVUS KK (I%ARG1 (I%VVAL R1))) (VUS KK (I%ARG2 (I%VVAL R1)))) (RAMENER (APFOR
BIN (SIG (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK))) KK O NN)) RR))

"J'ENVISAGE LE PLAN " SIG115
"??????????????????? VOYONS SI " "METHODE 5"
*** RR DECLARED FLUID
"J'ENVISAGE LE PLAN " SIG119
"??????????????????? VOYONS SI " "METHODE 9"
"J'ENVISAGE LE PLAN " SIG120
```

solution page 122/2

Evaluer $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1} C_{(n,k)}$

(Serv 99)

fig. 3-6

```
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2 R3 R4))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%TERMES VCNP (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT 1 (PLUS
 KK 1)) (VCNP NN KK)))))
*** SERVO100 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (CHOIX 4 (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1101) (KK) (I%ARG2 (I%TETE
(I%VVAL R1))))) (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1102) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))) (I%AT
TRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1103) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))) (I%ATTRIB (I%VAL R2) (I%VU
FORMTERM (I%ALGEB FORMLIN1104) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1101) (KK) (I%ARG2 (I%TETE (I%VVAL R
1)))))
*** SERVO101 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1102) (KK) (I%ARG2 (I%TETE (I%VVAL R
1)))))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1103) (KK) (I%ARG2 (I%TETE (I%VVAL R
1)))))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1104) (KK) (I%ARG2 (I%TETE (I%VVAL R
1)))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"******ECHEC SUR " (CHOIX 4 (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1101) (KK) (I%ARG2 (I%TETE (I%VVAL R
1))))) (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1102) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))) (I%ATTRIB (I%V
AL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1103) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))) (I%ATTRIB (I%VAL R2) (I%VUFORMTERM
(I%ALGEB FORMLIN1104) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))))
"J'ENVISAGE LE PLAN " SIG116 *
"??????????????????? VOYONS SI " "METHODE 6"
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2 R3 R4))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%VUFORMTERM (I%ALGEB FCNP6) (NN) (TIMES (EXPT (MINUS 1) KK) (TIM
ES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK)))))
*** SERVO104 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (TIMES (EXPT (MINUS 1) KK) (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP N
N KK))))
*** SERVO105 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R4) (I%APPLISUBST (I%TQT (I%VVAL R1)) (PLUS N 1)))
*** SERVO107 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R4)
R4  SERVO107  (PLUS NN 1)

NN + 1
" "
".... PROBLEME A RESOUDRE " (POURTOUT OC (I%VVAL R1) (I%ATTRIB (I%VAL R2) (I%REMPL (I%TETE OC) (I%APPLISUBST (I%TQ
OC) (QUOTIENT (VCNP (PLUS N 1) (PLUS M 1)) (PLUS N 1))) (I%VVAL R2))))
" "
".... PROBLEME A RESOUDRE " (RAMENER (APFORBIN (SIG (I%SIMPLIF (TIMES (I%VVAL R2) (I%VVAL R4))) KK O NN)) (I%VAL R3)
))

"J'ENVISAGE LE PLAN " SIG115
"??????????????????? VOYONS SI " "METHODE 5"
"J'ENVISAGE LE PLAN " SIG119
"??????????????????? VOYONS SI " "METHODE 9"
"J'ENVISAGE LE PLAN " SIG120 *
"??????????????????? VOYONS SI " "METHODE 10"
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2 R3 R4))
" "
```

122/3

③

```
" "
".... PROBLEME A RESOUDRE " (CHOIX 4 (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1101) (KK) (I%ARG2 (I%TETE
(I%VVAL R1))))) (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1102) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))) (I%AT
TRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1103) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))) (I%ATTRIB (I%VAL R2) (I%VU
FORMTERM (I%ALGEB FORMLIN1104) (KK) (I%ARG2 (I%TETE (I%VVAL R1))))))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%VUFORMTERM (I%ALGEB FORMLIN1101) (KK) (I%ARG2 (I%TETE (I%VVAL R
1)))))
*** SERVO109 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%TT (I%VVAL R2)))
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R2)
R2  SERVO109  (PLUS KK 1)

KK + 1
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R3) (I%NOUVAR))
*** SERVO110 DECLARED FLUID
" "
(4) ".... PROBLEME A RESOUDRE " (EFFECTUER (CHGVARLIE (SIG (TIMES (EXPT (MINUS 1) KK) (VCNP (PLUS NN 1) (PLUS KK 1))) K
K 0 NN) (I%VAL R3) (I%VVAL R2)) (I%VAL R4))

"J'ENVISAGE LE PLAN " SIG121
"??????????????????? VOYONS SI " "METHODE 11"
" "
".... PROBLEME A RESOUDRE " (SUC 5 (I%CREER (R1 R2)) (I%ATTRIB (I%VAL R1) (I%FORMEXPLICITE (PLUS KK 1) (KK) (I%ALGE
B FORMLIN1101))) (I%ATTRIB (I%VAL R2) (I%ARG2 (PLUS KK 1))) (I%ATTRIB SERVO111 (SIG (I%SIMPLIF (I%REMPL KK (DIFFERE
NCE SERVO110 (I%VVAL R2)) (TIMES (EXPT (MINUS 1) KK) (VCNP (PLUS NN 1) (PLUS KK 1))))) SERVO110 (I%SIMPLIF (PLUS (I
%VVAL R2) 0)) (I%SIMPLIF (PLUS (I%VVAL R2) NN)))) (I%LIBERER (R1 R2)))
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%FORMEXPLICITE (PLUS KK 1) (KK) (I%ALGEB FORMLIN1101)))
*** SERVO113 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%ARG2 (PLUS KK 1)))
*** SERVO114 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO111 (SIG (I%SIMPLIF (I%REMPL KK (DIFFERENCE SERVO110 (I%VVAL R2)) (TIMES
 (EXPT (MINUS 1) KK) (VCNP (PLUS NN 1) (PLUS KK 1))))) SERVO110 (I%SIMPLIF (PLUS (I%VVAL R2) 0)) (I%SIMPLIF (PLUS (
I%VVAL R2) NN))))
*** SERVO111 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2))
"LE PLAN " 121 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "METHODE 11"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R4)
R4  SERVO111  (SIG (MINUS (TIMES (EXPT (MINUS 1) SERVO110) (VCNP (PLUS NN 1) SERVO110))) SERVO110 1 (PLUS NN 1))

(5)
              SERVO110
SIG( - ( - 1)         *VCNP(NN + 1,SERVO110),SERVO110,1,NN + 1)
" "
".... PROBLEME A RESOUDRE " (EVALUER (I%VVAL R4) SERVO106)

"J'ENVISAGE LE PLAN " SIG106
"??????????????????? VOYONS SI " "SIG -F K K1 K2 ==>-SIG..."
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1))
" "
(6) ".... PROBLEME A RESOUDRE " (EVALUER (SIG (TIMES (EXPT (MINUS 1) SERVO110) (VCNP (PLUS NN 1) SERVO110)) SERVO110 1
(PLUS NN 1)) (I%VAL R1))
```

Fig 3-4

```
"J'ENVISAGE LE PLAN " SIG110
"???????????????????? VOYONS SI " "SIG K*F,O,N = SIG K*F,1,N"
"J'ENVISAGE LE PLAN " SIG102
"???????????????????? VOYONS SI " "SIG A*F = A*SIG F"
" "
".... PROBLEME A RESOUDRE " (SI (NONVUS SERVO110 (EXPT (MINUS 1) SERVO110)) (SUC 2 (EVALUER (SIG (VCNP (PLUS NN 1)
SERVO110) SERVO110 1 (PLUS NN 1)) SERVO115) (I%ATTRIB SERVO115 (I%SIMPLIF (TIMES (I%ALGEB SERVO115) (EXPT (MINUS 1)
 SERVO110))))))
" "
".... PROBLEME A RESOUDRE " (SI (NONVUS SERVO110 (VCNP (PLUS NN 1) SERVO110)) (SUC 2 (EVALUER (SIG (EXPT (MINUS 1)
SERVO110) SERVO110 1 (PLUS NN 1)) SERVO115) (I%ATTRIB SERVO115 (I%SIMPLIF (TIMES (I%ALGEB SERVO115) (EXPT (MINUS 1)
 SERVO110))))))
ECHEC SUR TOUTES LES ETAPES POSSIBLES
"J'ENVISAGE LE PLAN " SIG101
"???????????????????? VOYONS SI " "SIG F  K A A"
"J'ENVISAGE LE PLAN " SIG106
"???????????????????? VOYONS SI " "SIG -F K K1 K2 ==>-SIG..."
"J'ENVISAGE LE PLAN " SIG109
"???????????????????? VOYONS SI " "SIG 1 N+1==> SIG O N"
" "
".... PROBLEME A RESOUDRE " (I%CREER R1)
" "
(7) ".... PROBLEME A RESOUDRE " (RAMENER (APFORBIN (SIG (TIMES (EXPT (MINUS 1) SERVO110) (VCNP (PLUS NN 1) SERVO110)) S
ERVO110 O (PLUS NN 1))) (I%VAL R1))

"J'ENVISAGE LE PLAN " SIG115
"???????????????????? VOYONS SI " "METHODE 5"
(8) J'AI PU APPLIQUER 13
*** SERVO116 DECLARED FLUID
"LE PLAN " 115 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "METHODE 5"
" "
(9) (10) ".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO115 (I%SIMPLIF (DIFFERENCE (I%VVAL R1) (I%REMPL SERVO110 O (TIMES (EXPT
(MINUS 1) SERVO110) (VCNP (PLUS NN 1) SERVO110))))))
*** SERVO115 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER R1)
"LE PLAN " 109 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "SIG 1 N+1==> SIG O N"
" "
(11) ".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO106 (I%SIMPLIF (MINUS (I%VVAL R1))))
*** SERVO106 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1))
"LE PLAN " 106 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "SIG -F K K1 K2 ==>-SIG..."
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2 R3 R4))
"LE PLAN " 120 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "METHODE 10"
" "
(12) ".... PROBLEME A RESOUDRE " (I%ATTRIB RR (TIMES (QUOTIENT 1 (I%VVAL R4)) (I%VVAL R3)))
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2 R3 R4))
"LE PLAN " 116 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "METHODE 6"
" "
".... PROBLEME A RESOUDRE " (I%LIBERER R1)
"LE PLAN " 103 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "METHODE 3 "
" "
(13) ".... PROBLEME A RESOUDRE " (I%ECRIRE (I%ALGEB RR))
```

$$(NN + 1)^{(-1)}$$

* Le problème du noeud 3 est ainsi obtenu. Le moteur envisage alors le plan 120 (méthode 10). Cette méthode indique que :

> Pour se ramener à l'application de la formule du binome lorsqu'on évalue (SIG exp k k1 k2)
>
> > Si exp contient une combinaison dont le second argument est une forme linéaire de type + k + p
> >
> > > Alors faire un changement de variable muette (liée) en prenant ce deuxième argument pour nouvelle variable. Evaluer ensuite le résultat obtenu.

* Nous arrivons ainsi au noeud 4 qui préconise d'effectuer le changement S = K +1. Le plan 121 effectue ce changement de variable en notant R1 la substitution qui permet d'identifier le 2ème argument de la combinaison à la forme linéaire FORMLIN1101 (k+p).

* Le problème du noeud 5 est posé ; La nouvelle variable s'appelle SERV104. La formule 5 à évaluer est encadrée dans la figure 9.

* L'application du plan 106 permet de sortir le signe - de l'expression sous SIG.

* L'évaluation du noeud 6 ainsi obtenu est effectuée avec le plan 109: La méthode consiste ici à étendre le champ de variation de l'indice muet pour se ramener d'une variation (1,n+1) à une variation (0,n+1). L'écriture est du type
SIG (exp(s),s,1,n+1) = = > SIG (exp(s),s,0,n+1)-exp(0)

* Sur le noeud 7 auquel on arrive, le plan 115 s'applique bien. Ce plan reconnait la formule du binôme comme formule connue (la règle 13 s'applique). Le terme en SIG du problème du noeud 7 se voit donc attribuer la valeur 0

* Il reste à terminer tous les plans commencés : l'essentiel des étapes qui restent assurent la libération des notations et la remontée des valeurs (traits pointillés de la figure 8). Le (-1) qui apparait dans les étapes 9 et 10 vient du terme complémentaire exp(0) de l'étape 7. Le signe - sorti à l'étape 5 se combine avec ce -1 pour remonter le 1 jusqu'à l'étape 3. Le 1/(n+1) mis en facteur par le plan 116 (étape 2)se combine avec la valeur 1 qui remonte comme résultat de l'application de la formule du binôme. Le résultat définitif 1/N+1 est affecté à RR dans l'étape 12. L'étape 13 correspond à l'écriture de ce résultat (deuxième étape du SUC initial).

Notation utilisée pour décrire les solutions

La solution de ce problème nous permet d'introduire la notation que nous utiliserons pour décrire globalement les solutions des différents exercices. Nous notons les solutions de la façon suivante :

(n lorsque le plan numéro n commence.
) lorsqu'un plan termine.

* Cette notation permet de ramener l'expression de la partie utile des graphes solution à l'écriture d'une formule dont le parenthésage permet d'apprécier rapidement la complexité de la démarche suivie.
Pour ce problème nous arrivons à l'écriture suivante :
(103 (116 (120 (121) (106 (109 (115))))))

* Nous ne retenons dans cette notation que les parties utiles du graphe : celles qui amènent au succès. Les essais amenant un échec pourraient être inclus. Il faudrait trouver une typographique particulière (souligné par exemple) pour marquer qu'il s'agit de noeuds n'ayant pas permis d'aboutir. La notation pour le graphe précédent deviendrait :

(110 102 101 106 109 107 104 108 103 (115 119 120 116 (115 119 120 (121) (106 (109 (115)))))

Cette notation plus lourde aurait le mérite de faire apparaître l'effort fait par CAMELIA pour résoudre un problème. La notation que nous utilisons ne fait apparaître que la solution retenue par CAMELIA.

* Pour faciliter la lecture des feuilles de résultats lorsqu'un plan utile commence nous notons dans la marge et lorsqu'un plan utilise s'achève nous notons dans la marge (cf figure 9). Cette technique facilite nettement l'écriture de la formule décrivant la solution.

* Il serait envisageable de faire écrire la formule de description de la solution par le moteur d'inférence : On concevrait facilement un algorithme qui passe du graphe de la solution à l'écriture de la formule correspondante.

Remarques sur cet exemple

* Cet exercice a été résolu par CAMELIA dans un esprit particulier qu'il convient de préciser :

* CAMELIA est un système ESSENTIEL qui n'a pas du tout été envisagé au départ pour résoudre des problèmes particuliers d'évaluation de sommations de formules combinatoire. Les connaissances nécessaires pour résoudre de tels problèmes ont été décrites par M. BARON (BARON 81-82) et il nous a semblé intéressant de voir dans quelles conditions il était possible d'écrire pour CAMELIA, le chapitre de connaissances nécessaires pour travailler dans un domaine nouveau. Cela a représenté pour nous un bon test pour vérifier la souplesse et l'extensibilité du langage d'expression des connaissances, évaluer l'effort de codage nécessaire.

Les observations que nous pouvons faire sont les suivantes :

- Nous n'avons pas codé toutes les connaissances décrites par M. BARON. Ce travail a été limité pour des raisons de temps aux connaissances nécessaires pour résoudre une quinzaine de problèmes mais nous sommes convaincu que les connaissances supplémen-

taires qu'il faudrait ajouter ne posent pas de problème particulier. Les exemples que nous avons pris sont suffisamment significatifs des difficultés du domaine.

- Nous n'avons rencontré aucune difficulté au niveau du langage d'expression des connaissances. Il nous a fallu l'étendre avec trois actions immédiates.
-!%TERME (op,ex) qui sort les termes dont l'opérateur principal est op dans l'expression ex.
-!%VUFORMTERME (forme, para, ex) sort les sous termes vérifiant le pattern formé (avec conditons, paramètres, éventuellement) de l'expression ex. Le résultat est une liste de substitutions décrivant les occurences de forme dans ex.
-!%SOUSTERME(term, ex) qui vérifie si term est un sous terme de ex (résultat booléen).

Ces trois actions d'intérêt général n'ont rien de spécifique au domaine, leur introduction n'a pas posé de problème et nous a convaincu qu'il n'était pas difficile d'ajoindre de nouvelles actions dans le langage d'expression de connaissances.

- Le chapitre de connaissances nécessaire représente 300 lignes de code pour écrire 18 plans et 60 lignes de code pour écrire les règles : formule du binome, simplifications (c(n,n)-> 1, c(n,1)->n, >n,c(n,n-1) -> n,....), formes linéaires + k x + p).

- Le travail a pu être mené a bien en cinq jours à temps complet. Ce temps comprend l'écriture et le test de procédure liées aux 3 actions immédiates ci-dessus. Le travail a pu être fait en trois listings de la première frappe aux premiers résultats ; c'est dire qu'avec un entrainement à écrire dans ce langage, on fait peu de fautes d'écritures et que la mise au point se fait sans trop de problèmes de corrections.

- La partie la plus délicate s'avère être le test de modules de connaissances complexes. - Notons que nous n'avons pas travaillé au niveau méta-connaissance dans ce chapitre (c'est ce qui explique qu'il y a beaucoup d'essais inutiles qui sont tentés).

- Le détail des connaissances de ce chapitre est donné en annexe D5.

IV.3.3.5. <u>Exemple 5</u> :

* Soit Pb (94) à calculer le développement limité de la fonction :

```
                 x**2
        f(x) = (e     -1) - Log (1+x) / x
```

au voisinage de 0 à l'ordre 3.

* La figure 10 donne la solution obtenue par CAMELIA. Cet exemple permet de bien voir comment CAMELIA fait apparaître à l'utilisateur les calculs intermédiaires qu'il fait.

* Le problème est posé ligne 1 et SOLUT aborde le problème du développement proprement dit (ligne 2).

* CAMELIA a vu un opérateur particulier : LOG et essaie quelques plans mettant en oeuvre cet opérateur. Cela ne marche pas. Après avoir tenté des choses simples (polynome ?, monome ?), il aborde ligne 3 le plan DEVLIM90. Le dénominateur de f(x) est effectivement un monome (de degré 1), il commence donc le développement du numérateur à l'ordre 3+1 (soit 4). Ligne 5 il entreprend le développement de la différence qui est au numérateur et lignes 6-7-8 il aborde le développement de la différence (premier terme du numérateur). Les lignes 9 à 15 montrent le développement de e ** x ** 2 comme développement de e ** u(x) avec u(x) qui est vu comme le polynome x ** 2. La ligne 16 fait voir le résultat trouvé. Les lignes 17 et 18 représentent le développement évident du 1 et l'on arrive ligne 20 à voir le développement du premier terme de la différence qui est au numérateur de f(x).

En ligne 21, le développement du LOG commence. Le plan DEVLIM92 (DEVLIM LOG u(x)) échoue car ce plan général n'est tenté que si u(x) n'est pas un polynôme. Le plan 83 est alors choisi (ligne 22). Ce plan aboutit ligne 23 et le résultat trouvé est visible ligne 24. La ligne 25 annonce alors la fin du calcul pour le numérateur. Le résultat obtenu est visible en ligne 26. Il reste à diviser par x pour achever l'éxécution du plan 90 lancé initialement. Le résultat final est obtenu ligne 26.

La notation définie dans l'exemple 4 précédent pour décrire les solutions permet de donner la structure du cheminement par l'écriture :

```
(90   (75   (75   (78(95  (73))    (71))    (83) )  )
                  +___________+
                  Dev e**(x**2)    +___+
                                   Dev 1   +_____+
                                           Dev LOG
            +____________________________+
                  Dev 1er terme
      +_________________________________________+
                       Dev numérateur
```

* <u>Conclusion sur cet exemple</u>

Cet exemple fait apparaître comment CAMELIA utilise les

Dev $\frac{(e^x - 1) - \log(1+x)}{x^2}$ $x=0$

fig. 10-1

```
---------DEV74 SUR SY2 ----------------
 RESOUD(94,94,"T",NIL)$
*******************************
** PB(94) ** =====> (SUC 2 (CALCUL (DEVLIM (QUOTIENT (DIFFERENCE (DIFFERENCE (EXPT E (EXPT X 2)) 1) (LOG (PLUS 1 X)
)) X) X 0 3) RR) (%ECRIRE (%ALGEB RR)))
*******************************

TIME: 19786 MS
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM (QUOTIENT (DIFFERENCE (DIFFERENCE (EXPT E (EXPT X 2)) 1) (LOG (PLUS 1 X
))) X) X 0 3) RR)

"J'ENVISAGE LE PLAN " DEVLIM94
"???????????????? VOYONS SI " "DEV LOG A-LOG B = DEV LOG(A/B)"
"J'ENVISAGE LE PLAN " DEVLIM98
"???????????????? VOYONS SI " "DEVELOPPEMENT DE FRACTION RATIONNELLE"
"J'ENVISAGE LE PLAN " DEVLIM93
"???????????????? VOYONS SI " "DEVLIM LOG(A/B) "
"J'ENVISAGE LE PLAN " DEVLIM73
"???????????????? VOYONS SI " "DEV D'UN POLYNOME:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM72
"???????????????? VOYONS SI " "DEV D'UN MONOME:EVIDENT"
" "
".... PROBLEME A RESOUDRE " (I%CREER R1)
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%MONOMEP (QUOTIENT (DIFFERENCE (DIFFERENCE (EXPT E (EXPT X 2))
1) (LOG (PLUS 1 X))) X) (X)))
*** SERVO104 DECLARED FLUID
"******ECHEC SUR " (I%ATTRIB (I%VAL R1) (I%MONOMEP (QUOTIENT (DIFFERENCE (DIFFERENCE (EXPT E (EXPT X 2)) 1) (LOG (P
LUS 1 X))) X) (X)))
"J'ENVISAGE LE PLAN " DEVLIM90
"???????????????? VOYONS SI " "DEVLIM U(X)/A*X**P "
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%MONOMEP X (X)))
*** SERVO105 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM (DIFFERENCE (DIFFERENCE (EXPT E (EXPT X 2)) 1) (LOG (PLUS 1 X))) X 0 (I
%SIMPLIF (PLUS 3 (I%TQ (I%VVAL R1))))) (I%VAL R2))

"J'ENVISAGE LE PLAN " DEVLIM94
"???????????????? VOYONS SI " "DEV LOG A-LOG B = DEV LOG(A/B)"
"J'ENVISAGE LE PLAN " DEVLIM73
"???????????????? VOYONS SI " "DEV D'UN POLYNOME:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM84
"???????????????? VOYONS SI " "DEVLIM LOG(A-U(X)) "
"J'ENVISAGE LE PLAN " DEVLIM75
"???????????????? VOYONS SI " "DEV LIM D'UNE DIFFERENCE:FACILE"
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2))
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM (DIFFERENCE (EXPT E (EXPT X 2)) 1) X 0 4) (I%VAL R1))

"J'ENVISAGE LE PLAN " DEVLIM73
"???????????????? VOYONS SI " "DEV D'UN POLYNOME:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM75
"???????????????? VOYONS SI " "DEV LIM D'UNE DIFFERENCE:FACILE"
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2))
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM (EXPT E (EXPT X 2)) X 0 4) (I%VAL R1))
```

126/1

dénominateur : degré 1.

↑ on développe le numérateur à l'ordre 3 + (degré du dénominateur)

```
"?????????????????? VOYONS SI " "DEV D'UN POLYNOME:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM72
"?????????????????? VOYONS SI " "DEV D'UN MONOME:EVIDENT"
" "
".... PROBLEME A RESOUDRE " (I%CREER R1)
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%MONOMEP (EXPT E (EXPT X 2)) (X)))
*** SERVO111 DECLARED FLUID
"******ECHEC SUR " (I%ATTRIB (I%VAL R1) (I%MONOMEP (EXPT E (EXPT X 2)) (X)))
"J'ENVISAGE LE PLAN " DEVLIM78
"?????????????????? VOYONS SI " "DEV E**U(X) "
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1 R2 R3 R4))
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM (EXPT X 2) X 0 4) (I%VAL R1))

"J'ENVISAGE LE PLAN " DEVLIM71
"?????????????????? VOYONS SI " "DEV D'UNE CSTE:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM95
"?????????????????? VOYONS SI " "DEVELOPPEMENT U(X) **N AVEC N ENTIER"
" "
".... PROBLEME A RESOUDRE " (I%CREER (R1))
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM X X 0 4) (I%VAL R1))

"J'ENVISAGE LE PLAN " DEVLIM71
"?????????????????? VOYONS SI " "DEV D'UNE CSTE:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM73
"?????????????????? VOYONS SI " "DEV D'UN POLYNOME:EVIDENT"
*** SERVO116 DECLARED FLUID
"LE PLAN " 73 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEV D'UN POLYNOME:EVIDENT"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R1)
R1  SERVO116  X

X
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO112 (I%ALGEVAL (COUPE (EXPT (I%VVAL R1) 2) X 4)))
*** SERVO112 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER R1)
"LE PLAN " 95 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEVELOPPEMENT U(X) **N AVEC N ENTIER"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R1)
R1  SERVO112  (EXPT X 2)

 2
X
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%SIMPLIF (I%REMPL X 0 (I%VVAL R1))))
*** SERVO113 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R2)
R2  SERVO113  0

0
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R3) (I%ALGEVAL (DLEXP V 4)))
*** SERVO114 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R3)
R3  SERVO114  (PLUS (TIMES (EXPT V 4) EPS) (TIMES (QUOTIENT 1 24) (EXPT V 4)) (TIMES (QUOTIENT 1 6) (EXPT V 3)) (TI
```

Fig 10-3

```
MES (QUOTIENT 1 2) (EXPT V 2)) V 1)

 4          4       3       2
V *EPS + 1/24*V + 1/6*V + 1/2*V + V + 1
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R4) (I%SIMPLIF (I%REMPL V (DIFFERENCE (I%VVAL R1) (I%VVAL R2)) (I%VVAL
 R3))))
*** SERVO115 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R4)
R4  SERVO115  (PLUS (TIMES (EXPT X 8) EPS) (TIMES (QUOTIENT 1 24) (EXPT X 8)) (TIMES (QUOTIENT 1 6) (EXPT X 6)) (TI
MES (QUOTIENT 1 2) (EXPT X 4)) (EXPT X 2) 1)

 8          8       6       4     2
X *EPS + 1/24*X + 1/6*X + 1/2*X + X + 1
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO109 (I%SIMPLIF (TIMES (I%ALGEVAL (COUPE (I%VVAL R4) X 4)) (EXPT E (I%VVA
L R2)))))
*** SERVO109 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2 R3 R4))
"LE PLAN " 78 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEV E**U(X) "
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R1)
R1  SERVO109  (PLUS (TIMES (QUOTIENT 1 2) (EXPT X 4)) (EXPT X 2) 1)

     4    2
1/2*X  + X  + 1
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM 1 X O 4) (I%VAL R2))

"J'ENVISAGE LE PLAN " DEVLIM71
"?????????????????? VOYONS SI " "DEV D'UNE CSTE:EVIDENT"
*** SERVO110 DECLARED FLUID
"LE PLAN " 71 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEV D'UNE CSTE:EVIDENT"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R2)
R2  SERVO110  1

1
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO107 (I%ALGEVAL (DIFFERENCE (I%VVAL R1) (I%VVAL R2))))
*** SERVO107 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2))
"LE PLAN " 75 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEV LIM D'UNE DIFFERENCE:FACILE"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R1)
R1  SERVO107  (PLUS (TIMES (QUOTIENT 1 2) (EXPT X 4)) (EXPT X 2))

     4    2
1/2*X  + X
" "
".... PROBLEME A RESOUDRE " (CALCUL (DEVLIM (LOG (PLUS 1 X)) X O 4) (I%VAL R2))

"J'ENVISAGE LE PLAN " DEVLIM71
"?????????????????? VOYONS SI " "DEV D'UNE CSTE:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM73
"?????????????????? VOYONS SI " "DEV D'UN POLYNOME:EVIDENT"
"J'ENVISAGE LE PLAN " DEVLIM83
"?????????????????? VOYONS SI " "DEVLIM LOG(A+U(X)) "
" "
```

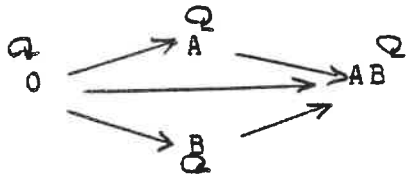
Handwritten margin marks: (14), (15), (16), 17, 18, (20), (21), (22)

```
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R1) (I%ALGEVAL (DLLOG1PX V 4)))
*** SERVO138 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (CHOIX 2 (SI (X EST X) (SUC 3 (I%ATTRIB (I%VAL R2) (I%SIMPLIF (I%REMPL V (QUOTIENT X 1)
 (I%VVAL R1)))) (I%MONTRER R2) (I%ATTRIB SERVO108 (I%ALGEVAL (PLUS (LOG 1) (I%VVAL R2)))))) (SUC 4 (CALCUL (DEVLIM
X X 0 4) (I%VAL R3)) (I%ATTRIB (I%VAL R2) (I%SIMPLIF (I%REMPL V (QUOTIENT (I%VVAL R3) 1) (I%VVAL R1)))) (I%MONTRER
R2) (I%ATTRIB SERVO108 (I%ALGEVAL (PLUS (LOG 1) (I%VVAL R2))))))
" "
".... PROBLEME A RESOUDRE " (SI (X EST X) (SUC 3 (I%ATTRIB (I%VAL R2) (I%SIMPLIF (I%REMPL V (QUOTIENT X 1) (I%VVAL
R1)))) (I%MONTRER R2) (I%ATTRIB SERVO108 (I%ALGEVAL (PLUS (LOG 1) (I%VVAL R2))))))
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB (I%VAL R2) (I%SIMPLIF (I%REMPL V (QUOTIENT X 1) (I%VVAL R1))))
*** SERVO139 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R2)
R2  SERVO139  (PLUS (TIMES (EXPT X 4) EPS) (MINUS (TIMES (QUOTIENT 1 4) (EXPT X 4))) (TIMES (QUOTIENT 1 3) (EXPT X
3)) (MINUS (TIMES (QUOTIENT 1 2) (EXPT X 2))) X)

 4            4        3        2
X *EPS - 1/4*X  + 1/3*X  - 1/2*X  + X
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO108 (I%ALGEVAL (PLUS (LOG 1) (I%VVAL R2))))
*** SERVO108 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2 R3))
"LE PLAN " 83 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEVLIM LOG(A+U(X)) "
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R2)
R2  SERVO108  (PLUS (TIMES (EXPT X 4) EPS) (MINUS (TIMES (QUOTIENT 1 4) (EXPT X 4))) (TIMES (QUOTIENT 1 3) (EXPT X
3)) (MINUS (TIMES (QUOTIENT 1 2) (EXPT X 2))) X)

 4            4        3        2
X *EPS - 1/4*X  + 1/3*X  - 1/2*X  + X
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB SERVO106 (I%ALGEVAL (DIFFERENCE (I%VVAL R1) (I%VVAL R2))))
*** SERVO106 DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2))
"LE PLAN " 75 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEV LIM D'UNE DIFFERENCE:FACILE"
" "
".... PROBLEME A RESOUDRE " (I%MONTRER R2)
R2  SERVO106  (PLUS (MINUS (TIMES (EXPT X 4) EPS)) (TIMES (QUOTIENT 3 4) (EXPT X 4)) (MINUS (TIMES (QUOTIENT 1 3) (
EXPT X 3))) (TIMES (QUOTIENT 3 2) (EXPT X 2)) (MINUS X))

   4            4        3        2
 - X *EPS + 3/4*X  - 1/3*X  + 3/2*X  - X
" "
".... PROBLEME A RESOUDRE " (I%ATTRIB RR (I%SIMPLIF (QUOTIENT (I%VVAL R2) (TIMES (I%TETE (I%VVAL R1)) (EXPT X (I%TQ
 (I%VVAL R1)))))))
*** RR DECLARED FLUID
" "
".... PROBLEME A RESOUDRE " (I%LIBERER (R1 R2))
"LE PLAN " 90 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "DEVLIM U(X)/A*X**P "
" "
".... PROBLEME A RESOUDRE " (I%ECRIRE (I%ALGEB RR))

   3            3        2
 - X *EPS + 3/4*X  - 1/3*X  + 3/2*X - 1
TEMPS EN MS :
TIME: 74214 MS
NIL
```

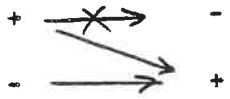
plans, comment ceux-ci s'imbriquent ; il montre comment il a été possible de décrire des méthodes fines dans le langage d'expression des connaissances et comment l'utilisateur peut les voir apparaître en même temps que tous les résultats intermédiaires.

IV.3.3.6. Exemple 6 : La compatibilité des groupes sanguins.

Cet exemple est destiné à montrer sur un exemple simple, l'universalité de l'approche. Il montre que CAMELIA peut être utilisé pour aborder des problèmes autres que les mathématiques pourvu que les connaissances du domaine d'activité soient exprimées dans le formalisme retenu. L'objectif est d'obtenir un programme (interactif) qui dise si deux sangs dont on connaît respectivement les groupes et les facteurs rhésus sont compatibles en termes de transfusion sanguine. Toutes les connaissances utiles sont contenues dans les graphes suivants : Si la flèche --> se lit : "Peut donner".



Compatibilité des groupes.



Compatibilité des RHESUS.

Ces graphes peuvent être décrits par un jeu de règles. Il suffit de disposer d'un prédicat "PEUT DONNER" noté PD dans les règles 11 à 18 décrites (figure 11-1). On voit sur cet exemple l'usage que nous faisons des règles conditionnelles. Ces connaissances "sûres" exprimées, il reste à décrire le savoir-faire, ce qu'il faut faire pour "dire que deux sangs sont compatibles". Cela amène les deux plans (numérotés 100 et 101). Le premier permet de vérifier des inclusions au sens ensembliste général (il n'a bien sûr rien de spécifique en direction de la transfusion sanguine mais il permettra de s'assurer que les groupes et rhésus fournis de façon interactive sont bien dans les ensembles respectifs (O,A,B,AB) et POS,NEG.

C'est là une façon de vérifier que les objets fournis sont acceptables. Le plan 101 est en fait le plan efficace. Il énonce que pour vérifier que deux sangs sont compatibles, on commence par vérifier que les données sont acceptables (nous avons utilisé les notations pratiques suivantes : GD, GR pour groupes donneur et receveur alors que RD, RR notent respectivement les rhésus donneur et receveur). Il suffit si (GD et GR) (A B AB O) et (RD RR) (POS NEG) de regarder si une règle s'applique. Si une règle s'applique cela signifie qu'il existe un arc sur les graphes et donc que les sangs sont compatibles.

L'écriture du plan (101) fait apparaître l'utilisation des opérateurs CHOIX et SUC. En particulier, les commandes NIL, secondes actions de SUC sont nécessaires pour faire échouer les CHOIX 2 et donc le SUC 3 initial lorsqu'il y a une anomalie dans les données. (L'action immédiate (!%ECRIRE...) amène en effet toujours un succès, même si c'est pour dire que quelque chose ne va pas!).

La procédure GRRH () décrite ne fait que gérer le dialogue et ne pose aucun problème : l'essentiel de son travail est d'appeler le moteur d'inférence SOLUT dès qu'il est en possession des données nécessaires.

Les exemples d'exécution qui suivent (figure 11, 2 à 4) montrent le comportement du programme.

Figure 11-1

```
00010 %$
00020 OUT REDUFICH$ OFF ECHO$ LISP$
00030 %=============================================================$
00040 OUT T$ WRITE "ENTREE DU CHAPITRE SANG     "$OUT REDUFICH$
00050 %=============================================================$
00060 CONNEC1 '(DIRE VERIFIER)$
00070 CONNEC2 '(APPARTENANCES)$
00080 CONNEC4 '(PD COMPATIBLE)$
00090 GLOBAL'(SANG11,SANG12,SANG13,SANG14,SANG15,SANG16,SANG17,SANG18)$
00100 REG(11):='(SANG11 (DESCRIPTEURS VARSUBST (X Y)))$
00110 REG(12):='(SANG12 (DESCRIPTEURS VARSUBST (Y)))$
00120 REG(13):='(SANG13 (DESCRIPTEURS VARSUBST (X Y)
00130                    CONDIT (OU (Y EST A)(Y EST AB))))$
00140 REG(14):='(SANG14 (DESCRIPTEURS VARSUBST (X Y)
00150                    CONDIT (OU (Y EST A)(Y EST AB))))$
00160 REG(15):='(SANG15 (DESCRIPTEURS VARSUBST (X Y)
00170                    CONDIT (OU (Y EST B)(Y EST AB)) ))$
00180 REG(16):='(SANG16 (DESCRIPTEURS VARSUBST (Y)
00190                    CONDIT (OU (Y EST B)(Y EST AB)) ))$
00200 REG(17):='(SANG17 (DESCRIPTEURS VARSUBST (X)))$
00210 REG(18):='(SANG18 (DESCRIPTEURS))$
00220 ALGEBRAIC$ OPERATOR PD$
00230 SANG11:= REEC ( PD (O,X,Y,POS), T )$
00240 SANG12:= REEC ( PD (O,NEG,Y,NEG), T )$
00250 SANG13:= REEC ( PD (A,X,Y,POS), T )$
00260 SANG14:= REEC ( PD (A,NEG,Y,NEG), T )$
00270 SANG15:= REEC ( PD (B,X,Y,POS), T )$
00280 SANG16:= REEC ( PD (B,NEG,Y,NEG), T )$
00290 SANG17:= REEC ( PD (AB,X,AB,POS), T )$
00300 SANG18:= REEC ( PD (AB,NEG,AB,NEG), T )$
00310 LISP$
00320 PLAN(100):='(VERIF100 (DESCRIPTEURS VARSUBST (L1 L2)
00330                        MSGECHEC "L1 NON INCLUS DANS L2"
00340                       )
00350          POUR (VERIFIER (APPARTENANCES L1 L2)) ESSAYER
00360     (CHOIX 2
00370        (POURTOUT X L1 (!%VUS X L2))
00380        (SUC 2 (!%ECRILIS (L1 " N'EST PAS INCLUS DANS " L2))
00390               NIL)
00400     ))$
00410 PLAN(101):='(RHESUS101 (DESCRIPTEURS VARSUBST (GD RD GR RR)
00420                        )
00430       POUR (DIRE (COMPATIBLE GD RD GR RR)) ESSAYER
00440     (SUC 3
00450        (CHOIX 2
00460            (VERIFIER (APPARTENANCES (GD GR)(A B O AB)))
00470            (SUC 2 (!%ECRIRE "GROUPE ILLEGAL") NIL) )
00480        (CHOIX 2
00490            (VERIFIER (APPARTENANCES (RD RR)(POS NEG)))
00500            (SUC 2 (!%ECRIRE "RHESUS ILLEGAL") NIL ) )
00510        (CHOIX 2
00520            (SI (!%RESULTATCONNU (PD GD RD GR RR) SANG)
00530                (!%ECRIRE "DONNEUR ET RECEVEUR COMPATIBLES"))
00540            (!%ECRIRE "DONNEUR ET RECEVEUR INCOMPATIBLES"))
00550     ))$
00560 %$
00570 LISP PROCEDURE GRRH() $
00580 BEGIN SCALAR GD,RD,GR,RR,REP$          REP:=T$
00590 WHILE REP DO
00600   <<WRITE "SANG DU DONNEUR ? (GROUPE PUIS RHESUS) "$TERPRI()$
00610     GD:=READ() $ RD:=READ() $
00620     WRITE "SANG DU RECEVEUR ? (GROUPE PUIS RHESUS) "$TERPRI()$
00630     GR:=READ() $ RR:=READ() $
00640     SOLUT LIST ('DIRE,LIST ('COMPATIBLE,GD,RD,GR,RR))$
00650     WRITE "VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)"$TERPRI()$
00660     REP:=READ()$
00670   >>$
00680 END$
```

Figure II-2

```
        END$

    lirchap 'turban;
        5: %$

        OUT REDUFICH$

        ENTREE DU CHAPITRE SANG     FIN DU SANG
(1) grrh();
        6: SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
    o
    pos
        SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
    ab
    pos
        NO DU PLAN A UTILISER ?
    999

        "J'ENVISAGE LE PLAN " RHESUS101
        " "
        ".... PROBLEME A RESOUDRE " (CHOIX 2 (VERIFIER (APPARTENANCES (O AB)
     (A
        B O AB))) (SUC 2 (!%ECRIRE "GROUPE ILLEGAL") NIL))
        " "
(2)     ".... PROBLEME A RESOUDRE " (VERIFIER (APPARTENANCES (O AB) (A B O A
    B)))

        NO DU PLAN A UTILISER ?
    999

        "J'ENVISAGE LE PLAN " VERIF100
        " "
        ".... PROBLEME A RESOUDRE " (POURTOUT X (O AB) (!%VUS X (A B O AB)))
        "LE PLAN " 100 " A PERMIS DE CONCLURE  "
        " "
        ".... PROBLEME A RESOUDRE " (CHOIX 2 (VERIFIER (APPARTENANCES (POS P
    OS)
        (POS NEG))) (SUC 2 (!%ECRIRE "RHESUS ILLEGAL") NIL))
        " "
(3)     ".... PROBLEME A RESOUDRE " (VERIFIER (APPARTENANCES (POS POS) (POS
    NEG)
        ))
        NO DU PLAN A UTILISER ?
    999

        "J'ENVISAGE LE PLAN " VERIF100
        " "
        ".... PROBLEME A RESOUDRE " (POURTOUT X (POS POS) (!%VUS X (POS NEG)
    ))
        "LE PLAN " 100 " A PERMIS DE CONCLURE  "
        " "
        ".... PROBLEME A RESOUDRE " (CHOIX 2 (SI (!%RESULTATCONNU (PD O POS
    AB P
        OS) SANG) (!%ECRIRE "DONNEUR ET RECEVEUR COMPATIBLES")) (!%ECRIRE "D
    ONNE
        UR ET RECEVEUR INCOMPATIBLES"))
        " "
        ".... PROBLEME A RESOUDRE " (SI (!%RESULTATCONNU (PD O POS AB POS) S
    ANG)
         (!%ECRIRE "DONNEUR ET RECEVEUR COMPATIBLES"))
(4)     J'AI PU APPLIQUER 11

(5)     DONNEUR ET RECEVEUR COMPATIBLES
        "LE PLAN " 101 " A PERMIS DE CONCLURE  "
        VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
```

figure II - 3

```
navigavue:=nil$msglevel:=3$grrh();
    7:
    8:
    9: SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
a
neg
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
ab
neg


    J'AI PU APPLIQUER 14

    DONNEUR ET RECEVEUR COMPATIBLES
    VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
(t)
    SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
ab
pos
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
a
neg


    DONNEUR ET RECEVEUR INCOMPATIBLES
    VOULEZ-VOUS CONTINUER? .(T=OUI,NIL=NON)
(t)
    SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
a
pos
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
b
pos


    DONNEUR ET RECEVEUR INCOMPATIBLES
    VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
(t)
    SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
a
neg
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
ab
neg


    J'AI PU APPLIQUER 14

    DONNEUR ET RECEVEUR COMPATIBLES
    VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
(nil)

    NIL
```

figure 11-4

```
    NIL

msglevel:=3$navigavue:=nil$  grrh();
    7:
    8:
    9: SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
q
zut
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
a
pos

    "ECHEC SUR PBTOUT : " (POURTOUT X (Q A) (!%VUS X (A B O AB)))
    "PB ACTUEL " (!%VUS Q (A B O AB))
    (Q A)  N'EST PAS INCLUS DANS  (A B O AB)
    "******ECHEC SUR " NIL
    ECHEC SUR TOUTES LES ETAPES POSSIBLES
    ***** ECHEC SUR (VERIFIER (APPARTENANCES (Q A) (A B O AB)))

(1) GROUPE ILLEGAL
    "******ECHEC SUR " NIL
    ECHEC SUR TOUTES LES ETAPES POSSIBLES
    "******ECHEC SUR " (CHOIX 2 (VERIFIER (APPARTENANCES (Q A) (A B O AB
)))
    (SUC 2 (!%ECRIRE "GROUPE ILLEGAL") NIL))
    ***** ECHEC SUR (DIRE (COMPATIBLE Q ZUT A POS))

    VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
(t)
    SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
a
pos
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
x
r

    "ECHEC SUR PBTOUT : " (POURTOUT X (A X) (!%VUS X (A B O AB)))
    "PB ACTUEL " (!%VUS X (A B O AB))
    (A X)  N'EST PAS INCLUS DANS  (A B O AB)
    "******ECHEC SUR " NIL
    ECHEC SUR TOUTES LES ETAPES POSSIBLES
    ***** ECHEC SUR (VERIFIER (APPARTENANCES (A X) (A B O AB)))

    GROUPE ILLEGAL
    "******ECHEC SUR " NIL
    ECHEC SUR TOUTES LES ETAPES POSSIBLES
    "******ECHEC SUR " (CHOIX 2 (VERIFIER (APPARTENANCES (A X) (A B O AB
)))
    (SUC 2 (!%ECRIRE "GROUPE ILLEGAL") NIL))
    ***** ECHEC SUR (DIRE (COMPATIBLE A POS X R))

    VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
(t)
    SANG DU DONNEUR ? (GROUPE PUIS RHESUS)
a
pos
    SANG DU RECEVEUR ? (GROUPE PUIS RHESUS)
b
z

    "ECHEC SUR PBTOUT : " (POURTOUT X (POS Z) (!%VUS X (POS NEG)))
    "PB ACTUEL " (!%VUS Z (POS NEG))
    (POS Z)  N'EST PAS INCLUS DANS  (POS NEG)
    "******ECHEC SUR " NIL
    ECHEC SUR TOUTES LES ETAPES POSSIBLES
    ***** ECHEC SUR (VERIFIER (APPARTENANCES (POS Z) (POS NEG)))

(2) RHESUS ILLEGAL
    "******ECHEC SUR " NIL
    ECHEC SUR TOUTES LES ETAPES POSSIBLES
    "******ECHEC SUR " (CHOIX 2 (VERIFIER (APPARTENANCES (POS Z) (POS NE
G)))
     (SUC 2 (!%ECRIRE "RHESUS ILLEGAL") NIL))
    ***** ECHEC SUR (DIRE (COMPATIBLE A POS B Z))

    VOULEZ-VOUS CONTINUER? (T=OUI,NIL=NON)
nil

    NIL
```

Conclusion

Cet exemple montre bien ce que l'on entend par programmer avec des règles. Il illustre bien la séparation que nous avons faite dans CAMELIA entre règles qui représentent des connaissances sûres et filtres qui représentent des savoir-faire, des démarches.

IV.4. ASPECTS INTERFACE HOMME-MACHINE

Le probème de l'interface homme-machine avec les systèmes experts est un problème difficile et mal résolu si l'on souhaite effectivement un accès d'utilisateurs non avertis. Ce problème est également important pour l'expert qui doit entrer des connaissances nouvelles. Cette partie à l'intention de faire apparaître différentes facettes de ce problème.

IV.4.1. Généralités sur inter-action homme-machine sous contrôle d'un système expert.

- Observation : la relation est essentiellement tripartite comme le montre la figure 12 : les partenaires en présence sont le concepteur, l'(es) expert(s) fournissant les connaissances, le(s) utilisateur(s). Les problèmes d'ergonomie logicielle (et nous ne parlerons que d'eux bien que les problèmes d'ergonomie des matériels soient importants) comportent donc 3 aspects :

i) l'interaction avec le concepteur du système.

ii) l'interaction avec les experts amenés à fournir des connaissances pour le système, le premier ayant un rôle prépondérant.

iii) les utilisateurs.

IV.4.2. Interface avec le concepteur du système :

Nous nous intéressons surtout ici aux systèmes experts ouverts qui comme CAMELIA autorisent le déclenchement des procédures à partir de décisions prises dans les règles. La mise en place d'un système expert de ce type est une tâche rude et il y a lieu de se doter des outils logiciels essentiels pour permettre l'enrichissement du système sur différents aspects.

* Facilités pour enrichir le langage d'écriture des connaissances (déclaration de nouvelles connectives, nouvelles actions immédiates).

* Facilités pour incorporer les procédures associées aux actions, possiblités de changer de procédure en gardant le nom de l'action (on peut ainsi changer d'exécution sans retoucher la base de connaissances mais en changeant la sémantique de l'action), possibilités de synonymes (actions de noms différents qui se réalisent par la même procédure) intéressantes pour la rédaction des connaissances : on peut être plus proche du langage de l'expert dans certains cas. Dans CAMELIA nous nous sommes dotés d'outils comme AJOUTACTIM pour ajouter des actions immédiates, CONNEC1,...,CONNECV pour déclarer des connectives nouvelles - cf. Annexe B - paragraphe "ENRICHIR l'ENVIRONNEMENT" et d'outils pour vérifier le contexte de travail à un instant donné (procédure CONTEXTE).

IV.4.3. Interface avec l'(es) expert(s)

IV.4.3.1. Interface avec le premier expert :

Le premier expert qui travaille en liaison étroite avec le concepteur (c'est souvent le même individu qui

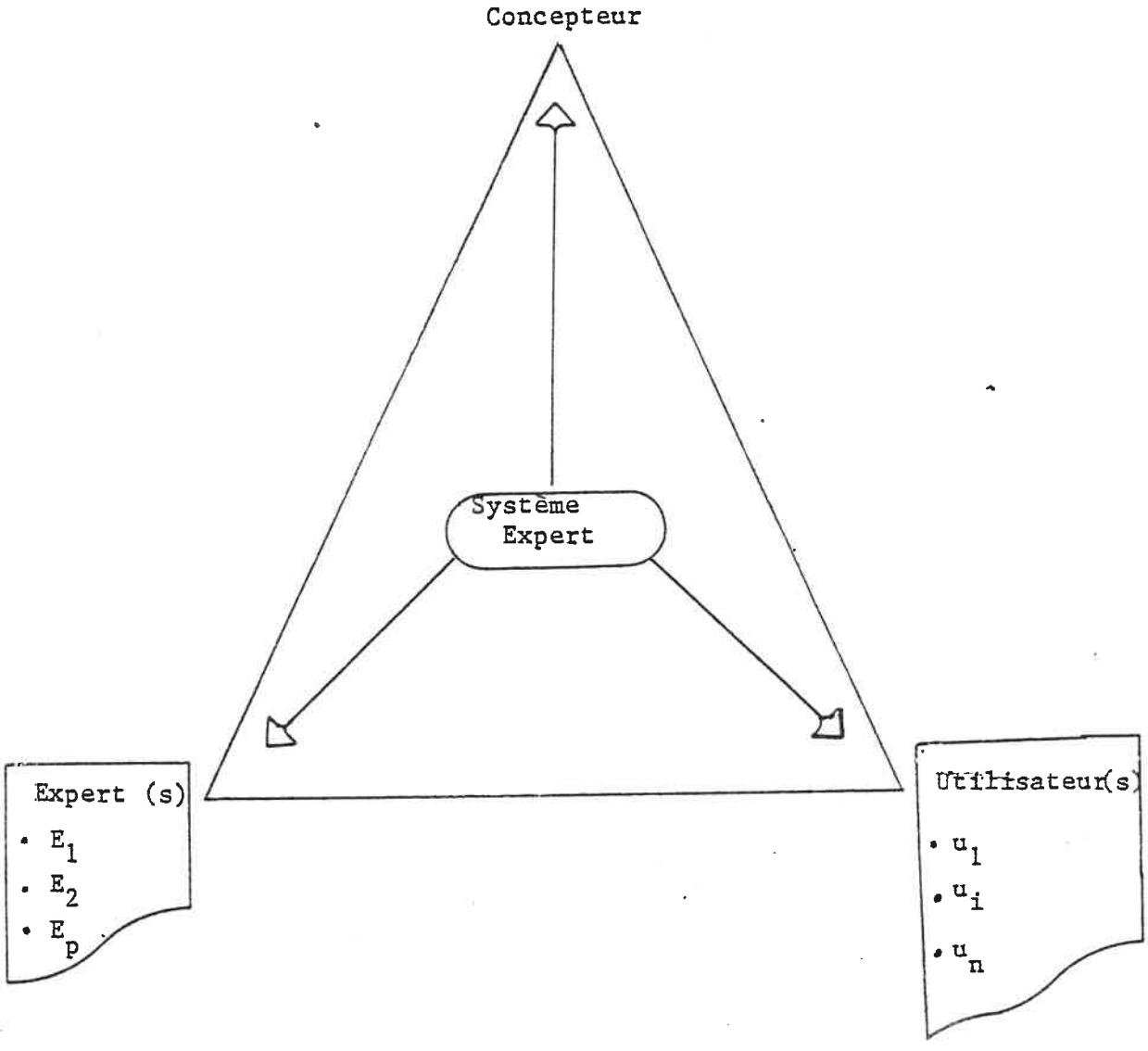


Figure 12 : Inter-action Homme-machine Systèmes experts

cumule les fonctions actuellement !) a pour rôle de fournir les connaissances initiales qui commencent le remplissage de la base. Il doit disposer (sinon définir par des exigences justifiées) d'outils adaptés. Cela concerne les outils de mise à jour, rectification, correction, de la base, les facilités d'édition d'une façon générale. Nous abordons là des problèmes qui relèvent du génie logiciel spécifique pour les systèmes experts. Toute une étude serait intéressante à conduire pour définir des outils permettant de gérer des ITEMS de connaissances données "en VRAC" par un expert (localisation, standardisation, ajout, suppression, correction, ...,. Tout ce travail fait autant que possible avec des accès, par contenu plutôt que par adresse .
La gestion d'un "VRAC" ne pose pas trop de problèmes pour un système expert mais la gestion externe du VRAC doit rester de dimension humaine : lorsque le nombre de règles devient grand (et c'est là que ça devient intéressant!) le listing avec des règles écrites dans un VRAC absolu n'est plus un outil suffisant au niveau humain. A notre sens, le développement des systèmes experts passe par le développement d'outils adaptés à ce travail.

IV.4.3.2. Interface avec d'autres experts.

Un système n'a de sens que si sa base de connaissances est ouverte, complètable par ajouts de nouveaux items apportés éventuellement par d'autres experts du domaine. A ce point se posent les problèmes d'homogénéité de langages (possibilités de définir des synonymes et plus généralement exprimer un concept d'un expert dans les termes d'un autre expert. A ce stade plus qu'ailleurs, se posent les problèmes de cohérence, de redondance. Les idées de prétraitement étendues à l'idée de compilation de connaissances pour essayer d'atteindre un langage intermédiaire commun sont-elles suffisantes pour traiter de problème ? La question reste ouverte en l'absence d'expérimentations adaptées.

Actuellement dans CAMELIA nous avons juste possibilité de définir des synonymes ou d'ajouter, supprimer de nouveaux items avec l'éditeur. Aucune vérification de cohérence de la base n'est faite.

Il est intéressant d'observer que le langage d'expression des connaissances est suffisamment souple pour traduire la même idée avec des écritures très différentes.

Exemples : Exprimer que la différence de deux fonctions impaires est impaire peut se traduire avec les écritures suivantes (toutes possibles dans CAMELIA).

E1.

```
%UNE DIFFERENCE DE FONCTIONS IMPAIRES EST IMPAIRE ;
PLAN(21):='(IMPAIRE21 (DESCRIPTEURS VARSUBST (F X)
                                     OTOPEX ( DIFFERENCE)

                      POUR (PRV (IMPAIRE F X)) ESSAYER
                           (SI (( !%OP F) EST DIFFERENCE)
                               (SUC 2
                                 (PRV (IMPAIRE (!%ARG1 F) X))
                                 (PRV (IMPAIRE (!%ARG2 F) X))) ));
```

E2.

```
PLAN(21):='(IMPAIRE21 (DESCRIPTEURS VARSUBST (A B X)
                                    OTOPEX (DIFFERENCE)
                       )
            POUR (PRV (IMPAIRE (DIFFERENCE A B X )) ESSAYER
                 (SUC 2
                   (PRV (IMPAIRE A X))
                   (PRV (IMPAIRE B X))  ));
```

E3.

```
PLAN(21):='(IMPAIRE21 (DESCRIPTEURS VARSUBST (A B X)
                                    OTOPEX (DIFFERENCE)
                       )
            POUR (PRV (IMPAIRE (DIFFERENCE A B) X )) ESSAYER
                  (SI (ET (IMPAIRE A X) (IMPAIRE B X)) T )) ;
```

E4.

```
PLAN(21):='(IMPAIRE21 (DESCRIPTEURS VARSUBST (F X)
                                    OTOPEX (DIFFERENCE)
                       )
            POUR (PRV (IMPAIRE F X )) ESSAYER
                  (SI (ET ((!%OP F) EST DIFFERENCE)
                          (TOUSVRAI (!%INTERVAL    1    (!%ORDROP
                          F))
                          (IMPAIRE (!%ARGI !? F ) X )) )
                          T) );
```

Ces écritures équivalentes ne sont pas traitées par le système de façon identique :

- E1, E2, E3, E4 sont choisies dès lors que le mot DIFFERENCE apparaît dans un problème de caractéristique (PRV IMPAIRE).

. mais pour E2 et E3 l'unification peut échouer très vite si DIFFERENCE n'est pas l'opérateur principal de la fonction que l'on cherche à établir comme IMPAIRE. Dans ce cas aucune substitution ne sera faite.

-. par contre E1 et E4 amènent le report de F dans les plan-types correspondants même avec le problème :

(PRV (IMPAIRE (TIMES (DIFFERENCE X 3) X) X))

L'unification marche. L'échec est vu beaucoup plus tard.

- Les écritures E3 et E4 font appel à la notion de condition problème (cf.IV.2.2.3.) avec une écriture simple en E3 (parce que DIFFERENCE est une connective binaire).

- L'écriture E4 donne une écriture acceptable pour des connectives variaires : elle dit que tous les arguments de la différence doivent être des fonctions impaires : l'écriture générale (!%INTERVAL 1 (!%ORDROP F)) génère l'ensemble fini (1 2) F ayant un opérateur principal d'ordre 2 et INTERVAL (I,J) renvoyant l'ensemble (I I+1 ... I+P J). Le joker !? de la condition TOUSVRAI balaie cet ensemble (1 2) pour que l'on passe en revue tous les arguments de F et que l'on vérifie que tous sont des fonctions impaires en X.

Nous voyons là la diversité des écritures acceptables mais nous voyons aussi que ces écritures ne sont pas absolument équivalentes du point de vue de l'efficacité du traitement qu'elles amènent. Les questions sont alors : faut-il donner le maximum de moyens d'expressions aux experts au risque de certaines inefficacités ? Quelles formes préfèrent les individus ? Préfèrent-ils tous la même ? Un système peut-il se rendre compte que ces formes expriment la même connaissance de base ? Un expert peut avoir envie de donner E2 parce qu'il n'a pas vu qu'un autre a fourni E4 : comment voir cette redondance? (qui ne gène d'ailleurs pas le système).
Mais si nous voulons un jour travailler avec de très grosses bases de connaissances, il ne faut sans doute pas tolérer de les grossir trop par des redondances qui les augmentent artificiellement sans être véritablement un apport de connaissances.

IV.4.4. Décisions concernant le contrôle.

Même si le système est conçu pour être autonome sur un problème, il est important que l'utilisateur puisse, s'il le désire garder le contrôle des connaissances utilisées. Dans CAMELIA nous avons mis en place trois possibilités (cf.IV.3.2.3.) :

- Autonomie complète (NAVIGAVUE = nil)
- Mode semi-automatique (autonomie sur des sous-problèmes, l'utilisateur prenant les décisions dans des choix critiques).
- Mode manuel - l'utilisateur étant sollicité pour tout choix de méthode.

i) Nous ne redéveloppons pas ici ces aspects. Notons seulement qu'il pourrait être intéressant que le système en mode manuel fasse des suggestions à l'utilisateur (dise, les choix possibles, ce qu'il ferait s'il était en mode automatique). L'utilisateur gardant la décision finale serait assisté dans sa prise de décision. Nous retrouvons l'idée de MAO (Mathématiques Assistées par Ordinateur). Cette possibilité n'est pas disponible dans l'état actuel de CAMELIA mais ne poserait pas de problème d'implémentation.

ii) Appel à l'aide : Un problème plus intéressant concerne les possibilités d'appel à l'aide ; le système peut en effet, dans certains cas, demander à l'utilisateur de l'aider. Un plan a été écrit pour cela et c'est une méthode parmi d'autres qu'a le système de demander qu'on l'aide. Ce plan a l'écriture très simple suivante :

```
%PLAN DEMANDANT UN SOUS-PROBLEME D'AIDE ;
PLAN(2):='(AIDESPB1 (DESCRIPTEUR VARSUBST (NIMPORTEQUOI)
                    )
          POUR NIMPORTEQUOI ESSAYER
          (SUC 6       (!%CREER       R1)(!%ECRIRE       (QUOTE
          NIMPORTEQUOI))
                 (!%ECRIRE "AIDEZ MOI ")
                 (!%ATTRIB (!%VAL R1) (!%ENTREE ) )
                 (!%VVAL R1)
                 (!%LIBERER R1)
                 NIMPORTEQUOI            )) ;
```

La seule difficulté concerne le type d'aide et l'aide effective que l'utilisateur peut apporter au système. Ce qui est délicat, c'est d'aider effectivement le système lorsqu'il appelle à l'aide (on peut penser en effet que si le système n'a plus que cette solution, c'est que le problème est difficile pour lui, sans doute aussi pour l'utilisateur).

- Actuellement, l'aide peut être apportée en fournissant à CAMELIA un sous-problème à résoudre, sous-problème censé apporter des éléments de solution. D'autres formes d'aide seraient intéressantes à prévoir : fournir une règle, fournir une méta-règle, fournir un plan. Mais dans ces conditions, c'est l'utilisateur qui devient expert fournissant les connaissances : nous dénaturons l'esprit de ce genre de système.

- La difficulté essentielle n'est cependant pas là. Elle est dans le fait que pour apporter une aide efficace, pertinente, l'utilisateur doit comprendre où en est le système. On retrouve l'importance des justifications apportées par le système de ce qu'il fait. La pertinence des messages (en qualité et en quantité) est essentielle à ce point. Nous avons traité ce problème en mettant à disposition de l'utilisateur un niveau de message variable. La variable globale MSGLEVEL prend ses valeurs dans l'intervalle (0,9) et plus la valeur attribuée est élevée plus le système est bavard. L'inconvénient de cette programmation est que si MGSLEVEL est trop faible (<3), on n'a pas assez de renseignements pour suivre et aider le système en cas de besoin. Si la valeur est élevée (>7), le système apparaît comme trop bavard si tout se passe bien et l'information utile pour apporter de l'aide peut être difficile à apercevoir. Notons que avec un MSGLEVEL faible, tous les messages sont stockés (dans un tableau de taille limitée mais ce pourrait être sur disque) et que c'est un filtrage à l'affichage qui se fait ; l'utilisateur peut donc avoir une trace, en temps différé de ce qui s'est fait dans le problème. La procédure JUSTIFIER x permet de sortir cette trace comme si MSGLEVEL avait été fixé à x en "temps réel". Cette possibilité de justification en temps différé à des degrés de finesse choisis par l'utilisateur n'est elle-même pas suffisante à notre sens.

Il apparaît qu'une bonne solution à ce problème passe par la synthèse de messages intelligents : cela suppose que le système soit capable de faire le point sur l'état d'avancement du travail, et de synthétiser (pourquoi pas en langue naturelle!) un message significatif dans lequel on trouverait l'explication de la démarche suivie, des points déjà résolus, des sous-problèmes qui restent à résoudre, de la nature du blocage, de la nature de l'aide apportée. La question reste ouverte. C'est un aspect des systèmes experts qui demanderait à être plus travaillé.

En guise de conclusion sur ce point, disons que : Mathematiques Assistées par Ordinateur (MAO) signifie transférer une partie de la compétence de l'utilisateur traditionnel vers les système (et on sait que la difficulté est de formuler les connaissances et le problème) ; pour CAMELIA demandant de l'aide, ce sont les Mathématiques Assistées par l'Utilisateur (MAU) qui posent problème.

IV.4.4.2. Décisions concernant les messages.

a) L'implantation des messages distribués par un système expert acceptant des règles déclenchant des procédures pose un certain nombre de questions fondamentales :

1) Qui décide de ce qui est affichable ? demandable ?

- est-ce l'expert lors de la rédaction des connaissances ? Dans ce cas, les messages sont dans les plans, dans les règles. L'attitude est alors du type "l'expert, celui qui sait, sait qu'à ce stade d'un problème de ce type, cette information est généralement intéressante et il décide de la donner à l'utilisateur".

- est-ce l'utilisateur lors de l'usage ? Après tout c'est lui qui sait ce qu'il veut, qui est concerné par le problème qui sera effectivement résolu.

- est-ce le rédacteur du système ? Dans ce cas, les messages peuvent se retrouver en partie dans les procédures déclenchées par les règles.

2) Qui juge de l'importance d'afficher quelque chose ?

- l'expert peut estimer qu'une indication est importante ; l'utilisateur peut ne pas s'y intéresser et réciproquement,

- les messages systèmes (rédigés par le concepteur) sont un bruit par rapport au problème (débordement de pile, de tableau, procédure redéfinie, variables déclarées par le système (générateur de variables...). Ces messages doivent-ils être fournis ? Comment peut réagir l'utilisateur "naïf"? Le dilemne de base reste que c'est l'expert qui sait et que c'est l'utilisateur qui doit décider !

b) Exemples : La procédure RESULTATCONNU consulte les règles correspondant à une caractéristique donnée (classe de règles) pour voir si une expression s'instancie avec le membre gauche d'une règle de cette classe. Si aucune règle ne marche, NIL est retourné sinon on renvoie le membre droit instancié. Le message annonçant le numéro de la règle qui marche et le fait que le résultat est connu peut être dans la procédure. Il peut aussi etre dans la règle d'appel avec une écriture comme :

```
(SI (RESULTATCONNU (PAIRE F X) PAIRE)
          (!%ECRIRE ("RESULTAT CONNU :" F "PAIRE" )))
```

On pourrait également considérer que RESULTATCONNU renvoie NIL ou bien une liste constituée du numéro de la règle et de la substitution nécessaire pou obtenir le membre droit. Au niveau du plan, nous aurions ainsi accès au numéro de la règle concernée. Des messages prévus dans les descripteurs de la règle seraient alors exploitables depuis le plan.

Nous voyons que les choix possibles lors de l'implantation sont très nombreux ; il n'est pas toujours évident qu'une solution vraiment meilleure s'impose.

c) Parallèle avec des situations rencontrées en EAO (Enseignement Assisté par Ordinateur).

Nous retrouvons là des situations semblables à ce que l'on rencontre lors de la rédaction de logiciels d'EAO dans lesquels l'adéquation des messages prévus par l'enseignant ne correspondent pas toujours au besoin de l'élève.

- le message proposé par le professeur n'est pas forcément celui qui convient à l'élève,

- l'élève peut avoir besoin d'une petite explication que le professeur n'a pas prévu de donner à ce stade,

- le message du professeur peut être incompréhensible ou non significatif pour l'élève au moment où il est délivré.

- ...

d) Quelques idées :

- Il n'est pas très difficile de mettre en place un dispositif qui permette à l'utilisateur d'agir sur le niveau des messages distribués : . en quantité : volumes de messages
. en qualité : justification des faits utilisés, des règles utilisées, des plans (raisonnements).

- Il nous paraîtrait plus intéressant que le système puisse manipuler un modèle de l'utilisateur qu'il a en face de lui pour savoir comment réagir à son égard. Il y a là une voie intéressante à creuser.

IV.4.5. Conclusion :

La qualité du dialogue homme-machine est fondamentale avec un tel système. Deux points apparaîssent se dégager et mériter des études plus approfondies :

1) Il s'agit en entrée du problème de la compilation des connaissances fournies par l'expert à l'aide d'outils logiciels adaptés.

2) Il s'agit en sortie d'avoir un niveau d'interaction intéressant avec l'utilisateur. Cela passe sans doute par le maintien à jour d'un modèle de l'utilisateur.

IV.5. CONCLUSION

CAMELIA est une réalisation qui a permis d'expérimenter des idées essentielles pour construire des systèmes experts, en particulier pour réaliser un système expert en mathématique. Le moteur d'inférence, la définition de la structure des bases de connaissances,la définition des modes de contrôle ont atteint un niveau satisfaisant. Il reste beaucoup de travail pour com-

pléter les chapitres de connaissances des domaines qui nous ont servi de support et expérimenter ce logiciel dans d'autres domaines mathématiques.

CAMELIA peut être complété au niveau fondamental par un certain nombre d'aspects que nous discutons dans le chapitre V. D'ores et déjà, il pourraît être transporté sur un méga-micro-ordinateur et être aménagé pour devenir un didacticiel en mathématique ou un "mathématicien de bureau" destiné à des laboratoires ou des bureaux d'études.

CHAPITRE V

PROLONGEMENTS POSSIBLES

CHAPITRE V

PROLONGEMENTS POSSIBLES

L'objet de ce chapitre est de dégager les voies possibles pour prolonger le travail engagé. Ces prolongements peuvent relever du développement (par exemple compléter l'écriture des chapitres de connaissances existants et envisager l'écriture de chapitres dans d'autres domaines) ou bien d'une recherche plus fondamentale (par exemple en direction de la compilation de connaissances du calcul parallèlme ou de moteurs d'inférence à inférence variable).

V.1. Rappel des prolongements évoqués dans le chapitre IV

La forme actuelle de CAMELIA peut être enrichie assez facilement par des apports techniques ou des apports de connaissances mathématiques nouvelles.

V.1.1 Apport de connaissances nouvelles

* Le premier travail de développement à exécuter serait de compléter les chapitres existants de la base de connaissances (parité, limites, développements limités intégrales). En particulier, le chapitre sur les limites est incomplet. Rappelons que notre objectif a été de montrer que le moteur est indépendant du contexte mathématique dans lequel il travaille et que toutes les connaissances sont dans les règles et méta-règles. Nous voulions aussi montrer que les connaissances d'un chapitre peuvent faire appel aux connaissances d'autres chapitres. Ces raisons nous ont fait écrire plusieurs chapitres dans des domaines différents pouvant être correllés (par exemple appel à la parité pour calculer des intégrales, aux développements limités pour calculer des limites).

* CAMELIA doit être testé dans d'autres domaines. Des chapitres de connaissances seraient intéressants à écrire en résolution d'équations sur la base des travaux de BUNDY (BUNDY 79a,79b,81a,81b). Les connaissances sont là bien formalisées. Il s'agit d'un exercice d'écriture !. La résolution d'inéquations avec les méthodes de BLEDSOE (BLEDSOE 79a) serait également un bon domaine : il fait fortement appel à des mécanismes d'inférences et à des mécanismes opératoires (calculs) intimement liés.

Ces travaux pourraient être envisagés dans le cadre de stages de DEA pour former des étudiants à l'usage de tels systèmes.

* Un domaine comme la manipulation de symboles avec des opérations non commutatives pourrait sans doute être abordé pour des calculs liés à la théorie des langages, le monoïde,...

V.1.2 Apports techniques

1) CAMELIA peut être amélioré en lui faisant gérer une base de faits qui permettrait d'alléger les appels répétés des fonctions d'accès aux objets ou sous objets d'un objet (opérateur principal, arguments, ...). Cette amélioration devrait permettre le stockage des problèmes déjà étudiés par CAMELIA (au cours d'une session, et pourquoi pas au cours de sessions précédentes avec une mémoire à long terme). Cela éviterait au système de s'attaquer deux fois au même problème sans s'en rendre compte. La difficulté est évidemment de caractériser les problèmes que l'on décide d'archiver : caractérisation ni trop lâche (amenant une explosion de la zone de stockage des problèmes résolus) ni trop sévère (laissant disparaître des problèmes qu'il peut être judicieux de conserver). Il faudrait faire là un travail comparable à ce qu'a fait PITRAT (PITRAT 66) pour l'archivage des "théorèmes intéressants" d'une théorie.

Il y aurait lieu là encore de prendre garde de ne pas stocker différentes formes d'un même problème : en effet, un problème peut être une instanciation d'un problème déjà résolu, des problèmes peuvent ne différer que par le nom d'un indice muet (indice de sommation, variable par rapport à laquelle on intègre, on dérive...). Il faudrait définir des classes d'équivalences d'objets, ne garder qu'un exemplaire de chaque classe. Ce travail n'est possible que si les algorithmes pour voir que deux problèmes sont équivalents, ne sont pas trop exigeants en ressources.

2) Amélioration du dialogue avec l'utilisateur

- Dans une perspective "Mathématiques Assistées par Ordinateur", il serait facile, lorsque le système travaille en mode manuel (conduit par l'utilisateur : NAVIGAVUE = VRAI) que CAMELIA fasse des suggestions à l'utilisateur (dise ce qu'il tenterait s'il était en mode autonome ; l'utilisateur prenant la décision lui-même de la voie à prendre (cf.IV.4.4i).

- Une amélioration beaucoup plus importante concernerait la synthèse de messages intelligents en particulier pour le cas où CAMELIA est appelé à demander de l'aide à l'utilisateur (cf.IV.4.4.ii). Ce travail mériterait d'être fait en particulier avec une perspective d'usage de CAMELIA comme système pour enseigner le calcul algébrique et la résolution de problèmes mathématiques. Un aspect plus fondamental concernerait la manipulation par le système de connaissances concernant l'utilisateur. On pourrait envisager effectivement que CAMELIA adapte ses messages en tenant compte d'un modèle de l'utilisateur qui est au clavier (modèle utilisateur étudiant auquel on justifierait des

théorèmes ou des démarches, auquel on demanderait ce qu'il faut faire sur un tel problème, la réponse étant alors d'autres suggestions justifiées, ..., modèle utilisateur nul en calcul pour qui il faut donner les calculs intermédiaires, modèle utilisateur qui n'attend que le résultat sans se soucier des façons de l'atteindre,...). Ce travail plus délicat pourrait être envisagé dans le cadre de thèse type docteur ingénieur.

V.2. COMPILATION DE CONNAISSANCES

V.2.1. Position du problème

La difficulté essentielle pour réaliser et utiliser un système expert est dans la gestion de la base de connaissances.

Pour des raisons pratiques évidentes, il est important que les conaissances soient fournies de la façon la plus modulaire possible, en items indépendants, dans un langage déclaratif proche de celui de l'expert qui les décrits. Ces souplesses offertes à l'utilisateur compliquent le travail du moteur d'inférence et pour des raisons d'efficacité, il peut y avoir intérêt à assurer des prétraitements fournissant un certain nombre de liens facilitant les accès (cf. IV.2.6. pour voir ce que nous faisons dans CAMELIA).

Partant du fait que les bases de connaissances sont coûteuses à écrire et que certaines pourraient être utilisées à des fins diverses, une idée qui pourrait être développée consisterait à définir des Bases de Connaissances Générales (BCG ou encyclopédies), écrites dans un langage orienté utilisateur avec toutes les facilités-utilisateurs évoquées ci-dessus. l'utilisation de ces BCG nécéssiterait alors une compilation dans un langage orienté système d'inférence. Le résultat de la compilation serait une base de connaissances spécialisée (BCS) utile avec un moteur et une finalité d'application donnés. Cette approche sera d'autant plus intéressante que l'on sera amenés à utiliser de "grosses" bases de connaissances. Nous pouvons ainsi séparer plus facilement les niveaux d'interventions des contraintes :

- Rédaction de l'encyclopédie : satisfaction des contraintes humaines (utilisateur / expert) : nodularité, langage déclaratif, ...
- Constitution de la BCS dans un langage objet satisfaisant les contraintes systèmes (efficacité par rapport à un moteur donné - homogénéité, cohérence vérifiée de la base de connaissances utilisée ...).

V.2.2. Structures envisageables

. On peut imaginer disposer d'encyclopédies diverses :

* par disciplines : mathématiques, médecine, géologie ... langue naturelle (compréhension génération)

* par type d'activités : - enseignement (règles pédagogiques)

- usage professionnel (aide au diagnostic, aide à la découverte, ...)
- communication (règles psychologiques décrivant des comportements, des dialogues adaptés à des utilisateurs différents, ...).

Ainsi les connaissances pour reconnaître ou engendrer de la parole sont de même nature - disposer d'une base générale que l'on spécialise pour faire de la synthèse ou de la reconnaissance peut être intéressant.

Des connaissances pour traiter des textes en langue naturelle peuvent être communes : par compilation on pourrait obtenir les connaissances nécessaires pour un système expert reconnaissant ou synthétisant un discours en langue naturelle, ...

. La possession de telles encyclopédies générales permettrait d'envisager par compilation des bases de connaissances spécialisées permettant d'atteindre des usages différents.

* système expert pour enseigner les mathématiques (EAO avec guidage effectif de l'étudiant),

* système expert pour assister un utilisateur dans la recherche d'une solution à un problème difficile.

. L'architecture de tels systèmes serait telle que le laisse apparaître la figure 13.

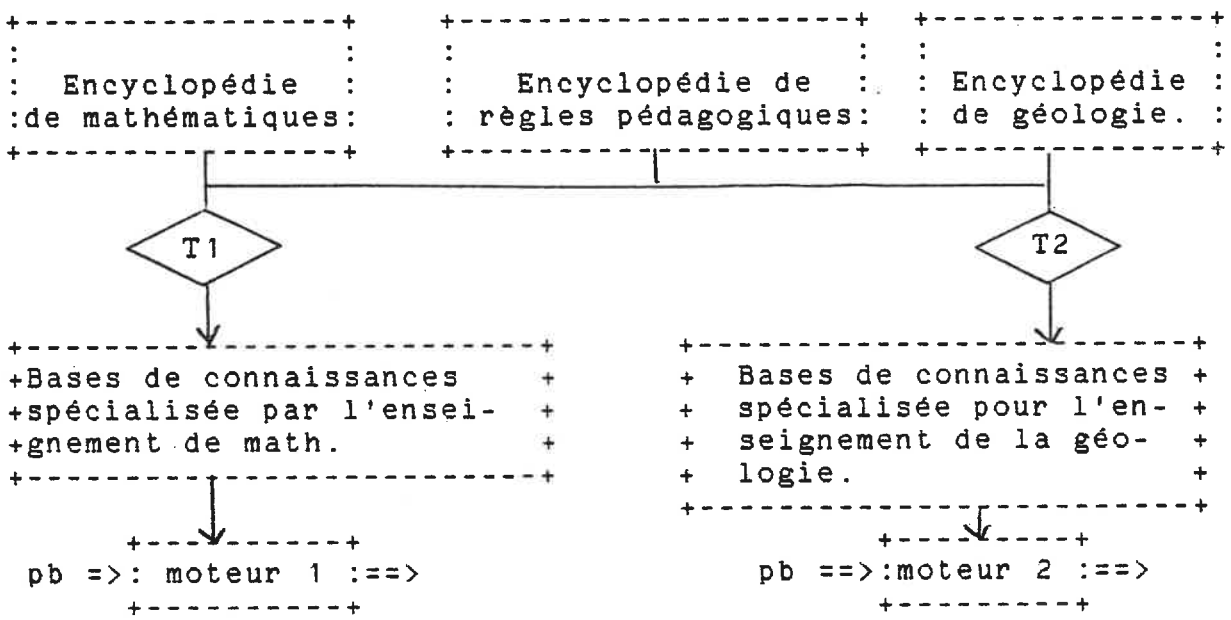


figure 13 : génération de base de connaissances à partir d'encyclopédie.

V.2.3. Caractéristiques de la gestion de connaissances dans l'encyclopédie

* Connaissances faciles à écrire, comprendre, modifier pour un spécialiste du domaine.

* Items de connaissances écrits dans un langage déclaratif, aussi proche que possible de celui des experts.

* Connaissances et méta-connaissances sont fournies en "VRAC".

* Les langages peuvent être hétérogènes lorsque les connaissances sont fournies par plusieurs experts (synonymes, ...)

V.2.4. Caractéristiques des connaissances dans une BCS.

* Les items de la BCS doivent avoir une forme adaptée aux objectifs et au moteur d'inférence utilisé. Ainsi dans CAMELIA nous avons des items sous forme de filtres et de plans. Pour un système destiné à enseigner les mathématiques, il y aurait lieu que les items comportent beaucoup plus d'informations liées aux pourquoi, comment, ..., à la justification des démarches, des choix, ...

* La forme des items doit être standard même si à la source elle a été fournie par des spécialistes différents. Un lien vers l'item source dans l'encyclopédie permet alors que les justifications soient fournies dans le langage de l'expert.

* La cohérence de l'information disponible doit être vérifiée (contradiction entre items révélées, révélation du fait qu'un item est plus général ou particulier qu'un autre fourni précédemment).

* La méta-connaissance doit être isolée et préparée pour faciliter la gestion faite par le moteur : la méta-connaissance doit avoir la même forme que les connaissances mais elle est gérée différemment.

V.2.5. Conclusion

Ces idées liées à la compilation de connaisssances n'ont pas été expérimentées à notre connaissance. Elles nous paraissent importantes pour démarrer de nouvelles recherches dans ce domaine : les systèmes experts seront surtout intéressants lorsqu'ils pourront prendre en compte de grandes bases de connaissances. Ces bases devront être partagées au maximum pour avoir un intérêt économique acceptable.

V-3 MOTEUR A INFERENCE VARIABLE

V.3.1 Le moteur d'inférence de CAMELIA est entièrement programmé (environ 250 lignes). Nous avons vu (IV.2-2.1) que l'opération SUC était actuellement interprétée comme une succes-

sion d'étapes. Pourtant, dans certains cas, ces actions peuvent être traitées de façon indépendante, l'essentiel étant qu'elles soient toutes exécutées. En fait l'ordre d'exécution n'est imposé que lorsqu'une étape fabrique un résultat utile pour une autre étape. Formellement cela peut apparaître facilement dans notre langage en observant les variables de notation : les actions !%CREER doivent bien sûr précéder les actions !%LIBERER. En se souvenant que l'accès au nom de la notation par exemple R1, se fait avec l'action (!%VAL R1) et que l'accès à la valeur notée R1 se fait avec l'action (!%VVAL R1), on pourrait fabriquer une règle explicitant que :
"Si dans un plan, une action 1 référe à (!%VAL x) et une action 2 référe à (!%VVAL x), alors l'action 1 doit être exécutée avant l'action 2".

. Ce type de règle devrait être conditionnelle puisque ceci n'a de sens que si la notation R1 n'est définie qu'une fois (un seul (!%VAL R1) dans le plan). Ceci interdit que la même notation soit utilisée pour des objets différents au sein d'un meme plan. Nous retrouvons le problème de l'ordre des affectations dans les langages de programmation.

. A partir de cet exemple, nous concevons bien que par un jeu de telles règles, il est sans doute possible d'établir des précédences entre certaines actions d'un plan, les autres actions pouvant s'exécuter dans un ordre indifférent au niveau logique ; Mais si l'ordre est indifférent, on peut alors espérer avec des heuristiques appropriées trouver pour chaque problème un ordre préférable (celui qui fait échouer très vite s'il doit y avoir échec, celui qui amène le plus vite l'assurance de succès s'il doit y avoir succès). Rappelons l'exemple vu en IV-2-2-1 sur l'intégration d'une différence : on peut chercher à intégrer d'abord le terme le plus compliqué.

. Ainsi, rien que pour l'interprêtation du (SUC...), on voit qu'il serait possible d'avoir une interprêtation moins rigide que l'interprêtation programmée actuelle mais qu'on pourait atteindre une souplesse intéressante en écrivant des règles d'interprêtation. Le moteur d'inférence doit donc lui-même pouvoir devenir système expert.

V.3.2 Quelques idées sur les moteurs-experts

1) L'idée à laquelle nous arrivons est donc que le moteur d'inférence soit lui-même un système expert, doté de connaissances sur la façon d'inférer. En quelque sorte, le moteur devient un noyau auquel on explicite formellement des règles du type "Dans tel domaine, avec tel type de plan, raisonner comme cela....".

La figure 14.1 rappelle la structure générale d'un système expert, la figure 14.2 fait apparaître la structure à laquelle nous arriverions

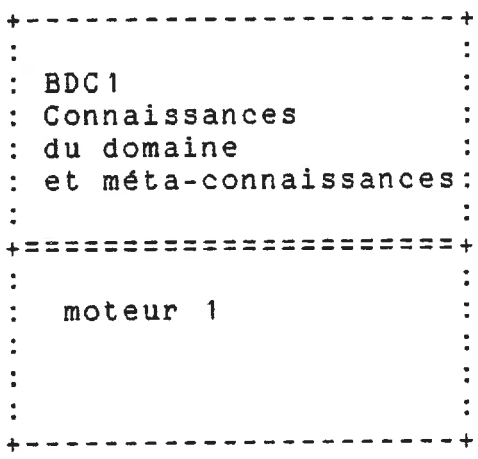


figure 14-1

Système expert au sens usuel

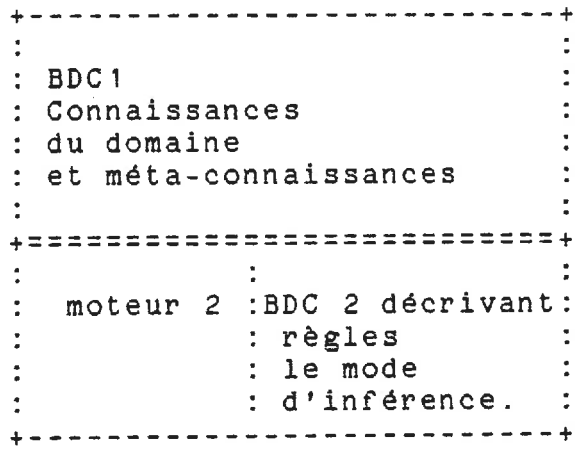


figure 14-2

système expert avec moteur à inférence variable

2°) Il apparait clairement que ce type de moteur à inférence variable serait doté de deux types fondamentaux de règles :
-les règles décrivant les connaissances du domaine et les méta-connaissances nécessaires pour les manipuler.
- les règles décrivant le fonctionnement du moteur.

Des travaux ont été réalisés avec ce type de perspectives.Citons le travail de D.FIESCHI à Marseille.

V-3-3 Conclusion

Il nous semble qu'un travail intéressant pourait être entrepris avec ces idées. CAMELIA étant un terrain d'expérimentation possible. Les questions qui se posent alors concernent :
i) le niveau auquel on s'arrête, le noyau moteur 2 pouvant peut-être lui aussi devenir système expert.

ii) la possibilité de faire des implémentations suffisamment efficaces de ces idées.
On peut voir que ces questions apparaissent comme de même nature que ce qu'on a vu pour l'utilisation de la méta-connaissance.

V-4 PARALLELISME

La possibilité effective de mettre en place des traitements parallèles reste un des problèmes et des objectifs essentiels actuellement <HUET 82> . Des travaux importants sont menés tant du point de vue matériel que logiciel. S'il est vrai que l'aboutissement de tels travaux pourra constituer un apport fondamental, en particulier pour l'intelligence artificielle, il est aussi vrai que les travaux en intelligence artificielle peuvent contribuer à la clarification des concepts et outils nécessaires.

La conception modulaire qui caractérise les systèmes experts en particulier au niveau des bases de connaissances et de faits en font sans doute des terrains propres à l'exploration de possibilités de parallélisme. Envisageons quelques aspects visibles à partir de CAMELIA.

V-4-1 Parallélisme lors de la sélection d'items

Les règles (filtres) qui figurent dans les différents chapîtres de la base sont indépendantes et lors de la sélection des plans applicables pour traiter un problème (PB) donné, on conçoit bien qu'un certain nombre de tâches puissent être conduites en parallèle - par exemple :

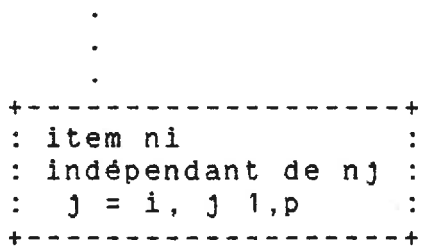
i) lors de la sélection grossière des filtres possibles (cf IV-2.4.1), l'élimination des plans qui imposent la présence d'opérateurs ne figurant pas dans le texte du problème pourrait être traitée simultanément pour tous les plans. (cf :mécanismes OTOPEX, O1OPEX).

Nous sommes dans une organisation du type suivant :

- on dispose d'une liste de numéros référençant des items.
- Tous ces items sont indépendants.
- Un problème fait (PB) doit être confronté à tous ces items.
- Le travail se poursuit lorsque toute la liste a été traitée
- Le même traitement est appliqué à tous les items.

Un schéma possible est le suivant :

LISTE : (n1 n2 - ni - - np)



FAIT : PB

Travail T Répété pour tous les éléments de la liste (ici : confrontation de PB et item ni i (1,p) pour décider si ni doit etre conservé ou non)

Résultat : liste modifiée par effect du travail T. (ici : liste initiale dont on a enlevé les éléments non significatifs).

Figure 15.

ii) Lors de la sélection fine, l'application des méta-règles (cf IV-3.2.4) amène une évaluation puis un classement des plans possibles pour aborder un problème. Après l'application des méta-règles de type 1 (modification éventuelle de la liste des plans possibles) nous sommes revenus dans une organisation telle que décrite figure 15 : cette fois, la liste de départ est la liste des numéros de plans, le travail T est l'évaluation des plans nj par l'application des méta-règles ; Toutes les évaluations peuvent être traitées en parallèle. Les valeurs associées à chaque plan permettent alors le classement désiré. Une condition de synchronisation peut être d'attendre que toutes les évaluations soient faites pour exécuter le tri.

V-4-2 Parallélisme lors de l'exécution des plans;

i) Le parallélisme est possible lors de l'exécution d'étapes d'un plan sous SUC (à condition d'avoir indépendance entre les étapes : cf IV.2.2.1 et V.3), sous CHOIX, sous POURTOUT et sous POURUN. Dans ce cas la liste d'items est la liste des étapes, le travail T à faire sur chaque étape est l'interprétation de cette étape. La liste "résultat" est une liste des retours (valeurs retournées : échec - succès de l'étape, valeur symbolique). Les contraintes de synchronisation peuvent être variables et exprimés formellement : ainsi, sous SUC de p étapes ou POURTOUT (travaillant avec un ensemble de p éléments) on peut arrêter l'exécution de (p-1) tâches et déclarer un échec dès qu'une tâche renvoie un échec. De même sous CHOIX ou POURUN, le premier succès rencontré permet de conclure et poursuivre (autorisant l'arrêt des autres tâches).

ii) Le parallélisme est également possible lors de l'exécution des actions conditionnelles (si (cond) (action)). L'évaluation de la condition et l'exécution de la partie action peuvent être lancées simultanément. Les contraintes de synchronisation sont alors : "on s'arrête et on renvoie faux dès que (cond) ou (action) renvoie NIL" : il se peut en effet que l'évaluation de (cond) soit plus longue que l'exécution de (action) et que l'on puisse décider que l'action échouera même si la condition s'avère être vraie. Ce peut être le cas pour les conditions problèmes en particulier.

" On renvoie un succès si les deux tâches aboutissent à un succès. En particulier : le résultat vrai de (cond) permet de valider le travail de (action).

Le problème qui reste en suspend concerne évidemment la restauration de l'état antérieur si des actions non encore validées modifient des structures internes (cas de l'affectation en particulier). Ce type d'approche nécessite de dater les objets et de gérer le temps - Ces problèmes de gestion de temps, de datations des événements , objets, faits dans les systèmes experts sont à notre connaissance encore mal étudiés mais constituent sans doute une voie où il convient d'engager des travaux.

V-4-3 Conclusion

L'étude des possiblités de traitement parallèle dans CAMELIA n'a pas été l'objet d'expérimenations. Il nous semble malgré tout que la figure 16 suivante peut constituer une approche de ce problème : un premier objectif serait de faire piloter k processeurs esclaves effectuant le travail pour les p items par un processeur maître.

liste départ LD : (n_1 n_2 ... n_j ..... n_p)

+----------------------+
: item n° n_i :
: independant de :
: n_j $\forall i \in \{1,p\}$ $j \neq i$:
: :
+----------------------+

fait : F

Travail : T - à faire pour tous les éléments de LD.
- T produit une valuation vi du fait F pour tout item ni ($i \in (1,p)$)
La valuation peut être
- une décision concernant nj :
(par exemple vj = 1 si nj doit être conservé.
vj = 0 si nj doit être éliminé.

- une valeur calculée
- une marque de succès (1) ou d'échec (0) si T est une interprétation d'étape de plan, les items étant des étapes indépendantes.

Résultat : Liste finale LD : (n1 v1 n2 n2...nj vj... np vp).
(liste de départ pour lequel chaque élémént est associé à la valuation produite par T).

Contraintes : synchronisation, gestion des différents processus;
- arrêt lorsque LD est entièrement décrite.
- arrêt dès qu'une conditon est atteinte sur une valuation (arrêt dès qu'un échec (vi = 0) ou dès qu'un succès (vi = 1)...).

figure 16

Le processeur maître aurait à gérer la distribution des tâches, les valeurs retournées par les esclaves, la datation des objets. La détermination précise de k serait délicate. L'habitude que nous avons acquise avec CAMELIA laisse penser qu'un k = 5 serait un bon ordre de grandeur. Plusieurs raisons à cela :

- le nombre d'étapes d'un plan dans une succession SUC ou un CHOIX dépasse rarement 10, exceptionnellement 15 : Ainsi en attaquant sous SUC avec les 5 premières étapes, une partie non négligeable du travail est faite lorsque l'étape la plus lente renvoie son résultat. Si le plan doit échouer on peut souvent le savoir dans ce même temps (très souvent un plan échoue rapidement s'il doit échouer, il est plus rare qu'une méthode démarre bien et échoue beaucoup plus loin).

- pour tout ce qui est filtrage par exemple, on sait que lorsqu'il y a échec, très souvent la décision est prise rapidement. Les mécanismes d'unification sont lents surtout pour les unifications qui aboutissent. Mettre quelques processeurs en parallèle pour effectuer un filtrage entre un fait F et un ensemble E de règles doit permettre d'atteindre des équilibres du type : "un ou deux processeurs sont "occupés" à une unification qui

"dure", les autres éliminant très vite toutes les autres règles qui ne s'appliquent pas".Des expériences mériteraient d'être faites pour voir les types d'équilibres qui peuvent être atteints. Même si les résultats ne sont pas concluants, cela aura sûrement le mérite d'obliger à développer quelques outils de mesures de performances qui font actuellement cruellement défaut autour des systèmes experts.

CHAPITRE VI

CONCLUSIONS

VI - CONCLUSIONS

Notre travail a permis de prendre du recul par rapport à la possibilité d'avoir une activité mathématique avec les ordinateurs.

. Un premier constat a pu être fait : celui de la séparation quasi totale jusqu'à maintenant entre l'activité relevant de la démonstration automatique de théorèmes et l'activité relevant du calcul algébrique. Ces deux activités se sont déroulées sur des voies parallèles au point de s'ignorer bien souvent.

. Un second constat concerne les approches avec lesquelles ces activités ont été abordées : l'approche algorithmique et l'approche heuristique ont été utilisées à parts sensiblement égales, bien souvent sur des problèmes semblables et avec des succès comparables. Rares ont été les systèmes qui ont essayer de combiner ces deux approches pour tenter de tirer parti, au mieux, des avantages de chacune des approches. Une mention particulière de MACSYMA doit être faite à ce sujet pour le calcul algébrique : il apparaît que c'est le seul système où l'intégration de ces approches se soit faite de façon significative.

Notre idée centrale est que pour faire des mathématiques avec un ordinateur, ces aspects ne doivent pas apparaître comme exclusifs, à fortiori concurents, mais comme complémentaires. Ils doivent coopérer et donc cohabiter, ce qui signifie qu'il est indispensable qu'un système mathématicien puisse manipuler à la fois l'inférence, des preuves et le calcul symbolique, en faisant appel à des algorithmes et à des heuristiques. Il apparaît comme fondamental :

i) qu'un calcul symbolique puisse se dérouler sous contrôle d'un mécanisme d'inférence capable de prouver des propriétés sur les objets manipulés ;

ii) qu'un raisonnement, qu'une chaîne déductive puisse faire appel pendant un temps donné à des mécanismes puissants de calcul.

iii) que l'on puisse faire appel à des algorithmes puissants lorsque c'est possible. L'efficacité et la sécurité ainsi atteintes doivent être prises en compte. Par contre, les conditions de déclenchement de ces algorithmes (décision d'appliquer un algorithme, choix de l'algorithme, choix du moment où l'on déclenche l'algorithme, ...) doivent pouvoir être déterminées au mieux. Ceci ne peut être fait que par des procédés heuristiques utilisant les données du problème à résoudre.

La composition de ces différents aspects correspond à ce que nous avons voulu exprimer en parlant de comportement du mathématicien.

L'analyse du comportement des savoirs et des savoir-faire du mathématicien nous a amené très tôt (VIVET 77) à l'idée de ce que l'on appelle communément maintenant les systèmes experts. Une partie essentielle de notre travail a alors

été de rassembler les outils nécessaires pour concevoir et construire un système montrant que ces idées sont réalisables.
Ainsi sont arrivés REDUCE et R-LISP. Ainsi est né CAMELIA.

. La conception de CAMELIA se justifie d'elle-même par les éléments que nous venons de décrire :

* CAMELIA intègre effectivement des possibilités de preuves et de calculs capables de coopérer : ainsi dans un processus d'intégration de fractions rationnelles où apparaissent des lignes trigonométriques, le moteur d'inférence est capable de prouver qu'une intégrande en x est impaire pour décider du changement de variable u = cos x. Nous n'avons pas cherché la complexité dans les preuves conduites. Les travaux de Dominique PASTRE (PASTRE 77-79-82) montrent clairement que des preuves parfois très difficiles peuvent être conduites avec les méthodes que nous utilisons. Il nous a suffit de montrer qu'une preuve pourrait intervenir au cours d'un calcul.

* Cela nous a conduit à l'idée importante de condition-problème qui revient à dire que dans un système qui incorpore des mécanismes de preuve, toute propriété p, prouvable par le système, peut devenir condition et être utilisée avec des écritures du type SI p ALORS ...

* CAMELIA permet de contrôler avec des heuristiques, le déclenchement d'algorithmes pour résoudre des problèmes ; les règles (filtres) exprimées dans la base de connaissances peuvent décrire les conditions d'application d'un algorithme et commander, si les conditions requises sont satisfaites, le déclenchement de cet algorithme. L'exploitation du résultat étant décrite dans la règle, ce résultat peut faire l'objet d'un traitement heuristique (qui peut lui-même commander le déclenchement d'un algorithme).

. La réalisation de CAMELIA a permis de faire un travail fondamental en termes de systèmes experts :

- Tout d'abord la définition d'un langage pour exprimer l'expertise du mathématicien. Ce langage permet de décrire des savoirs (connaissances "sûres") mais aussi des "savoir-faire", des méthodes. Ce langage, extensible, s'est montré suffisamment souple et flexible pour exprimer des connaissances dans des domaines variés : calcul intégral, sommations de formules combinatoires, preuves de parité, détermination de limites, calcul de développements limités (mais aussi vérification de compatibilité de sang en transfusion sanguine).

- Un moteur d'inférence a été construit pour interpréter ces connaissances lors de la résolution d'un problème.

- Ce moteur est caractérisé par le mécanisme à deux niveaux de sélection des connaissances utiles pour résoudre un problème. Un niveau de sélection grossière qui fait un choix rapide des choses envisageables (essentiellement par élimination radicale des choses qu'il n'y a pas lieu de prendre en compte) et surtout un niveau de sélection fine. Celle-ci est

faite par une manipulation effective de méta-connaissances données formellement au système sous forme de méta-règles. Ces méta-règles expriment des critères permettant une évaluation puis la sélection des connaissances qui sont applicables. A ce stade, nous avons pu utiliser récursivement le moteur pour faire les évaluations numériques de COUT et d'ESPOIR qui peuvent être accordées aux méthodes pouvant intervenir pour résoudre un problème. Un usage particulier de la méta-connaissance a pu être fait pour écarter des problèmes dits "absurdes" pour raison sémantique.

Le problème difficle du dialogue homme-machine avec un système expert a pu être mieux cerné.Un apport essentiel de l'approche système expert est évidemment que le système peut justifier à l'utilisateur ses étapes intermédiaires de calcul ou de raisonnement. Les justifications se font dans le langage de l'expert et même si les méthodes décrites et utilisées par le système sont "savantes", elles apparaissent dans des termes compréhensibles aux humains. Les seuls cas échappant à cette possibilité sont ceux qui relèvent de l'appel d'algorithmes puissants puisqu'alors l'aspect "boite noire" cache tous les calculs intermédiaires. Ce qui est alors malgré tout intéressant c'est que les raisons qui ont amené le déclenchement de cet algorithme peuvent avoir été expliquées. Concernant le problème de la distribution du contrôle, du pilotage d'un problème entre l'utilisateur et le système, il est évident que même dans le cas où l'utilisateur souhaite garder le contrôle du choix des méthodes, il est intéressant de disposer d'un système qui suggère à l'utilisateur des choix possibles en exprimant les meilleures possibilités qu'il envisagerait s'il était en mode automatique complet.

- Les résultats de CAMELIA sont encourageants. Les quelques 140 exercices traités dans des domaines différents montrent la généralité de l'approche et sa faisabilité. Le travail de codage des connaissances apparaît comme très important en volume et il n'est pas pensable que ce travail puisse être fait par un seul individu. La diversité des exercices abordés et des chapitres de connaissances que nous avons écrits, montre que la voie est praticable et donne des résultats satisfaisants en particulier avec des utilisateurs peu avertis.

. Le travail avec CAMELIA a, en particulier, permis d'apercevoir des pistes de travail avec les systèmes experts. Les recherches les plus intéressantes et les plus fondamentales qui pourraient être initialisées à partir de là pourraient concerner les moteurs à inférence variable et la prise en compte des possibilités de parallélisme (avec un aspect gestion du temps et datation des évènements intéressant par lui-même).

. Un aspect non négligeable du travail avec cette démarche est d'ordre pédagogique. La nécessité d'isoler l'expertise du mathématicien, d'en assurer un codage et la possibilité de séparer les connaissances sûres (résultats connus, faits mathématiques) des connaissances, incertaines généralement, de type savoir-faire, démarche, intéressent le pédagogue des mathématiques. Cela revient dans le prolongement des travaux de POLYA à permettre l'expression complète de démarches pour résoudre des problèmes, en faisant apparaître comment intervient le plau-

sible, l'incertain, l'heuristique en mathématique. Un autre aspect concerne l'enseignement du calcul algébrique. Isoler et formaliser les connaissances acquises sur la mise en oeuvre d'algorithmes puissants peut être d'un grand secours pour amener les utilisateurs à un meilleur usage des systèmes de calcul formel.

Puisse être CAMELIA le départ de travaux destinés à réinvestir les résultats acquis en démonstration de théorèmes comme en calcul symbolique dans une perspective réelle de Mathématiques Assistées par Ordinateurs .

ANNEXES A

ANNEXE A1 VARIABLES GLOBALES DU SYSTEME

NAVIGAVUE : variable qui permet de piloter les plans "à la main".Si sa valeur est NIL,le système est autonome entièrement sur le problème. Si sa valeur n'est pas NIL, chaque fois que le moteur d'inférence se pose un problème nécessitant la mise en oeuvre d'un plan, le système demande le plan à utiliser. On peut répondre par le numéro d'un plan ou par 999 (ce qui laisse alors le système se débrouiller seul sur le sous-problème en cours).

MSGLEVEL variable qui fixe le niveau de message .Valeur numérique entre 0 et 10.Si MSGLEVEL est faible, le système fournit peu d'explications sur ce qu'il fait. Si MSGLEVEL est élevé, le système justifie ses démarches (et peut devenir très bavard). La valeur 5 donne un niveau raisonnable de messages en fonctionnement normal.

OPORDRE 1 : liste des opérateurs d'ordre 1
OPORDRE 2 : liste des opérateurs d'ordre 2
OPORDRE 3 : liste des opérateurs d'ordre 3
OPORDRE 4 : liste des opérateurs d'ordre 4
OPORDRE V : liste des opérateurs variaires
PLAN : tableau des plans
REG : tableau des règles
ACTIMM : liste des actions immédiates
MACTIMM : liste des M-actions immédiates
ACTIMMFONCT : liste de paires n1 n2
n1 : nom de l'action immédiate
n2 : nom de la procédure exécutant l'action immédiate correspondante.
MACTIMMFONCT : idem ACTIMMFONCT mais pour les M.ACTIONS
COUTACTIMM : liste servant de table (tarif) associant pour chaque action immédiate une estimation de coût (Actions coûteuses, actions peu coûteuses,..) Cette table fournie par l'utilisateur donne une appréciation sur les ressources nécessaires pour exécuter l'action : par exemple, aller chercher l'argument d'une expression est une action peut coûteuse, par contre, effectuer un changement de variable peut être coûteuse.
COUT : variables dans lesquelles s'accumulent les coûts.
ESPOIR : respectivement les espoirs à associer à un plan lors du jugement par les méta-règles. Ces éléments servent à calculer la note permettant d'apprécier l'intérêt d'appliquer un plan pour résoudre un problème.
TABRARETE : table associant à chaque opérateur une appréciation sur sa rareté dans les calculs. Cela facilite le choix de plans avec des méta-règles. L'idée est la suivante : si un plan impose un opérateur rare et si l'expression contient cet opérateur rare, augmenter l'espoir que l'on met dans ce plan.

ANNEXE A 2. - SYNTAXE DES PLANS

```
<Plan> = '(< nom plan> (DESCRIPTEURS <liste P.descripteurs>)
                        POUR <pbtype> ESSAYER
                             <plantype>)
 <nomplan> : = A...A 9...9 où A sont des lettres
                           9 sont des chiffres
 <liste P.descripteurs>   :=   <P.descripteur>   <liste   P.descrip-
 teurs>
 <P.descripteur> : = VARSUBST (<listvar>)
                   / OTOPEX (<listop>)
                   / O1OPEX (<listop>)
                   / COUTMETHODE <num>
                   / MSGSUCCES <mess>
                   / MSGECHEC <mess>
 <listvar> := <identificateur> <listvar>
 <listop>  := <opérateur> <listop>
 <num>     := <nombre entier>
 <mess>    := message (chaîne de caractères entre ")
 <Pb type> := <type Pb terminal>
 <Plantype>:= (CHOIX n <plantype 1>...<plantype n>)
            / (SUC   n <plantype 1>...<plantype n>)
            / (POURUN   <var> <liste> <plantype>)
            / (POURTOUT <var> <liste> <plantype>)
            / (SI <condition> <plantype>)
            / <Pb type>
            / <Pb terminal>
 <type Pb terminal> := (PRV (PAIRE   f x ))
                     / (PRV (IMPAIRE f x ))
                     / (CALCUL (INT f x) R)
                     / (CALCUL (LIM    f x pt ) R)
                     / (CALCUL (DEVLIM f x pt n) R)
                     / (EVALUER (SIG F K K1 K2) R)
                     / (RAMENER (APFORBIN (SIG F K K1 K2)) R)
                     / (EFFECTUER   (CHGVARLIE  (  SIG  F  K K1 K2 )
                     NV EX) R)
```

Note : la paire composée des 2 mots clés qui caractérisent le problème sera appelée "Problème caractéristique".
exemple : (PRV PAIRE) (CALCUL PRIM) (CALCUL LIM) sont des problèmes caractéristiques.

<Pbterminal> : <type Pb terminal> dans lequel il ne reste plus de variables substituables : problème complètement défini.
<condition> : cf annexe A5

ANNEXE A 3.-

SYNTAXE DES REGLES

Une règle est définie en MODE LISP par :
<règle>:= (<nom reg><liste R.descripteurs>)
<nom reg>:= A...A9...9 où (A sont des lettres
(9 sont des chiffres

Notation : la sous-chaîne de caractères AA....A de <nom reg> sera appelée "caractéristique" de la règle.

<liste R.descripteurs>:= (<R.descripteur><liste R.descripteurs>)
<R.descripteur>:= VARSUBST (<listvar>)
/ PARAM (<listevar>)
/ CONDIT (<condition>)

La réécriture associée à <règle> est définie en MODE ALGEBRAIC par :

<nom reg> := REEC (<mg>, <md>) où <nomreg> est le même que le premier élément de la règle.
REEC est déclaré comme un opérateur algébrique. Ceci permet de donner les règles sous forme algébrique complète usuelle au lieu de les fournir sous forme de listes préfixes.
<mg> := expression algébrique, membre gauche de la règle
<md> := expression algébrique, membre droit de la règle.

La réécriture peut également être définie en MODE SYMBOLIC par :
<nom reg> : = ' (REEC <mg> <md>)
<mg> et <md> sont alors l'expression des membres gauche (resp. droit) de la règle, donnés sans forme de listes préfixes.

ANNEXE A 4.-

SYNTAXE DES META-REGLES

Les Méta-règles agissent sur des variables globales (COUT, ESPOIR, LPP, NEWLPP) pour évaluer l'intérêt d'appliquer un plan PL à un problème PB (COUT et ESPOIR sont des variables dans lesquelles on estime le coût et l'espoir associés au plan PL pour aborder le problème PB. NEWLPP est la liste de plans possibles modifiée (plans rajoutés ou écartés) LPP est la liste des plans a priori possibles).
Elles ont la forme :

```
<M-reg> : = '(SI <cond 1> <estimation 1>)
          / '(SI <cond 2> <estimation 2>)
          / '(SI <cond 3> <estimation 3>)
<estimation 1> : = (!%ATTRIB NEWLPP <fct LPP-PB>)
<estimation 2> : = (!%ATTRIB COUT   <fct cout 2>)
                 / (!%ATTRIB ESPOIR <fct espoir 2>)
<estimation 3> : = (!%ATTRIB COUT   <fct cout 3>)
                 / (!%ATTRIB ESPOIR <fct espoir 3>)
<cond1> : = <condition >
<cond2> : = <condition >
<cond3> : = <condition >
```

Les méta-règles se trouvent classées en 3 types :

Type 1: correspondant à <estimation 1> règles qui manipulent la liste des plans possibles (enlèvent ou ajoutent des plans, et calculent NEWLPP = f (LPP,PB)

Type 2 : <fct cout 2> et <fct espoir 2> et <cond 2> ne dépendent que du plan PL que l'on examine.

Type 3 : <fct cout 3> et <fct espoir 3> et <cond 3> dépendent à la fois du plan PL et du problème PB

Note. Les conditions <cond 1> <cond 2> <cond 3> peuvent faire intervenir des Méta-Actions (actions pouvant agir sur les actions immédiates des plans).

ANNEXE A 5.-

FORME DES CONDITIONS DANS CAMELIA

<condition> : = (OU <condition> <condition> ... <condition>)
/ (ET <condition> <condition> .... <condition>)
/ (NON <condition>)
/ (<predicat> <args>)
/ (<obj> EST <obj>)
/ (<obj> NESTPAS <obj>)

<predicat> : nom d'une fonction renvoyant une valeur logique NIL pour FAUX, tout autre chose que NIL pour VRAI

<args> : liste des arguments nécéssaires pour l'évaluation du prédicat.

<obj> : atome ou liste

Note : <args> ou <obj> peuvent faire intervenir des actions immédiates ou des M.actions immédiates.

Les prédicats actuellement utilisables sont :

ENTIER <x> : vrai si <x> est un nombre entier
POSITIF <x> : vrai si <x> est un nombre positif
NEGATIF <x> : " négatif

TRIGO <ex> : vrai s'il y a une ligne trigonométrique dans <ex>
TRIGOH <ex> : " hyperbolique "
LOGEXP <ex> : vrai s'il y a un logarithme ou une exponentielle dans <ex>

VUS <x> <ex> : vrai si le symbole <x> apparait dans <ex>
NONVUS <x> <ex> : vrai si <x> n'apparait pas dans <ex>

VUFORMTERM <F> <PARA> <ex> : renvoie NIL si aucun terme de la forme F n'apparaît dans <ex>. La forme <F> est donnée explicitement par son "pattern", ses descripteurs (variables substituables, paramètres formels, condition) éventuels. Si la forme <F> apparait au moins une fois, VUFORMTERM renvoie une liste de paires ((TT1 S1) (TT2 S2)...(TTn Sn)) où les TTi sont les termes trouvés dans <ex> de la forme du "pattern" de <F>. Les Si sont les substitutions nécessaires. Elles ont la forme (T (V1 t1) (V2 t2)....(Vn tn) NIL) où Vi sont les variables substituables et ti les termes à substituer.

RESULTATCONNU <e> <nom> : NIL si <e> n'apparait pas comme membre gauche d'une règle (tableau REG) parmi les règles caractérisées par le nom <nom>. Sinon, le membre droit de la règle concernée est retourné avec les substitions faites.

LINEAIRE <ex> <x> : renvoie NIL si <ex> n'est pas linéaire en <x>. Sinon, <ex> est une forme a. <x> + b et la paire (a b) instanciée est retournée

MONOMEP <ex> <x> : renvoie NIL si <ex> n'est pas un monôme en <x>. Sinon, <ex> est une forme a. <x> ** n et la paire

(a n) instanciée est retournée.

POLYNOMEP <ex> <x> : renvoie NIL si <ex> n'est pas un polynome en <x>, renvoie T sinon.

FORME <ex> <Nom de forme> <paramètres> : vrai si <ex> s'instancie avec la forme donnée dans <Nom de forme>. Les paramètres sont des variables substituables pour lesquelles on impose la substitution. <Nom de forme> défini une règle conditionnelle.

Exemple : avec

```
LISP ;
MONOMAXN : = '(FORMAXN (DESCRITEURS VARSUBST (A X N)
                                     PARAM    (X)
                                     CONDIT (ET (NONVUS X A)
                                                (NONVUS X N))) ;
ALGEBRAIC ; FORMAXN:=A * X ** N ;
l'appel FORME   ('(TIMES   (SIN   PHI)   (EXPT   X   (PLUS   KK   33)),
                MONOMAXN, '(X));
renvoie la liste de substitutions
(T (A (SIN PHI)) (X X) (N (PLUS KK 33))
```

FORMEXPLICITE <ex> <param> <modèle> : marche comme FORME donné ci-dessus mais la forme est donnée explicitement par <modèle> alors que dans FORME la forme est donnée par un nom de variable à laquelle est affectée la forme.

Cas particulier : prédicats manipulant des prédicats

Notons <appel> une forme (<predicat> <args>) dans laquelle un des arguments de <args> est un joker noté (?). Si <L> est une liste d'éléments (atomes ou listes), les prédicats suivants sont disponibles.

TOUSVRAI <L> <appel> est vrai si le prédicat de l'appel est vrai pour toute valeur du joker choisie dans <L>

UNVRAI <L> <appel> est vrai si le prédicat de l'appel est vrai pour une valeur du joker choisie dans <L>

AUCUNVRAI <L> <appel> est vrai si le prédicat de l'appel est faux pour toute valeur du joker choisie dans <L>

exemple : voir qu'une expression ex est un polynôme en x peut se vérifier en disant

```
(OU   (MONOMEP EX  X)
(ET   (TOUSVRAI (CDR EX) ( MONOMEP ? X ))
(VUS  (% OP EX) ('PLUS 'DIFFERENCE 'TIMES)) ))
```

Cas particulier : Les conditions problèmes :

Elles ont la forme d'une propriété P, prouvable par le système (par exemple : (PAIRE f x)). Dans ce cas, le mot clé (ici PAIRE) n'est pas associé à une procédure, mais la condition exprime un prédicat que l'interpréteur va prouver. La preuve renvoie alors NIL (interprété comme faux) ou une valeur autre

que NIL (interprétée comme vrai). Dans ce cas, l'évaluation de condition lance l'exécution du moteur d'inférence sur la propriété à prouver au lieu de lancer le mécanisme d'évaluation d'expressions logiques.

Usage : Cela permet d'écrire dans le langage des plans des conditions comme (si (ET (PAIRE f x) (IMPAIRE g x))
(actions ....))
De telles conditions peuvent apparaître dans les plans ou dans les règles .

ANNEXE B

PROCEDURES DE CAMELIA

UNIFIP

ARGUMENTS
RG : Règle
ex : Expression
MGRG : Membre gauche de la règle

RESULTATS
Substitution si la règle s'applique
(T((V1 T1) (V2 T2)...(Vn Tn))
(NIL (...) .....)

EXEMPLE
UNIFIP (RG,ex,FMG(RG))

UNIFIPQ

ARGUMENTS
LVSVBST : Liste de variables substituables
ex : Expression
MGRG : Membre gauche de la règle

RESULTATS
Substitution si la règle s'applique
(T((V1 T1)(V2 T2)...(Vn Tn)))
(NIL (...) .....
Chaque couple (Vi Ti) donne le terme Ti qui doit se substituer à la variable Vi

EXEMPLE
UNIFIPQ ('(u v w), ex, MRG)

UNIFIE

ARGUMENTS
RG : Règle
ex : Expression
MGR : Membre gauche de la règle
MDR : Membre droit de la règle

RESULTATS
MDR modifié par la substitution liée à l'unification de ex avec MGR
ou NIL si échec

EXEMPLE
UNIFIE (Plan (4),PB (21), PBTYPE (Plan (4), PLANTYPE (Plan 4))

EXUNIFIP

ARGUMENTS
ex : expression
Param : liste de paramètres actuels
RG : règle pouvant être conditionnelle, avec paramètres, ...

RESULTATS
(T (vi ti)...)
ou
(NIL (vi ti)...)

EXEMPLE
EXUNIFIP (ex,'(u), Reg (23))

EXUNIFIE

ARGUMENTS
ex : expression
Param : liste de paramètres actuels

RG : Règle conditionnelle paramètrée

RESULTATS NIL si échec
ou
membre droit de la règle modifié par la substitution ex MG(RG)

EXEMPLE EXUNIFIE (ex,'(u), Reg (28))

APPLISUBST

ARGUMENTS Subs : substitution (telle que renvoyée par les procédures d'unification : forme générale :
(T (V T) (V T)...(V T) NIL)
1 1 2 2 n n
ex : expression contenant des variables substituables (Vi d'une substitution).

RESULTATS ex modifiée par application de la substitution subs.

FORME

ARGUMENTS ex : expression
Nom de forme : nom associé à un pattern
PARAM : liste de paramètres

RESULTATS NIL
ou bien
liste (T(vi ti)...) donnant les substitutions

EXEMPLE
```
LISP MONOMAX : = '(FORMAXN (DESCRIPTEURS
VARSUBST ( ) PARAM ( ) CONDIT ( ) ));
ALGEBRAIC FORMAXN : = A*X**N ;
FORME (ex,MONOMAX,'(X)) ;
```

FORMEXPLICITE

ARGUMENTS ex : expression
paramet : liste de paramètres actuels
modèle : forme (pattern (DESCRIPTEURS VARSUBST ( ) PARAM ( ) CONDIT ( )))

RESULTATS (T ((vi ti) ..))
NIL
substitution

EXEMPLE
```
mode : = '((Times A (expt x n))
(DESCRIPTEURS.
          VARSUBST (x A n)
          PARAM   (x)
          CONDIT  (ET(NONVUS  x  A)(NONVUS  x
          n)))) ;
FORMEXPLICITE(ex,'(u),mode) ;
```

APPLIPAQUET1

ARGUMENTS nom : nom attaché aux règles à essayer
param : liste de paramètres actuels
ex : expression

RESULTATS (NIL (exp)) si échec de toute règle
ou
(nr (exp))
nr : n° de la première règle qui s'est appliquée
exp : résultat de l'application de la règle à ex.

EXEMPLE APPLIPAQUET1 ('LIMSOMME,'(X),ex) ;

RESULTATCONNU

ARGUMENTS ex : expression
NOM : nom du paquet de règles où l'on cherche

RESULTATS NIL ou bien valeur du résultat

EXEMPLE M RESULTATCONNU ('PAIRE(cos X)X),'paire);

VALCONDITION

ARGUMENTS C : condition

RESULTATS NIL ou T

MONOMEP

ARGUMENTS ex : expression courante
x : liste d'un élément (l'indéterminée)

RESULTATS NIL si ex n'est pas monome en x
n
((A) (n)) si ex est monome de la forme $A\ x$

POLYNOMEP

ARGUMENTS ex : expression courante
x : indéterminée du polynome

RESULTATS NIL ou T

TOUSVRAI

ARGUMENTS L : liste d'éléments
appel : forme '(Fct a1 ai-1 !? ai+1 an)
dans laquelle un arg est un jocker (!?) par courant L

RESULTATS Vrai si V !? L
Fct (a1, a2, ai-1, ai+1 an) est vrai

UNVRAI

ARGUMENTS L : liste d'éléments
appel : cf TOUSVRAI

RESULTATS Vrai si !? L pour lequel
Fct (a1, a2, ai-1,!?, ai+1, an) est différent de NIL

AUCUNVRAI

ARGUMENTS L : liste d'éléments
appel : cf TOUSVRAI

RESULTATS Vrai si !? L
FCT (a1, a2, , ai-1,!?, ai+1, an) = NIL

REARRANGE

ARGUMENTS LPP : liste des plans possibles
PB : pb en cours

RESULTATS Liste des plans possibles triés par ordre de mérite en tenant compte de la note attribuée par application des méta-règles.

JUGEMENT

ARGUMENTS PL : plan à juger
PB : problème pour lequel on juge le plan
LPP : liste des autres plans possibles
LISMREG : liste des numéros de méta-règles permettant d'effectuer le jugemet.

RESULTATS Renvoie LPP modifié par les méta-règles.
. Les variables COUT et ESPOIR ont été modifiés pendant l'application des méta-règles.

NOTEFINALE

ARGUMENTS COUT : cout du plan calculé par les méta-règles
ESPOIR : espoir associé au plan (calculé par les méta-règles).

RESULTATS Note calculée pour estimer l'intérêt d'appliquer le plan PL pour résoudre le problème PB.

FONCTIONS DE SERVICE D'EXTRACTION D'INFORMATION UTILE
]]]

OBJECTIFSAVOIR

ARGUMENTS L : identificateur forme AAA. A999....9

RESULTATS extraie AA...A (généralement le nom d'un concept manipulé par les règles ou les plans).

EXEMPLE PAIRE 4 (4ème règle du chapitre PARITE, INT36 (36ème plan du chapitre intégration.

OBJECTIFPB

ARGUMENTS PB problème terminal

RESULTATS Objectif du problème : forme
(PRV PAIRE), (CALCUL LIM), (CALCUL INT),...

EXEMPLE OBJECTIFPB (PB (60)) ;

COUTMETHODE

ARGUMENTS PL : plan

RESULTATS Valeur du cout méthode associée à un plan dans la Base de connaissances.

FVSUBST

ARGUMENTS R : règle ou plan

RESULTATS Accès aux variables substituables d'une règle ou d'un plan

OTOPEX

ARGUMENTS PL : plan.

RESULTATS liste des opérateurs qui doivent être effectivement présents dans l'expression courante pour que le plan mérite d'être tenté

O1OPEX

ARGUMENTS PL : plan

RESULTATS liste des opérateurs dont un au moins doit être effectivement présent dans l'expression courante pour que le lan mérite d'être tenté.

OPRINCEX

ARGUMENTS PB : problème terminal

RESULTATS opérateur principal de l'expression du problème.

EXEMPLE l'intégrande pour une intégrale, fonction dont on cherche a calculer la limite, le développement limite...

CUMULCOUTACTIMM

ARGUMENTS PL : plan

RESULTATS Cumul du coût des actions immédiates d'un plan établi à partir d'un "tarif"des actions.

RAROTO

ARGUMENTS PL : plan

RESULTATS cumul des raretés des opérateurs imposés par le plan (OTOPEX)

PBTYPE

ARGUMENTS PL : plan

RESULTATS Problème type du plan

VOLPBTYPE

ARGUMENTS PL : plan problème.

RESULTATS : volume du problème type du plan

PLANTYPE

ARGUMENTS PL : plan

RESULTATS plan type du plan

VOLPLANTYPE

ARGUMENTS PL : plan

RESULTATS Volume du plan type du plan

ACTION

ARGUMENTS I : n° de l'action
PL : plan

RESULTATS i ième action d'un plan si le plan est un (SUC...) ou (CHOIX..)
si i=1 on retourne PL
note si l'action résultante est une action immédiate, on supprime le % pour éviter l'exécution de cette action par EXECACTIMM.

ORDROP

ARGUMENTS ex

RESULTATS ordre de l'opérateur principal de ex

OP

ARGUMENTS L

RESULTATS Opérateur principal de l'expression L

EXEMPLE La vérification sémantique que L est effectivement un opérateur n'est pas faite

ARG1

ARGUMENTS L

RESULTATS Argument 1 de l'expression L

ARG2

ARGUMENTS L

RESULTATS Argument 2 de l'expression L

ARG3

ARGUMENTS L

RESULTATS Argument 3 de l'expression L

ARGI

ARGUMENTS I L

RESULTATS Ie argument de l'expression L

FMG

ARGUMENTS r : règle

RESULTATS accès au membre gauche d'une règle

FMD

ARGUMENTS r : règle

RESULTATS accès au membre droit d'une règle.

OPERATEURS

ARGUMENTS ex : expression

RESULTATS liste des opérateurs présents dans l'expression.

TERMES

ARGUMENTS OPER : opérateur principal des termes cherchés
ex : expression dans laquelle se fait la recherche

RESULTAT liste des termes (sous termes de ex) commençant par l'opérateur OPER.

FONCTIONS D'INTERET GENERAL

]]]]]]]]]]]]]]]]]]]]]]]]]]]

LISTCAR

ARGUMENTS u : liste d'association
forme ((x1 a1)...(xi ai) (xn an))

RESULTATS renvoie la liste des CAR
(x1.....xi.....xn)

ENTIER

ARGUMENTS x

RESULTATS renvoie T si x est un nombre entier

NEGATIF

ARGUMENTS x

RESULTATS renvoie T si x est un nombre négatif

POSITIF

ARGUMENTS x

RESULTATS renvoie T si x est un nombre positif

INTERVAL

ARGUMENTS n1,n2 : 2 nombres entiers (n1 < n2)

RESULTATS liste des nombres de l'intervale (n1,n2)

ENLEVE

ARGUMENTS L : liste
A : élément (atome ou liste)

RESULTATS Liste L dont on a enlevé la 1ère occurrence de A.

ENLEVELIST

ARGUMENTS L : liste
LA : liste d'éléments

RESULTATS chaque élément de LA voit sa première occurence retiré de L ; le résultat est L ainsi réduit.

LOOK

ARGUMENTS L : liste de paires ((a aa)..)
a : élément

RESULTATS renvoie le CDR de la paire (a aa) qui à a comme CAR

SUIVANT

ARGUMENTS L : liste d'éléments (a aa b bb...)
a : élément

RESULTATS donne l'élément aa suivant a dans L . NIL sinon

MEMBREP

ARGUMENTS x : élément ou liste
L : liste

RESULTATS T si x L
NIL sinon.

NBOC ARGUMENTS x : élément
ex : expression (liste)

RESULTATS nombre (entier) d'occurrences de l'élément x dans l'expression ex.

SOUSTERME

ARGUMENTS x : sous terme cherché
L : expression dans laquelle on cherche si on trouve le sous terme x.

RESULTAT T si X est un sous terme de L
NIL sinon

EXEMPLE SOUSTERME ('(CNP N K),'(TIMES (EXPT(MINUS 1) KK) (CNP N K))); renvoie vrai (T).

SURTERME

ARGUMENTS x : terme
ex : expression (liste)

RESULTATS liste des termes de EX qui ont x comme fils

VUFORMTERM

ARGUMENTS modèle : modèle (idem FORMEXPLICITE) de la forme '(pattern (DESCRIPTEURS VARSUBST() PARAM ()CONDIT ()))
EX : expression

RESULTAT renvoie les occurences dans EX du modèle (terme, substitutions nécessaires). Ceci est fait en tenant compte de la compatibilité entre les paramètres du modèle et actuels (variables substituables dont on impose l'instanciation) et en s'assurant que la condition (CONDIT) associée au modèle est vraie pour les substitutions résultantes. Le résultat se présente sous la forme (T1 s1)...(Ti si)...(Tn sn)) où les Ti sont les termes trouvés, les si les substitutions associées.

EXEMPLES : Si F : ='((TIMES (QUOTIENT 1 (PLUS M 1)) (CNP N M)) (DESCRIPTEURS VARSUBST (N M) PARAM (N))) ;

et EX :='(TIMES 3(TIMES (QUOTIENT 1(PLUS K 1) (CNP NN K)))

Alors VUFORMTERM (F,'(NN),EX) ; renvoie
((TIMES (QUOTIENT 1 (PLUS K 1)(CNP NN K)))(T
(N NN)(M K) nil))

si F : = '((PLUS K P)(DESCRIPTEURS VARSUBST
(K P) PARAM (K) CONDIT (NONVUS K P))) ;
EX :='(PLUS (PLUS KK Q)(PLUS KK (PLUS KK
QQ))) ;
alors VUFORMTERM(F,'(KK), EX) ; renvoie

((PLUS KK Q)(T (K KK) (P Q) nil)) ((PLUS KK
QQ)(T (K KK)(P QQ) nil)))
Sans la condition dans F on récupère en plus
le résultat ((PLUS KK (PLUS KK QQ))(T (K
KK)(P (PLUS KK QQ)) nil))

NVUS

ARGUMENTS x : atome
L : liste

RESULTATS nombre d'apparitions de x dans L.

FVOL

ARGUMENTS ex : expression

RESULTATS nombre (entier) de symboles (volume) de l'expression.

VOL ARGUMENTS L : liste

RESULTATS nombre d'atomes volume de la liste L.

REMPLACE

ARGUMENTS u
v liste ou élément
w liste

RESULTATS remplace u par v dans w (au 1er niveau seulement)

REMPL ARGUMENTS u
v listes ou élément
w

RESULTATS idem REMPLACE mais à toute profondeur.

PROCEDURES POUR RAJOUTER DES CONNAISSANCES

ENRICHIR L'ENVIRONNEMENT

CONNEC1

ARGUMENTS u : nom de connective unaire à déclarer

RESULTATS toutes les connectives unaires

EXEMPLE CONNEC1 ARCSIN

CONNEC2

ARGUMENTS u : nom de connective binaire à déclarer

RESULTATS toutes les connectives binaires

EXEMPLE CONNEC2 EGAL

CONNEC3

ARGUMENTS u : nom de connective ternaire à déclarer

RESULTATS toutes les connectives ternaires

EXEMPLE CONNEC 3 EQUATION

CONNEC4

ARGUMENTS u : nom de connective 4-aire à déclarer

RESULTATS toutes les connectives 4-aires

EXEMPLE etc....

CONNECV

ARGUMENTS u : nom de connective variaire à déclarer

RESULTATS toutes les connectives variaires

EXEMPLE etc....

REGLEBIENCONNUE

ARGUMENTS Descrip : descripteur de la règle
Propriété : réécriture affectée au CAR du descripteur

RESULTATS Règles reclassées par objectif

EXEMPLE
```
REGLEBIENCONNUE(''(nom(DESCRIPTEURS VARSBST()
                                    PARAM ()
                                    CONDIT()
                                    )),
                '(REEC<mg><md>)) ;
```

Les règles sont reclassées par objectifs.

LIRCHAP

ARGUMENTS L: atome ou liste d'atomes : les atomes sont les DDNAME de chapitres de connaissances à lire

RESULTAT Le(s) chapitre(s) de connaissances est (sont) lu(s). Le prétraitement est fait.

EXEMPLE LIRCHAP 'PARITE ; LIRCHAP '(LIMITES INTEGR) ;

AJOUTACTIMM

ARGUMENTS NA : nom de l'action
NP : nom de la procédure

RESULTATS la paire (NA NP) est ajoutée à ACTIMMFONCT-ACTIMM (liste des actions immédiates) est recalculé.

NOTE : Si NA est NIL alors le nom de l'action, est le nom de la procédure précéde de !%.

PROCEDURES POUR TESTER L'ENVIRONNEMENT
]]]

CONTEXTE

ARGUMENTS Pas d'argument

RESULTATS affiche NAVIGAVUE : variable globale définissant le mode de pilotage
MSGLEVEL : niveau de messages distribués.
PLANOBJ : plans associés aux objectifs accessibles.
REGOBJ : règles associées aux objectifs ou concepts manipulés.

PROCEDURE D'ENTREE/SORTIE
]]]]]]]]]]]]]]]]]]]]]]]]]

ENTREE ARGUMENTS rien

RESULTATS renvoie ce qui est frappé au clavier

ECRIRE ARGUMENTS L

RESULTATS écrit dans la file LISPOUT l'évaluation de L et retourne T

AFFICHE ARGUMENTS niv : niveau
L : atome ou liste de sortie

RESULTATS affiche l'évaluation des éléments de L si niv < MSGLEVEL

MONTRER ARGUMENTS L : atome ou lite d'atomes (ces atomes sont des notations utilisées dans les plans).

RESULTATS écrit dans la LISPOUT le nom de la notation sa VAL (SERV..) et sa VVAL (valeur actuellement associée à la notation).

INTERACTIONS MODE ALGEBRIQUE/MODE LISP

ALGEB ARGUMENTS X

RESULTATS permet d'utiliser la variable LISP x en mode algébrique.

ALGEVAL

ARGUMENTS l liste (<nom proc> <args>) ou <nom proc> est le nom d'une procédure de mode ALGEBRAIC <args> est la liste d'arguments nécessaires à la procédure.

RESULTATS évaluation de la procédure <nomproc> avec les arguments <args>

EXEMPLE ALGEVAL (DLEXP V 4) calcul de développement limité de e**v à l'ordre 4 au voisinage de 0 à l'aide de la fonction algébrique DLEXP.

GESTION DES NOTATIONS

CREER

ARGUMENTS L : atome ou liste d'atomes.
Ces atomes sont des notations utilisées localement dans les plans.

RESULTATS Créé un représentant actuel pour chaque notation. Création de variables de service SERV...

EXEMPLE CREER R1
CREER (R1 R2 R3)

LIBERER

ARGUMENTS L : idem CREER

RESULTATS libère le représentant actuel pour chaque notation.

EXEMPLE LIBERER (R1 R2)

VAL

ARGUMENTS L : notation

RESULTATS accès au nom (SERV..) du représentant actuel de la notation L

VVAL

ARGUMENTS L : notation

RESULTATS accès à la valeur actuellement notée L.

Annexe C

liste des actions immédiates

```
prettyprint actimm$
    14: (!%OP !%ORDROP !%ARG1 !%ARG2 !%ARG3 !%ARGI !%RESULTATCONNU
            !%ATTRIB !%CREER !%LIBERER !%NOUVAR !%VAL !%VVAL !%ECRIRE
            !%MONTRER !%ECRILIS !%ENTREE !%TETE !%QUEUE !%TQ !%TTQ !%TQT
            !%TT !%TQTQ !%TQQTQ !%VALNUM !%OBJ !%ALGEB' !%ALGEVAL
            !%ENTIER !%RESTEDIV !%INTERVAL !%LINEAIRE !%MONOMEP
            !%DEVBINOM !%POLYNOMEP !%COUPE !%REMPL !%NONVUS !%VUS !%NBOC
            !%VOL !%FVOL !%SURTERME !%SIMPLIF !%DERIV !%APPLIPAQUET1
            !%APPLIC !%IPPE !%LIMPLUS !%CHGVARINT !%TERMES !%VUFORMTERM
            !%APPLISUBST
            !%FORMEXPLICITE !%ACTION !%OPRINCEX !%OBJECTIFPB !%OTOPEX
            !%O1OPEX !%PTLIM !%PTDEVLIM !%COUTMETHODE !%CUMULCOUTACTIMM
            !%ENLEVE !%ENLEVELIST)
```

Coût des actions immédiates

```
prettyprint coutactimm$
    15: (COUTACT !%OP 5 !%ORDROP 6 !%ARG1 5 !%ARG2 5 !%ARG3 5 !%ARGI 6
            !%RESULTATCONNU 14 !%ATTRIB 5 !%CREER 5 !%LIBERER 5 !%NOUVAR
            20 !%VAL 5 !%VVAL 5 !%ECRIRE 2 !%MONTRER 2 !%ENTREE 50
            !%ECRILIS 2 !%TETE 4 !%QUEUE 4 !%TQ 4 !%TTQ 4 !%TQT 4 !%TT 4
            !%TQTQ 4 !%TQQTQ 4 !%OBJ 4 !%ALGEB 4 !%ALGEVAL 15 !%VALNUM 4
            !%ENTIER 4 !%RESTEDIV 4 !%INTERVAL 5 !%LINEAIRE 6 !%MONOMEP
            6 !%DEVBINOM 8 !%POLYNOMEP 10 !%COUPE 15 !%REMPL 6 !%NONVUS
            5 !%VUS 5 !%NBOC 8 !%FVOL 8 !%VOL 8 !%SIMPLIF 15 !%DERIV 8
            !%SURTERME 15 !%APPLIPAQUET1 30 !%APPLIC 15 !%LIMPLUS 10
            !%IPPE 10 !%CHGVARINT 15 !%TERMES 10 !%VUFORMTERM 25
            !%APPLISUBST 25 !%FORMEXPLICITE 25)
```

liste des Méta-actions

```
prettyprint mactimm$
    16: (!%!%VOL
            !%!%ACTION
            !%!%COUTMETHODE
            !%!%CUMULCOUTACTIMM
            !%!%VOLPLANTYPE
            !%!%VOLCONDACT1 !%!%VOLPBTYPE !%!%TETE !%!%TQ !%!%OTOPEX
            !%!%O1OPEX !%!%PBTYPE !%!%PLANTYPE !%!%RARETECOMMUNE)
```

Annexe D1

```
IGNORANCE UNIVERSELLE:NE RIEN SAVOIR FAIRE                              $       00000670
PLAN(0):='(PLANNIL ( DESCRIPTEURS VARSUBST (NIMPORTEQUOI) )                     00000680
          POUR NIMPORTEQUOI ESSAYER                                             00000690
          (SI NIL NIL)  )$                                                      00000700
SAVOIR-FAIRE UNIVERSEL:SAVOIR TOUT FAIRE ...                            $       00000710
PLAN(1):='(PLANT ( DESCRIPTEURS VARSUBST (NIMPORTEQUOI) )                       00000720
          POUR NIMPORTEQUOI ESSAYER                                             00000730
          (SI T T))$                                                            00000740
PLAN DEMANDANT UN SOUS-PROBLEME D'AIDE                                  $       00000750
PLAN(2):='(AIDESP=1 ( DESCRIPTEURS VARSUBST (NIMPORTEQUOI)                      00000760
                    )                                                           00000770
          POUR NIMPORTEQUOI ESSAYER                                             00000780
          (SUC 6 (!%CREER R1)(!%ECRIRE (QUOTE NIMPORTEQUOI) )                   00000790
                 (!%ECRIRE "AIDEZ MOI ")                                        00000800
                 (!%ATTRIB (!%VAL R1) (!%ENTREE   ) )                           00000810
                 (!%VVAL R1)                                                    00000820
                 (!%LIBERER R1)                                                 00000830
                 NIMPORTEQUOI            ))$                                   00000840
```

```
OUT REDUFICH$ OFF ECHO$ LISP$                                                  00000010
%=============================================================$               00000020
OUT T$WRITE "ENTREE DU CHAPITRE INTEGR "$  OUT REDUFICH $                      00000030
%=============================================================$               00000040
%%%        RESULTATS CONNUS SUR LES PRIMITIVES                 $               00000050
%=============================================================$               00000060
                                                                               00000070
LISP$                                                                          00000080
REG(20):='(PRIM20 ( DESCRIPTEURS VARSUBST (X)))$                               00000090
REG(21):='(PRIM21 ( DESCRIPTEURS VARSUBST (X)))$                               00000100
REG(22):='(PRIM22 ( DESCRIPTEURS VARSUBST (X)))$                               00000110
REG(23):='(PRIM23 ( DESCRIPTEURS VARSUBST (X)))$                               00000120
REG(24):='(PRIM24 ( DESCRIPTEURS VARSUBST (X)))$                               00000130
REG(25):='(PRIM25 ( DESCRIPTEURS VARSUBST (X)))$                               00000140
REG(26):='(PRIM26 ( DESCRIPTEURS VARSUBST (X)))$                               00000150
REG(27):='(PRIM27 ( DESCRIPTEURS VARSUBST (X)))$                               00000160
REG(28):='(PRIM28 ( DESCRIPTEURS VARSUBST (X)))$                               00000170
REG(29):='(PRIM29 ( DESCRIPTEURS VARSUBST (X)))$                               00000180
REG(30):='(PRIM30 ( DESCRIPTEURS VARSUBST (X)))$                               00000190
REG(31):='(PRIM31 ( DESCRIPTEURS VARSUBST (X)))$                               00000200
REG(32):='(PRIM32 ( DESCRIPTEURS VARSUBST (X)))$                               00000210
REG(33):='(PRIM33 ( DESCRIPTEURS VARSUBST (X)))$                               00000220
REG(34):='(PRIM34 ( DESCRIPTEURS VARSUBST (X)))$                               00000230
REG(35):='(PRIM35 ( DESCRIPTEURS VARSUBST (X)))$                               00000240
REG(36):='(PRIM36 ( DESCRIPTEURS VARSUBST (X A M)                              00000250
                    CONDIT (ET (NONVUS X A)(NONVUS X M)(M NESTPAS -1))))$      00000260
REG(37):='(PRIM37 ( DESCRIPTEURS VARSUBST (X)))$                               00000270
REG(38):='(PRIM38 ( DESCRIPTEURS VARSUBST (X)))$                               00000280
REG(39):='(PRIM39 ( DESCRIPTEURS VARSUBST (X)))$                               00000290
REG(40):='(PRIM40 ( DESCRIPTEURS VARSUBST (X)))$                               00000300
REG(41):='(PRIM41 ( DESCRIPTEURS VARSUBST (X)))$                               00000310
REG(42):='(PRIM42 ( DESCRIPTEURS VARSUBST (X)))$                               00000320
REG(43):='(PRIM43 ( DESCRIPTEURS VARSUBST (X)))$                               00000330
REG(44):='(PRIM44 (DESCRIPTEURS VARSUBST (X M)                                 00000340
                    CONDIT (NONVUS X M)  ))$                                   00000350
REG(45):='(PRIM45 (DESCRIPTEURS VARSUBST (X A)                                 00000360
                    CONDIT (NONVUS X A)  ))$                                   00000370
REG(46):='(PRIM46 (DESCRIPTEURS VARSUBST (X A B M)                             00000380
                    CONDIT (ET (NONVUS X A)(NONVUS X B)                        00000390
                      (NONVUS X M)(M NESTPAS -1))  ))$                         00000400
REG(47):='(PRIM47 (DESCRIPTEURS VARSUBST (X A)                                 00000410
                    CONDIT (ET (NONVUS X A)(POSITIF A))  ))$                   00000420
REG(48):='(PRIM48 (DESCRIPTEURS VARSUBST (X H)                                 00000430
                    CONDIT (NONVUS X H)  ))$                                   00000440
REG(49):='(PRIM49 (DESCRIPTEURS VARSUBST (X A)                                 00000450
                    CONDIT (NONVUS X A) ))$                                    00000460
REG(50):='(PRIM50 (DESCRIPTEURS VARSUBST (X A)                                 00000470
                    CONDIT (NONVUS X A) ))$                                    00000480
REG(51):='(PRIM51 (DESCRIPTEURS VARSUBST (X A)                                 00000490
                    CONDIT (NONVUS X A) ))$                                    00000500
REG(52):='(PRIM52 (DESCRIPTEURS VARSUBST (X A)                                 00000510
                    CONDIT (NONVUS X A) ))$                                    00000520
REG(53):='(PRIM53 (DESCRIPTEURS VARSUBST (X A)                                 00000530
                    CONDIT (NONVUS X A) ))$                                    00000540
REG(54):='(PRIM54 (DESCRIPTEURS VARSUBST (X A)                                 00000550
                    CONDIT (NONVUS X A) ))$                                    00000560
%$                                                                             00000570
ALGEBRAIC $                                                                    00000580
OPERATOR PRIM $                                                                00000590
%SCALAR PRIM20,PRIM21,PRIM22,PRIM23,PRIM24,PRIM25,PRIM26,PRIM27,PRIM28$        00000600
%SCALAR PRIM29,PRIM30,PRIM31,PRIM32,PRIM33,PRIM34,PRIM35,PRIM36$               00000610
%SCALAR PRIM37,PRIM38,PRIM39,PRIM40,PRIM41,PRIM42,PRIM43,PRIM44,PRIM45$        00000620
%SCALAR PRIM46,PRIM47,PRIM48,PRIM49,PRIM50,PRIM51,PRIM52,PRIM53,PRIM54$        00000625
OFF DIV$                                                                       00000630
PRIM20:=REEC(PRIM(X,X),X**2 / 2)$                                              00000640
PRIM21:=REEC(PRIM(1/X,X),LOG(X))$                                              00000650
PRIM22:=REEC(PRIM(E**X,X),E**X)$                                               00000660
PRIM23:=REEC(PRIM(COS(X),X),SIN(X))$                                           00000670
PRIM24:=REEC(PRIM(SIN(X),X),-COS(X))$                                          00000680
PRIM25:=REEC(PRIM(1+TG(X)**2,X),TG(X))$                                        00000690
PRIM26:=REEC(PRIM(1/COS(X)**2,X),TG(X))$                                       00000700
PRIM27:=REEC(PRIM(1/SIN(X)**2,X),-COTG(X))$                                    00000710
PRIM28:=REEC(PRIM(1/(1+X**2),X),ATAN(X))$                                      00000720
PRIM29:=REEC(PRIM(1/(1-X**2),X),ATH(X))$                                       00000730
PRIM30:=REEC(PRIM(RAC2(X),X),(2/3)*X**(3/2))$                                  00000740
PRIM31:=REEC(PRIM(1/RAC2(X),X),2*RAC2(X))$                                     00000750
PRIM32:=REEC(PRIM(SH(X),X),CH(X))$                                             00000760
PRIM33:=REEC(PRIM(CH(X),X),SH(X))$                                             00000770
PRIM34:=REEC(PRIM(1-TH(X)**2,X),TH(X))$                                        00000780
PRIM35:=REEC(PRIM(1/RAC2(1-X**2),X),ASIN(X))$                                  00000790
PRIM36:=REEC(PRIM(A/(X**M),X)  ,(A/(1-M)) / X**(M-1) )$                        00000800
PRIM37:=REEC(PRIM(1/RAC2(1+X**2),X),ACOS(X))$                                  00000810
PRIM38:=REEC(PRIM(1/RAC2(X**2+1),X),ASH(X))$                                   00000820
PRIM39:=REEC(PRIM(1/RAC2(X**2-1),X),ACH(X))$                                   00000830
PRIM40:=REEC(PRIM(1/SIN(X),X),LOG(TG(X/2)))$                                   00000840
PRIM41:=REEC(PRIM(1/COS(X),X),LOG(TG(X/2+PI/4)))$                              00000850
PRIM42:=REEC(PRIM(TG(X),X),-LOG(COS(X)))$                                      00000860
PRIM43:=REEC(PRIM(COTG(X),X),LOG(SIN(X)))$                                     00000870
PRIM44:=REEC(PRIM(X**M,X),X**(M+1)/(M+1))$                                     00000880
PRIM45:=REEC(PRIM( 1 / (A+X)  , X),LOG(A+X))$                                  00000890
PRIM46:=REEC(PRIM((A*X+B)**M,X),(1/A)*((A*X+B)**(M+1))/(M+1))$                 00000900
PRIM47:=REEC(PRIM(A**X,X),(A**X)/LOG(A))$                                      00000910
PRIM48:=REEC(PRIM(1/RAC2(X**2+H),X),LOG(X+RAC2(X**2+H)))$                      00000920
PRIM49:=REEC(PRIM(1/(X**2+A**2),X),ATAN(X/A)/A )$                              00000930
PRIM50:=REEC(PRIM(1/(A**2-X**2),X),LOG((A+X)/(A-X))/(2*A))$                    00000940
PRIM51:=REEC(PRIM(1/RAC2(X**2+A**2),X),ASH(X/A))$                              00000950
PRIM52:=REEC(PRIM(1/RAC2(X**2-A**2),X),ACH(X/A))$                              00000960
PRIM53:=REEC(PRIM(X/(X**2+A),X),LOG(X**2+A)/2)$                                00000970
PRIM54:=REEC(PRIM(X/RAC2(X**2+A),X),RAC2(X**2+A))$                             00000980
%$                                                                             00000990
LISP$                                                                          00001000
%=============================================================$               00001010
%                CALCUL DE PRIMITIVES                       $                 00001020
%=============================================================$               00001030
%RETROUVER UN RESULTAT CONNU $                                                 00001040
PLAN(20):='(INT20 (DESCRIPTEURS VARSUBST (F X R)                               00001050
                   COUTMETHODE 5                                               00001060
                   MSGSUCCES "RESULTAT USUEL "                                 00001070
                   MSGECHEC  "JE NE SAIS PAS TOUT PAR COEUR "                  00001080
                  )                                                            00001090
            POUR (CALCUL (PRIM F X) R) ESSAYER                                 00001100
              (!%ATTRIB R                                                      00001110
                        (!%RESULTATCONNU (PRIM F X) PRIM) )                    00001120
             )$                                                                00001130
%$                                                                             00001140
% INTEGRER UNE DIFFERENCE $                                                    00001150
PLAN(21):='(INT21 (DESCRIPTEURS VARSUBST (F G X R)                             00001160
                   DTOPEX (DIFFERENCE)                                         00001170
                   COUTMETHODE 5                                               00001180
                   MSGSUCCES "PRIMITIVE DE DIFFERENCE"                         00001190
                   MSGECHEC "JE N'AI PAS SU INTEGRER LA DIFFERENCE"            00001200
                  )                                                            00001210
            POUR (CALCUL (PRIM (DIFFERENCE F G) X) R) ESSAYER                  00001220
              (SUC 5                                                           00001230
                (!%CREER (R1 R2))                                              00001240
                (CALCUL (PRIM F X) (!%VAL R1))                                 00001250
                (CALCUL (PRIM G X) (!%VAL R2))                                 00001260
                (!%ATTRIB R (DIFFERENCE (!%VVAL R1)(!%VVAL R2)))               00001270
                (!%LIBERER (R1 R2))                )) $                        00001280
%$                                                                             00001290
% INTEGRER UNE SOMME N-AIRE  $                                                 00001300
PLAN(22):='(INT22 (DESCRIPTEURS VARSUBST (F X R)                               00001310
```

Annexes D2 - D3 -

- filtres - plans
- Résultats connus -

- pour intégrer -
- plans ayant servis pour intégrer dans PB(150) en particulier -

```
                    OTOPEX (PLUS)                                          00001320
                    COUTMETHODE 5                                          00001330
                    MSGSUCCES "PRIMITIVE DE SOMME"                         00001340
                    MSGSUCCES "JE N'AI PAS SU INTEGRER LES TERMES          00001350
                               DE LA SOMME"                                00001360
                  )                                                        00001370
            POUR (CALCUL (PRIM F X) R) ESSAYER                             00001380
            (SI ((!%OP F)  EST PLUS)                                       00001390
               (SUC 3                                                      00001400
                  (!%CREER R1)(!%ATTRIB R 0)                               00001410
                  (POURTOUT I (!%INTERVAL 1 (!%ORDROP F))                  00001420
                     (SUC 3 (CALCUL (PRIM (!%ARGI I F) X)(!%VAL R1))       00001430
                            (!%ATTRIB R (PLUS (!%OBJ R)(!%VVAL R1)))       00001440
                            (!%MONTRER R1)) )                              00001450
                  (!%LIBERER R1) ) ) )$                                    00001460
%$                                                                         00001470
%* INTEGRER (A+B) ** CSTE    INT48 EST LA VERSION AVEC -  $                00001480
PLAN(23):='(INT23 (DESCRIPTEURS VARSUBST (F G X N R)                       00001490
                    COUTMETHODE 8                                          00001500
                    MSGSUCCES " DEVELOPPEMENT DU BINOME"                   00001510
                    OTOPEX (EXPT PLUS)                                     00001520
                  )                                                        00001530
            POUR (CALCUL (PRIM (EXPT (PLUS F G) N)X) R) ESSAYER            00001540
           (SI    (!%ENTIER N)                                             00001550
            (SUC 4 (!%CREER R1)                                            00001560
                   (!%ATTRIB (!%VAL R1)(!%DEVBINOM (EXPT (PLUS F G) N)))00001570
                   (!%MONTRER R1)                                          00001580
                   (CALCUL (PRIM (!%VVAL R1) X)R)                          00001590
                   (!%LIBERER R1)   )))$                                   00001600
%$                                                                         00001610
%* INTEGRER (A1+A2+...)/C $                                                00001620
PLAN(24):='(INT24 (DESCRIPTEURS VARSUBST (F G X R)                         00001630
                    OTOPEX (QUOTIENT PLUS)                                 00001640
                    COUTMETHODE 8                                          00001650
                    MSGSUCCES "(A1+A2+...+AN) / C "                        00001655
                  )                                                        00001660
            POUR (CALCUL (PRIM (QUOTIENT F G) X) R) ESSAYER                00001670
           (SI       ((!%OP F) EST PLUS)                                   00001680
              (SUC 4                                                       00001690
                   (!%CREER (R1 ))(!%ATTRIB R 0)                           00001700
                   (POURTOUT I (!%INTERVAL 1 (!%ORDROP F))                 00001710
                    (SUC 3                                                 00001720
                    (CALCUL(PRIM (!%SIMPLIF(QUOTIENT( !%ARGI I F)          00001730
                                 G)) X)( !%VAL R1))                        00001740
                   (!%MONTRER R1)                                          00001750
                   (!%ATTRIB R (PLUS (!%VVAL R1)(!%OBJ R)) )  ))          00001760
                   (!%LIBERER (R1 ))                                       00001770
           ) )  )$                                                         00001780
%$                                                                         00001790
%INTEGRER ((A+B)**N / C $                                                  00001800
PLAN(25):='(INT25 (DESCRIPTEURS VARSUBST (U N D X R)                       00001810
                    OTOPEX (QUOTIENT EXPT PLUS)                            00001820
                    COUTMETHODE 10                                         00001830
                    MSGSUCCES "(A+B)**N / C "                              00001835
                  )                                                        00001840
            POUR (CALCUL (PRIM (QUOTIENT (EXPT U N)D) X ) R) ESSAYER       00001850
            (SI (ET ((!%OP U) EST PLUS)                                    00001860
                    ((!%ORDROP U) EST 2)                                   00001870
                    (!%ENTIER N) )                                         00001880
                (SUC 5                                                     00001890
                  (!%CREER (R1 R2))                                        00001900
                  (!%ATTRIB (!%VAL R1)(!%DEVBINOM (EXPT U N)))             00001910
                  (!%ATTRIB (!%VAL R2) (QUOTIENT (!%VVAL R1)D))            00001920
                  (!%MONTRER R2)                                           00001930
                  (CALCUL (PRIM ( !%VVAL R2) X) R)                         00001940
                  (!%LIBERER (R1 R2))                                      00001950
            )))$                                                           00001960
%$                                                                         00001970
%INT(A*F(X) DX)$                                                           00001980
PLAN(26):='(INT26 (DESCRIPTEURS VARSUBST (F A X R)                         00001990
                    OTOPEX (TIMES)                                         00002000
                    COUTMETHODE 8                                          00002010
                    MSGSUCCES "PRODUIT D'UNE CSTE ET D'UNE FCT"            00002020
                  )                                                        00002030
            POUR (CALCUL (PRIM (TIMES A F) X) R) ESSAYER                   00002040
   (CHOIX 2 (SI (NONVUS X A)                                               00002050
                (SUC 4                                                     00002060
                  (!%CREER R2)                                             00002070
                  (CALCUL (PRIM F X)(!%VAL R2))                            00002080
                  (!%ATTRIB R (TIMES A (!%VVAL R2)))                       00002090
                  (!%LIBERER R2)                                           00002100
                ))                                                         00002102
            (SI (NONVUS X F)                                               00002104
               (SUC 4 (!%CREER R1)                                         00002106
                  (CALCUL (PRIM A X)(!%VAL R1))                            00002108
                  (!%ATTRIB R (TIMES F (!%VVAL R1)))                       00002110
                  (!%LIBERER R1) ))                                        00002112
    ))$                                                                    00002114
%$                                                                         00002120
%INTEGRER UNE CONSTANTE       $                                            00002130
PLAN(27):='(INT27 (DESCRIPTEURS VARSUBST (F X R)                           00002140
                    COUTMETHODE 3                                          00002150
                    MSGSUCCES "FONCTION CSTE PAR RAPPORT A X "             00002160
                  )                                                        00002170
            POUR (CALCUL (PRIM F X) R) ESSAYER                             00002180
            (SI (!%NONVUS X F)                                             00002190
                (!%ATTRIB R (!%SIMPLIF (TIMES F X)))                       00002200
            ))$                                                            00002210
%$                                                                         00002220
%INTEGRER  X**N     $                                                      00002230
PLAN(28):='(INT28 (DESCRIPTEURS VARSUBST (F X N R)                         00002240
                    OTOPEX (EXPT)                                          00002250
                    COUTMETHODE 4                                          00002260
                    MSGSUCCES "MONOME X**N INTEGRE / X"                    00002270
                  )                                                        00002280
          POUR (CALCUL (PRIM (EXPT X N) X) R) ESSAYER                      00002290
          (SI(ET (N NESTPAS -1) (NONVUS X N))                              00002300
            (!%ATTRIB R (!%SIMPLIF (QUOTIENT (EXPT X (PLUS N 1))         00002310
                                     (PLUS N  1))))))$                     00002320
%$                                                                         00002330
%INTEGRER UN MONOME $                                                      00002340
PLAN(29):='(INT29 (DESCRIPTEURS VARSUBST (F X R)                           00002350
                    COUTMETHODE 6                                          00002360
                    MSGSUCCES "INTEGRATION D'UN MONOME "                   00002370
                  )                                                        00002380
              POUR (CALCUL (PRIM F X) R) ESSAYER                           00002390
               (SUC 4                                                      00002400
                  (!%CREER R1)                                             00002410
                  (!%ATTRIB (!%VAL R1)(!%MONOMEP F (X)))                   00002420
                  (!%ATTRIB R (!%SIMPLIF (QUOTIENT                         00002430
                              (TIMES (!%TETE (!%VVAL R1))                  00002440
                                     (EXPT X                               00002450
                                           (PLUS (!%TQ (!%VVAL R1))        00002460
                                                 1)))                      00002470
                              (PLUS (!%ARG1 (!%VVAL R1)) 1) )) )           00002480
                  (!%LIBERER R1)                                           00002490
              ))$                                                          00002500
%$                                                                         00002510
%INTEGRATION D'UN PRODUIT N-AIRE                    $                      00002512
% R1<== CSTE , R2<==MONOMES , R3<== LE RESTE       $                      00002514
% R<== R1*PRIM(R2*R3    X)                         $                      00002516
PLAN(30):='(INT30 (DESCRIPTEURS VARSUBST (F X R)                           00002540
```

```
                    OTOPEX (TIMES)                                          00002550
                    COUTMETHODE 8                                           00002560
                    MSGSUCCES "INTEGRATION DE PRODUIT N-AIRE"               00002570
                  )                                                         00002580
          POUR (CALCUL (PRIM F X) R) ESSAYER                                00002590
         (SI (ET ((!%OP F) EST TIMES)(PGE (!%ORDROP F) 3))                  00002600
          (SUC 14 (!%CREER (R1 R2 R3 R4))                                   00002602
            (!%ATTRIB (!%VAL R1) 1)                                         00002604
            (!%ATTRIB (!%VAL R2) 1)                                         00002606
            (!%ATTRIB (!%VAL R3) 1)                                         00002608
            (POURTOUT I (!%INTERVAL 1 (!%ORDROP F))                         00002610
              (CHOIX 3                                                      00002612
                (SI (NONVUS X (!%ARGI I F))                                 00002614
                    (!%ATTRIB (!%VAL R1)(TIMES (!%VVAL R1)                  00002616
                                               (!%ARGI I F))))              00002618
                (SI (MONOMEP (!%ARGI I F) (X))                              00002620
                    (!%ATTRIB (!%VAL R2)(TIMES (!%VVAL R2)                  00002622
                                               (!%ARGI I F))))              00002624
                (!%ATTRIB (!%VAL R3)(TIMES (!%VVAL R3)                      00002626
                                           (!%ARGI I F)))  ))               00002628
            (!%ATTRIB (!%VAL R1)(!%SIMPLIF (!%VVAL R1)) )                   00002630
            (!%ATTRIB (!%VAL R2)(!%SIMPLIF (!%VVAL R2)) )                   00002632
            (!%ATTRIB (!%VAL R3)(!%SIMPLIF (!%VVAL R3)) )                   00002634
            (!%MONTRER R1)(!%MONTRER R2)(!%MONTRER R3)                      00002636
            (CALCUL (PRIM (TIMES (!%VVAL R2)(!%VVAL R3)) X)                 00002638
                    (!%VAL R4))                                             00002640
            (!%ATTRIB R (!%SIMPLIF (TIMES (!%VVAL R1)                       00002642
                                          (!%VVAL R4))))                    00002644
            (!%LIBERER (R1 R2 R3 R4))                                       00002646
          )))$                                                              00002648
                                                                            00002770
%$                                                                          00002780
%CHOIX DES TERMES A INTEGRER PAR PARTIES $                                  00002790
PLAN(31):='(INT31 (DESCRIPTEURS VARSUBST (U V X R)                          00002800
                    OTOPEX (TIMES)                                          00002810
                    COUTMETHODE 13                                          00002820
                    MSGSUCCES "PRODUIT INTEGRE PAR PARTIES"                 00002830
                  )                                                         00002840
          POUR (CALCUL (PRIM (TIMES U V) X) R) ESSAYER                      00002845
         (SI (ET (VUS X U)(VUS X V))                                        00002850
          (CHOIX 2                                                          00002860
            (CALCUL (INTEGRERPARPARTIE (TIMES U V) X)R)                     00002870
            (CALCUL (INTEGRERPARPARTIE (TIMES V U) X)R) )))$                00002880
%$                                                                          00002890
%INTEGRER PAR PARTIES LE LOG D'UN POLYNOME $                                00002900
PLAN(32):='(INT32 (DESCRIPTEURS VARSUBST (U X R)                            00002910
                    OTOPEX (LOG)                                            00002920
                    COUTMETHODE 3                                           00002930
                    MSGSUCCES "INTEGRATION DU LOG PAR PARTIES"              00002940
                  )                                                         00002950
          POUR (CALCUL (PRIM (LOG U) X) R) ESSAYER                          00002960
          (SI (POLYNOMEP U X)                                               00002970
            (CALCUL (INTEGRERPARPARTIE (TIMES (LOG U) 1) X) R)              00002980
          ))$                                                               00002990
%$                                                                          00003000
%INTEGRATION PAR PARTIE DU PRODUIT D'UN MONOME ET D'UNE FONCTION $           00003010
% LOG-EXP OU TRIGO                                                         $ 00003020
PLAN(33):='(INT33 (DESCRIPTEURS VARSUBST (U V X R)                          00003030
                    OTOPEX (TIMES)                                          00003040
                    O1OPEX (LOG EXPT SIN COS TG)                            00003050
                    COUTMETHODE 10                                          00003060
                    MSGSUCCES "INTEGRATION DU PRODUIT PAR PARTIE"           00003070
                    MSGECHEC "ECHEC SUR PRODUIT PAR PARTIES"                00003080
                  )                                                         00003090
          POUR (CALCUL (PRIM (TIMES U V) X )R) ESSAYER                      00003100
     (CHOIX 2                                                               00003110
      (SI (ET (MONOMEP U (X))                                               00003115
              (VUS X V))                                                    00003116
        (CHOIX 2                                                            00003120
        (SI (OU ((!%OP V) EST EXPT)(TRIGO V))                               00003130
         (CALCUL (INTEGRERPARPARTIE (TIMES U V) X) R))                      00003132
        (SI ((!%OP V) EST LOG)                                              00003133
         (CALCUL (INTEGRERPARPARTIE (TIMES V U) X) R))))                    00003140
      (SI (ET (MONOMEP V (X))                                               00003145
              (VUS X U))                                                    00003146
        (CHOIX 2                                                            00003150
        (SI (OU ((!%OP U) EST EXPT)(TRIGO U))                               00003152
         (CALCUL (INTEGRERPARPARTIE (TIMES U V) X) R))                      00003153
        (SI ((!%OP V) EST LOG)                                              00003160
         (CALCUL (INTEGRERPARPARTIE (TIMES V U) X) R)) ))))$                00003170
%$                                                                          00003180
%INTEGRATION D'UNE FRACTION RATIONELLE $                                    00003190
PLAN(34):='(PLAN34 (DESCRIPTEURS VARSUBST (U V X R)                         00003200
                    OTOPEX (QUOTIENT)                                       00003210
                    COUTMETHODE 15                                          00003220
                    MSGSUCCES "FRACTION RATIONNELLE PAR RISCH "             00003230
                    MSGECHEC "PAS SU INTEGRER LA FRACTION"                  00003240
                  )                                                         00003250
          POUR (CALCUL (PRIM (QUOTIENT U V) X)R) ESSAYER                    00003260
          (SI (ET (POLYNOMEP U X)(POLYNOMEP V X))                           00003270
            (SUC 2                                                          00003280
              (!%ATTRIB R (!%ALGEVAL (INT (QUOTIENT U V) X)))               00003290
              T ))                                                          00003300
          )$                                                                00003310
%$                                                                          00003320
%CHANGEMENTS DE VARIABLES DOMAINE TRIGO.EX:INT(SIN(X)*LOG(COS(X))$          00003330
PLAN(35):='(PLAN35 (DESCRIPTEURS VARSUBST (U V X R)                         00003340
                    OTOPEX (TIMES)                                          00003350
                    O1OPEX (SIN COS)                                        00003360
                    COUTMETHODE 8                                           00003370
                    MSGSUCCES "CHG-VAR-INT DANS U*LOG V"                    00003380
                  )                                                         00003390
          POUR (CALCUL (PRIM (TIMES U (LOG V)) X) R ) ESSAYER               00003400
          (SI (TOUSVRAI (!%SURTERME X V)(VUS (!%OP ?)(COS)))                00003410
            (SUC 6 (!%CREER (R1 R2 R3))                                     00003420
               (!%ATTRIB (!%VAL R1)(!%NOUVAR))                              00003430
               (!%ATTRIB (!%VAL R2)(!%REMPL (TIMES U (LOG V))               00003440
                                            (TG X)(QUOTIENT (SIN X)(COS X))))00003450
               (!%ATTRIB (!%VAL R3)(!%CHGVARINT (!%VVAL R2) X               00003460
                                                (!%VVAL R1)(COS X)))        00003470
               (!%MONTRER R3)                                               00003480
               (CALCUL (PRIM (!%VVAL R3)(!%VVAL R1) ) R)                    00003490
               (!%LIBERER (R1 R2 R3))                                       00003500
               )))$                                                         00003510
%$                                                                          00003520
% INT(U/V.X) EST INT(U*(1/V).X) $                                          00003530
PLAN(36):='(INT36 (DESCRIPTEURS VARSUBST (U V X R)                          00003540
                    OTOPEX (QUOTIENT)                                       00003550
                    COUTMETHODE 25                                          00003560
                    MSGSUCCES "QUOTIENT U/V RAMENE A U*(1/V)"               00003570
                  )                                                         00003580
          POUR (CALCUL (PRIM (QUOTIENT U V) X) R) ESSAYER                   00003585
         (SI (U_NESTPAS_1)                                                  00003590
          (CALCUL (PRIM (TIMES U (QUOTIENT 1 V)) X) R)                      00003600
          ))$                                                               00003610
%$                                                                          00003620
%INTEGRATION DE U**N * U' $                                                 00003630
PLAN(37):='(INT37 (DESCRIPTEURS VARSUBST (U N V X R)                        00003640
                    OTOPEX (EXPT TIMES)                                     00003650
                    COUTMETHODE 5                                           00003660
                    MSGSUCCES "INTEGRATION DE U**N * U'"                    00003670
                  )                                                         00003680
          POUR (CALCUL (PRIM (TIMES (EXPT U N) V) X ) R)ESSAYER             00003690
```

```
          (SI ( ET (ENTIER N)                                                  00003690
                   ((!%DERIV U X) EST V) )                                     00003700
            (!%ATTRIB R (QUOTIENT (EXPT U (PLUS N 1))                          00003710
                                  (PLUS N 1)))   ))$                           00003720
%$                                                                             00003730
%INTEGRATION DE U*U' $                                                         00003740
PLAN(38):='(INT38 (DESCRIPTEURS VARSUBST (U V X R)                             00003750
                  OTOPEX (TIMES)                                               00003760
                  COUTMETHODE 1                                                00003770
                  MSGSUCCES "INTEGRATION DE U*U'  "                            00003780
                 )                                                             00003790
          POUR (CALCUL (PRIM (TIMES U V) X) R) ESSAYER                         00003800
          (CHOIX 2                                                             00003810
             (SI ((!%DERIV U X) EST V)                                         00003820
                 (!%ATTRIB R (QUOTIENT (EXPT U 2) 2)))                         00003830
             (SI ((!%DERIV V X) EST U)                                         00003840
                 (!%ATTRIB R (QUOTIENT (EXPT V 2) 2))) ))$                     00003850
%                                                                              00003860
%CALCUL DE INT F/(A+B)**2  EN INTEGRANT PAR PARTIE A PARTIR DE    $            00003870
% 1/(A+B)                                                        $            00003880
%  R1(X)<==-F/[(A+B)**(N-2) * (A'+B')]                           $            00003890
%  R2(X)<== 1/(A+B)                                              $            00003900
%  R3(X)<== INT(R1(X)' / (A+B) , X)                              $            00003910
%  RR   <== R1*R2-R3                                             $            00003920
PLAN(39):='(INT39 (DESCRIPTEURS VARSUBST (F A B N X RR)                        00003930
                  OTOPEX (QUOTIENT EXPT PLUS)                                  00003940
                  COUTMETHODE 5                                                00003950
                  MSGSUCCES "INT PAR PARTIE A PARTIR DE 1/(A+B)"               00003960
                 )                                                             00003970
          POUR (CALCUL (PRIM (QUOTIENT F (EXPT (PLUS A B) N)) X)               00003980
                       RR) ESSAYER                                             00003990
         (SUC 8 (!%CREER (R1 R2 R3))                                           00004000
           (!%ATTRIB (!%VAL R1)(!%SIMPLIF (QUOTIENT (MINUS F)                  00004010
             (TIMES (EXPT (PLUS A B)(DIFFERENCE  N 2))                         00004020
                    (PLUS (!%DERIV A X)(!%DERIV B X)) ))))                     00004030
           (!%MONTRER R1)                                                      00004040
           (!%ATTRIB (!%VAL R2)(QUOTIENT 1 (PLUS A B)))                        00004050
           (SI (PPQ (!%VOL (!%VVAL R1))                                        00004060
                    (!%VOL (QUOTIENT F (EXPT (PLUS A B) N))))                  00004070
               (CALCUL (PRIM (!%SIMPLIF (QUOTIENT (!%DERIV (!%VVAL            00004080
                R1) X)(PLUS A B))) X)(!%VAL R3)))                              00004090
           (!%MONTRER R3)                                                      00004100
           (!%ATTRIB RR (!%SIMPLIF (DIFFERENCE (TIMES (!%VVAL R1)              00004110
                                (!%VVAL R2))(!%VVAL R3))))                     00004120
           (!%LIBERER (R1 R2 R3 ))                                             00004130
         ))$                                                                   00004140
%$                                                                             00004150
%CALCUL DE INT F/(A-B)**2  EN INTEGRANT PAR PARTIE A PARTIR DE    $            00004160
% 1/(A-B)                                                        $            00004170
%  R1(X)<==-F/[(A-B)**(N-2) * (A'-B')]                           $            00004180
%  R2(X)<== 1/(A-B)                                              $            00004190
%  R3(X)<== INT(R1(X)' / (A-B) , X)                              $            00004200
%  RR   <== R1*R2-R3                                             $            00004210
PLAN(40):='(INT40 (DESCRIPTEURS VARSUBST (F A B N X RR)                        00004220
                  OTOPEX (QUOTIENT EXPT DIFFERENCE)                            00004230
                  COUTMETHODE 5                                                00004240
                  MSGSUCCES "INT PAR PARTIE A PARTIR DE 1/(A-B)"               00004250
                 )                                                             00004260
          POUR (CALCUL (PRIM (QUOTIENT F (EXPT (DIFFERENCE A B) N)) X)         00004270
                       RR) ESSAYER                                             00004280
         (SUC 8 (!%CREER (R1 R2 R3))                                           00004290
           (!%ATTRIB (!%VAL R1)(!%SIMPLIF (QUOTIENT (MINUS F)                  00004300
             (TIMES (EXPT (DIFFERENCE A B)(DIFFERENCE  N 2))                   00004310
                    (DIFFERENCE (!%DERIV A X)(!%DERIV B X)) ))))               00004320
           (!%MONTRER R1)                                                      00004330
           (!%ATTRIB (!%VAL R2)(QUOTIENT 1 (DIFFERENCE A B)))                  00004340
           (SI (PPQ (!%VOL (!%VVAL R1))                                        00004350
                    (!%VOL (QUOTIENT F (EXPT (DIFFERENCE A B) N))))            00004360
               (CALCUL (PRIM (!%SIMPLIF (QUOTIENT (!%DERIV (!%VVAL            00004370
                R1) X)(DIFFERENCE A B))) X)(!%VAL R3)))                        00004380
           (!%MONTRER R3)                                                      00004390
           (!%ATTRIB RR (!%SIMPLIF (DIFFERENCE (TIMES (!%VVAL R1)              00004400
                                (!%VVAL R2))(!%VVAL R3))))                     00004410
           (!%LIBERER (R1 R2 R3 ))                                             00004420
         ))$                                                                   00004430
%$                                                                             00004440
%INTEGRATION AVEC DES FONCTIONS TRIGONOMETRIQUES-CHG VAR U=SIN(X)$             00004450
%  R6<= NOUVAR                                                  $             00004460
%  R1<== F                                                      $             00004470
%  R2<==SIMPLIF(REMPL (X, - X,R1))                              $             00004480
%CHOIX 3 SI R1 EST -R2                                          $             00004490
%         R3<== CHGVAR(F,X,R6,COS(X))                          $             00004500
%       SINON R2<==SIMPLIF (REMPL (X,PI-X,R1))                 $             00004510
%       SI R1 EST -R2                                          $             00004520
%          R3<== CHGVAR(F,X,R6,SIN(X))                         $             00004530
%       SINON R2<==SIMPLIF (REMPL (X,PI+X,R1))                 $             00004540
%       SI R1 EST R2                                           $             00004550
%          R3<== CHGVAR(F,X,R6,TG(X))                          $             00004560
%  MONTRER R3                                                  $             00004570
%  R4<== INT(R3,R6)                                            $             00004580
%MONTRER R4                                                    $             00004590
%R<== REMPL (R6,R5,R4)                                         $             00004600
PLAN(41):='(INT41 (DESCRIPTEURS VARSUBST (F X R)                               00004610
                  01OPEX (SIN COS TG)                                          00004620
                  COUTMETHODE 4                                                00004630
                  MSGSUCCES "CHG VARIABLE FONCTION TRIGONOMETRIQUE"            00004640
                 )                                                             00004650
          POUR (CALCUL (PRIM F X) R) ESSAYER                                   00004660
    (SUC 13 (!%CREER (R1 R2 R3 R4 R5 R6))                                      00004670
     (!%ATTRIB (!%VAL R6)(!%NOUVAR))                                           00004680
     (!%ATTRIB (!%VAL R1) F)                                                   00004690
     (CHOIX 3                                                                  00004700
       (SI (IMPAIRE F X)                                                       00004701
           (SUC 2                                                              00004702
             (!%ATTRIB (!%VAL R5)(COS X))                                      00004703
             (!%ATTRIB (!%VAL R3)                                              00004704
                       (!%CHGVARINT F X (!%VAL R6)(COS X))) ))                 00004705
       (SUC 3 (!%ATTRIB (!%VAL R2)(!%SIMPLIF (!%REMPL X                        00004710
                        (DIFFERENCE PI X)(!%VVAL R1))))                        00004720
              (!%MONTRER R2)                                                   00004730
              (SI ((!%SIMPLIF (PLUS (!%VVAL R1)(!%VVAL R2))) EST 0)            00004740
                  (SUC 2                                                       00004750
                     (!%ATTRIB (!%VAL R5)(SIN X))                              00004760
                     (!%ATTRIB (!%VAL R3)                                      00004770
                               (!%CHGVARINT F X (!%VAL R6)(SIN X))) ))         00004780
       (SUC 3 (!%ATTRIB (!%VAL R2)(!%SIMPLIF (!%REMPL X                        00004870
                        (PLUS PI X)(!%VVAL R1))))                              00004880
              (!%MONTRER R2)                                                   00004890
              (SI (( !%VVAL R2) EST  (!%VVAL R1))                              00004900
                  (SUC 2                                                       00004910
                     (!%ATTRIB (!%VAL R5)(TG X))                               00004920
                     (!%ATTRIB (!%VAL R3)                                      00004930
                               (!%CHGVARINT F X (!%VAL R6)(TG X))) ))          00004940
     )                                                                         00004950
     (!%ATTRIB (!%VAL R3) (!%SIMPLIF (!%VVAL R3)) )                            00004960
     (!%MONTRER R3)                                                            00004970
     (CALCUL (PRIM (!%VVAL R3)(!%VAL R6))(!%VAL R4))                           00004980
     (!%MONTRER R4)                                                            00004990
     (!%ATTRIB R (!%REMPL (!%VAL R6)(!%VVAL R5)                                00005000
                          (!%VVAL R4)))                                        00005010
     (!%LIBERER (R1 R2 R3 R4 R5 R6))                                           00005020
     ))$                                                                       00005030
```

```
%$                                                                      00005040
%* INT(COS(X)**N,X) $                                                   00005050
%* SI N EST IMPAIR R1(X)<==(1-SIN(X)**2)**(N-1)/2 * COS(X)         $    00005060
%* SI N EST   PAIR R1(X)<==(1+COS(2*X))/2  **N/2                   $    00005070
PLAN(42):='(INT42   (DESCRIPTEURS VARSUBST (U N X R)                    00005080
                    OTOPEX   (EXPT COS)                                 00005090
                    COUTMETHODE 5                                       00005100
                    MSGSUCCES "FORME COS(X) ** N "                      00005110
                   )                                                    00005120
           POUR (CALCUL (PRIM (EXPT (COS U) N) X) R) ESSAYER            00005130
           (CHOIX 2                                                     00005140
             (SI ((!%RESTEDIV N 2) EST 1)                               00005150
               (SUC 5 (!%CREER R1)                                      00005160
                  (!%ATTRIB (!%VAL R1)(TIMES (COS U)                    00005170
                     (!%SIMPLIF (EXPT (DIFFERENCE 1 (EXPT(SIN U)2))     00005180
                                      (QUOTIENT(DIFFERENCE N 1)2)) )))  00005190
                  (!%MONTRER R1)                                        00005200
                  (CALCUL (PRIM (!%VVAL R1) X) R)                       00005210
                  (!%LIBERER R1)                                        00005220
                 )                                                      00005230
               (SI ((!%RESTEDIV N 2) EST 0)                             00005240
                 (SUC 5 (!%CREER R1)                                    00005250
                   (!%ATTRIB (!%VAL R1)(!%SIMPLIF (!%REMPL              00005260
                        (EXPT (COS U) 2)                                00005270
                        (QUOTIENT (PLUS 1 (COS(TIMES 2 U)))2)           00005280
                        (EXPT(EXPT(COS U) 2)(QUOTIENT N 2)))))          00005290
                   (!%MONTRER R1)                                       00005300
                   (CALCUL (PRIM (!%VVAL R1) X) R)                      00005310
                   (!%LIBERER R1)                                       00005320
                 ))                                                     00005330
           ))$                                                          00005340
                                                                        00005350
%$                                                                      00005360
%* INT(SIN(X)**N,X) $                                                   00005370
%* SI N EST IMPAIR R1(X)<==(1-COS(X)**2)**(N-1)/2 * SIN(X)         $    00005380
%* SI N EST   PAIR R1(X)<==(1+SIN(2*X))/2  **N/2                   $    00005390
PLAN(43):='(INT43   (DESCRIPTEURS VARSUBST (U N X R)                    00005400
                    OTOPEX   (EXPT SIN)                                 00005410
                    COUTMETHODE 5                                       00005420
                    MSGSUCCES "FORME SIN(X) ** N "                      00005430
                   )                                                    00005440
           POUR (CALCUL (PRIM (EXPT (SIN U) N) X) R) ESSAYER            00005450
           (CHOIX 2                                                     00005460
             (SI ((!%RESTEDIV N 2) EST 1)                               00005470
               (SUC 5 (!%CREER R1)                                      00005480
                  (!%ATTRIB (!%VAL R1)(TIMES (SIN U)                    00005490
                     (!%SIMPLIF (EXPT (DIFFERENCE 1 (EXPT(COS U)2))     00005500
                                      (QUOTIENT(DIFFERENCE N 1)2)) )))  00005510
                  (!%MONTRER R1)                                        00005520
                  (CALCUL (PRIM (!%VVAL R1) X) R)                       00005530
                  (!%LIBERER R1)                                        00005540
                 )                                                      00005550
               (SI ((!%RESTEDIV N 2) EST 0)                             00005560
                 (SUC 5 (!%CREER R1)                                    00005570
                   (!%ATTRIB (!%VAL R1)(!%SIMPLIF (!%REMPL              00005580
                        (EXPT (SIN U) 2)                                00005590
                        (QUOTIENT (PLUS 1 (SIN(TIMES 2 X)))2)           00005600
                        (EXPT(EXPT(SIN U) 2)(QUOTIENT N 2)))))          00005610
                   (!%MONTRER R1)                                       00005620
                   (CALCUL (PRIM (!%VVAL R1) X) R)                      00005630
                   (!%LIBERER R1)                                       00005640
                 ))                                                     00005650
           ))$                                                          00005660
%$                                                                      00005670
%INT (A/B,X) AVEC B CSTE $                                              00005680
PLAN(44):='(INT44 (DESCRIPTEURS VARSUBST (A B X R)                      00005690
                    OTOPEX   (QUOTIENT)                                 00005700
                    COUTMETHODE 5                                       00005710
                    MSGSUCCES "FRACTION AVEC DENOMINATEUR CONSTANT"     00005720
                   )                                                    00005730
           POUR (CALCUL (PRIM (QUOTIENT A B) X) R) ESSAYER              00005740
           (SI (NONVUS X B)                                             00005750
             (SUC 4 (!%CREER R1)                                        00005760
               (CALCUL (PRIM A X)(!%VAL R1))                            00005770
               (!%ATTRIB R (QUOTIENT (!%VVAL R1) B))                    00005780
               (!%LIBERER R1)                                           00005790
           )))$                                                         00005800
%$                                                                      00005810
%INT(   (A1+A2+...+AN)*B , X) $                                         00005820
PLAN(45):='(INT45 (DESCRIPTEURS VARSUBST (A B X R)                      00005830
                    OTOPEX   (TIMES PLUS)                               00005840
                    COUTMETHODE 3                                       00005850
                    MSGSUCCES "PRODUIT DONT UN FACTEUR EST SOMME N-AIRE" 00005860
                   )                                                    00005870
           POUR (CALCUL (PRIM (TIMES A B) X) R) ESSAYER                 00005880
          (CHOIX 2                                                      00005890
           (SI ((!%OP A) EST PLUS)                                      00005900
             (SUC 4 (!%CREER R1)(!%ATTRIB R 0)                          00005910
               (POURTOUT I (!%INTERVAL 1 (!%ORDROP A))                  00005920
                   (SUC 3                                               00005930
                       (CALCUL (PRIM (TIMES (!%ARGI I A) B)X)(!%VAL R1))00005940
                       (!%MONTRER R1)                                   00005950
                       (!%ATTRIB R (PLUS (!%OBJ R)(!%VVAL R1))) ))      00005960
               (!%LIBERER R1)                                           00005970
             ))                                                         00005980
           (SI ((!%OP B) EST PLUS)                                      00005990
             (SUC 4 (!%CREER R1)(!%ATTRIB R 0)                          00006000
               (POURTOUT I (!%INTERVAL 1 (!%ORDROP B))                  00006010
                   (SUC 3                                               00006020
                       (CALCUL (PRIM (TIMES (!%ARGI I B) A)X)(!%VAL R1))00006030
                       (!%MONTRER R1)                                   00006040
                       (!%ATTRIB R (PLUS (!%OBJ R)(!%VVAL R1))) ))      00006050
               (!%LIBERER R1)                                           00006060
           )) ) ) $                                                     00006070
%$                                                                      00006080
%INTEGRATION AVEC DES FONCTIONS HYPERBOLIQUES- CHG VAR U=SH(X)       $  00006090
%* R6<= NOUVAR                                                       $  00006100
%* R1<==SUBST(TH(X),SH(X)/CH(X),CH(X),COS(X),SH(X),SIN(X) )          $  00006110
%* R2<==SIMPLIF(REMPL (X, - X,R1))                                   $  00006120
%CHOIX 3 SI R1 EST -R2                                               $  00006130
%*        R3<== CHGVAR(F,X,R6,CH(X))                                 $  00006140
%*      SINON R2<==SIMPLIF (REMPL (X,PI-X,R1))                       $  00006150
%*      SI R1 EST -R2                                                $  00006160
%*        R3<== CHGVAR(F,X,R6,SH(X))                                 $  00006170
%*      SINON R2<==SIMPLIF (REMPL (X,PI+X,R1))                       $  00006180
%*      SI R1 EST R2                                                 $  00006190
%*        R3<== CHGVAR(F,X,R6,TH(X))                                 $  00006200
%* MONTRER R3                                                        $  00006210
%* R4<== INT(R3,R6)                                                  $  00006220
%MONTRER R4                                                          $  00006230
%R<== REMPL (R6,R5,R4)                                               $  00006240
PLAN(46):='(INT46 (DESCRIPTEURS VARSUBST (F X R)                        00006250
                    O1OPEX (SH CH TH)                                   00006260
                    COUTMETHODE 6                                       00006270
                    MSGSUCCES "CHG VARIABLE FONCTION HYPERBOLIQUE"       00006280
                   )                                                    00006290
           POUR (CALCUL (PRIM F X) R) ESSAYER                           00006300
    (SUC 13 (!%CREER (R1 R2 R3 R4 R5 R6))                               00006310
     (!%ATTRIB (!%VAL R6)(!%NOUVAR))                                    00006320
     (!%ATTRIB (!%VAL R1) F)                                            00006330
     (!%ATTRIB (!%VAL R1)(!%REMPL (TH X)(QUOTIENT (SH X)                00006340
               (CH X))(!%VVAL R1)))                                     00006350
     (!%ATTRIB (!%VAL R1)(!%REMPL (CH X)(COS X)                         00006360
```

```
                                (!%VVAL R1)))                                   00006360
    (!%ATTRIB (!%VAL R1)(!%SIMPLIF(!%REMPL (SH X)(SIN X)                        00006370
                                   (!%VVAL R1))))                               00006380
    (!%MONTRER R1)                                                              00006390
    (CHOIX 3                                                                    00006400
      (SUC 3 (!%ATTRIB (!%VAL R2)(!%SIMPLIF (!%REMPL X                          00006410
                       (DIFFERENCE PI X)(!%VVAL R1))))                          00006420
             (!%MONTRER R2)                                                     00006430
             (SI ((!%SIMPLIF (PLUS (!%VVAL R1)(!%VVAL R2))) EST 0)              00006440
                 (SUC 2                                                         00006450
                    (!%ATTRIB (!%VAL R5)(SH X))                                 00006460
                    (!%ATTRIB (!%VAL R3)                                        00006470
                              (!%CHGVARINT F X (!%VAL R6)(SH X))) )))           00006480
      (SUC 3 (!%ATTRIB (!%VAL R2)(!%SIMPLIF (!%REMPL X (MINUS X)                00006490
                                   (!%VVAL R1))))                               00006500
             (!%MONTRER R2)                                                     00006510
             (SI ((!%SIMPLIF (PLUS (!%VVAL R1)(!%VVAL R2))) EST 0)              00006520
                 (SUC 2                                                         00006530
                    (!%ATTRIB (!%VAL R5)(CH X))                                 00006540
                    (!%ATTRIB (!%VAL R3)                                        00006550
                              (!%CHGVARINT F X (!%VAL R6)(CH X))) )))           00006560
      (SUC 3 (!%ATTRIB (!%VAL R2)(!%SIMPLIF (!%REMPL X                          00006570
                       (PLUS PI X)(!%VVAL R1))))                                00006580
             (!%MONTRER R2)                                                     00006590
             (SI (( !%VVAL R2) EST  (!%VVAL R1))                                00006600
                 (SUC 2                                                         00006610
                    (!%ATTRIB (!%VAL R5)(TH X))                                 00006620
                    (!%ATTRIB (!%VAL R3)                                        00006630
                              (!%CHGVARINT F X (!%VAL R6)(TH X))) )))           00006640
    )                                                                           00006650
    (!%ATTRIB (!%VAL R3) (!%SIMPLIF (!%VVAL R3)) )                              00006660
    (!%MONTRER R3)                                                              00006670
    (CALCUL (PRIM (!%VVAL R3)(!%VAL R6))(!%VAL R4))                             00006680
    (!%MONTRER R4)                                                              00006690
    (!%ATTRIB RR (!%REMPL (!%VAL R6)(!%VVAL R5)                                 00006700
                          (!%VVAL R4)))                                         00006710
    (!%LIBERER (R1 R2 R3 R4 R5 R6))                                             00006720
    ))$                                                                         00006730
                                                                                00006740
%$                                                                              00006750
% INTEGRER (A-B)/C $                                                           00006760
PLAN(47):='(INT47 (DESCRIPTEURS VARSUBST (F G D X R)                            00006770
                  OTOPEX (QUOTIENT DIFFERENCE)                                  00006780
                  MSGSUCCES " FORME  (A-B)/C "                                  00006790
                  COUTMETHODE 8                                                 00006800
                 )                                                              00006810
          POUR (CALCUL (PRIM (QUOTIENT (DIFFERENCE F G) D) X) R) ESSAYER        00006820
              (SUC 5                                                            00006830
                 (!%CREER (R1 R2))                                              00006840
                 (CALCUL (PRIM (!%SIMPLIF (QUOTIENT F                           00006850
                                  D)) X)( !%VAL R1))                            00006860
                 (CALCUL (PRIM (!%SIMPLIF (QUOTIENT G                           00006870
                                  D)) X)( !%VAL R2))                            00006880
                  (!%ATTRIB R (DIFFERENCE (!%VVAL R1)(!%VVAL R2)) )             00006890
                  (!%LIBERER (R1 R2))                                           00006900
           )  )$                                                                00006910
%$                                                                              00006920
%INTEGRER ((A-B)**N / C $                                                       00006930
PLAN(48):='(INT48 (DESCRIPTEURS VARSUBST (F G N C X R)                          00006940
                  OTOPEX (QUOTIENT EXPT DIFFERENCE)                             00006950
                  MSGSUCCES "FORME (A-B)**N / C"                                00006960
                  COUTMETHODE 10                                                00006970
                 )                                                              00006980
 POUR (CALCUL (PRIM (QUOTIENT (EXPT (DIFFERENCE F G) N) C)X ) R) ESSAYER        00006990
   (SI (!%ENTIER N)                                                             00007000
             (SUC 5                                                             00007010
                (!%CREER (R1 R2))                                               00007020
                (!%ATTRIB (!%VAL R1)(!%DEVBINOM                                 00007030
                          (EXPT (DIFFERENCE F G) N) ))                          00007040
                (!%ATTRIB (!%VAL R2) (QUOTIENT (!%VVAL R1) C ))                 00007050
                (CALCUL (PRIM (!%VVAL R2) X) R)                                 00007060
                (!%LIBERER (R1 R2))                                             00007070
           )))$                                                                 00007080
%$                                                                              00007090
% INTEGRER (A-B) ** CSTE    $                                                  00007100
PLAN(49):='(INT49 (DESCRIPTEURS VARSUBST (F G N X R)                            00007110
                  COUTMETHODE 8                                                 00007120
                  MSGSUCCES "INTEGRATION PAR DEVELOPPEMENT DU BINOME"           00007130
                  OTOPEX (EXPT DIFFERENCE)                                      00007140
                 )                                                              00007150
           POUR (CALCUL (PRIM (EXPT(DIFFERENCE F G)N) X) R) ESSAYER             00007160
           (SI (!%ENTIER N)                                                     00007170
           (SUC 4 (!%CREER R1)                                                  00007180
                  (!%ATTRIB (!%VAL R1)(!%DEVBINOM                               00007190
                            (EXPT (DIFFERENCE F G) N) ))                        00007200
                  (!%MONTRER R1)                                                00007210
                  (CALCUL (PRIM (!%VVAL R1) X)R)                                00007220
                  (!%LIBERER R1)   )))$                                         00007230
%$                                                                              00007240
%CALCUL INT(CSTE/F(X),X) EST CSTE*INT(1/F(X),X)$                                00007250
PLAN(50):='(INT50 (DESCRIPTEURS VARSUBST( A B X R)                              00007260
                  OTOPEX (QUOTIENT)                                             00007270
                  COUTMETHODE 5                                                 00007280
                  MSGSUCCES "INT(C/F(X),X) EST C*INT(1/F(X),X"                  00007290
                 )                                                              00007300
           POUR (CALCUL (PRIM (QUOTIENT A B) X) R) ESSAYER                      00007310
           (SI (ET (NONVUS X A)(A NESTPAS 1) )                                  00007320
               (SUC 4 (!%CREER R1)                                              00007330
                  (CALCUL (PRIM (QUOTIENT 1 B) X)(!%VAL R1))                    00007340
                  (!%ATTRIB R (!%SIMPLIF (TIMES A (!%VVAL R1))))                00007350
                  (!%LIBERER R1)                                                00007360
           )) )$                                                                00007370
%$                                                                              00007380
% CALCUL INT(COS(A*X+B) , X)                                          $         00007390
PLAN(51):='(INT51 (DESCRIPTEURS VARSUBST (U X R)                                00007400
                    OTOPEX (PLUS TIMES)                                         00007410
                    OTOPEX (COS       )                                         00007420
                    COUTMETHODE 5                                               00007430
                    MSGSUCCES "INT(COS(A*X+B).X)  "                             00007440
                  )                                                             00007450
           POUR (CALCUL (PRIM (COS U) X) R) ESSAYER                             00007460
            (SUC 4 (!%CREER R1)                                                 00007470
              (!%ATTRIB (!%VAL R1)(!%LINEAIRE U (X)) )                          00007480
              (!%ATTRIB R (TIMES (QUOTIENT 1 (!%TETE (!%VVAL R1)))              00007490
                                 (SIN U)))                                      00007500
              (!%LIBERER R1)                                                    00007510
            ) )$                                                                00007520
%$                                                                              00007530
% CALCUL INT(SIN(A*X+B) , X)                                          $         00007540
PLAN(52):='(INT52 (DESCRIPTEURS VARSUBST (U X R)                                00007550
                    OTOPEX (PLUS TIMES)                                         00007560
                    OTOPEX (SIN       )                                         00007570
                    COUTMETHODE 5                                               00007580
                    MSGSUCCES "INT(SIN(A*X+B).X)  "                             00007590
                  )                                                             00007600
           POUR (CALCUL (PRIM (SIN U) X) R) ESSAYER                             00007610
            (SUC 4 (!%CREER R1)                                                 00007620
              (!%ATTRIB (!%VAL R1)(!%LINEAIRE U (X)) )                          00007630
              (!%ATTRIB R (TIMES (QUOTIENT 1 (!%TETE (!%VVAL R1)))              00007640
                                 (MINUS (COS U))))                              00007650
              (!%LIBERER R1)                                                    00007660
            ) )$                                                                00007670
%$
```

```
%INT(-F,X) EST - INT(F,X)$                                                      00007680
PLAN(53):='(INT53 (DESCRIPTEURS VARSUBST (F X R)                                00007690
                  OTOPEX (MINUS)                                                00007700
                  COUTMETHODE 5                                                 00007710
                  MSGSUCCES "INT(-F,X) EST -INT(F,X) "                          00007720
                 )                                                              00007730
         POUR (CALCUL (PRIM (MINUS F) X) R) ESSAYER                             00007740
         (SUC 4 (!%CREER R1)                                                    00007750
              (CALCUL (PRIM F X)(!%VAL R1))                                     00007760
              (!%ATTRIB R (!%SIMPLIF (MINUS (!%VVAL R1))))                      00007770
              (!%LIBERER R1) ))$                                                00007771
%INT(-F*G,X) EST -INT(F*G,X) $                                                  00007780
PLAN(54):='(INT54 (DESCRIPTEURS VARSUBST (F G X R)                              00007790
                  OTOPEX (TIMES MINUS)                                          00007800
                  COUTMETHODE 5                                                 00007810
                  MSGSUCCES "INT(-F*G,X) EST -INT(F*G,X) "                      00007820
                 )                                                              00007830
         POUR (CALCUL (PRIM (TIMES (MINUS F) G) X) R) ESSAYER                   00007840
         (SUC 4 (!%CREER R1)                                                    00007850
              (CALCUL (PRIM (TIMES F G) X)(!%VAL R1))                           00007860
              (!%ATTRIB R (!%SIMPLIF (MINUS (!%VVAL R1))))                      00007870
              (!%LIBERER R1) ))$                                                00007875
%INT(F*-G,X) EST -INT(F*G,X) $                                                  00007880
PLAN(55):='(INT55 (DESCRIPTEURS VARSUBST (F G X R)                              00007890
                  OTOPEX (TIMES MINUS)                                          00007900
                  COUTMETHODE 5                                                 00007910
                  MSGSUCCES "INT(F*-G,X) EST -INT(F*G,X) "                      00007920
                 )                                                              00007930
         POUR (CALCUL (PRIM (TIMES F (MINUS G)) X) R) ESSAYER                   00007940
         (SUC 4 (!%CREER R1)                                                    00007950
              (CALCUL (PRIM (TIMES F G) X)(!%VAL R1))                           00007960
              (!%ATTRIB R (!%SIMPLIF (MINUS (!%VVAL R1))))                      00007970
              (!%LIBERER R1) ))$                                                00007975
%... INTEGRATION PAR PARTIES ... $                                              00007980
PLAN(60):='(INT60 (DESCRIPTEURS VARSUBST (U V X R)                              00007990
                   OTOPEX (TIMES)                                               00008000
                   COUTMETHODE 20                                               00008010
                   MSGSUCCES "INTEGRATION PAR PARTIES"                          00008020
                  )                                                             00008030
           POUR (CALCUL (INTEGRERPARPARTIE (TIMES U V) X) R) ESSAYER            00008040
           (SUC 10                                                              00008050
            (!%CREER (VIPP ))                                                   00008060
            (CALCUL (PRIM V X)(!%VAL VIPP))                                     00008070
            (!%MONTRER VIPP)                                                    00008080
            (!%CREER (DUIPP))                                                   00008090
            (!%ATTRIB (!%VAL DUIPP)(!%DERIV U X))                               00008100
            (!%CREER R1)                                                        00008110
            (CALCUL (PRIM (!%SIMPLIF(TIMES (!%VVAL VIPP)(!%VVAL DUIPP))        00008120
                          ) X) (!%VAL R1))                                      00008130
            (!%ATTRIB R (!%SIMPLIF (DIFFERENCE (TIMES U                         00008140
                                                (!%VVAL VIPP))( !%VVAL R1))))   00008150
            (!%LIBERER ( DUIPP VIPP ))                                          00008160
            (!%LIBERER R1)                                                      00008170
           ))$                                                                  00008180
%=============================================================$                00008190
%%------------------ CALCUL DE PRIMITIVES ---------------$                      00008200
PB(140):='(SUC 2 (CALCUL (PRIM (COS X) X) RR)(!%ECRIRE (!%ALGEB RR)) )$        00008210
PB(141):='(SUC 2 (CALCUL (PRIM (PLUS (SIN X)(COS X)) X) RR)                     00008220
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008230
PB(142):='(SUC 2 (CALCUL (PRIM (EXPT(PLUS(SIN X)(COS X))3) X) RR)               00008240
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008250
PB(143):='(SUC 2 (CALCUL (PRIM (QUOTIENT(PLUS 1 X)X) X) RR)                     00008260
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008270
PB(144):='(SUC 2 (CALCUL (PRIM (EXPT(PLUS X 1)2) X) RR)                         00008280
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008290
PB(145):='(SUC 2 (CALCUL (PRIM (TIMES(PLUS 1 A)(QUOTIENT 1 X)) X) RR)           00008300
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008310
PB(146):='(SUC 2 (CALCUL (PRIM (TIMES X (COS X)) X) RR)                         00008320
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008330
PB(147):='(SUC 2 (CALCUL (PRIM (TIMES (SIN PHI)(PLUS X 1)) X) RR)               00008340
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008350
PB(148):='(SUC 2 (CALCUL (PRIM (TIMES (EXPT X (PLUS 33 AA))                     00008360
                                (SIN BB) ) X) RR)                               00008370
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008380
PB(149):='(SUC 2 (CALCUL (PRIM (EXPT X (PLUS N 5)) X) RR)                       00008390
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008400
PB(150):='(SUC 2 (CALCUL (PRIM (EXPT (TG X) 2) X) RR)                           00008410
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008420
PB(151):='(SUC 2 (CALCUL (PRIM  X  X) RR)                                       00008430
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008440
PB(152):='(SUC 2 (CALCUL(PRIM(TIMES(TIMES 3(EXPT X 2))(LOG X))                  00008450
                           X)RR)                                                00008460
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008470
PB(152):='(SUC 2 (CALCUL(PRIM (TIMES 3(EXPT X 2)(LOG X))                        00008471
                           X)RR)                                                00008472
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008473
PB(153):='(SUC 2 (CALCUL(PRIM(TIMES (EXPT X 3)(SIN X))X)RR)                     00008480
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008490
PB(154):='(SUC 2 (CALCUL(PRIM(LOG(PLUS (EXPT X 2) 1))X)RR)                      00008500
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008510
PB(155):='(SUC 2 (CALCUL(PRIM(TIMES (SIN X)(LOG(COS X)))X)RR)                   00008520
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008530
PB(156):='(SUC 2 (CALCUL(PRIM(TIMES (TG X)(LOG(COS X)))X)RR)                    00008540
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008550
%INT(CH(X)*(1-(SH(X)**2)) / SH(X)*(1-(CH(X)**2)) ,X)$                           00008560
PB(157):='(SUC 2 (CALCUL (PRIM (QUOTIENT                                        00008570
              (TIMES (CH X)                                                     00008580
                     (DIFFERENCE 1 (EXPT (SH X) 2)))                            00008590
              (TIMES (SH X)                                                     00008600
                     (DIFFERENCE 1 (EXPT (CH X) 2))) ) X) RR)                   00008610
            (!%ECRIRE (!%ALGEB RR)) )$                                          00008620
%INT ( X**2 / (COS(X) + X*SIN(X))**2 , X )$                                     00008630
PB(158):='(SUC 2 (CALCUL (PRIM (QUOTIENT                                        00008640
              (EXPT X 2)                                                        00008650
              (EXPT (PLUS (COS X)(TIMES X (SIN X))) 2)) X) RR)                  00008660
            (!%ECRIRE (!%ALGEB RR)) )$                                          00008670
PB(159):='(SUC 2 (CALCUL (PRIM (QUOTIENT (EXPT (PLUS X 2) 3)X) X)RR)            00008680
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008690
PB(160):='(SUC 2 (CALCUL (PRIM (EXPT (COS X) 3) X)RR)                           00008700
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008710
PB(161):='(SUC 2 (CALCUL (PRIM (EXPT (COS X) 2) X)RR)                           00008720
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008730
PB(162):='(SUC 2 (CALCUL (PRIM (COS (TIMES 2 X)) X) RR)                         00008740
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008750
PB(163):='(SUC 2 (CALCUL (PRIM (SIN (PLUS (TIMES K X) H)) X)RR)                 00008760
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008770
PB(164):='(SUC 2 (CALCUL (PRIM (SIN (PLUS (TIMES (SIN K) X) (COS H))) X)        00008780
                          RR)                                                   00008790
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008800
PB(165):='(SUC 2(CALCUL (PRIM (TIMES (EXPT X 3) 5) X) RR)                       00008810
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008820
PB(166):='(SUC 2 (CALCUL (PRIM (QUOTIENT 1                                      00008830
                   (PLUS (EXPT (SIN U) 2)(EXPT X 2))) X )RR)                    00008840
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008850
PB(167):='(SUC 2 (CALCUL (PRIM (QUOTIENT 1                                      00008860
                   (RAC2 (PLUS (EXPT (COS FI) 2)(EXPT X 2))))X)RR)              00008870
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008880
PB(168):='(SUC 2 (CALCUL (PRIM (QUOTIENT X                                      00008890
                   (RAC2 (PLUS (EXPT (LOG F) 2)(EXPT X 2)))) X)RR)              00008900
                 (!%ECRIRE (!%ALGEB RR)) )$                                     00008910
PB(169):='(SUC 2 (CALCUL (PRIM (QUOTIENT 1                                      00008920
                   (RAC2 (PLUS 9 (EXPT X 2) ))) X) RR)                          00008930
```

```
                (!%ECRIRE (!%ALGEB RR)) )$                                    00008940
PB(170):='(SUC 2 (CALCUL (PRIM (TIMES (SH X)(CH X)) X) RR)                    00008950
                (!%ECRIRE (!%ALGEB RR)) )$                                    00008960
PB(171):='(SUC 2 (CALCUL (PRIM (QUOTIENT (EXPT (SH X) 3)                      00008970
                                          (EXPT (CH X) 2)) X ) RR)            00008980
                (!%ECRIRE (!%ALGEB RR)) )$                                    00008990
PB(172):='(SUC 2 (CALCUL (PRIM (QUOTIENT (CH X)                               00009000
                  (DIFFERENCE 2 (EXPT (CH X) 2))) X) RR)                      00009010
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009020
PB(173):='(SUC 2 (CALCUL (PRIM (EXPT (TG X) 3) X) RR)                         00009030
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009040
PB(174):='(SUC 2 (CALCUL (PRIM                                                00009050
                (TIMES (COS X)(EXPT (TG X) 2)) X) RR)                         00009060
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009070
PB(175):='(SUC 2 (CALCUL (PRIM                                                00009080
                (QUOTIENT (SIN X)(EXPT (COS X) 3)) X) RR)                     00009090
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009100
PB(176):='(SUC 2 (CALCUL (PRIM                                                00009101
                (TIMES 3 (EXPT (SIN X) 2)(COS X)) X) RR)                      00009102
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009103
PB(177):='(SUC 2 (CALCUL (PRIM                                                00009104
                (TIMES 3 (SIN X)(EXPT (COS X) 2)) X) RR)                      00009105
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009106
PB(178):='(SUC 2 (CALCUL (PRIM                                                00009107
                (EXPT (SIN X) 3) X) RR)                                       00009108
                (!%ECRIRE (!%ALGEB RR)) )$                                    00009117
%=================================================================$           00009118
OUT T $ WRITE "==>FIN DU CHAPITRE INTEGRATION " $ OUT REDUFICH   $            00009120
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&$           00009130
END $                                                                         00009140
```

liste des Méta-règles

Annexe D4 a

```
1 8620 9570
08620 %-------------------------------------------------------------$
08630 % METAREGLES :CE SONT DES PROBLEMES (SI (COND) ACT)          $
08640 %JE PEUX LES APPLIQUER PAR APPEL DE SOLUT                    $
08650 %$
08660 %META-REGLES DE TYPE 1 :NEWLPP=F(LPP,PB)$
08670 %SI LE PROBLEME EST TROP GROS,IL Y A PEU DE CHANCES QUE CE SOIT $
08680 % UN RESULTAT CONNU                                          $
08690 % CALCUL DEVLIM                                              $
08700   MREG(1):='(SI (GREATERP (!%VOL PB) 10)
08710                 (!%ATTRIB NEWLPP (!%ENLEVE LPP 71))      )$
08720 %SI LE PROBLEME EST TROP GROS,IL Y A PEU DE CHANCES QUE CE SOIT $
08730 % UN RESULTAT CONNU                                          $
08740 % CALCUL LIMITES                                             $
08750    MREG(2):='(SI (GREATERP (!%VOL PB) 10)
08760                  (!%ATTRIB NEWLPP (!%ENLEVELIST LPP (39 40)) )
08770                )$
08780 MREG(3):='(SI ((!%OBJECTIFPB PB) EST (CALCUL LIM))
08790               (CHOIX 3
08800                 (SI ((!%PTLIM PB) EST 0)
08810                      (!%ATTRIB NEWLPP (!%ENLEVELIST
08820                                        LPP (46 47 48))))
08830                 (SI ((!%PTLIM PB) EST PINF)
08840                      (!%ATTRIB NEWLPP (47)))
08850                 (SI ((!%PTLIM PB) EST MINF)
08860                      (!%ATTRIB NEWLPP (48)))
08870                 ))$
08880 % CALCUL DEVLIM                                              $
08890  MREG(4):='(SI ((!%PTDEVLIM PB) EST 0)
08900                  (!%ATTRIB NEWLPP (!%ENLEVELIST LPP (70))) )$
08910 %GESTION DE PROBLEME ABSURDE:NE LAISSE QUE LE PLAN NIL A TENTER$
08920 % PRV PAIRE                                                  $
08930  MREG(5):='(SI (!%RESULTATCONNU (!%REMPL PAIRE IMPAIRE
08940                                  (!%TQ PB)) IMPAIRE)
08950               (SUC 2 (!%ECRIRE "PB ABSURDE:F(X) IMPAIRE")
08960                      (!%ATTRIB NEWLPP (0)) ) )$
08970 % PRV IMPAIRE                                                $
08980  MREG(6):='(SI (!%RESULTATCONNU (!%REMPL IMPAIRE PAIRE
08990                                  (!%TQ PB)) PAIRE)
09000               (SUC 2 (  !%ECRIRE "PB ABSURDE:F(X) PAIRE")
09010                      (!%ATTRIB NEWLPP (0)) ) )$
09020 %VERIFICATION SEMANTIQUE POUR LA PARITE $
09030 % PRV PAIRE                                                  $
09040  MREG(7):='(SI (PGQ (!%SIMPLIF (ABS (VALNUM (DIFFERENCE
09050                  (!%REMPL (!%TQQTQ PB) 1 (!%TQTQ PB))
09060                  (!%REMPL ( !%TQQTQ PB) -1 (!%TQTQ PB)))))) 0.00000
01)
09070               (SUC 2 (!%ECRIRE "F(X) NON PAIRE:F(1)#F(-1)")
09080                      (!%ATTRIB NEWLPP (0) ). ))$
09090 %  PRV IMPAIRE                                                $
09100  MREG(8):='(SI (PGQ (!%SIMPLIF (ABS (VALNUM (PLUS
09110                  (!%REMPL (!%TQQTQ PB) 1 (!%TQTQ PB))
09120                  (!%REMPL (!%TQQTQ PB) -1 (!%TQTQ PB)))))) 0.000000
1)
09130               (SUC 2 (!%ECRIRE "F(X) NON IMPAIRE:F(1)#-F(-1)")
09140                      (!%ATTRIB NEWLPP (0) ) ))$
09150 %$
09160 %SI LE PROBLEME EST TROP GROS,IL Y A PEU DE CHANCES QUE CE SOIT $
09170 % UN RESULTAT CONNU OU UNE CONSTANTE                          $
09180 % CALCUL INTEGRALES                                          $
09190 MREG(9):='(SI (GREATERP (!%VOL PB) 16)
09200                  (!%ATTRIB NEWLPP (!%ENLEVELIST LPP (20 27 28)) )
09210                )$
```

```
09240 % CALCUL DE LIMITES $
09250 MREG(10):='(SI (TRIGO PB)
09260                (!%ATTRIB NEWLPP (!%ENLEVELIST LPP (39 64))))$
09270 % CALCUL DE DEVLIM $
09280 MREG(11):='(SI (TRIGO PB)
09290                (!%ATTRIB NEWLPP (!%ENLEVELIST LPP (72 73))))$
09300 % META-REGLES DE TYPE 2 :(COUT,ESPOIR)=F(PL)                    $
09310 % PENALITES POUR VOLUME DE GROS PLANS                           $
09320 MREG(20):='(SI (GREATERP (!%!%VOLPLANTYPE PL) 40)
09330               (!%ATTRIB COUT (PLUS (!%OBJ COUT) 4)) )$
09340 %COUT USUEL DE LA METHODE      ET POIDS DES ACTIONS IMMEDIATES $
09350 MREG(21):='(SI T
09360               (!%ATTRIB COUT (PLUS (!%OBJ COUT) (!%!%COUTMETHODE P
L))))$
09370 %ESPOIR LIE A UN PLAN FORTEMENT CONTRAINT                       $
09380 MREG(22):='(SI (GREATERP (!%!%VOLCONDACT1 PL) 5)
09390                (!%ATTRIB ESPOIR (PLUS (!%OBJ ESPOIR) 5)) )$
09400 %ESPOIR LIE A UN PLAN DONT LE PBTYPE EST GROS:C'EST LA MEME IDEE $
09410 % QUE POUR LES GROSSES CONDITIONS                               $
09420 MREG(23):='(SI T (!%ATTRIB ESPOIR
09430                     (PLUS (!%OBJ ESPOIR) (!%!%VOLPBTYPE PL))))$
09440 %$
09450  %META-REGLES DE TYPE 3 :(COUT,ESPOIR)=F(PB,PL)$
09460 %ESPOIR LIE A LA PRESENCE DE L'OPERATEUR PRINCIPAL DANS OTOPEX$
09470 MREG(30):='(SI (VUS (!%OPRINCEX PB)(!%!%OTOPEX PL))
09480               (!%ATTRIB ESPOIR (PLUS (!%OBJ ESPOIR) 50))  )$
09490 %ESPOIR LIE A LA PRESENCE DE L'OPERATEUR PRINCIPAL DANS O1OPEX$
09500 MREG(31):='(SI (VUS (!%OPRINCEX PB)(!%!%O1OPEX PL))
09510               (!%ATTRIB ESPOIR (PLUS (!%OBJ ESPOIR) 20))  )$
09520 %AUGMENTER L'ESPOIR D'UN PLAN IMPOSANT DES OPERATEURS RARES SI LE$
09530 % PROBLEME COMPORTE LUI MEME CES OPERATEURS                     $
09540 MREG(32):='(SI T (!%ATTRIB ESPOIR
09550                    (PLUS (!%OBJ ESPOIR)(!%!%RARETECOMMUNE PL PB))
09560                   ))$
09570 %$
```

Annexe D4, b Gestion des méta-règles. Aspects méta-méta

```
05560 %=============================================================$
05570 %          GESTION DES META-REGLES                            $
05580 %=============================================================$
05590 LISP PROCEDURE REARRANGE (LPP,PBPB)$
05600 %REARRANGE LPP PAR ORDRE DECROISSANT DE NOTES CALCULEES A PARTIR $
05610 % DE COUT ET ESPOIR FIXES PAR JUGEMENT.JUGEMENT PEUT MODIFIER LPP$
05620 % NEWLPP:VARIABLE QUI RECUPERE CE QUE JUGEMENT A FAIT DE LPP     $
05630 % MREGAP:MREG UTILES DANS UNE SITUATION DONNEE                   $
05640 % OBJPB :OBJECTIF DU PROBLEME PBPB                               $
05650 BEGIN SCALAR LONG,NUMPLAN,MAXI,RMAXI,OX,NOTE$
05660       SCALAR NEWLPP,COUT,ESPOIR,MREGAP,OBJPB$
05670       TABTRI:=MKVECT(20)$
05680     COUT:=99 $ ESPOIR:=99$
05690 %APPLIQUER LES M-REGLES DE TYPE (1) NEWLPP=F(PB.LPP)              $
05700 OBJPB:=OBJECTIFPB PBPB$
05710 MREGAP:=COND(((OBJPB='(CALCUL DEVLIM)) '(1 1 4)    ),
05720              ((OBJPB='(CALCUL LIM)    ) '(1 2 3)    ),
05730              ((OBJPB='(PRV PAIRE )    ) '(1 5 7)    ),
05740              ((OBJPB='(PRV IMPAIRE)   ) '(1 6 8)    ),
05750              (      T                   '(1    )    ))$
05760 NEWLPP:=JUGEMENT(NIL,PBPB,LPP, MREGAP)$
05770 LONG:=LENGTH NEWLPP$
05780 IF GREATERP (LONG,20) THEN <<WRITE "DEBORDEMENT TABTRI "$
05790                            LONG:=20$  >>$
05800 FOR I:=1:LONG DO
05810   <<NUMPLAN:=CAR NEWLPP$
05820     COUT:=1 $ ESPOIR:=50 $
05830     %APPLIQUER LES M-REGLES TYPE (2) (COUT,ESPOIR)=F(PL)$
05840     NEWLPP:=JUGEMENT(PLAN(NUMPLAN),NIL,NEWLPP,
05850                      '(2  10 11 12 ) )$
05860     %APPLIQUER LES M-REGLES TYPE (3) (COUT,ESPOIR)=F(PB,PL)$
05870     NEWLPP:=JUGEMENT(PLAN(NUMPLAN),PBPB,NEWLPP,
05880                      '(3  20 21)  )$
05890     %CALCUL DE LA NOTE FINALE$
05900     NOTE:=NOTEFINALE(SIMPLIF ESPOIR,SIMPLIF COUT)$
05910     PUTV(TABTRI,I,LIST(NOTE,NUMPLAN))$
05920     NEWLPP:=CDR NEWLPP$
05930   >>$
05940 %TRIONS.....$
05950 FOR I:=1:LONG-1 DO
05960   <<MAXI:=CAR GETV(TABTRI,I)$ RMAXI:=I$
05970     FOR J:=I+1 : LONG DO
05980        <<IF GREATERP (CAR GETV(TABTRI,J) , MAXI)
05990              THEN <<MAXI:=CAR GETV(TABTRI,J) $ RMAXI:=J$ >>$
06000        >>$
06010     OX:=GETV(TABTRI,I)$
06020     PUTV(TABTRI,I,GETV(TABTRI,RMAXI) )$
06030     PUTV(TABTRI,RMAXI,OX)$
06040   >>$
06050 FOR I:=1:LONG DO
06060   <<OX:=GETV(TABTRI,I)$
06070     AFFICHE(5,'("TABTRI(" I ") : " OX) )$
06080   >>$
06090 %RECONSTRUISONS LPP $
06100 NEWLPP:=CDR GETV(TABTRI,1)$
06110 FOR I:=2:MIN2(LONG,99) DO
06120    <<NEWLPP:=APPEND(NEWLPP,CDR GETV(TABTRI,I))$
06130    >>$
06140 RETURN NEWLPP$
06150 END$
06160 %$
```

```
%$
OUT REDUFICH$ OFF ECHO$ LISP$
%=================================================================$
OUT T$ WRITE "ENTREE DU CHAPITRE SOMMES DE COMBINES "$OUT REDUFICH$
%=================================================================$
CONNEC1 '(APFORBIN APFORCOL)$
CONNEC2 '(EVALUER RAMENER EFFECTUER CNP VCNP)$
CONNEC3 '(CHGVARLIE)$
CONNEC4 '(SIG)$
ALGEBRAIC OPERATOR VCNP.SIG$
ALGEBRAIC OPERATOR CNP$
PLAN(101):='(SIG101 (DESCRIPTEURS VARSUBST (F K K1 K2 R)
                                  COUTMETHODE 5
                                  MSGSUCCES "SIG F  K A A"
                    )
            POUR (EVALUER (SIG F K K1 K2) R) ESSAYER
            (SI (K1 EST K2)
                (!%ATTRIB R (!%SIMPLIF (!%REMPL K K1 F)))) )$
%$
%EVALUER SIG A*F K K1 K2 EST A*SIG F K K1 K2 $
PLAN(102):='(SIG102 (DESCRIPTEURS VARSUBST (F G K K1 K2 R)
                                  OTOPEX (TIMES)
                                  COUTMETHODE 5
                                  MSGSUCCES "SIG A*F = A*SIG F"
                    )
            POUR (EVALUER (SIG (TIMES F G) K K1 K2) R) ESSAYER
           (CHOIX 2
             (SI (NONVUS K F)
                 (SUC 2 (EVALUER (SIG G K K1 K2) R)
                        (!%ATTRIB R (!%SIMPLIF (TIMES (!%ALGEB R)
                                                      F)))) )
              (SI (NONVUS K G)
                  (SUC 2 (EVALUER (SIG F K K1 K2) R)
                         (!%ATTRIB R (!%SIMPLIF (TIMES (!%ALGEB R)
                                                       F)))) ) ))$
%$
PLAN(103):='(SIG103 (DESCRIPTEURS VARSUBST (F K K1 K2 R)
                                  OTOPEX(VCNP)
                                  COUTMETHODE 9
                                  MSGSUCCES "METHODE 3 "
                    )
          POUR (EVALUER (SIG F K K1 K2) R) ESSAYER
         (SI ((!%NBOC VCNP F) EST 1)
             (SUC 5 (!%CREER (R1))
                (!%ATTRIB (!%VAL R1)(!%TETE (!%TERMES VCNP F)))
                (CHOIX 2
                   (SI (ET (NONVUS K (!%ARG1 (!%VVAL R1)))
                           (   VUS K (!%ARG2 (!%VVAL R1))) )
                       (RAMENER (APFORBIN (SIG F K K1 K2)) R) )
                     (SI (ET (   VUS K (!%ARG1 (!%VVAL R1)))
                             (NONVUS K (!%ARG2 (!%VVAL R1))) )
                         (RAMENER (APFORCOL (SIG F K K1 K2)) R) )
                )
                (!%LIBERER R1)
             )))$
%$
%EVALUATION D'UNE SOMME SIG A+B $
PLAN(104):='(SIG104 (DESCRIPTEURS VARSUBST (F K K1 K2 R)
                                  OTOPEX (PLUS)
                                  COUTMETHODE 5
                                  MSGSUCCES "METHODE 4"
                    )
          POUR (EVALUER (SIG F K K1 K2 ) R) ESSAYER
          (SI ((!%OP F) EST PLUS)
             (SUC 4 (!%CREER (R1 R2))
                (!%ATTRIB R O)
                (POURTOUT I (!%INTERVAL 1 (!%ORDROP F))
                  (SUC 3 (EVALUER (SIG (!%ARGI I F) K K1 K2)
                                  (!%VAL R2))
                         (!%MONTRER R2)
                         (!%ATTRIB R (PLUS (!%OBJ R)(!%VVAL R2)))))
                (!%LIBERER (R1 R2))  )) )$
%$
PLAN(105):='(SIG105 (DESCRIPTEURS VARSUBST (F G K K1 K2 R)
                                  OTOPEX (DIFFERENCE)
                                  COUTMETHODE 5
                                  MSGSUCCES "METHODE 4"
                    )
        POUR (EVALUER (SIG (DIFFERENCE F G) K K1 K2) R) ESSAYER
     (CHOIX 2
        (SI (F EST G)
            (!%ATTRIB R O))
        (SUC 5 (!%CREER (R1 R2))
           (EVALUER (SIG F K K1 K2)(!%VAL R1))
           (EVALUER (SIG G K K1 K2)(!%VAL R2))
           (!%ATTRIB R (!%SIMPLIF (DIFFERENCE (!%VVAL R1)
                                              (!%VVAL R2))))
           (!%LIBERER (R1 R2)) ))$
PLAN(106):='(SIG106 (DESCRIPTEURS VARSUBST (F K K1 K2 R)
                                  OTOPEX (MINUS)
                                  COUTMETHODE 5
                                  MSGSUCCES "SIG -F K K1 K2 ==>-SIG..."
                    )
        POUR (EVALUER (SIG (MINUS F) K K1 K2) R) ESSAYER
        (SUC 4 (!%CREER (R1 ))
           (EVALUER (SIG F K K1 K2)(!%VAL R1))
           (!%ATTRIB R (!%SIMPLIF (MINUS (!%VVAL R1))))
           (!%LIBERER (R1 )) ))$
%$
PLAN(107):='(SIG107 (DESCRIPTEURS VARSUBST (F K K2 R)
                     COUTMETHODE 10
                     MSGSUCCES "SIG F K 1 K2+1==>SIG F K O K2"
                    )
        POUR (EVALUER (SIG F K 1 (PLUS K2 1)) R) ESSAYER
        (SUC 4 (!%CREER R1)
           (EVALUER (SIG F K O K2)(!%VAL R1))
           (!%ATTRIB R (!%SIMPLIF (DIFFERENCE (PLUS (!%VVAL R1)
                                  (!%REMPL K (PLUS K2 1) F) )
                                  (!%REMPL K O           F) ) ))
           (!%LIBERER R1)
        ))$
PLAN(108):='(SIG108 (DESCRIPTEURS VARSUBST (F K K2 R)
                                  OTOPEX (PLUS VCNP)
                                  COUTMETHODE 8
                     MSGSUCCES "SIG 1 N+1==> SIG O N"
                    )
          POUR (EVALUER (SIG F K 1 (PLUS K2 1)) R) ESSAYER
          (SI (VUFORMTERM ((VCNP (PLUS K2 1) K)
                           (DESCRIPTEURS    ))
                          NIL F)
              (SUC 6 (!%CREER (R1 R2))
                 (!%ATTRIB (!%VAL R1)(!%NOUVAR))
                 (EFFECTUER (CHGVARLIE (SIG F K 1 (PLUS K2 1))
                    (!%VVAL R1)(DIFFERENCE (PLUS K2 1)K) )
                    (!%VAL R2))
                 (!%MONTRER R2)
                 (RAMENER (APFORBIN (!%VVAL R2)) R)
                 (!%LIBERER (R1 R2))
              ) ) )$
%$
PLAN(109):='(SIG109 (DESCRIPTEURS VARSUBST (F K K2 R)
```

Annexe D5 à

Le chapitre de connaissances consacré à l'évaluation de sommations de formules combinatoires.

- Connaissances nécessaires en particulier pour résoudre Pb (200).

```
                                OTOPEX (PLUS VCNP)                      00001101
                                COUTMETHODE 8                           00001102
                        MSGSUCCES "SIG 1 N+1==> SIG O N"                00001103
                       )                                                00001104
              POUR (EVALUER (SIG F K 1 (PLUS K2 1)) R) ESSAYER          00001105
              (SUC 4 (!%CREER R1)                                       00001106
                 (RAMENER (APFORBIN (SIG F K O (PLUS K2 1)))(!%VAL R1))  00001107
                 (!%ATTRIB R (!%SIMPLIF (DIFFERENCE (!%VVAL R1)         00001108
                                        (!%REMPL K O F))))              00001109
                 (!%LIBERER R1)                                         00001110
               ))$                                                      00001111
%$                                                                      00001112
PLAN(110):='(SIG110 (DESCRIPTEURS VARSUBST(F G K K2 R)                  00001113
                    OTOPEX (TIMES)                                      00001114
                    COUTMETHODE 8                                       00001115
                    MSGSUCCES "SIG K*F,O,N = SIG K*F,1,N"               00001116
                   )                                                    00001117
              POUR (EVALUER (SIG (TIMES F G) K O K2) R) ESSAYER         00001118
             (SI (OU (F EST K)(G EST K))                                00001119
                 (EVALUER (SIG (TIMES F G) K 1 K2) R)                   00001120
             ))$                                                        00001121
%LE PLAN(111) AMENE UN BOUCLAGE TEL QU¢IL EST ECRIT$                    00001122
%PLAN(111):='(SIG111 (DESCRIPTEURS VARSUBST (F K K1 K2 R)               00001123
                    COUTMETHODE 20                                      00001124
                    MSGSUCCES "SIMPLIFICATION POSSIBLE "                00001125
                   )                                                    00001126
           POUR (EVALUER (SIG F K K1 K2) R) ESSAYER                     00001127
        (EVALUER (SIG (!%SIMPLIF F) K (!%SIMPLIF K1)(!%SIMPLIF K2))    00001128
                 R)   )$                                                00001129
%$                                                                      00001130
%.....................$                                                00001221
PLAN(115):='(SIG115 (DESCRIPTEURS VARSUBST (F K K2 R)                   00001222
                                OTOPEX(VCNP)                            00001223
                                COUTMETHODE 5                           00001224
                                MSGSUCCES "METHODE 5"                   00001225
                               )                                        00001226
           POUR (RAMENER (APFORBIN (SIG F K O K2)) R) ESSAYER           00001227
           (SI (SOUSTERME (VCNP K2 K) F)                                00001228
              (!%ATTRIB R (!%RESULTATCONNU (SIG (!%SIMPLIF F) K O K2)   00001229
                                           BINEWTON) )                  00001230
           ))$                                                          00001231
%$                                                                      00001232
PLAN(116):='(SIG116 (DESCRIPTEURS VARSUBST (F K K1 K2 R)                00001233
                                OTOPEX (QUOTIENT VCNP PLUS)             00001234
                                COUTMETHODE 5                           00001240
                                MSGSUCCES "METHODE 6"                   00001250
                               )                                        00001260
              POUR (RAMENER (APFORBIN (SIG F K K1 K2)) R) ESSAYER       00001270
              (SUC 10 (!%CREER (R1 R2 R3 R4))                           00001280
                 (!%ATTRIB (!%VAL R1)(!%VUFORMTERM (!%ALGEB FCNP6)      00001290
                                                   (K2) F))             00001300
                 (!%ATTRIB (!%VAL R2)F)                                 00001310
                 (!%ATTRIB (!%VAL R4)(!%APPLISUBST                      00001320
                    (!%TOT (!%VVAL R1)) (PLUS N 1)) )                   00001330
                 (!%MONTRER R4)                                         00001340
                 (POURTOUT OC (!%VVAL R1)                               00001350
                    (!%ATTRIB (!%VAL R2) (!%REMPL (!%TETE OC)           00001360
                                (!%APPLISUBST (!%TO OC)                 00001370
                                  (QUOTIENT (VCNP(PLUS N 1)(PLUS M 1))  00001380
                                            (PLUS N 1))) (!%VVAL R2))) )00001390
                 (RAMENER (APFORBIN (SIG (!%SIMPLIF                     00001400
                                (TIMES (!%VVAL R2)(!%VVAL R4)))         00001410
                               K K1 K2)) (!%VAL R3))                    00001420
                 (!%ATTRIB R (TIMES (QUOTIENT 1 (!%VVAL R4))            00001430
                                    (!%VVAL R3)) )                      00001440
                 (!%LIBERER (R1 R2 R3 R4))                              00001450
               ))$                                                      00001460
PLAN(117):='(SIG117 (DESCRIPTEURS VARSUBST (F K K1 K2 R)                00001470
                                OTOPEX(VCNP DIFFERENCE)                 00001480
                                COUTMETHODE 5                           00001490
                                MSGSUCCES "METHODE 7"                   00001500
                               )                                        00001510
           POUR (RAMENER (APFORBIN (SIG F K K1 K2)) R) ESSAYER          00001520
           (SI (VUFORMTERM (!%ALGEB FCNP7) (K2) F)                      00001530
               (!%ATTRIB R (!%REMPL (VCNP K2 (DIFFERENCE K2 K))         00001540
                                    (VCNP K2 K)   F) )                  00001550
           ))$                                                          00001560
%$                                                                      00001570
PLAN(118):='(SIG118 (DESCRIPTEURS VARSUBST (F K K1 K2 R)                00001571
                                OTOPEX (VCNP )                          00001572
                                COUTMETHODE 5                           00001573
                                MSGSUCCES "METHODE 8"                   00001574
                               )                                        00001575
              POUR (RAMENER (APFORBIN (SIG F K K1 K2)) R) ESSAYER       00001576
              (SUC 10 (!%CREER (R1 R2 R3 R4))                           00001577
                 (!%ATTRIB (!%VAL R1)(!%VUFORMTERM                      00001578
                             ((TIMES M (VCNP KK2 M)   )                 00001579
                               (DESCRIPTEURS VARSUBST (KK2 M)           00001580
                                             PARAM (KK2) ))             00001581
                            (K2) F))                                    00001582
                 (!%ATTRIB (!%VAL R2)F)                                 00001583
                 (!%ATTRIB (!%VAL R4)(!%APPLISUBST                      00001584
                    (!%TOT (!%VVAL R1)) (DIFFERENCE M 1)) )             00001585
                 (!%MONTRER R4)                                         00001586
                 (POURTOUT OC (!%VVAL R1)                               00001587
                    (!%ATTRIB (!%VAL R2) (!%REMPL (!%TETE OC)           00001588
                                (!%APPLISUBST (!%TO OC)                 00001589
                                  (VCNP (DIFFERENCE K2 1)               00001590
                                        (DIFFERENCE M 1)))(!%VVAL R2))))00001591
                 (RAMENER (APFORBIN (SIG (!%SIMPLIF (!%VVAL R2))        00001592
                               K K1 K2)) (!%VAL R3))                    00001593
                 (!%ATTRIB R (TIMES K2                                  00001594
                                    (!%VVAL R3)) )                      00001595
                 (!%LIBERER (R1 R2 R3 R4))                              00001596
               ))$                                                      00001597
PLAN(119):='(SIG119 (DESCRIPTEURS VARSUBST (F K K1 K2 R)                00001598
                                OTOPEX(VCNP)                            00001599
                                COUTMETHODE 5                           00001600
                                MSGSUCCES "METHODE 9"                   00001610
                               )                                        00001620
          POUR (RAMENER (APFORBIN (SIG F K K1 K2)) R) ESSAYER           00001630
         (SI (ET (POSITIF K1)                                           00001640
                 (SOUSTERME (VCNP K2 K) F)                              00001650
                 (NON (!%VUFORMTERM (!%ALGEB FCNP9) (K) F))             00001660
                 (PGQ (!%VOL (!%SIMPLIF (QUOTIENT F K)))(!%VOL F))      00001670
                )                                                       00001680
             (SUC 7 (!%CREER (R1 R2))                                   00001690
                 (EVALUER (SIG F K O K2)(!%VAL R1))                     00001700
                 (!%MONTRER R1)                                         00001710
                 (EVALUER (SIG F K O (!%SIMPLIF (DIFFERENCE K1 1)))     00001720
                          (!%VAL R2))                                   00001730
                 (!%MONTRER R2)                                         00001740
                 (!%ATTRIB R (!%SIMPLIF (DIFFERENCE (!%VVAL R1)         00001750
                                                    (!%VVAL R2))))      00001760
                 (!%LIBERER (R1 R2))                                    00001770
            )) )$                                                       00001780
%$                                                                      00001790
PLAN(120):='(SIG120 (DESCRIPTEURS VARSUBST (F K K1 K2 R)                00002490
                                OTOPEX(VCNP)                            00002500
                                O1OPEX (PLUS DIFFERENCE)                00002510
                                COUTMETHODE 5                           00002520
                                MSGSUCCES "METHODE 10"                  00002530
```

D5 b

```
                    )
              POUR (RAMENER (APFORBIN (SIG F K K1 K2)) R) ESSAYER
              (SUC 10 (!%CREER (R1 R2 R3 R4))
                 (!%ATTRIB (!%VAL R1)(!%TERMES VCNP F))
                 (CHOIX 4 (!%ATTRIB(!%VAL R2)(!%VUFORMTERM
                 (!%ALGEB FORMLIN11101)(K)(!%ARG2(!%TETE (!%VVAL R1)))))
                       (!%ATTRIB(!%VAL R2)(!%VUFORMTERM
                 (!%ALGEB FORMLIN11102)(K)(!%ARG2(!%TETE (!%VVAL R1)))))
                       (!%ATTRIB(!%VAL R2)(!%VUFORMTERM
                 (!%ALGEB FORMLIN11103)(K)(!%ARG2(!%TETE (!%VVAL R1)))))
                       (!%ATTRIB(!%VAL R2)(!%VUFORMTERM
                 (!%ALGEB FORMLIN11104)(K)(!%ARG2(!%TETE (!%VVAL R1)))))
                 )
                 (!%ATTRIB (!%VAL R2)(!%TT (!%VVAL R2)))
                 (!%MONTRER R2)
                 (!%ATTRIB (!%VAL R3)(!%NOUVAR))
                 (EFFECTUER (CHGVARLIE (SIG F K K1 K2)
                                       (!%VAL R3)(!%VVAL R2))
                            (!%VAL R4))
                 (!%MONTRER R4)
                 (EVALUER (!%VVAL R4) R)
                 (!%LIBERER (R1 R2 R3 R4))
                 ) )$
%
PLAN(121):='(SIG121 (DESCRIPTEURS $
                     VARSUBST (F K K1 K2 NV EX R)
                     OTOPEX (SIG)
                     COUTMETHODE 5
                     MSGSUCCES "METHODE 11"
                    )
              POUR (EFFECTUER (CHGVARLIE (SIG F K K1 K2) NV EX) R)
       ESSAYER (CHOIX 3
        (SUC 5 (!%CREER (R1 R2))
           (!%ATTRIB (!%VAL R1)(!%FORMEXPLICITE EX (K)
                                (!%ALGEB FORMLIN11101)))
           (!%ATTRIB (!%VAL R2)(!%ARG2 EX))
           (!%ATTRIB R (SIG (!%SIMPLIF(!%REMPL K (DIFFERENCE NV
                                                 (!%VVAL R2)) F))
                       NV (!%SIMPLIF (PLUS (!%VVAL R2) K1))
                          (!%SIMPLIF (PLUS (!%VVAL R2) K2))) )
           (!%LIBERER (R1 R2))
           )
        (SI (EX EST (DIFFERENCE K K1))
            (!%ATTRIB R (SIG (!%REMPL K (PLUS NV K1) F)
                             NV 0 (!%SIMPLIF (DIFFERENCE K2 K1))))
            )

        (SI (EX EST (DIFFERENCE K2 K))
            (!%ATTRIB R (SIG (!%REMPL K (DIFFERENCE K2 NV)F)
                             NV 0 (!%SIMPLIF (DIFFERENCE K2 K1))))
            ) ))$
%$
ALGEBRAIC$
FOR ALL N LET VCNP(N,-1) = 0 , VCNP(N,N+1) = 0 , VCNP(N-1,N) = 0,
              VCNP(N,N  ) = 1 , VCNP(N,1   ) = N , VCNP(N,N-1) = N $
FOR ALL N LET VCNP(0,N) = 0 , VCNP(N,0) = 1 $
OFF EXP$
LISP$
%$
%FORMULES DU BINOME $
ALGEBRAIC OFF EXP,MCD$
REG(11):='(BINEWTON11 (DESCRIPTEURS VARSUBST (X Y K N)
                       CONDIT (ET (NONVUS K X)(NONVUS K Y))  ))$
ALGEBRAIC BINEWTON11:=REEC(SIG (X**K * Y**(N-K) * VCNP(N,K),K,0,N),
                           (X+Y)**N)$
REG(12):='(BINEWTON12 (DESCRIPTEURS VARSUBST (X P K N)
                       CONDIT (ET (NONVUS K X)(NONVUS K P)) ))$
ALGEBRAIC BINEWTON12:=REEC(SIG(X**(K+P) * VCNP(N,K),K,0,N),
                           (X**P)*(1+X)**N)$
REG(13):='(BINEWTON13 (DESCRIPTEURS VARSUBST (X K N)
                                    CONDIT (NONVUS K X)
                       )  )$
ALGEBRAIC BINEWTON13:=REEC(SIG (X**K * VCNP(N,K),K,0,N) ,
                           (1+X)**N )$
REG(14):='(BINEWTON14 (DESCRIPTEURS VARSUBST (K N)
                                    CONDIT (NONVUS K N)
                       )  )$
ALGEBRAIC BINEWTON14:=REEC(SIG (VCNP(N,K),K,0,N) ,
                           2**N )$
REG(15):='(BINEWTON15 (DESCRIPTEURS VARSUBST (X P K N)
                       CONDIT (ET (NONVUS K X)(NONVUS K P)) ))$
ALGEBRAIC BINEWTON15:=REEC(SIG(X**(K-P) * VCNP(N,K),K,0,N),
                           (X**(-P))*(1+X)**N)$
REG(16):='(BINEWTON16 (DESCRIPTEURS VARSUBST (X P K N)
                       CONDIT (ET (NONVUS K X)(NONVUS K P)) ))$
%SCALAR BINEWTON16$
BINEWTON16:='(REEC (SIG (TIMES(EXPT X (DIFFERENCE K P))(VCNP N K))
                        K 0 N)
                   (TIMES (EXPT X (MINUS P))
                          (EXPT (PLUS 1 X) N)) )$
%FORMES RECHERCHEES POUR LES EVALUATIONS DE SIG $
%SCALAR FCNP6,FCNP7,FCNP8,FCNP9$
%SCALAR FORMLIN11101,FORMLIN11102,FORMLIN11103,FORMLIN11104$
FCNP6:='((TIMES (QUOTIENT 1 (PLUS M 1))(VCNP N M))
         (DESCRIPTEURS VARSUBST (N M) PARAM (N) ))$
FCNP7:='((VCNP N (DIFFERENCE N K))
         (DESCRIPTEURS VARSUBST (N K) PARAM (N) ))$
FCNP8:='((TIMES M (VCNP N M))
         (DESCRIPTEURS VARSUBST(M N) PARAM (N) ))$
FCNP9:='((QUOTIENT 1 (DIFFERENCE K P))
         (DESCRIPTEURS VARSUBST(K P) PARAM (K) ))$
FORMLIN11101:='((PLUS K P)
          (DESCRIPTEURS VARSUBST(K P) PARAM (K)
                        CONDIT (NONVUS K P) ))$
FORMLIN11102:='((DIFFERENCE K P)
          (DESCRIPTEURS VARSUBST(K P) PARAM (K)
                        CONDIT (NONVUS K P) ))$
FORMLIN11103:='((PLUS (MINUS K) P)
          (DESCRIPTEURS VARSUBST(K P) PARAM (K)
                        CONDIT (NONVUS K P) ))$
FORMLIN11104:='((PLUS (MINUS K)(MINUS P))
          (DESCRIPTEURS VARSUBST(K P) PARAM (K)
                        CONDIT (NONVUS K P) ))$
%$
%LISTE DE PB POUR TESTS$
PB(201):='(SUC 2 (EVALUER (SIG (TIMES (VCNP (PLUS N 1) U)
                                      (EXPT (MINUS 1)(DIFFERENCE U 1)))
                              U 0 0) RR)
          (!%ECRIRE (!%ALGEB RR)) )$
PB(202):='(SUC 2 (EVALUER (SIG (TIMES (PLUS NN 1)(VCNP NN KK))
                               KK 0 NN) RR)
          (!%ECRIRE (!%ALGEB RR)) )$
PB(203):='(SUC 2 (EVALUER (SIG (TIMES(EXPT(MINUS 1)U)
                                     (VCNP (PLUS NN 1) U))
                               U 1 (PLUS NN 1)) RR)
          (!%ECRIRE (!%ALGEB RR)) )$
PB(205):='(SUC 2 (EVALUER (SIG (DIFFERENCE
                        (TIMES (EXPT (MINUS 1)(DIFFERENCE U 1))
                               (VCNP (PLUS NN 1) U ))
                        (TIMES (EXPT (MINUS 1)(DIFFERENCE U 1))
                               (VCNP (PLUS NN 1) U)) )
                           U 0 (PLUS NN 1) ) RR)
          (!%ECRIRE (!%ALGEB RR)) )$
```

D5c

règles.

Problèmes.

```
PB(206):='(SUC 2 (EVALUER (SIG (MINUS (TIMES (EXPT (MINUS 1) KK)                00003860
                                              (VCNP (PLUS NN 1) KK)))           00003870
                          KK 1 (PLUS NN 1))  RR)                                00003880
              (!%ECRIRE (!%ALGEB RR)) )$                                        00003890
PB(215):='(SUC 2 (RAMENER (APPORBIN (SIG (TIMES (EXPT (MINUS 1)                00003900
                                                       U )                      00003910
                                                 (VCNP (PLUS NN 1)U))           00003920
                                        U O (PLUS NN 1)))RR)                    00003930
              (!%ECRIRE (!%ALGEB RR)) )$                                        00003940
PB(216):='(SUC 2 (RAMENER (APPORBIN (SIG (TIMES (EXPT(MINUS 1)KK)               00003950
                                    (TIMES(QUOTIENT 1(PLUS KK 1))               00003960
                                          (VCNP NN KK)))                        00003970
                                  KK O NN)) RR)                                 00003980
              (!%ECRIRE (!%ALGEB RR)) )$                                        00003990
PB(217):='(SUC 2 (RAMENER (APPORBIN                                             00004000
                (SIG (TIMES 4 (VCNP NN (DIFFERENCE NN I)))                      00004010
                     I O NN)) RR)                                               00004020
              (!%ECRIRE (!%ALGEB RR)) )$                                        00004030
PB(219):='(SUC 2 (RAMENER (APPORBIN                                             00004040
                (SIG (TIMES (EXPT (MINUS 1)U)                                   00004050
                            (VCNP (PLUS NN 1) U ))                              00004060
                  U 1 (PLUS NN 1))) RR)                                         00004070
              (!%ECRIRE (!%ALGEB RR)) )$                                        00004080
PB(219):='(SUC 2 (RAMENER (APPORBIN                                             00004090
         (SIG (MINUS (TIMES (EXPT (MINUS 1)U)                                   00004100
                            (VCNP (PLUS NN 1) U )) )                            00004110
                  U 1 (PLUS NN 1))) RR)                                         00004120
              (!%ECRIRE (!%ALGEB RR)) )$                                        00004130
PB(220):='(SUC 2 (RAMENER (APPORBIN                                             00004140
                (SIG (TIMES (EXPT (MINUS 1) KK)(VCNP (PLUS NN 1)                00004150
                                                     (PLUS KK 1)))              00004160
                  KK O NN)) RR)                                                 00004170
              (!%ECRIRE (!%ALGEB RR)) )$                                        00004180
PB(221):='(SUC 2 (EFFECTUER (CHGVARLIE                                          00004190
                (SIG (TIMES (EXPT (MINUS 1) KK)(VCNP (PLUS NN 1)                00004200
                                                     (PLUS KK 1)))              00004210
                  KK O NN) SERVOX (PLUS KK 1))  RR)                             00004220
              (!%ECRIRE (!%ALGEB RR)) )$                                        00004230
PB(222):='(SUC 2 (EVALUER (SIG                                                  00004240
                (VCNP NN I) I O NN ) RR)                                        00004250
                (!%ECRIRE (!%ALGEB RR)) )$                                      00004260
PB(223):='(SUC 2 (EVALUER (SIG                                                  00004270
                (TIMES(EXPT(MINUS 1)I)(VCNP NN I)) I O NN ) RR)                 00004280
                (!%ECRIRE (!%ALGEB RR)) )$                                      00004290
PB(224):='(SUC 2 (EVALUER (SIG                                                  00004300
                (VCNP  NN (PLUS I 1)) I O NN ) RR)                              00004310
                (!%ECRIRE (!%ALGEB RR)) )$                                      00004320
PB(225):='(SUC 2 (EVALUER (SIG                                                  00004330
                (VCNP (PLUS NN 1)(PLUS I 1)) I O NN ) RR)                       00004340
                (!%ECRIRE (!%ALGEB RR)) )$                                      00004350
PB(226):='(SUC 2 (EVALUER (SIG                                                  00004360
                (TIMES I (VCNP NN I)) I O NN ) RR)                              00004370
                (!%ECRIRE (!%ALGEB RR)) )$                                      00004380
PB(200):='(SUC 2 (EVALUER (SIG                                                  00004390
            (TIMES (EXPT (MINUS 1)KK)(TIMES (QUOTIENT 1 (PLUS KK 1))            00004400
                                            (VCNP NN KK)) )                     00004410
                  KK O NN)  RR)                                                 00004420
              (!%ECRIRE (!%ALGEB RR)) )$                                        00004430
%.....................................................$                        00004440
SHUT REDUFICH$ OUT T $ TERPRI()$ WRITE "--->FIN DES COMBINES" $                 00004450
%.....................................................$                        00004460
ON ECHO$                                                                        00004470
END$                                                                            00004480
% EXEMPLES$                                                                    00004490
LISP $ %SCALAR EX $                                                             00004500
EX:='(TIMES 12 (VCNP N M)(PLUS (VCNP K H) 5));                                  00004510
TERMES ('VCNP,EX);                                                              00004520
SOUSTERME('(VCNP K H),EX);                                                      00004530
SOUSTERME('(VCNP N M),EX);                                                      00004540
EX:='(TIMES (TIMES 3 4) 5 (TIMES 6 7));                                         00004550
TERMES ('TIMES,EX);                                                             00004560
%;                                                                              00004570
%SCALAR F;                                                                      00004580
F:=FCNP6;                                                                       00004590
EX:='(TIMES 3 (TIMES (QUOTIENT 1 (PLUS K 1))(VCNP NN K)));                      00004600
SOUSTERME('(PLUS K 1),EX);                                                      00004610
VUFORMTERM(F,'(NN),EX);                                                         00004620
EX:='(TIMES (TIMES (QUOTIENT 1 (PLUS K 1))(VCNP NN K))                          00004630
            (TIMES (QUOTIENT 1 (PLUS H 1))(VCNP NN H)) );                       00004640
SOUSTERME('(QUOTIENT 1 (PLUS H 1)),EX);                                         00004650
VUFORMTERM(F,'(NN),EX);                                                         00004660
F:=FORMLIN1101;                                                                 00004670
EX:='(PLUS (PLUS KK Q)(PLUS KK (PLUS KK QQ)));                                  00004680
VUFORMTERM(F,'(KK),EX);                                                         00004690
%=====================================================$                       00004700
```

D5 d

tests des procédures ajoutées.

ANNEXE E1

FICHE TECHNIQUE
REPARTITION DU CODE DANS CAMELIA

-CAMELIA : Janvier 1984 : environ 6 000 lignes
(réparties en 2 500 de procédures. 3 500 dans les chapitres de la base de connaissances).

-Moteur d'inférence : 200 lignes

-Procédures d'unifications, filtrage : 300 lignes

-Procédures de service (évaluateur de conditions, prétraitement, exécuteur d'actions immédiates,...) initialisations, tables, déclarations..... 1 000 lignes

-Code de procédures représentant les actions immédiates d'intérêt général : 800 lignes

- gestion de méta-connaissances 100 lignes

-Méta-règles 100 lignes

-Chapitres de connaissances

PARITE : 350 lignes

LIMITES + Developpements limités : 1 800 lignes

INTEGRATION 900 lignes

COMBINAISONS 450 lignes

ANNEXE F

RESULTATS - LISTES DE PROBLEMES RESOLUS

Cette annexe contient une liste de problèmes effectivement résolus par CAMELIA avec la base de connaissances actuelle dans les domaines suivants :

- preuve de parité : 22 problèmes
- évaluation de combinaisons : 17 problèmes
- calcul de primitives : 38 problèmes
- calcul de limites : 28 problèmes
- calcul de développement limités : 23 problèmes.

- Présentation des résultats

* Chaque problème est repéré par son numéro PB(i) suivi de :.
Sur la même ligne nous avons noté le temps CPU en ms sur unité SY1 (NAS 9080 du CIRCE-CNRS).
Les temps précédés d'un signe - correspondent à des temps obtenus sur SY2 et doivent être réduit d'environ 36 % pour être comparés aux temps obtenus surSY1. Ces temps ne sont donnés qu'à titre indicatif comme ordre de grandeur.

* Les temps donnés ont été tous obtenus avec leur version interprétée de REDUCE. Une version compilée de REDUCE existe et a pu être testée sur des exemples simples de calcul algébrique. Sur le calcul de factorielle (50) répétés 100 fois, on a pu observer qu'avec la version compilée, il est possible de gagner environ 25% de temps à l'exécution. Cette possibilité devrait être mise en oeuvre si CAMELIA devait entrer en exploitation. La compilation est coûteuse en temps et pour un programme en mouvance permanente il n'était pas utile de compiler chaque fois.

* Dans la marge nous avons décrit la structure des appels de plans nécessaires pour résoudre le problème. La notation parenthésée a été expliquée en IV.3.3.4. Elle permet d'avoir une idée des imbrications de plans et de la complexité des solutions qui peuvent être apportées. Lorsqu'une liste de numéro est soulignée et marquée CP, cela signifie qu'il s'agit d'une condition problème c'est-à-dire d'une preuve que le moteur a été amené à établir pour pouvoir conduire le calcul.

NOTE.

Eléments ayant permis de comparer les performances sur les deux processeurs SY1 et SY2 du CIRCE.
Temps de chargement sur SY1 : 1272,1328,1306,1264.
Moyenne : 1292 ms
Temps de chargement sur SY2 : 2750,2776,2742,2729,2712.
Moyenne : 2741 ms

- Pour les opérations de chargement de CAMELIA et de la base de connaissances concernant la parité le rapport des performances s'établit à o,47 entre SY1 et SY2. Ces opérations nécessitent beaucoup d'entrées/sorties.

- Pour des exécutions, les problèmes 60 à 63 (consacrés au calcul de limites mais de difficulté différente) ont été amenés à passer sur les deux processeurs, les solutions fournies étant les mêmes. Dans les résultats, le temps SY1 est noté entre parenthèse et le temps SY2 est précédé du signe -.
Sur ces 4 exercices, le rapport est très stable et on peut considérer qu'il permet une bonne appréciation des performances relatives des 2 processeurs.
Les mesures sont les suivantes :

	SY1	SY2	RAPPORT
PB(60) :	15100	23659	0,6382
PB(61) :	27168	42348	0,6415
PB(62) :	3218	5005	0,6430
PB(63) :	9394	14646	0,6414

```
NIL
PB(9):  11498
(PRV (PAIRE (TIMES (COS X) (VABS X)) X))


PB(10):  1318
(PRV (IMPAIRE (!%ARG2 (EXPT Y X)) X))


PB(11):  678
(PRV (PAIRE X X))


PB(12):  16168
(PRV (PAIRE (QUOTIENT (PLUS (EXPT E X) (EXPT E (MINUS X))) 2) X))


PB(13):  3985
(PRV (PAIRE (EXPT E X) X))


PB(14):  16671
(PRV (PAIRE (COS (TIMES K X)) X))


PB(15):  7677
(PRV (PAIRE (SIN (COS X)) X))


PB(16):  5531
(PRV (PAIRE (COS (SIN X)) X))


PB(17):  4862
(SI
   (UNVRAI ((SIN X) (COS X)) (PAIRE !? X))
   (!%ECRIRE "EXEMPLE DE CONDITION PROBLEME SOUS UNVRAI"))


PB(18):  1982
(SI
   (AUCUNVRAI ((COS X) (EXPT X 2)) (IMPAIRE !? X))
   (!%ECRIRE "NI COS(X) NI X**2 N'EST IMPAIRE"))


PB(19):  3261
(CHOIX
   2
   (SI
      (TOUSVRAI ((SIN X) (TG X)) (PAIRE !? X))
      (!%ECRIRE "RIDICULE:NI SIN(X) NI TG(X) NE SONT PAIRES"))
   (SI
      (AUCUNVRAI ((SIN X) (TG X)) (PAIRE !? X))
      (!%ECRIRE "BON:NI SIN(X) NI TG(X) NE SONT PAIRES EN X")))
```

0000000
0000000
0000000 (5 (6 (14)) (3))
0000000
0000000
0000000
0000000
0000000 (14)
0000000
0000000
0000000
0000000
0000000 niveau méta écarte le problème.
0000000
0000000
0000000
0000000
0000000 (5 (9)(8))
0000000
0000000
0000000
0000000
0000000 problème absurde!-
0000000
0000000
0000000
0000000
0000000 (6 (16 (8)(14)))
0000000
0000000
0000000
0000000
0000000 (10 (6 (14)))
0000000
0000000
0000000
0000000
0000000 (6 (14))
0000000
0000000
0000000
0000000
0000000 (3) condition problème -
0000000
0000000
0000000
0000000
0000000
0000000
0000000 & conditions - problèmes -
0000000
0000000
0000000
0000000
0000000
0000000
0000000 & conditions - problèmes sous CHOIX
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000

PB(20): 15458
(PRV (PAIRE (PLUS (TIMES X X) (COS X)) X))

PB(21): 10926
(PRV (PAIRE (DIFFERENCE (COS X) (VABS X)) X))

PB(22): -17975
(PRV (PAIRE (EXPT E (PLUS (COS (EXPT X 3)) (EXPT X 2))) X))

PB(23): -12041
(PRV (PAIRE (EXPT (PLUS (TG X) X) 4) X))

PB(24): -16978
(PRV (PAIRE (TIMES (EXPT (SIN X) 5) (TG (QUOTIENT X 2))) X))

PB(25): -5237
(PRV (PAIRE (EXPT E (EXPT X 2)) X))

PB(26): -5752
(PRV (PAIRE (EXPT (COS X) 4) X))

PB(30): 21545
(PRV (IMPAIRE (PLUS (TIMES X (VABS X)) (EXPT (SIN X) 3)) X))

PB(31): 5375
(PRV (IMPAIRE (SIN (TG X)) X))

PB(32): 29814
(PRV
  (IMPAIRE
    (QUOTIENT (DIFFERENCE (EXPT E X) (EXPT E (MINUS X))) 2)
    X))

PB(33): 4305
(PRV (IMPAIRE (EXPT X 3) X))

PB(200): -52782
(SUC
  2
  (EVALUER
    (SIG

Evaluer $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1} \binom{n}{k}$

```
            (EXPT (MINUS 1) KK)
            (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK)))
         KK
         O
         NN)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(201):-10431 — Evaluer $\sum_{u=0}^{0} (-1)^{u-1} C_{n+1,u}$ (101)

```
(SUC
   2
   (EVALUER
      (SIG
         (TIMES
            (VCNP (PLUS N 1) U)
            (EXPT (MINUS 1) (DIFFERENCE U 1)))
         U
         O
         O)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(202):-13754 — Evaluer $\sum_{k=0}^{n} (n+1) C_{n,k}$ (102 (103 (115)))

```
(SUC
   2
   (EVALUER (SIG (TIMES (PLUS NN 1) (VCNP NN KK)) KK O NN) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(203):-29727 — Evaluer $\sum_{u=1}^{n+1} (-1)^{u} C_{n+1,u}$ (109 (115))

```
(SUC
   2
   (EVALUER
      (SIG
         (TIMES (EXPT (MINUS 1) U) (VCNP (PLUS NN 1) U))
         U
         1
         (PLUS NN 1))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(205): 7871 — Evaluer $\sum_{u=0}^{n+1} (-1)^{u-1} C_{n+1,u} - (-1)^{n-1} C_{n+1,u}$ (105)

```
(SUC
   2
   (EVALUER
      (SIG
         (DIFFERENCE
            (TIMES
               (EXPT (MINUS 1) (DIFFERENCE U 1))
               (VCNP (PLUS NN 1) U))
            (TIMES
               (EXPT (MINUS 1) (DIFFERENCE U 1))
               (VCNP (PLUS NN 1) U)))
         U
         O
         (PLUS NN 1))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(206):-38951 Evaluer $\sum_{k=1}^{n+1} (-1)^k C_{n+1,k}$ (106 (109 (115)))

```
(SUC
   2
   (EVALUER
      (SIG
         (MINUS (TIMES (EXPT (MINUS 1) KK) (VCNP (PLUS NN 1) KK)))
         KK
         1
         (PLUS NN 1))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(215):-4079 Ramener apforbin $\sum_{u=0}^{n+1} (-1)^u C_{n+1,u}$ (115)

```
(SUC
   2
   (RAMENER
      (APFORBIN
         (SIG
            (TIMES (EXPT (MINUS 1) U) (VCNP (PLUS NN 1) U))
            U
            0
            (PLUS NN 1)))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(216):-57487 Ramener apforbin $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1} C_{n,k}$ (116 (120 (121) (106 (109 (115)))))

```
(SUC
   2
   (RAMENER
      (APFORBIN
         (SIG
            (TIMES
               (EXPT (MINUS 1) KK)
               (TIMES (QUOTIENT 1 (PLUS KK 1)) (VCNP NN KK)))
            KK
            0
            NN))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(217):-5799 Ramener apforbin $\sum_{i=0}^{n} 4.C_{n,n-i}$ (117)

```
(SUC
   2
   (RAMENER
      (APFORBIN (SIG (TIMES 4 (VCNP NN (DIFFERENCE NN I))) I 0 NN))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(219): 38401 Ramener apforbin $\sum_{u=1}^{n+1} -\left((-1)^u C_{n+1,u}\right)$ (119 (106 (103 (115))) (106 (101)))

```
(SUC
   2
   (RAMENER
      (APFORBIN
         (SIG
            (MINUS (TIMES (EXPT (MINUS 1) U) (VCNP (PLUS NN 1) U)))
```

```
            1
            (PLUS NN 1)))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(220): -31467

Ramener apforbin $\sum_{k=0}^{n} (-1)^k C_{n+1,k+1}$ (120 (121)(106 (109 (115))))

```
(SUC
   2
   (RAMENER
      (APFORBIN
         (SIG
            (TIMES
               (EXPT (MINUS 1) KK)
               (VCNP (PLUS NN 1) (PLUS KK 1)))
            KK
            0
            NN))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(221): -2609

effectuer(chgvar $\sum_{k=0}^{h} (-1)^k C_{n+1,k+1}$, s, $k+1$) (121)

```
(SUC
   2
   (EFFECTUER
      (CHGVARLIE
         (SIG
            (TIMES
               (EXPT (MINUS 1) KK)
               (VCNP (PLUS NN 1) (PLUS KK 1)))
            KK
            0
            NN)
         SERVOX
         (PLUS KK 1))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

Evaluer $\sum_{k=0}^{n} C_{n,k}$

PB(222): -5547

(SUC 2 (EVALUER (SIG (VCNP NN I) I 0 NN) RR) (!%ECRIRE (!%ALGEB RR))) (103 (115))

PB(223): -9077

Evaluer $\sum_{k=0}^{n} (-1)^k C_{n,k}$ (103 (115))

```
(SUC
   2
   (EVALUER (SIG (TIMES (EXPT (MINUS 1) I) (VCNP NN I)) I 0 NN) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(224): -29672

Evaluer $\sum_{k=0}^{n} C_{n,k+1}$ (103 (120 (121)(107 (103 (115)))))

```
(SUC
   2
   (EVALUER (SIG (VCNP NN (PLUS I 1)) I 0 NN) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(225): -24464

Evaluer $\sum_{k=0}^{n} C_{n+1,k+1}$ (103 (120 (121)(109 (115))))

```
(SUC
   2
   (EVALUER (SIG (VCNP (PLUS NN 1) (PLUS I 1)) I O NN) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(226): -41485 — Evaluer $\sum_{k=0}^{n} k.C_{n,k}$

```
(SUC
   2
   (EVALUER (SIG (TIMES I (VCNP NN I)) I O NN) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 0000000 0000000 0000000 0000000 0000000 0000000 (110 (103 (118 (120 (121) (103 (115))))))

NIL
PB(140): -4894 — $\int \cos x \, dx$

```
(SUC 2 (CALCUL (PRIM (COS X) X) RR) (!%ECRIRE (!%ALGEB RR)))
```

0000000 (20)

PB(141): -17509 — $\int (\sin x + \cos x) \, dx$

```
(SUC
   2
   (CALCUL (PRIM (PLUS (SIN X) (COS X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (22 (20)(20))

PB(142): — $\int (\sin x + \cos x)^3 \, dx$

```
(SUC
   2
   (CALCUL (PRIM (EXPT (PLUS (SIN X) (COS X)) 3) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(143): -22462 V 7376 — $\int \frac{1+x}{x} \, dx$

```
(SUC
   2
   (CALCUL (PRIM (QUOTIENT (PLUS 1 X) X) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (24 (20)(27)) V RISCH(34)

PB(144): -36724 V 20610 — $\int (x+1)^2 \, dx$

```
(SUC
   2
   (CALCUL (PRIM (EXPT (PLUS X 1) 2) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (23 (22 (28) (31 (60 (20)(27))) (27)))
0000000 V (23(22 (28) (25 (20)) (27)))

PB(145): -22210 V 11800 — $\int (1+A) \frac{1}{x} \, dx$

```
(SUC
   2
   (CALCUL (PRIM (TIMES (PLUS 1 A) (QUOTIENT 1 X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (31 (60 (20)(27)))
0000000 V (25 (20))

PB(146): -22348 V 14187 — $\int x.\cos(x) \, dx$

```
(SUC
   2
   (CALCUL (PRIM (TIMES X (COS X)) X) RR)
```

0000000 (33(60 (20)(20)))

PB(147): 14282 V -26316 $\int \sin\varphi\,(x+1)\,dx$ (26 (22 (20)(27))) V (31 (60 (20)(27)(27)))

```
(SUC
   2
   (CALCUL (PRIM (TIMES (SIN PHI) (PLUS X 1)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(148): 16667 $\int x^{33+AA} \sin BB\,dx$ (33 (60 (27)))

```
(SUC
   2
   (CALCUL (PRIM (TIMES (EXPT X (PLUS 33 AA)) (SIN BB)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(149): 3138 $\int x^{5+n}\,dx$ (28)

```
(SUC
   2
   (CALCUL (PRIM (EXPT X (PLUS N 5)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(150): 18358 $\int tg^2 x\,dx$ (41 (34))

```
(SUC 2 (CALCUL (PRIM (EXPT (TG X) 2) X) RR) (!%ECRIRE (!%ALGEB RR)))
```

PB(151): 1790 $\int x\,dx$ (20)

```
(SUC 2 (CALCUL (PRIM X X) RR) (!%ECRIRE (!%ALGEB RR)))
```

PB(152): 26488 $\int 3\cdot x^2 \cdot \log x\,dx$ (30 (33 (60 (28)(44 (28)))))

```
(SUC
   2
   (CALCUL (PRIM (TIMES 3 (EXPT X 2) (LOG X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(153): 84623 $\int x^3 \sin x\,dx$ (33 (60 (20)(53 (30 (33 (60 (20) (30 (33 (60 (20) (53 (20))))))))))))

```
(SUC
   2
   (CALCUL (PRIM (TIMES (EXPT X 3) (SIN X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(154): +19915 $\int \log(x^2+1)\,dx$ (38 (60 (27) (34)))

```
(SUC
   2
   (CALCUL (PRIM (LOG (PLUS (EXPT X 2) 1)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(155): +26603 $\int \sin x \log(\cos x)\,dx$ (31 (60 (20)(20)))

```
(SUC
   2
   (CALCUL (PRIM (TIMES (SIN X) (LOG (COS X))) X) RR)
```

```
   (!%ECRIRE (!%ALGEB RR)))
```

PB(156): −129530 V 39607 — $\int tg(x)\, Log(\cos x)\, dx$ — (31 (60 (20)(41 (16(15(14)(10 (3)))(3)) (36 (54 (38)))))) v (41(19)(36(54(35)

```
(SUC
   2
   (CALCUL (PRIM (TIMES (TG X) (LOG (COS X))) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(157): −36030 — $\int \frac{ch\,x\,(1-sh^2x)}{sh\,x\,(1-ch^2x)}\,dx$ — (46 (34))

```
(SUC
   2
   (CALCUL
      (PRIM
         (QUOTIENT
            (TIMES (CH X) (DIFFERENCE 1 (EXPT (SH X) 2)))
            (TIMES (SH X) (DIFFERENCE 1 (EXPT (CH X) 2))))
         X)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(158): −45724 — $\int \frac{x^2}{(\cos x + x\sin x)^2}\,dx$ — (39 (50 (20)))

```
(SUC
   2
   (CALCUL
      (PRIM
         (QUOTIENT
            (EXPT X 2)
            (EXPT (PLUS (COS X) (TIMES X (SIN X))) 2))
         X)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(159): −32023 — $\int \frac{(x+2)^3}{x}\,dx$ — (25 (34))

```
(SUC
   2
   (CALCUL (PRIM (QUOTIENT (EXPT (PLUS X 2) 3) X) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(160): 48360 — $\int \cos^3 x\,dx$ — (42 (41 (22 (53 (28)) (27))))

```
(SUC 2 (CALCUL (PRIM (EXPT (COS X) 3) X) RR) (!%ECRIRE (!%ALGEB RR)))
```

PB(161): 36665 — $\int \cos^2 x\,dx$ — (42 (22 (51)(27))))

```
(SUC 2 (CALCUL (PRIM (EXPT (COS X) 2) X) RR) (!%ECRIRE (!%ALGEB RR)))
```

PB(162): −11104 — $\int \cos 2x\,dx$ — (51)

```
(SUC
   2
   (CALCUL (PRIM (COS (TIMES 2 X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

```
(SUC
   2
   (CALCUL (PRIM (SIN (PLUS (TIMES K X) H)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(164): -14684 $\int \sin(\sin(k)x + \cos h)\,dx$ (52)

```
(SUC
   2
   (CALCUL (PRIM (SIN (PLUS (TIMES (SIN K) X) (COS H))) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(165): -28832 $\int 5x^3\,dx$ (33 (60 (26 (28))))

```
(SUC
   2
   (CALCUL (PRIM (TIMES (EXPT X 3) 5) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(166): -18192 V 10960 $\int \frac{dx}{x^2+\sin^2 u}$ (34) Risch V (20)

```
(SUC
   2
   (CALCUL
      (PRIM (QUOTIENT 1 (PLUS (EXPT (SIN U) 2) (EXPT X 2))) X)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(167): 10816 $\int \frac{dx}{\sqrt{x^2+\cos^2 \varphi}}$ (20)

```
(SUC
   2
   (CALCUL
      (PRIM
         (QUOTIENT 1 (RAC2 (PLUS (EXPT (COS FI) 2) (EXPT X 2))))
         X)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(168): 8559 $\int \frac{x^2\,dx}{\sqrt{x^2+\operatorname{Log} F}}$ (20)

```
(SUC
   2
   (CALCUL
      (PRIM (QUOTIENT X (RAC2 (PLUS (EXPT (LOG F) 2) (EXPT X 2)))) X)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(169): 7928 $\int \frac{dx}{\sqrt{x^2+9}}$ (20)

```
(SUC
   2
   (CALCUL (PRIM (QUOTIENT 1 (RAC2 (PLUS 9 (EXPT X 2)))) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(170): -8832 $\int \operatorname{sh} x \operatorname{ch} x\,dx$ (38)

```
(SUC
```

```
   2
   (CALCUL (PRIM (TIMES (SH X) (CH X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(171): -20081 $\int \frac{sh^3 x}{ch^2 x}\,dx$ (46 (34))

```
(SUC
   2
   (CALCUL (PRIM (QUOTIENT (EXPT (SH X) 3) (EXPT (CH X) 2)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(172): -31368 $\int \frac{ch\,x}{2-ch^2 x}\,dx$ (46 (50 (34)))

```
(SUC
   2
   (CALCUL
      (PRIM (QUOTIENT (CH X) (DIFFERENCE 2 (EXPT (CH X) 2))) X)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(173): -31987 $\int tg^3 x\,dx$ (41 (18 (14)) (34)) CP

```
(SUC 2 (CALCUL (PRIM (EXPT (TG X) 3) X) RR) (!%ECRIRE (!%ALGEB RR)))
```

PB(174): 31146 $\int \cos(x) tg^2 x\,dx$ (41 (34))

```
(SUC
   2
   (CALCUL (PRIM (TIMES (COS X) (EXPT (TG X) 2)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(175): 27841 $\int \frac{\sin x}{\cos^3 x}\,dx$ (41 (16 (14)(9)) (50 (20))) CP

```
(SUC
   2
   (CALCUL (PRIM (QUOTIENT (SIN X) (EXPT (COS X) 3)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(176): 33077 $\int 3\sin^2 x \cos x\,dx$ (41 (33 (60 (28)(27))))

```
(SUC
   2
   (CALCUL (PRIM (TIMES 3 (EXPT (SIN X) 2) (COS X)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(177): -62066 V 37836 $\int 3\sin x \cos^2 x\,dx$ (41 (16 (14)) (53 (33 (60 (28)(27))))) CP

V (41 (19) (53 (33 (30 (28)(27))))) CP

```
(SUC
   2
   (CALCUL (PRIM (TIMES 3 (SIN X) (EXPT (COS X) 2)) X) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(178): -89502 V 32011 $\int \sin^3 x\,dx$ (43 (41 (16 (14)(4 (12 (7 (3))) (8))) (22 (28) (53 (27)))))

V (43 (41 (19) (56)) CP

```
(SUC 2 (CALCUL (PRIM (EXPT (SIN X) 3) X) RR) (!%ECRIRE (!%ALGEB RR)))
```

```
(SUC 2 (CALCUL (LIM (LOG (COS X)) X 0) RR) (!%ECRIRE (!%ALGEB RR)))
```

0000000

PB(51): 21707 $\lim_{x\to 0} e^{1-\cos x}$

```
(SUC
   2
   (CALCUL (LIM (EXPT E (DIFFERENCE 1 (COS X))) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (61 (42 (40)(41)))

PB(52): 15074 $\lim_{x\to 0} |\sin(0) - 1|$

```
(SUC
   2
   (CALCUL (LIM (ABS (DIFFERENCE (SIN X) 1)) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (49 (43 (40)(41)))

PB(53): 13194 $\lim_{x\to 0} -\operatorname{tg}(2x)$

```
(SUC
   2
   (CALCUL (LIM (MINUS (TG (TIMES 2 X))) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (49 (54 (64)))

PB(54): 5915 $\lim_{x\to 0} (2x^2+x+3)$

```
(SUC
   2
   (CALCUL (LIM (PLUS (TIMES 2 (EXPT X 2)) X 3) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (64)

PB(55): 12069 $\lim_{x\to 0} (\sin x - \operatorname{tg} x)$

```
(SUC
   2
   (CALCUL (LIM (DIFFERENCE (SIN X) (TG X)) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (43 (40)(40))

PB(56): 12485 $\lim_{x\to 0} \operatorname{tg}(e^x)$

```
(SUC 2 (CALCUL (LIM (TG (EXPT E X)) X 0) RR) (!%ECRIRE (!%ALGEB RR)))
```

0000000 (54 (40))

PB(57): 12618 $\lim_{x\to\infty} \frac{e.x}{1+x}$

```
(SUC
   2
   (CALCUL (LIM (QUOTIENT (TIMES E X) (PLUS 1 X)) X PINF) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (47 (45 (41 (64))))

PB(58): $\lim_{x\to 0} \frac{\sin(2x^3)}{\cos^3 x - 1}$

```
(SUC
   2
   (CALCUL
      (LIM
         (QUOTIENT (SIN (EXPT X 3)) (DIFFERENCE (EXPT (COS X) 3) 1))
         X
         0)
```

0000000

```
   RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(59): 39594

$$\lim_{x\to 0} \frac{\sin(x^7)}{x^7}$$

```
(SUC
   2
   (CALCUL (LIM (QUOTIENT (SIN (EXPT X 7)) (EXPT X 7)) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(45 (52 (64)) (64) (69 (90 (79 (73)))))

PB(60): -23659 (15900)

$$\lim_{x\to 0} (e^x + \log x)$$

```
(SUC
   2
   (CALCUL (LIM (PLUS (EXPT E X) (LOG X)) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(48 (40) (40))

PB(61): -48348 (27168)

$$\lim_{x\to 0} \frac{e^x}{\sin x + \cos x}$$

```
(SUC
   2
   (CALCUL (LIM (QUOTIENT (EXPT E X) (PLUS (SIN X) (COS X))) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(45 (40) (48 (40) (40)))

PB(62): -5005 (3818)

$$\lim_{x\to 0} (\cos x)$$

```
(SUC 2 (CALCUL (LIM (COS X) X 0) RR) (!%ECRIRE (!%ALGEB RR)))
```

(40)

PB(63): -14646 (9394)

$$\lim_{x\to 1} \cos(x-1)$$

```
(SUC
   2
   (CALCUL (LIM (COS (DIFFERENCE X 1)) X 1) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(46 (40))

PB(64): -29780

$$\lim_{x\to 0} (\sin x + \cos x)^3$$

```
(SUC
   2
   (CALCUL (LIM (EXPT (PLUS (SIN X) (COS X)) 3) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(60 (48 (40) (40)))

PB(65): -18864

$$\lim_{x\to 0} \sin(x) * \cos(x)$$

```
(SUC
   2
   (CALCUL (LIM (TIMES (SIN X) (COS X)) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(44 (40) (40))

PB(66): -6833

$$\lim_{x\to \infty} \sin\left(\frac{1}{x}\right)$$

```
(SUC
   2
   (CALCUL (LIM (SIN (QUOTIENT 1 X)) X PINF) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(47 (40))

```
(SUC                                                                    x→n    sinπx
   2
   (CALCUL
      (LIM
         (TIMES
            (EXPT -1 N)
            (QUOTIENT (DIFFERENCE X N) (SIN (TIMES PI X))))
         X
         N)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

$\lim_{x\to 0} \cos(\sin x)$

PB(68): 8272 V 7829

```
(SUC 2 (CALCUL (LIM (COS (SIN X)) X 0) RR) (!%ECRIRE (!%ALGEB RR)))
```

(49 (52)) V (53 (40))

PB(69): 22839

$\lim_{x\to 0} \frac{\sin x}{x}$

```
(SUC
   2
   (CALCUL (LIM (QUOTIENT (SIN X) X) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(50 (45 (40)(41))) L'HOSPITAL

PB(70): 31282

$\lim_{x\to 0} \sqrt{|e^x - 1|}$

```
(SUC
   2
   (CALCUL (LIM (RAC2 (ABS (DIFFERENCE (EXPT E X) 1))) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(55 (56 (43 (40)(41))))

PB(71): 16487

$\lim_{x\to 0} \cos(x)(\operatorname{tg} 2x)$

```
(SUC
   2
   (CALCUL (LIM (TIMES (COS X) (TG (TIMES 2 X))) X 0) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

(44 (40) (54 (64)))

PB(72):

$\lim_{x\to\infty} x^2\left(e^{\frac{1}{x}} - e^{\frac{1}{x+1}}\right)$

```
(SUC
   2
   (CALCUL
      (LIM
         (TIMES
            (EXPT X 2)
            (DIFFERENCE
               (EXPTE(QUOTIENT 1 X))
               (EXPTE(QUOTIENT 1 (PLUS 1 X)))) )
         X
         PINF)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(73):

$\lim_{x\to\frac{\pi}{6}} \left(\operatorname{tg}\frac{3x}{2}\right)^{\operatorname{tg} 3x}$

```
(SUC
   2
   (CALCUL
      (LIM
```

```
        (EXPT (TG (QUOTIENT (TIMES 3 X) 2)) (TG (TIMES 3 X)))
        X
        (QUOTIENT PI 6))
     RR)
  (!%ECRIRE (!%ALGEB RR)))
```

PB(74):

$$\lim_{x\to 0} \frac{24(1-\cos x)-12x^6+\sin(x^4)}{tg^6 x}$$

```
(SUC
   2
   (CALCUL
      (LIM
         (QUOTIENT
            (PLUS
               (DIFFERENCE
                  (TIMES 24 (DIFFERENCE 1 (COS X)))
                  (TIMES 12 (EXPT X 2)))
               (SIN (EXPT X 4)))
            (EXPT (TG X) 6))
         X
         0)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(75):

$$\lim_{x\to \frac{\pi}{2}} tg(x) * tg(2x)$$

```
(SUC
   2
   (CALCUL
      (LIM (TIMES (TG X) (TG (TIMES 2 X))) X (QUOTIENT PI 2))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(76):

$$\lim_{x\to \frac{1}{2}} \left(x-\frac{1}{2}\right) tg\,\pi x$$

```
(SUC
   2
   (CALCUL
      (LIM
         (TIMES (DIFFERENCE X (QUOTIENT 1 2)) (TG (TIMES PI X)))
         X
         (QUOTIENT1 2))
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(77):

$$\lim_{x\to\infty} x^2\left(1+\frac{1}{x}\right)^x - e\cdot x^3 \log\left(1+\frac{1}{x}\right)$$

```
(SUC
   2
   (CALCUL
      (LIM
         (DIFFERENCE
            (TIMES (EXPT X 2) (EXPT (PLUS 1 (QUOTIENT 1 X)) X))
            (TIMES
               E
               (TIMES (EXPT X 3) (LOG (PLUS 1 (QUOTIENT 1 X))))))
         X
         PINF)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(88): -19906

$DEV_0^5 \sin(e^x)$

```
(SUC
   2
   (CALCUL (DEVLIM (SIN (EXPT E X)) X O 5) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (79 (78 (73)))

PB(89): -18150

$DEV_0^5 \sin(\cos x)$

```
(SUC
   2
   (CALCUL (DEVLIM (SIN (COS X)) X O 5) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (79 (80))

PB(90): -7767

$DEV_0^3 (1+x)^4$

```
(SUC
   2
   (CALCUL (DEVLIM (EXPT (PLUS 1 X) 4) X O 3) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (91)

PB(91): -12042

$DEV_0^2 (Q+\sin(x))^3$

```
(SUC
   2
   (CALCUL (DEVLIM (EXPT (PLUS Q (SIN X)) 3) X O 2) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (85)

PB(92): -26360

$DEV_0^6 (e^x + e^{-x})$

```
(SUC
   2
   (CALCUL (DEVLIM (PLUS (EXPT E X) (EXPT E (MINUS X))) X O 6) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (74 (78 (73)) (78 (73)))

PB(93): 13533

$DEV_0^4 \sin(x)\cos(2x)$

```
(SUC
   2
   (CALCUL (DEVLIM (TIMES (SIN X) (COS (TIMES 2 X))) X O 4) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (76 (79 (90 (73))))

PB(94): -61271 V -60550

$DEV_0^3 \frac{(e^{x^2}-1)-\log(1+x)}{x}$

```
(SUC
   2
   (CALCUL
      (DEVLIM
         (QUOTIENT
            (DIFFERENCE
               (DIFFERENCE (EXPT E (EXPT X 2)) 1)
               (LOG (PLUS 1 X)))
            X)
         X
         O
         3)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

0000000 (77 (75 (75 (78 (73)) (71)) (83)) (73)) V

0000000 (90 (75 (75 (78 (73)) (71)) (83)))

```
PB(88):
(SUC
  2
   (CALCUL (DEVLIM (SIN (EXPT E X)) X O 5) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

−19906 $\underset{0}{\overset{5}{DEV}} \sin(e^{x})$ (79 (78 (73)))

```
PB(89):
(SUC
  2
   (CALCUL (DEVLIM (SIN (COS X)) X O 5) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

−12150 $\underset{0}{\overset{5}{DEV}} \sin(\cos x)$ (79 (80))

```
PB(90):
(SUC
  2
   (CALCUL (DEVLIM (EXPT (PLUS 1 X) 4) X O 3) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

−7767 $\underset{0}{\overset{3}{DEV}} (1+x)^{4}$ (91)

```
PB(91):
(SUC
  2
   (CALCUL (DEVLIM (EXPT (PLUS Q (SIN X)) 3) X O 2) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

−12042 $\underset{0}{\overset{2}{DEV}} (Q+\sin(x))^{3}$ (85)

```
PB(92):
(SUC
  2
   (CALCUL (DEVLIM (PLUS (EXPT E X) (EXPT E (MINUS X))) X O 6) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

−26360 $\underset{0}{\overset{6}{DEV}} (e^{x}+e^{-x})$ (74 (78 (73)) (78 (73)))

```
PB(93):
(SUC
  2
   (CALCUL (DEVLIM (TIMES (SIN X) (COS (TIMES 2 X))) X O 4) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

13533 $\underset{0}{\overset{4}{DEV}} \sin(x)\cos(2x)$ (76 (79 (70 (73))))

```
PB(94):
(SUC
  2
   (CALCUL
      (DEVLIM
         (QUOTIENT
            (DIFFERENCE
               (DIFFERENCE (EXPT E (EXPT X 2)) 1)
               (LOG (PLUS 1 X)))
            X)
         X
         O
         3)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

−61271 V −60 550 $\underset{0}{\overset{3}{DEV}} \frac{(e^{x^2}-1)-\mathrm{Log}(1+x)}{x}$ (77(75(75(78(73))(71))(83))(73)) V (90 (75(75 (78(73))(71)) (83)))

PB(95): 18932

$DEV_0^2\ \sqrt{1+x}\ \sin(x)$ (76 (87)(79))

```
(SUC
   2
   (CALCUL (DEVLIM (TIMES (RAC2 (PLUS 1 X)) (SIN X)) X 0 3) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(96): 3560

$DEV_0^7\ (\cos(x))^n$ (85) formule de TAYLOR.

```
(SUC
   2
   (CALCUL (DEVLIM (EXPT (COS X) N) X 0 7) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(97): 20402

$DEV_0^4\ \frac{x}{e^x-1}$ (77 (73)(75 (78 (73))(71)))

```
(SUC
   2
   (CALCUL (DEVLIM (QUOTIENT X (DIFFERENCE (EXPT E X) 1)) X 0 4) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(98): 21022

$DEV_0^6\ \frac{\sin x}{e^x-1}$ (77 (79) (75 (78 (73)) (71)))

```
(SUC
   2
   (CALCUL
      (DEVLIM (QUOTIENT (SIN X) (DIFFERENCE (EXPT E X) 1)) X 0 6)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(99): 34M3

$DEV_0^4\ \log\frac{x}{e^x-1}$ (93 (75 (78 (73))(71)) (85))

```
(SUC
   2
   (CALCUL
      (DEVLIM (LOG (QUOTIENT X (DIFFERENCE (EXPT X) 1))) X 0 4)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(100): -44787

$DEV_0^4\ \sqrt{1+\sqrt{1+x^2}}-\sqrt{2}$ (75 (87 (87 (73))) (71))

```
(SUC
   2
   (CALCUL
      (DEVLIM
         (DIFFERENCE
            (RAC2 (PLUS 1 (RAC2 (PLUS 1 (EXPT X 2)))) )
            (RAC2 2))
         X
         0
         4)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

PB(101): -18475

$DEV_0^3\ \frac{1}{\sqrt{1+x}}$ (86 (87))

```
(SUC
   2
   (CALCUL (DEVLIM (QUOTIENT 1 (RAC2 (PLUS 1 X))) X 0 3) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

```
PB(102):   -31646  si écrit (PLUS (Log ...)(minus (Log ...)))
(SUC   V - 87559
   2
   (CALCUL
      (DEVLIM
         (DIFFERENCE (LOG (PLUS 2 X)) (LOG (PLUS 2 (TIMES 2 X))))
         X
         O
         2)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

$\overset{2}{\underset{0}{\mathrm{DEV}}} \left(\log(2+x) - \log(2+2x) \right)$

(75 (73) (83 (73)))
(94 (77 (73) (73)) (92 (85 (83 (85)))))

```
PB(103):  -16833
(SUC
   2
   (CALCUL (DEVLIM (TG (EXPT E X)) X O 3) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

$\overset{3}{\underset{0}{\mathrm{DEV}}} \; tg(e^x)$

(81 (78 (73)))

```
PB(104):   -23437
(SUC
   2
   (CALCUL
      (DEVLIM (QUOTIENT (EXPT E X) (DIFFERENCE (SIN X) 1)) X O 6)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

$\overset{6}{\underset{0}{\mathrm{DEV}}} \; \frac{e^x}{\sin(x) - 1}$

(85) TAYLOR.

```
PB(105):  -17640
(SUC
   2
   (CALCUL (DEVLIM (QUOTIENT (SIN (EXPT X 2)) (EXPT X 2)) X O 5) RR)
   (!%ECRIRE (!%ALGEB RR)))
```

$\overset{5}{\underset{0}{\mathrm{DEV}}} \; \frac{\sin x^2}{x^2}$

(90 (79 (73)))

```
PB(106):  -29524
(SUC
   2
   (CALCUL
      (DEVLIM
         (QUOTIENT (DIFFERENCE (COS (EXPT X 2)) 1) (EXPT X 2))
         X
         O
         4)
      RR)
   (!%ECRIRE (!%ALGEB RR)))
```

$\overset{4}{\underset{0}{\mathrm{DEV}}} \; \frac{\cos(x^2) - 1}{x^2}$

(90 (75 (80) (71)))

```
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
0000000
```

ANNEXE G

EXEMPLES DE SOLUTIONS

$\int tg^3(x)\,dx$

```
solut pb(173);
    9: " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (EXPT (TG X) 3) X) RR)
    NO DU PLAN A UTILISER ?
41
    "J'ENVISAGE LE PLAN " INT41
    "??????????????????? VOYONS SI " "CHG VARIABLE FONCTION TRIGONOMETRIQ
UE"

    " "
    ".... PROBLEME A RESOUDRE " (!%CREER (R1 R2 R3 R4 R5 R6))
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R6) (!%NOUVAR))
    *** SERV0103 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (EXPT (TG X) 3))
    *** SERV0100 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (CHOIX 3 (SI (IMPAIRE (EXPT (TG X) 3) X)
 (SU
    C 2 (!%ATTRIB (!%VAL R5) (COS X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT
(EXP
    T (TG X) 3) X (!%VAL R6) (COS X))))) (SUC 3 (!%ATTRIB (!%VAL R2) (!%
SIMP
    LIF (!%REMPL X (DIFFERENCE PI X) (!%VVAL R1)))) (!%MONTRER R2) (SI (
(!%S
    IMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VA
L R5
    ) (SIN X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (EXPT (TG X) 3) X (!%VA
L R6
    ) (SIN X)))))) (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL X (MI
NUS
    X) (!%VVAL R1)))) (!%MONTRER R2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (
!%VV
    AL R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (COS X)) (!%ATTRIB (!%VA
L R3
    ) (!%CHGVARINT (EXPT (TG X) 3) X (!%VAL R6) (COS X)))))) (SUC 3 (!%A
TTRI
    B (!%VAL R2) (!%SIMPLIF (!%REMPL X (PLUS PI X) (!%VVAL R1)))) (!%MON
TRER
     R2) (SI ((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (!%ATTRIB (!%VAL R5) (
TG X
    )) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (EXPT (TG X) 3) X (!%VAL R6) (T
G X)
    ))))))
    " "
    ".... PROBLEME A RESOUDRE " (SI (IMPAIRE (EXPT (TG X) 3) X) (SUC 2 (
!%AT
    TRIB (!%VAL R5) (COS X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (EXPT (TG
 X)
    3) X (!%VAL R6) (COS X)))))

    "CONDITION PROBLEME ===> " (IMPAIRE (EXPT (TG X) 3) X)
    NO DU PLAN A UTILISER ?
18
    "J'ENVISAGE LE PLAN " IMPAIRE18
    "??????????????????? VOYONS SI " "A**F AVEC A IMPAIRE ET F IMPAIR"
    NO DU PLAN A UTILISER ?
14
    "J'ENVISAGE LE PLAN " IMPAIRE14
    "??????????????????? VOYONS SI " " PROPRIETE EVIDENTE "
```

```
    J'AI PU APPLIQUER 6

    RESULTAT CONNU
    "LE PLAN " 14 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" " PROPRIETE EVIDENTE "
    "LE PLAN " 18 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "A**F AVEC A IMPAIRE ET F IMPAIR"
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R5) (COS X))
    *** SERV0104 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R3) (!%CHGVARINT (EXPT
(TG
    X) 3) X (!%VAL R6) (COS X)))
    *** SERV0102 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R3) (!%SIMPLIF (!%VVAL
R3))
    )
    " "
    ".... PROBLEME A RESOUDRE " (!%MONTRER R3)
    R3  SERV0102  (QUOTIENT (PLUS (EXPT SERV0105 2) (MINUS 1)) (EXPT SER
V010
    5 3))

             2              3
    (SERV0105  - 1)/SERV0105
    " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (!%VVAL R3) (!%VAL R6)) (!
%VAL
     R4))
    NO DU PLAN A UTILISER ?
34
    "J'ENVISAGE LE PLAN " PLAN34
    "???????????????? VOYONS SI " "FRACTION RATIONNELLE PAR RISCH "
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB SERV0103 (!%ALGEVAL (INT (QUOT
IENT
     (PLUS (EXPT SERV0105 2) (MINUS 1)) (EXPT SERV0105 3)) SERV0105)))
    *** SERV0103 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " T
    "LE PLAN " 34 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "FRACTION RATIONNELLE PAR RISCH "
    " "
    ".... PROBLEME A RESOUDRE " (!%MONTRER R4)
    R4  SERV0103  (QUOTIENT (PLUS (TIMES 2 (LOG SERV0105) (EXPT SERV0105
 2))
     1) (TIMES 2 (EXPT SERV0105 2)))

                                2                  2
    (2*LOG(SERV0105)*SERV0105  + 1)/(2*SERV0105 )
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%REMPL (!%VAL R6) (!%VVAL
 R5)
     (!%VVAL R4)))
    " "
    ".... PROBLEME A RESOUDRE " (!%LIBERER (R1 R2 R3 R4 R5 R6))
    "LE PLAN " 41 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "CHG VARIABLE FONCTION TRIGONOMETRIQUE"
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

             2                                2
    (2*COS(X) *LOG(COS(X)) + 1)/(2*COS(X) )
```

$\int 3x^2 \operatorname{Log} x \, dx.$

```
solut pb(152);
    6: " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TIMES (TIMES 3 (EXPT X 2)
) (L
    OG X)) X) RR)
    NO DU PLAN A UTILISER ?
33
    "J'ENVISAGE LE PLAN " INT33
    "???????????????????? VOYONS SI " "INTEGRATION DU PRODUIT PAR PARTIE"
    " "
    ".... PROBLEME A RESOUDRE " (SI (ET (MONOMEP (TIMES 3 (EXPT X 2)) (X
)) (
    VUS X (LOG X))) (CHOIX 2 (SI (OU ((!%OP (LOG X)) EST EXPT) (TRIGO (L
OG X
    ))) (CALCUL (INTEGRERPARPARTIE (TIMES (TIMES 3 (EXPT X 2)) (LOG X))
X) R
    R)) (SI ((!%OP (LOG X)) EST LOG) (CALCUL (INTEGRERPARPARTIE (TIMES (
LOG
    X) (TIMES 3 (EXPT X 2))) X) RR))))
    " "
    ".... PROBLEME A RESOUDRE " (SI (OU ((!%OP (LOG X)) EST EXPT) (TRIGO
 (LO
    G X))) (CALCUL (INTEGRERPARPARTIE (TIMES (TIMES 3 (EXPT X 2)) (LOG X
)) X
    ) RR))
    " "
    ".... PROBLEME A RESOUDRE " (SI ((!%OP (LOG X)) EST LOG) (CALCUL (IN
TEGR
    ERPARPARTIE (TIMES (LOG X) (TIMES 3 (EXPT X 2))) X) RR))
    NO DU PLAN A UTILISER ?
60
    "J'ENVISAGE LE PLAN " INT60
    "???????????????????? VOYONS SI " "INTEGRATION PAR PARTIES"
    " "
    ".... PROBLEME A RESOUDRE " (!%CREER (VIPP))
    " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (TIMES 3 (EXPT X 2)) X) (!
%VAL
     VIPP))
    NO DU PLAN A UTILISER ?
29
    "J'ENVISAGE LE PLAN " INT29
    "???????????????????? VOYONS SI " "INTEGRATION D'UN MONOME "
    " "
    ".... PROBLEME A RESOUDRE " (!%CREER R1)
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%MONOMEP (TIMES 3
 (EX
    PT X 2)) (X)))
    *** SERV0094 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB SERV0093 (!%SIMPLIF (QUOTIENT
(TIM
    ES (!%TETE (!%VVAL R1)) (EXPT X (PLUS (!%TQ (!%VVAL R1)) 1))) (PLUS
(!%A
    RG1 (!%VVAL R1)) 1))))
    *** SERV0093 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%LIBERER R1)
    "LE PLAN " 29 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "INTEGRATION D'UN MONOME "
```

```
    " "
    ".... PROBLEME A RESOUDRE " (!%MONTRER VIPP)
    VIPP  SERV0093  (EXPT X 3)

     3
    X
    " "
    ".... PROBLEME A RESOUDRE " (!%CREER (DUIPP))
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL DUIPP) (!%DERIV (LOG X)
 X))

    *** SERV0095 DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%CREER R1)
    " "
    ".... PROBLEME A RESOUDRE " (CALCUL (PRIM (!%SIMPLIF (TIMES (!%VVAL
VIPP
    ) (!%VVAL DUIPP))) X) (!%VAL R1))
    NO DU PLAN A UTILISER ?
20
    "J'ENVISAGE LE PLAN " INT20
    "???????????????????? VOYONS SI " "RESULTAT USUEL "
    J'AI PU APPLIQUER 44
    *** SERV0096 DECLARED FLUID
    "LE PLAN " 20 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "RESULTAT USUEL "
    " "
    ".... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%SIMPLIF (DIFFERENCE (TIM
ES (
    LOG X) (!%VVAL VIPP)) (!%VVAL R1))))
    *** RR DECLARED FLUID
    " "
    ".... PROBLEME A RESOUDRE " (!%LIBERER (DUIPP VIPP))
    " "
    ".... PROBLEME A RESOUDRE " (!%LIBERER R1)
    "LE PLAN " 60 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "INTEGRATION PAR PARTIES"
    "LE PLAN " 33 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "INTEGRATION DU PRODUIT PAR PARTIE"
    " "
    ".... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

     3
    (X *(3*LOG(X) - 1))/3

    T

showtime;
    7:
    TIME: 4709 MS

    NIL
```

solut pb(170);

MSGLEVEL vaut 6 -

$\int sh(x).ch(x)\,dx$

```
    ".... PROBLEME A RESOUDRE " (CALCUL (INT (TIMES (SH X) (CH X)) X) RR
)

    NO DU PLAN A UTILISER ?
999

    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
    "TABTRI(" 1 ") : " (32666 38)
    "TABTRI(" 2 ") : " (16333 31)
    "TABTRI(" 3 ") : " (10894 46)
    "TABTRI(" 4 ") : " (9166 27)
    "TABTRI(" 5 ") : " (8166 26)
    "TABTRI(" 6 ") : " (6875 20)
    "TABTRI(" 7 ") : " (4583 29)
    "J'ENVISAGE LE PLAN " INT38
```

Mult' intégration par parties
Chgt variable trig. Hyperb.
intégrer une constante
$\int A.f(x)\,dx$
résultat connu.
monome ?

} Tri des plans possibles pour effectuer le calcul.

```
    ".... PROBLEME A RESOUDRE " (SI ((!%DERIV (SH X) X) EST (CH X)) (!%A
TTRIB RR (QUOTIENT (EXPT (SH X) 2) 2)))
    "LE PLAN " 38 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "INTEGRATION DE U*U'  "


    ".... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

         2
    SH(X) /2

    T
```

$\int \frac{sh^3 x}{ch^2 x}\, dx$

```
msglevel:=6$solut pb(171)$


    ".... PROBLEME A RESOUDRE " (CALCUL (INT (QUOTIENT (EXPT (SH X) 3) (
EXPT (CH X) 2)) X) RR)

    NO DU PLAN A UTILISER ?
999

    "TABTRI(" 1 ") : " (12375 44)
    "TABTRI(" 2 ") : " (9900 50)
    "TABTRI(" 3 ") : " (6500 34)
    "TABTRI(" 4 ") : " (6187 36)
    "TABTRI(" 5 ") : " (4583 29)
    "TABTRI(" 6 ") : " (3842 46)
    "J'ENVISAGE LE PLAN " INT44
    "J'ENVISAGE LE PLAN " INT50
    "J'ENVISAGE LE PLAN " PLAN34
    "###### EXPLICATION :" "JE N'AI PAS SU INTEGRER LA FRACTION
                          "
    "J'ENVISAGE LELAN " INT36

    NO DU PLAN A UTILISER ?
0
    "J'ENVISAGE LE PLAN " PLANNIL

    ***** ECHEC SUR (CALCUL (INT (TIMES (EXPT (SH X) 3) (QUOTIENT 1 (EXP
T (CH X) 2))) X) RR)

    "J'AI ESSAYE TOUS LES PLANS " NIL
    "J'ENVISAGE LE PLAN " INT29


    ".... PROBLEME A RESOUDRE " (!%CREER R1)
```

$\int A/B$ avec B casté

$\int u^k / f(x)$

fraction rationnelle.

$\int u/v \rightarrow \int u * \frac{1}{v}$

monome

chgt variable fonct. Hyperboliques.

```
    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%MONOMEP (QUOTIEN
T (EXPT (SH X) 3) (EXPT (CH X) 2)) (X)))
    *** SERV0576 DECLARED FLUID
    "******ECHEC SUR " (!%ATTRIB (!%VAL R1) (!%MONOMEP (QUOTIENT (EXPT (
SH X) 3) (EXPT (CH X) 2)) (X)))
    "J'ENVISAGE LE PLAN " INT46   b


    ".... PROBLEME A RESOUDRE " (!%CREER (R1 R2 R3 R4 R5 R6))

    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R6) (!%NOUVAR))
    *** SERV0617 DECLARED FLUID


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL r1) (QUOTIENT (EXPT (SH
 X) 3) (EXPT (CH X) 2)))
    *** SERV0612 DECLARED FLUID


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%REMPL (TH X) (QU
OTIENT (SH X) (CH X)) (!%VVAL R1)))


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%REMPL (CH X) (CO
S X) (!%VVAL R1)))


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%SIMPLIF (!%REMPL
 (SH X) (SIN X) (!%VVAL R1))))


    ".... PROBLEME A RESOUDRE " (!%MONTR R1)

    R1  SERV0612  (QUOTIENT (EXPT (SIN X) 3) (EXPT (COS X) 2))

          3       2
    SIN(X) /COS(X)


    ".... PROBLEME A RESOUDRE " (CHOIX 3 (SUC 3 (!%ATTRIB (!%VAL R2) (!%
SIMPLIF (!%REMPL X (DIFFERENCE PI X) (!%VVAL R1
    )))) (!%MONTRER R2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2)))
EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SH X)) (!%A
    TTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXPT (CH X)
 2)) X (!%VAL R6) (SH X)))))) (SUC 3 (!%ATTRIB
    (!%VAL R2) (!%SIMPLIF (!%REMPL X (MINUS X) (!%VVAL R1)))) (!%MONTRER
 R2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL
    R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (CH X)) (!%ATTRIB (!%VAL R3
) (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXPT
    (CH X) 2)) X (!%VAL R6) (CH X)))))) (SUC 3 (!%ATTRIB (!%VAL R2) (!%S
IMPLIF (!%REMPL X (PLUS PI X) (!%VVAL R1)))) (!
    %MONTRER R2) (SI ((!%VVAL R2) EST (!%VVAL R1)) (SUC 2 (!%ATTRIB (!%V
AL R5) (TH X)) (!%ATTRIB (!%VAL R3) (!%CHGVARIN
    T (QUOTIENT (EXPT (SH X) 3) (EXPT (CH X) 2)) X (!%VAL R6) (TH X)))))
))


    ".... PROBLEME A RESOUDRE " (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (
!%REMPL X (DIFFERENCE PI X) (!%VVAL R1)))) (!%M
    ONTRER R2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (S
UC 2 (!%ATTRIB (!%VAL R5) (SH X)) (!%ATTRIB (!%
    VAL R3) (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXPT (CH X) 2)) X (!
%VAL R6) (SH X))))))
```

```
   ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL
X (DIFFERENCE PI X) (!%VVAL R1))))
   *** SERV0613 DECLARED FLUID

   ".... PROBLEME A RESOUDRE " (!%MONTRER R2)

   R2  SERV0613  (QUOTIENT (EXPT (SIN X) 3) (EXPT (COS X) 2))

         3       2
   SIN(X) /COS(X)

   ".... PROBLEME A RESOUDRE " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVA
L R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SH
   X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXP
T (CH X) 2)) X (!%VAL R6) (SH X)))))
   "******ECHEC SUR " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) E
ST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (SH X)) (!%AT
   TRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXPT (CH X)
2)) X (!%VAL R6) (SH X)))))

   ".... PROBLEME A RESOUDRE " (SUC 3 (!%ATTRIB (!%VAL R2) (!%SIMPLIF (
!%REMPL X (MINUS X) (!%VVAL R1)))) (!%MONTRER R
   2) (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVAL R2))) EST 0) (SUC 2 (!%
ATTRIB (!%VAL R5) (CH X)) (!%ATTRIB (!%VAL R3)
   (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXPT (CH X) 2)) X (!%VAL R6)
 (CH X))))))

   ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R2) (!%SIMPLIF (!%REMPL
 X (MINUS X) (!%VVAL R1))))

   ".... PROBLEME A RESOUDRE " (!%MONTRER R2)

   R2  SERV0613  (QUOTIENT (MINUS (EXPT (SIN X) 3)) (EXPT (COS X) 2))

            3       2
   ( - SIN(X) )/COS(X)

   ".... PROBLEME A RESOUDRE " (SI ((!%SIMPLIF (PLUS (!%VVAL R1) (!%VVA
L R2))) EST 0) (SUC 2 (!%ATTRIB (!%VAL R5) (CH
   X)) (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTIENT (EXPT (SH X) 3) (EXP
T (CH X) 2)) X (!%VAL R6) (CH X)))))

   ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R5) (CH X))
   *** SERV0616 DECLARED FLUID

   ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R3) (!%CHGVARINT (QUOTI
ENT (EXPT (SH X) 3) (EXPT (CH X) 2)) X (!%VAL R
   6) (CH X)))
   *** SERV0614 DECLARED FLUID

   ".... PROBLEME A RESOUDRE " (!%MONTRER R3)

   R3  SERV0614  (QUOTIENT (DIFFERENCE (EXPT SERV0617 2) 1) (EXPT SERVO
617 2))
```

```
          2              2
    (SERV0617  - 1)/SERV0617


    ".... PROBLEME A RESOUDRE " (CALCUL (INT (!%VVAL R3) (!%VAL R6)) (!%
VAL R4))

    NO DU PLAN A UTILISER ?
999

    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
    "TABTRI(" 1 ") : " (12375 44)
    "TABTRI(" 2 ") : " (9900 50)
    "TABTRI(" 3 ") : " (7928 48)
    "TABTRI(" 4 ") : " (7923 47)
    "TABTRI(" 5 ") : " (7400 40)
    "TABTRI(" 6 ") : " (7375 21)
    "TABTRI(" 7 ") : " (6500 34)
    "TABTRI(" 8 ") : " (6187 36)
    "TABTRI(" 9 ") : " (5583 49)
    "TABTiI(" 10 ") : " (4583 29)
    "J'ENVISAGE LE PLAN " INT44
    "J'ENVISAGE LE PLAN " INT50
    "J'ENVISAGE LE PLAN " INT48

    "J'ENVISAGE LE PLAN " INT47


    ".... PROBLEME A RESOUDRE " (!%CREER (R1 R2))


    ".... PROBLEME A RESOUDRE " (CALCUL (INT (!%SIMPLIF (QUOTIENT (EXPT
SERV0617 2) (EXPT SERV0617 2))) SERV0617) (!%VA
    L R1))

    NO DU PLAN A UTILISER ?
999

    "TABTRI(" 1 ") : " (9166 27)
    "TABTRI(" 2 ") : " (6875 20)
    "TABTRI(" 3 ") : " (4583 29)
    "J'ENVISAGE LE PLAN " INT27
    *** SERV1298 DECLARED FLUID
    "LE PLAN " 27 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "FONCTION CSTE PAR RAPPORT A X "


    ".... PROBLEME A RESOUDRE " (CALCUL (INT (!%SIMPLIF (QUOTIENT 1 (EXP
T SERV0617 2))) SERV0617) (!%VAL R2))

    NO DU PLAN A UTILISER ?
999


    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
    "TABTRI(" 1 ") : " (12375 44)
    "TABTRI(" 2 ") : " (9900 50)
    "TABTRI(" 3 ") : " (9166 27)
    "TABTRI(" 4 ") : " (9000 28)
    "TABTRI(" 5 ") : " (6875 20)
    "TABTRI(" 6 ") : " (6500 34)
    "TABTRI(" 7 ") : " (6187 36)
    "TABTRI("08 ") : " (4583 29)
    "J'ENVISAGE LE PLAN " INT44
    "J'ENVISAGE LE PLAN " INT50
```

```
    ".... PROBLEME A RESOUDRE " (!%CREER R1)

    ".... PROBLEME A RESOUDRE " (CALCUL (INT (QUOTIENT 1 (EXPT SERV0617
2)) SERV0617) (!%VAL R1))

    NO DU PLAN A UTILISER ?
999

    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
    "TABTRI(" 1 ") : " (12375 44)
    "TABTRI(" 2 ") : " (9900 50)
    "TABTRI(" 3 ") : " (9166 27)
    "TABTRI(" 4 ") : " (9000 28)
    "TABTRI(" 5 ") : " (6875 20)
    "TABTRI(" 6 ") : " (6500 34)
    "TABTRI(" 7 ") : " (6187 36)
    "TABTRI(" 8 ") : " (4583 29)
    "J'ENVISAGE LE PLAN " INT44
    "J'ENVISAGE LE PLAN " INT50

    ".... PROBLEME A RESOUDRE " (!%CREER R1)

    ".... PROBLEME A RESOUDRE " (CALCUL (INT (QUOTIENT 1 (EXPT SERV0617
2)) SERV0617) (!%VAL R1))

    NO DU PLAN A UTILISER ?
20
    "J'ENVISAGE LE PLAN " INT20

    J'AI PU APPLIQUER 36
    *** SERV2096 DECLARED FLUID

    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
    "LE PLAN " 20 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "RESULTAT USUEL "

    ".... PROBLEME A RESOUDRE " (!%ATTRIB SERV1745 (!%SIMPLIF (TIMES 1 (
!%VVAL R1))))
    *** SERV1745 DECLARED FLUID

    ".... PROBLEME A RESOUDRE " (!%LIBERER R1)
    "LE PLAN " 50 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "INT(C/F(X),X) EST C*INT(1/F(X),X"

    ".... PROBLEME A RESOUDRE " (!%ATTRIB SERV1299 (!%SIMPLIF (TIMES 1 (
!%VVAL R1))))
    *** SERV1299 DECLARED FLUID

    ".... PROBLEME A RESOUDRE " (!%LIBERER R1)
    "LE PLAN " 50 " A PERMIS DE CONCLURE  "
    "###### JUSTIFICATION :" "INT(C/F(X),X) EST C*INT(1/F(X),X"

    ".... PROBLEME A RESOUDRE " (!%ATTRIB SERV0615 (DIFFERENCE (!%VVAL R
1) (!%VVAL R2)))
    *** SERV0615 DECLARED FLUID
```

```
"..... PROBLEME A RESOUDRE " (!%LIBERER (R1 R2))
"LE PLAN " 47 " A PERMIS DE CONCLURE  "


"..... PROBLEME A RESOUDRE " (!%MONTRER R4)

R4  SERV0615  (DIFFERENCE SERV0617 (QUOTIENT (MINUS 1) SERV0617))

          2
(SERV0617  + 1)/SERV0617


"..... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%REMPL (!%VAL R6) (!%VVAL
R5) (!%VVAL R4)))
*** RR DECLARED FLUID


"..... PROBLEME A RESOUDRE " (!%LIBERER (R1 R2 R3 R4 R5 R6))
"LE PLAN " 46 " A PERMIS DE CONCLURE  "
"##### JUSTIFICATION :" "CHG VARIABLE FONCTION HYPERBOLIQUE"


"..... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

      2
(CH(X)  + 1)/CH(X)
```

```
solut pb(158);
```

$$\int \frac{x^2}{(\cos x + x \sin x)^2} dx$$

```
"..... PROBLEME A RESOUDRE " (CALCUL (INT (QUOTIENT (EXPT X 2) (EXPT
(PLUS (COS X) (TIMES X (SIN X))) 2)) X) RR)

NO DU PLAN A UTILISER ?
999

DEBORDEMENT TABTRI
MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS

MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
"TABtRI(" 1 ") : " (19333 38) "TABTRI(" 2 ") : " (13166 33)
"TABTRI(" 3 ") : " (12375 44)
"TABTRI(" 4 ") : " (10142 37)
"TABTRI(" 5 ") : " (9900 50)
"TABTRI(" 6 ") : " (9666 31)
"TABTRI(" 7 ") : " (7333 39) integration par partie ∫ F dx/(A+B)^2 avec ∫ F * 1/(A+B)^2 dx
"TABTRI(" 8 ") : " (7000 51)
"TABTRI(" 9 ") : " (6500 34)
"TABTRI(" 10 ") : " (6187 36)
"TABTRI(" 11 ") : " (5650 25)
"TABTRI(" 12 ") : " (5500 23)
"TABTRI(" 13 ") : " (5263 24)
"TABTRI(" 14 ") : " (5090 22)
"TABTRI(" 15 ") : " (4833 26)
"TABTRI(" 16 ") : " (4600 43)
"TABTRI(" 17 ") : " (4600 42)
"TABTRI(" 18 ") : " (4583 29)
"TABTRI(" 19 ") : " (3933 45)
"TABTRI(" 20 ") : " (3523 35)
"J'ENVISAGE LE PLAN " INT38

"J'ENVISAGE LE PLAN " INT33
```

```
    "J'ENVISAGE LE PLAN " INT37

    "J'ENVISAGE LE PLAN " INT50
    "J'ENVISAGE LE PLAN " INT31

    "J'ENVISAGE LE PLAN " INT39


    ".... PROBLEME A RESOUDRE " (!%CREER (R1 R2 R3))


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R1) (!%SIMPLIF (QUOTIEN
T (MINUS (EXPT X 2)) (TIMES (EXPT (PLUS (COS X)
     (TIMES X (SIN X))) (DIFFERENCE 2 2)) (PLUS (!%DERIV (COS X) X) (!%D
ERIV (TIMES X (SIN X)) X))))))
    *** SERV3771 DECLARED FLUID


    ".... PROBLEME A RESOUDRE " (!%MONTRER R1)

    R1  SERV3771  (QUOTIENT (MINUS X) (COS X))

    ( - X)/COS(X)


    ".... PROBLEME A RESOUDRE " (!%ATTRIB (!%VAL R2) (QUOTIENT 1 (PLUS (
COS X) (TIMES X (SIN X)))))
    *** SERV3772 DECLARED FLUID


    ".@ (!%VOL (!%VVAL R1)) (!%VOL (QUO
TIENT (EXPT X 2) (EXPT (PLUS (COS X) (TIMES X (
    SIN X))) 2)))) (CALCUL (INT (!%SIMPLIF (QUOTIENT (!%DERIV (!%VVAL R1
) X) (PLUS (COS X) (TIMES X (SIN X))))) X) (!%V
    AL R3)))

    NO DU PLAN A UTILISER ?
999


    MESSAG PLEIN:PLUS DE PLACE POUR LES EXPLICATIONS
    "TABTRI(" 1 ") : " (12375 44)
    "TABTRI(" 2 ") : " (9900 50)
    "TABTRI(" 3 ") : " (6666 53)
    "TABTRI(" 4 ") : " (6500 34)
    "TABTRI(" 5 ") : " (6187 36)
    "TABTRI(" 6 ") : " (4600 42)
    "TABTRI(" 7 ") : " (4583 29)
    "J'ENVISAGE LE PLAN " INT44
    "J'ENVISAGE LE PLAN " INT50


    ".... PROBLEME A RESOUDRE " (!%CREER R1)


    ".... PROBLEME A RESOUDRE " (CALCUL (INT (QUOTIENT 1 (EXPT (COS X) 2
)) X) (!%VAL R1))

    NO DU PLAN A UTILISER ?
20
    "J'ENVISAGE LE PLAN " INT20

    J'AI PU APPLIQUER 26
    *** SERV4179 DECLARED FLUID
    "LE PLAN " 20 " A PERMIS DE CONCLURE  "
```

```
    "##### JUSTIFICATION :" "RESULT


    ".... PROBLEME A RESOUDRE " (!%ATTRIB SERV3773 (!%SIMPLIF (TIMES (MI
NUS 1) (!%VVAL R1))))
    *** SERV3773 DECLARED FLUID


    ".... PROBLEME A RESOUDRE " (!%LIBERER R1)
    "LE PLAN " 50 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "INT(C/F(X),X) EST C*INT(1/F(X),X"


    ".... PROBLEME A RESOUDRE " (!%MONTRER R3)

    R3  SERV3773  (MINUS (TG X))

     - TG(X)


    ".... PROBLEME A RESOUDRE " (!%ATTRIB RR (!%SIMPLIF (DIFFERENCE (TIM
ES (!%VVAL R1) (!%VVAL R2)) (!%VVAL R3))))


    ".... PROBLEME A RESOUDRE " (!%LIBERER (R1 R2 R3))
    "LE PLAN " 39 " A PERMIS DE CONCLURE  "
    "##### JUSTIFICATION :" "INT PAR PARTIE A PARTIR DE 1/(A+B)"


    ".... PROBLEME A RESOUDRE " (!%ECRIRE (!%ALGEB RR))

                                                    2
    (X*TG(X)*SIN(X)*COS(X) - X  + TG(X)*COS(X) )/(COS(X)*(X*SIN(X) + COS(
X)))

    T
```

BIBLIOGRAPHIE

NOTATIONS ET ABREVIATIONS

AI : Artificial Intelligence Journal
CACM : Communications of the Association for Computing Machinery (ACM)
JACM : Journal of the Association for Computing Machinery
IJCAI : International Joint Conference in Artificial Intelligence
SIGSAM : Special Interest Group on Symbolic and Algebraic Manipulations (ACM)
MI : Machine Intelligence (Edinburgh University Press)
SIGART : Special Interest Group in Artificial Intelligence (ACM)
ECAI : European Conference on Artificial Intelligence
CNRS/GR22 : Groupe de Recherche 22 du Centre National de la Recherche Scientifique - Couloir 45/46 - 3e étage - Université de Paris - Place Jussieu 75005 PARIS.
KBS : Knowledge Based Systems
PDIS : Pattern Directed Inference Systems - (New-York : Academic Press) by D.WATERMAN and F.HAYES-ROTH (eds).

REFERENCES

(ARNOLD 70) A.ARNOLD : Les mathématiques à la portée de l'ordinateur. Dunod - Paris (1970).

(BALLANTYNE 77) A.M.BALLANTYNE and W.W.BLEDSOE : Automatic proofs of theorems in analysis using non-standard techniques. JACM, 24(3), p. 353-374.

(BARON 81) M.BARON: Pilotage automatique du calcul formel en analyse combinatoire.Manipulation de formules sommatoires.Colloque Artificielle- TOULOUSE- Juillet 1981, Publication verte du CNRS-GR22-n°24, p.47-66.

(BARON 82a) M.BARON: un système pour exprimer et mettre en oeuvr des connaissances en manipulation formelle d' expressions - Thèse de 3ème cycle - PARIS VI - 20 décembre 1982.

(BARON 82b) M.BARON: un exemple d'utilisation de SEME -système pour exprimer et mettre en oeuvre des connaissances en manipulation formelle d'expression .Colloque Intelligence Artificielle -LE MANS-Septembre 1982, CNRS GR22 N°30 , p.149 186.

(BARSTOW 77) D.BARSTOW : A knowledge based organization for rules about programming. Proc. Workshop Pattern-Directed Inference Systems, SIGART Newsletter 63, (1977), p.18-22.

(BAXTER 73) L.D.BAXTER : An efficient unification algorithm, Rep.CS-73-23, University of Waterloo, Dept.of Analysis and Computer Science (1973).

(BOND 64) E.BOND, M.AUSSLENDER, S.GRISOFF, R.KENEY, M.MYSZEWSKI, J.E.SAMMET, R.G.TOBEY, S.ZILLES, FORMAC : An experimental FORmula MAnipulation Compiler, proceedings of the ACM national conference, Aug.1964.

(BELOVARI) G.BELOVARI,J.A.CAMPRELL : Generating contours of integration : an application of PROLOG in symbolic computing -SIGSAM.

(BERGMAN 73) M.BERGMAN : Résolution par la démonstration automatique de quelques problèmes en intégration symbolique sur ordinateur. Thèse 3ème cycle 1973 -Université de Marseille.

(BERGMAN 78) M.BERGMAN, KANOUI H. : SYCOPHANTE : système de calcu formel sur ordinateur. Rapport final ATP informatique. Group I.A. Université de Marseille-Luminy.

(BERLEKAMP 67) E.R.BERLEKAMP : Factoring polynomials over finite fields. Bell System Tech.J.46, (1967), p.1853-1859.

(BERLEKAMP 70) E.R.BERLEKAMP : Factoring polynomials over large finite fields. Math.Comp.24, (1970), p.713-735.

(BLEDSOE 71) W.W.BLEDSOE : Splitting and reduction heuristics in automatic theorem proving- AI-2 (1971) -p.55-77.

(BLEDSOE 72) W.W.BLEDSOE,R.S.BOYER, W.H.HENNEMAN : Computer proof of limit theorems AI 3 (1972) - p.27-60.

(BLEDSOE 74) W.W.BLEDSOE and P.BRUELL : A man-machine theorem-proving system. Artificial Int. 5, (1974), p.51-72.

(BLEDSOE 79a) W.W.BLEDSOE,P.BRUELL.R.SHOSTAK:A Prover for General Inequalities.IJCAI 79.TOKYO-August 1979.

(BLEDSOE 79b) W.W.BLEDSOE :A Resolution-based Prover for General Inequalities.University of Texas-Math Department.memo ATP 52-july1979.

(BLEDSOE 77) W.W.BLEDSOE : Non-resolution theorem proving. Artificial Intelligence, 9(1), p.1-36.

(BOLEY 83) H.BOLEY : Artificial Intelligence Languages and Machines. TSI, vol.2, n°3, (1983) p.145-166.

(BORNING 81) A. BORNING,A. BUNDY :Using matching in algebraic

equation solving,IJCAI 7 (1981),p.466-471.
(BOURGOIN 78) D.BOURGOIN : PARI : Un programme heuristique de résolution d'exercices d'arithmétique. Thèse 3ème cycle - PARIS VI (1978).
(BOURGOIN 79) D.BOURGOIN: Organisation des connaissances dans un programme résolvant des exercices d'arithmétique . Colloque Intelligence Artificielle-ROUEN-,Sept 79,publication verte du CNRS GR22-n°11,p.113-132.
(BOYER 79) R.S.BOYER and J.S.MOORE : A computational Logic. New-York : Academic Press.
(BRUNER 56) J.S.BRUNER, J.J.GOODNOW and G.A.AUSTIN : A study of thinking. Wiley, New-York, (1956).
(BUNDY 73) A.BUNDY : Doing arithmetic with diagrams, adv papers. IJCAI3 (1973) - p.383-387.
(BUNDY 75) A.BUNDY : Analysing mathematical proofs (or reading between the lines), DAI Research Paper n°2 (1975).
(BUNDY 79) A.BUNDY ET AL: Solving mechanics problems using meta-level inference .Proceeding IJCAI6,(1979),p.1017-1027.
(BUNDY 79a) A.BUNDY and B.WELHAM : Using meta-level descriptions for selective application of multiple rewrite rules in algebraic manipulation, DAI Research Paper n°121, University of Edinburg (1979).
(BUNDY 79b) A.BUNDY : A treatise on elementary equation solving, DAI working Paper n°51, University of Edinburgh (1979).
(BUNDY 81a) A.BUNDY, B.SILVER : Homogenization : preparing equations for change of unknown, IJCAI 81, Vancouver, (1981),p.551-553.
(BUNDY 81b) A.BUNDY,B.WELMAN: Using meta-level inference for selective application of multiple rewrite rules sets in algebraic manipulations ,ARTIFICIAL INTELLIGENCE,vol 16,n°2,may 1981,p.189-211.
(BUTHION 75) M.BUTHION : Un programme qui résout formellement des problèmes de construction géométrique. Thèse 3ème cycle -PARIS VI (1975).
(CAVINESS 70) B.CAVINESS : On canonical form and simplification, Journal of ACM, vol.17, n°2 (1970).
(CAVINESS 76) B.F.CAVINESS and R.J.FATEMAN : Simplification of radical expressions.Proc. SYMSAC 76, p.329-338.
(CHANG 70) C.L.CHANG : The unit proof and the input proof in theorem proving. J.ACM - Vol.17 - n°4 (oct.1970) - p.698-707.
(CHANG 79a) C.L.CHANG : Using rewriting rules for connection graphs to prove theorems. A.I.12(2).
(CHANG 79b) C.L.CHANG : Resolution plans in theorem proving. In IJCAI-6, p.143-148.
(CLAYBROOK 76) B.G.CLAYBROOK : A new approach to the symbolic factorisation of multivariate polynomials, AI 7, n°3, p.203-242.
(COLLINS 71a) G.E.COLLINS : The SAC-1 System : An introduction and survey, Proceedings SS-SAM, Los Angeles, March 71, p.144-157.
(COLLINS 71b) G.E.COLLINS : The calculation of multivariate polynomial resultants. JACM 18, (1971), p.515-532.
(COLLINS 73) G.E.COLLINS : Computer algebra of polynomial and rational functions, American Mathematical Monthly, vol.8, n°7, (1973), p.725-755.
(CORDIER 79) M. O. CORDIER: Commande d'un robot en langage naturel dans un domaine nécéssitant des connaissances pragmatiques : les recettes de cuisine, thèse de 3ème

cycle , LRI PARIS XI(1979).
(DALLARD 74) R.DALLARD : Présentation d'un programme de démonstration de théorèmes d'arithmétique. Thèse 3ème cycle -PARIS VI (1974).
(DAVENPORT 79a) J.H.DAVENPORT : On the integration of algebraic functions. Springer-Verlag Lecture Notes in Computer Science 102, Berlin-Heidelberg-New-York (1979).
(DAVENPORT 79b) J.H.DAVENPORT : Anatomy of an integral. SIGSAM Bulletin (November 1979).
(DAVENPORT 79c) J.H.DAVENPORT : Algorithms for the integration of algebraic functions. Proc.EUROSAM 79 p.415-425.
(DAVENPORT 80) J.H.DAVENPORT et R.D.JENKS : MODLISP - an introduction, Proc. LISP 80, The LISP Company, Stanford, California 1980.
(DAVENPORT 83b) J.H.DAVENPORT : VLSI et calcul formel -Rapport de recherche RR 357 - Université de Grenoble (1983).
(DAVENPORT 83a) J.H.DAVENPORT : PGCD et VLSI - Rapport de recherche RR 358 - Université de Grenoble (1983).
(DAVIS 60) DAVIS - PUTMAN : A computing procedure for quantification theory, J.ACM. (1960).
(DAVIS 76) R.DAVIS : Applications of Meta-Level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases. Doctoral dissertation, Stanford Artificial Intelligence Laboratory, Memo 283. (Reprinted in KBS.).
(DAVIS 77) R.DAVIS : Meta-level knowledge : overview and applications. In IJCAI-5, p.920-927.
(DAVIS 77a) R.DAVIS : Interactive transfer of expertise : Acquisition of new inference rules. Proc. 5th Int.Joint Conf.Artificial Intelligence, Cambridge, Massachusetts, (1977), p.321-328.
(DAVIS 77b) R.DAVIS and B.G.BUCHANAN : Meta level knowledge : overview and implications. Proc. 5th Int.Joint Conf.Artificial Intelligence, Cambridge, Massachusetts, (1977), p.920-927.
(DAVIS 77c) R.DAVIS, B.G.BUCHANAN and E.SHORTLIFFE : Production rules as a representation for a knowledge-based consultation program. J.Artificial Intelligence 8,(1977), p.15-45.
(DAVIS 77d) R.DAVIS : Knowledge acquisition in rule-based system : knowledge about representation as a basis for system construction and maintenance. In PDIS edited by D.A.Waterman and F.Hayes-Roth, (1978), p.99-134.
(DAVIS 80) R.DAVIS and D.LENAT : Knowledge-Based Systems in Artificial Intelligence. New-York : Mc Graw-Hill.
(DE BRUIJN 68) N.G.DE BRUIJN : The mathematical language AUTOMATH, its usage, and some of its extensions. Proceedings of symposium on automatic demonstration (Versailles 1968). Springer Lecture Notes in Mathematics n°125 (1970).
(DELHAYE 70) J.L.DELHAYE : DATAL : Un programme de démonstration automatique de théorèmes. Thèse 3ème cycle - Paris VI (1970)
(DERMOTT 72) D.MC DERMOTT and G.J.SUSSMAN, The CONNIVER Reference Manual, MIT AI Lab. Memo 259, May. (Rev., July 1973).
(DERMOTT 74) D.MC DERMOTT and G.SUSSMAN : The CONNIVER Reference Manual, Memo 259a, MIT AI Lab, Cambridge, Massachussets,(1974).
(DERMOTT 77) D.MC DERMOTT : A deductive model of control of a problem solver. Proc.Workshop Pattern-Directed

Inference Systems, SIGART Newsletter 63, (1977), p.2-7.

(DERMOTT 79) D. MC DERMOTT : Learning to use analogies. In IJCAI-6, p.568-576.

(DI SCALA 82) R.M.DI SCALA : A Pascal based computer algebra system for micro-computers. ACM-SIGSAM Bulletin 16(3), p.7, (1982) and Thèse 3e cycle - Université Grenoble (1982).

(DI SCALA 83) R.M.DI SCALA : Pascal as host language of computer algebra systems. Article présenté à SIGPLAN-ACM-Avril 83.

(DROUFFE 81) J.M.DROUFFE : AMP Language reference manual - version 6 - Doc. n°DP ht 81 - nnn-

(DUDA 78) R.O.DUDA et al. : Semantic network representations in rule-based inference systems. in PDIS, p.203-221.

(DURAND 75) A.DURAND : Un programme de démonstration d'exercices d'algèbre. Thèse 3ème Cycle - PARIS VI (1975).

(ERNST 69) G.W.ERNST and A.NEWELL : GPS : A case study in Generality and Problem Solving. New-York : Academic Press.

(FATEMAN 71) R.J.FATEMAN : The user-level semantic matching capability in MACSYMA, Proceedings SS-SAM, Los Angeles, March 71, p.311-323.

(FEIGENBAUM 63) E.FEIGENBAUM, R.FELDMAN : Computers and thought -Mc Graw Hill - (1963).

(FIKES 71) R.E.FIKES and N.J.NILSON : STRIPS : A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2(3/4), p.189-208.

(FIKES 72) R.E.FIKES, P.E.HART and N.J.NILSSON : Learning and executing generalized robot plans. Artificial Int.3,(1972), p.251.288.

(FINDLER 71) N.V.FINDLER and B.MELTZER (Eds.) : Artificial Intelligence and Heuristic Programming. New-York : American Elsevier.

(FRIEDLAND 81) P.E.FRIEDLAND : Acquisition of procedural knowledge from domain experts, IJCAI 81, Vancouver, (1981), p.856-861.

(GERLERNTER 59) M.GELERNTER : Realization of a geometry theorem-proving machine, Proc. Int. conf. information processing. PARIS - UNESCO (1959) - p.273-282.

(GILLET 79) M.GILLET: Un programme de démonstration automatique en théorie des groupes utilisant beaucoup de connaissances - Colloque Intelligence Artificielle-ROUEN-Sept.1979, Publication verte du CNRS GR22, n°11,p.133-142.

(GRANDBASTIEN 74) M.GRANDBASTIEN : Un programme qui résout formellement des équations trigonométriques par des procédés heuristiques. Thèse 3ème cycle - PARIS VI (1974).

(GREEN 69) C.GREEN : Application of theorem proving to problem solving. In IJCAI-1, p.219-239.

(GRIESMER 75) J.H.GRIESMER, R.D.JENKS, D.Y.Y.YUN : SCRATCHPAD User's Manual - IBM research publication. RA 70.June 1975.

(HADAMARD 59) J.HADAMARD : Essai sur la psychologie de l'invention dans le domaine mathématique. Librairie Albert Blanchard. Paris (1959).

(HARRINGTON 79) S.J.HARRINGTON : A new symbolic integration system in REDUCE. Computer Journal 22 (1979),p.127-131.

(HARRINGTON 79) S.J.HARRINGTON : A symbolic limit evaluation program in REDUCE. SIGSAM Bulletin 49 (Feb.1979), p.27-31.

(HEARN 71) A.C.HEARN : REDUCE 2 : A system and language for algebraic manipulation, Proceedings SS-SAM, Los Angeles, March 71, p.128-133.
(HEARN 73) A.C.HEARN : REDUCE-2 User's manual. Computing physics group, University of Utah, (1973).
(HEARN 76) A.C.HEARN : A new REDUCE model for algebraic simplification. Proc.SYMSAC 76, p.46-51.
(HEARN 79) A.C.HEARN : Non-modular computation of polynomial gcd using trial division. Proc.EUROSAM 79, p.227-239.
(HEIDRICK 76) C.L.HEDRICK : Learning production systems from examples. AI 7, (1976), p.21-49.
(HERBRAND 30) J.HERBRAND : Recherches sur la théorie de la démonstration - Thèse - Université de Paris (1930).
(HERZ 75) A.HERZ : Programme de démonstration de théorèmes formulables en logique des prédicats du 1er ordre avec égalité. Thèse 3ème cycle - PARIS VI (1975).
(HUET 76) G.HUET : Résolution d'équations dans des langages d'ordre 1,2,...Omega. Thèse d'Etat - Université de Paris VII - 1976.
(HUET 80) G.HUET, D.OPPEN : Equations and rewrite rules : a survey - In formal languages : perspectives and open problems. Ed. Book R. Academic Press - 1980.
(HOROWITZ 71) E.HOROWITZ : Modular arithmetic and finite field theory : a tutorial, Proceedings SS-SAM, Los Angeles, March 71, p.188-194.
(JACKSON 74) P.C.JACKSON : Introduction to Artificial Intelligence. New-York : Petrocelli Books.

(JENKS 79) R.D.JENKS, MODLISP : Proc. EUROSAM 79, p.466-480.
(KAHN 77) G.KAHN, D.B.MACQUEEN : Coroutines and networks of parallel processes. Proc. IFIP Congress 77 - North-Holland, Amsterdam - p.993-998 (1977).
(KAHN 81a) K.M.KAHN : UNIFORM : a language based upon unification,IJCAI 7 (1981),p.933-939.
(KAHN 81b) K.KAHN : UNIFORM : A language based upon unification which unifies (much of) LISP, PROLOG, and ACT 1 - UPMAIL - TR 2 -(1981) - Uppsala University.
(KAHN 82) K.KAHN : The implementation of UNIFORM, A knowledge representation programming language based upon equivalence of descriptions - UPMAIL - TR 9 (1982) -Uppsala University.
(KANOUI 73) H.KANOUI : Application de la démonstration automatique aux manipulations algébriques et à l'intégration formelle sur ordinateur. Thèse 3ème cycle 1973 - Université de Marseille.
(KANOUI 76) H.KANOUI : Some aspects of symbolic integration via predicate logic programming, ACM-SIGSAM - Bulletin - nov.1976.
(KLING 71) R.E.KLING : A paradigm for reasoning by analogy. A.I.,vol.2,n°2, p.147-178.
(KORPELA 77) J.KORPELA : Automatic generation of algorithms - Research Report n°6, Helsinki University of technology, Finland (1977).
(KOWALSKI 70) R.KOWALSKI : Search strategies for theorem-proving. In MI5, p.181-201.
(KOWALSKI 72) R.KOWALSKI : AND/OR Graphs, theorem-proving graphs, and bidirectional search. In MI7, p.94-167.

(KOWALSKI 79) R.KOWALSKI : Logic for Problem Solving. New-York : North-Holland.

(KUNG 81) H.T.KUNG : Use of VLSI in algebraic computation : some suggestions, Proceedings of the 1981 ACM Symposium on Symbolic and Algebraic Computation, SYMSAC 81, p.218-222.

(LAFON 80) J.C.LAFON : Conception d'algorithmes efficaces pour les calculs algébriques. Actes du congrès AFCET 1980 - Informatique. p.427-436.

(LANAM 81) D.H.LANAM : An algebraic front-end for the production and use of numeric programs, Proceedings of the 1981 ACM Symposium on Symbolic and Algebraic Computation, SYMSAC 81, p.223-227.

(LAPLACE 73) A.LAPLACE : FORmac DEsk CALculator :un outil de mise au point et d'aide au calcul formel sur ordinateur. thèse d'Etat. USMG. Grenoble, fevrier 1973.

(LAURENT 72) J.P.LAURENT : Un programme qui calcule des limites en levant les indéterminations par des procédés heuristiques. Thèse 3ème cycle -PARIS VI -(1972).

(LAURENT 73) J.P.LAURENT : A program that computes limits using heuristics to evaluate the indeterminate forms. AI 4 (1973), p.69-94.

(LAURIERE 76) J.L LAURIERE :Un langage et et un programme pour énoncer et résoudre des problèmes combinatoires. Thèse Etat -PARIS VI (1976).

(LAURIERE 78) J.L.LAURIERE : A language and a program for stating and solving combinatorial problems. AI 10 (1978). p.29-127.

(LAURIERE 79) J.L LAURIERE: Représentation et utilisation des connaissances - Colloque Intelligence Artificielle du CNRS GR22, n°11,p.3-112.

(LAURIERE 82) J.L.LAURIERE : Representation et utilisation des connaissances. TSI, vol.1, n°1 et 2, janvier-février 1982 p.25-42 et mars-avril 1982 p.109-133.

(LAZARD 81) D.LAZARD :Factorisation des polynomes. CALSYF 2 , Journées calcul formel - STRASBOURG 1979,POITIERS 1981, p.173-184.

(LENAT 75) D.LENAT : BEINGs : Knowledge as interacting experts. Proc.4th Int.Joint Conf.Artificial Intelligence, Tbilisi, USSR,(1975), p.126-133.

(LENAT 76) D.B.LENAT : AM : An artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search. Rep.. STAN-CS-76-570, Stanford University, Computer Science Dept. July. (Reprinted in KBS).

(LENAT 77a) D.LENAT : Automated theory formation in mathematics. Proc.5th Int.Joint Conf.Artificial Intelligence, Cambridge, Massachusetts, (1977), p.833-842.

(LENAT 77b) D.LENAT : The ubiquity of discovery : 1977 computers and thought lecture. Proc.5th Int.Joint Conf.Artificial Intelligence, Cambridge, Massachusetts, (1977), p.1093-1105.

(LENAT 77c) D.LENAT and J.Mc DERMOTT : Less than general production system architectures. Proc.5th Int.Joint Conf.Artificial Intelligence, Cambridge, Massachusetts, (1977), p.928-932.

(LENAT 77d) D.LENAT and G.HARRIS : Designing a rule system that searches for scientific discoveries. In PDIS edited by D.A.Waterman and F.Hayes-Roth, (1978), p.25-52.

(LENAT 77e) D.LENAT : The ubiquity of discovery - AI 9, vol.3, (1977), p.257-286.

(MANNA 71) Z.MANNA, R.WALDINGER : Toward automatic program synthesis. C.ACM - vol.14, n°3 - March 71 - p.151-165.

(MARTIN 71) W.A.MARTIN and R.J.FATEMAN : The MACSYMA system.Proc. ACM 2d Symposium on Symbolic and Algebraic Manipulation,Los Angeles, CA, p.23-25.
(MARTIN 71a) W.A.MARTIN, R.J.FATEMAN : The MACSYMA System, Proceedings SS-SAM, Los Angeles, March 71, p.59-75.
(MARTIN 71b) W.A.MARTIN : Computer INPUT/OUTPUT of mathematical expression, Proceedings SS-SAM, Los Angeles, March 71, p.78-99.
(MELTZER 69) B.MELTZER and D.MICHIE (Eds) : Machine Intelligence 4. Edinburgh : Edinburgh University Press.
(MELTZER 70) B.MELTZER and D.MICHIE (Eds) : Machine Intelligence 5. Edinburgh : Edinburgh University Press.
(MELTZER 71) B.MELTZER and D.MICHIE (Eds) : Machine Intelligence 6. Edinburgh : Edinburgh University Press.
(MELTZER 72) B.MELTZER and D.MICHIE (Eds) : Machine Intelligence 7. Edinburgh : Edinburgh University Press.
(MERIALDO 79) B.MERIALDO : Représentation des ensembles en démonstration automatique. Thèse 3ème cycle - PARIS VI (1979).
(MICHIE 68) D.MICHIE (Ed) : Machine Intelligence 3 :Edinburgh : Edinburgh University Press.
(MIGNOTTE 76) M.MIGNOTTE :Factorisation des polynômes sur un corps fini. Astérisque 38-39, (1976), p.149-157.
(MINKER 77) J.MINKER : Control structure of a pattern-directed search system. Proc.Workshop Pattern-Directed Inference Systems, SIGART Newsletter 63,(1977), p.7-14.
(MITCHELL 81) T.M.MITCHELL, P.E.UTGOFF, B.NUDEL, R.BANERJI : Learning problem solving heuristics through practice, IJCAI 81, Vancouver, (1981), p.127-134.
(MOORE 77) R.C.MOORE : Reasoning about knowledge and action. In IJCAI-5, p.223-227.
(MOORE 79) R.C. MOORE : Reasoning about knowledge and action. Tech. Note 191, SRI International, Artificial Intelligence Center, MenloPark, CA.
(MOSES 67) J.MOSES : Symbolic Intregration . MAC-TR-47, Projet MAC, Massachusetts Insitute of Technology, Cambridge, MA.
(MOSES 71) J.MOSES : Symbolic integration, the stormy decade. Communications ACM 14, (1971), p.548-560.
(MOSES 71a) J.MOSES : Algebraic simplification : a guide for the perplexed, Proceedings SS-SAM, Los Angeles, March 71, p.282-304.
(MOSES 71b) J.MOSES : Symbolic integration : the stormy decade, Proceedings SS-SAM, Los Angeles, March 71, p.427-440.
(MOSES 74) J.MOSES : The evolution of algebraic manipulation algorithms, IFIP 74, North-Holland Publishing Company, p.483 488.
(MUSSER 75) D.R.MUSSER : Multivariate polynomial factorization, Journal of ACM, vol.22, n°2, (1975), p.291-308.
(NEVINS 74) A.J.NEVINS : A human oriented logic for automatic theorem proving. J.ACM 21 - (1974) - p.606-621.
(NEVINS 75a) A.J.NEVINS : Plane geometry theorem proving using forward chaining. AI 6 - (1975) - p.1-23.
(NEVINS 75b) A.J.NEVINS : A relaxation approach to splitting in a automatic theorem prover. AI 6 - (1975) - p.25-39.
(NEWELL 63) A.NEWELL and H.A SIMON : GPS, a program that simulates human thought. In Computers and Thought, E.Feigenbaum et J.Feldman eds, p.279-293.

(NILSSON 71) N.J.NILSSON : Problem solving methods in artificial intelligence. Mc Graw Hill, New-York, (1971).
(NILSSON 79) N.J.NILSSON : A production system for automatic deduction. In MI 9.
(NORMAN 77) A.C.NORMAN and P.M.A.MOORE : Implementing the new RISCH integration algorithm. Proc. 4th.Int. Colloquium on advanced computing methods in theoretical physics, Marseilles (1977).
(NORMAN 78) A.C.NORMAN : Symbolic and algebraic modes in REDUCE. REDUCE Newsletter 3 (July 1978), p.5-9.
(NORTON 71) L.M.NORTON : Experiments with a heuristic theorem proving programm for predicate calculus with equality. AI 2 (1971) - p.261-284.
(PASTRE 76) D.PASTRE : Démonstration automatique de théorèmes en théorie des ensembles. Thèse 3ème cycle -PARIS VI (1976).
(PASTRE 78a) D.PASTRE : AUtomatic theorem proving in set theory, AI 10 (1978) p.1-27.
(PASTRE 78b) D.PASTRE : Observation du mathématicien : aide à l'enseignement et à la démonstration automatique de théorèmes. Educ. Studies in Mathematics 9 (1978), p.461-502.
(PASTRE 79) D.PASTRE: Représentation et structuration des connaissances en démonstration automatique de théorèmes .Colloque Intelligence Artificielle-ROUEN-Sept 79,publication verte du CNRS GR22-n°11, p.145-160.
(PASTRE 82a) D.PASTRE: un programme et deux mathématiciens face a quelques théorèmes, Colloque intelligence artificielle-LE MANS-septembre 1982,publication du CNRS/gr22-n°30,p.241-282.
(PASTRE 82b) . D.PASTRE: a language for expressing mathematical knowledge in automatic theorem proving ,ECAI,ORSAY-11/14 juillet 1982,p.116-118.
(PITRAT 66) J.PITRAT : Réalisation de programmes de démonstration de théorèmes utilisant des méthodes heuristiques. Thèse d'Etat - Paris - 1966.
(PITRAT 70) J.PITRAT : Un programme de démonstration de théoremes, Monographie AFCET 7, Dunod (1970).
(PITRAT 77) J.PITRAT : A chess combination program which uses plans. Artificial Intelligence 8(3),p.275-321.
(POLYA 54) G.POLYA : Mathematics and plausible reasoning, Vols.1,2. Princeton Univ.Press, Princeton, New Jersey, (1954).
(POLYA 57) G.POLYA : Comment poser et résoudre un problème. Dunod Editeur. Paris (1957).
(POLYA 58) G.POLYA : Les mathématiques et le raisonnement "plausible". Gauthier-Villars Editeur. Paris (1958).
(POPLE 77) H.E.POPLE : The formation of composite hypotheses in diagnostic problem solving : an exercise in synthetic reasoning. In IJCAI-5, p.1030-1037.
(PRAWITZ 60) PRAWITZ : An improved proof procedure. Theoria, vol.26 (1960).
(RISCH 69) R.H.RISCH : The solution of the problem of integration in finite terms. Trans.American Mathematical Society 139, (1969), p.167-189.
(RISCH 69) R.H.RISCH : The problem of integration in finite terms. Trans.A.M.S. 139, (1969), p.167-189 (MR 38 5759).
(RISCH 70) R.H.RISCH : The solution of the problem of integration in finite terms. Bulletin AMS 76, (1970), p.605-608.
(RICH 79) A.D.RICH et D.R.STOUTMEYER : Capabilities of

the MUMATH 79 Computer algebra system for the intel 8080 Microprocessor. Springer-Verlag LNCS 72 (1979), p.241.

(ROBINSON 65) J.A.ROBINSON : A machine-oriented logic based solution principle. JACM, 12(1),p.23-41.

(ROUSSEL 75) P.ROUSSEL : PROLOG : Manuel de référence et d'utilisation. Groupe d'Intelligence Artificielle, Marseille-Luminy ; September 1975).

(SACERDOTI 74) E.D.SACERDOTI : Planning in a hierarchy of abstraction spaces. AI 5 (1974) - p.115-135.

(SHIMURA 73) M.SHIMURA, F.H.GEORGE : Rule oriented methods in problem solving. AI 4 (1973) - p.203-223.

(SIKLOSSY 71) L:SIKLOSSY, V.MARINOV : Heuristic search vs.Exhaustive search, IJCAI 2 - (Londres 1971) - p. 601-607.

(SIKLOSSY 73) L.SIKLOSSY, RICH, MARINOV : Breadth first search : some surprising results. A.I 4 n°1 - Spring 1973 - p.1-28).

(SIKLOSSY 73) L.SIKLOSSY, J.ROACH : Proving the impossible is impossible is possible : disproofs based on hereditary partitions. IJCAI 4 (1973), p.383-387.

(SIRET 70) Y.SIRET : Contribution au calcul formel sur ordinateur. Thèse d'Etat - Grenoble (juillet 1970)

(SLAGLE 63) J.R.SLAGLE : A heuristic program that solves symboli integration problems in freshman calculus. In Computers and Thought, p.191-203. (Also in JACM, 1963, vol.10., p.507-520)

(SLAGLE 71) J.R.SLAGLE : Artificial Intelligence : The Heuristic programming Approach. New-York : Mc Graw-Hill.

(SLAGLE 73) J.R.SLAGLE, L.M.NORTON : Experiments with an automatic theorem prover having partial ordering inference rules. C.ACM. Vol.16 - n°11 (nov.1973) -p.682-688.

(SLAGLE 74) J.R.SLAGLE : Automated theorem-proving for theories with simplifiers, commutativity and associativity, J.ACM 21 (1974), p.622-642.

(TOURNIER 71) E.TOURNIER : Un exemple d'utilisation du calcul formel sur ordinateur. Méthode générale de localisation des racines d'une équation algébrique à coefficients complexes. Thèse 3ème cycle - USMG -Grenoble (septembre 1971).

(VAN VAALEN 75) J.VAN VAALEN : An extension of unification to substitutions with an application to automatic theorem proving. In IJCAI-4, p.77-82.

(VIVET 73) M.VIVET : Un programme qui vérifie des identités en utilisant le raisonnement par récurrence. Thèse 3ème cycle - PARIS VI (1973).

(VIVET 77) M.VIVET : Use of knowledge bases in automatic theorem proving and symbolic manipulations, fourth International Colloquium on advanced computing methods in theorical physics. St Maximin - March 1977 - p.122-134.

(VIVET 80) M.VIVET: Calcul formel et intelligence artificielle. Colloque intelligence artificielle -CAEN-sept83-publication du CNRS/gr22-n°20,p.5-40.

(VIVET 81) M.VIVET: Calcul algébrique et représentation de connaissances mathématiques , Congrès AFCET-IA, NANCY, septembre 1981, p.743-750.

(VIVET 82a) M.VIVET: Towards expert system in algebraic and symbolic manipulation -ECAI, 11/14 juillet 1982, Orsay.

(VIVET 82b) M.VIVET: Construction d'un système expert autour d'u système de calcul algébrique . Journées systèmes

experts , AFCET-ADI-Avignon-mai 1982.

(VIVET 82c) M.VIVET: CAMELIA : vers un programme mathématicien , colloque Intelligence Artificielle -LE MANS-septembre 1982, Publication du CNRS/GR22. n°30, p. 187-240.

(VIVET 83) M.VIVET: Un exemple d'utilisation de méta-règles :la sélection des plans dans CAMELIA.Colloque d'intelligence artificielle,CHAMBERY,septembre 83,publication du CNRS/GR22.

(WANG 71) S.H.WANG : Automatic computation of limits. 2nd Symposium on symbolic and algebraic manipulation. Mars 1971. p.458-464.

(WANG 76) P.S.WANG : Factoring multivariate polynomials over algebraic number fields. Math.Comp.30, (1976), p.324-336.

(WARREN 77) D.H.D.WARREN and L/M/PEREIRA : PROLOG : The language and its implementation compared with LISP. Proc of the symp. on Artificial Intelligence and Programming Languages(ACM); SIGPLAN notices,12(8);and SIGART Newsletter, no.64, p.109-115.

(WATERMAN 70) D.WATERMAN : Generalization learning techniques for automating the learning of heuristics, AI 1 (1970), p.121-170.

(WATERMAN 78) D.WATERMAN and F.HAYES-ROTH (Eds) : Pattern-Directed Inference Systems PDIS. New-York : Academic Press.